Stéphane Doncieux Benoît Girard
Agnès Guillot John Hallam
Jean-Arcady Meyer
Jean-Baptiste Mouret (Eds.)

LNAI 6226

From Animals to Animats 11

11th International Conference
on Simulation of Adaptive Behavior, SAB 2010
Paris - Clos Lucé, France, August 2010, Proceedings

Springer

Lecture Notes in Artificial Intelligence 6226

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Stéphane Doncieux Benoît Girard
Agnès Guillot John Hallam
Jean-Arcady Meyer Jean-Baptiste Mouret (Eds.)

From Animals to Animats 11

11th International Conference
on Simulation of Adaptive Behavior, SAB 2010
Paris – Clos Lucé, France, August 25-28, 2010
Proceedings

Springer

Volume Editors

Stéphane Doncieux
Benoît Girard
Agnès Guillot
Jean-Arcady Meyer
Jean-Baptiste Mouret

ISIR, Université Pierre et Marie Curie, Paris, France
E-mail: {stephane.doncieux, benoit.girard, agnes.guillot,
jean-arcady.meyer, jean-baptiste.mouret}@upmc.fr

John Hallam
Mærsk Mc-Kinney Møller Institute
University of Southern Denmark
Odense M, Denmark
E-mail: john@mip.sdu.dk

Library of Congress Control Number: 2010932527

CR Subject Classification (1998): I.2, F.1, I.4, I.5, J.3, H.3

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-15192-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-15192-7 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

This book contains the articles presented at the 11th International Conference on the Simulation of Adaptive Behavior (SAB 2010), which was held at the Museum d'Histoire Naturelle and at the University Pierre et Marie Curie of Paris during August 25–27, 2010. On August 28, the attendees moved to Le Clos Lucé – the last home of Leonardo da Vinci – where the 20th anniversary of the conference was celebrated.

The objective of the biennial SAB Conference is to bring together researchers in computer science, artificial intelligence, artificial life, complex systems, robotics, neurosciences, ethology, evolutionary biology, and related fields so as to further our understanding of the behaviors and underlying mechanisms that allow natural and artificial animals to adapt and survive in uncertain environments.

Adaptive behavior research is distinguished by its focus on the modelling and creation of complete animal-like systems, which – however simple at the moment – may be one of the best routes to understanding intelligence in natural and artificial systems. The conference is part of a long series that started with the first SAB Conference held in Paris in September 1990, which was followed by conferences in Honolulu 1992, Brighton 1994, Cape Cod 1996, Zürich 1998, Paris 2000, Edinburgh 2002, Los Angeles 2004, Rome 2006 and Osaka 2008. In 1992, the MIT Press introduced the quarterly journal *Adaptive Behavior,* now published by SAGE Publications. The establishment of the International Society for Adaptive Behavior (ISAB) in 1995 further underlined the emergence of adaptive behavior as a fully fledged scientific discipline. The present proceedings are a comprehensive and up-to-date resource for the latest progress in this exciting field.

The articles cover all main areas in animat research, including perception and motor control, action selection, motivation and emotion, internal models and representation, collective behavior, language evolution, evolution and learning. The authors focus on well-defined models, computer simulations or robotic models, that help to characterize and compare various organizational principles, architectures, and adaptation processes capable of inducing adaptive behavior in real animals or synthetic agents, the animats.

This conference and its proceedings would not exist without the substantial help of a wide range of people. Foremost, we would like to thank the members of the Program Committee, who thoughtfully reviewed all the submissions and provided detailed suggestions on how to improve the articles. We also acknowledge the significant contribution of Chantal Iannarelli to the conference's organization. And, once again, we warmly thank Jean Solé for the artistic conception of the SAB 2010 poster and the proceedings cover.

Organization

From Animals to Animats 11, the 11th International Conference on the Simulation of Adaptive Behavior (SAB2010) was organized by the Institut des Systèmes Intelligents et de Robotique of the UPMC/CNRS, and ISAB (International Society for Adaptive Behavior).

Conference Chairs

Jean-Arcady Meyer	ISIR, UPMC/CNRS, Paris, France
Agnès Guillot	ISIR, UPMC/CNRS, Paris, France
John Hallam	The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Odense, Denmark

Local Organizers

Stéphane Doncieux	ISIR, UPMC/CNRS, Paris, France
Benoît Girard	ISIR, UPMC/CNRS, Paris, France
Jean-Baptiste Mouret	ISIR, UPMC/CNRS, Paris, France

Program Committee

Hussein Abbass, Australia
Frederic Alexandre, France
Pierre Andry, France
Ronald Arkin, USA
Angelo Arleo, France
Minoru Asada, Japan
Gianluca Baldassarre, Italy
Christian Balkenius, Sweden
Ryad Benosman, France
Luc Berthouze, UK
Aude Billard, Switzerland
Josh Bongard, USA
Joanna Bryson, UK
Nicolas Bredeche, France
Seth Bullock, UK
Angelo Cangelosi, UK
Ricardo Chavarriaga, Switzerland
Francis Colas, Switzerland
Thomas Collett, UK
Luis Correia, UK
Nikolaus Correl, Switzerland
Kerstin Dautenhahn, UK
Ezequiel di Paolo, Spain
Julien Diard, France
Stéphane Doncieux, France
Marco Dorigo, Belgium
Richard Duro, Spain
Michael Dyer, USA
Jason Fleisher, USA
Luca Gambardella, Switzerland
Philippe Gaussier, France
Benoît Girard, France
Faustino Gomez, Switzerland
Osamu Hanagata, Japan
Inman Harvey, UK
Mark Humphries, France

Phil Husbands, UK
Fumiya Iida, Switzerland
Hiroyuki Iizuka, Japan
Auke Jan Ijspeert, Switzerland
Takashi Ikegami, Japan
Akio Ishiguro, Japan
Koji Ito, Japan
Naoto Iwahashi, Japan
Frederic Kaplan, Switzerland
Kuniaki Kawabata, Japan
Mehdi Khamassi, France
Toshiyuki Kondo, Japan
Robert Kozma, USA
Yasuo Kuniyoshi, Japan
Hanspeter Mallot, Germany
Davide Marocco, Italy
Alcherio Martinoli, Switzerland
Francois Michaud, Switzerland
Francesco Mondada, Switzerland
Jean-Baptiste Mouret, France
Kazuyuki Murase, Japan
Ryohei Nakano, Japan
Ulrich Nehmzow, Ireland
Stefano Nolfi, Italy
Tetsuya Ogata, Japan
Pierre-Yves Oudeyer, France
Pietro Pantano, Italy
Jan Peters, Germany
Rolf Pfeifer, Switzerland
Tony Prescott, UK
Mikhail Prokopenko, Australia
Marc Schoenauer, France
Michele Sebag, France
Noel Sharkey, UK
Denis Sheynikhovich, France
Olivier Sigaud, France
Olaf Sporns, USA
Kenneth Stanley, USA
Kenji Suzuki, Japan
Jun Tani, Japan
Charles Taylor, USA
Tim Taylor, UK
Vadim Tikhanoff, UK
Peter Todd, Germany
Elio Tuci, Italy
Eiji Uchibe, Japan
Richard Vaughan, Canada
Paul Vogt, The Netherlands
Hiroaki Wagatsuma, Japan
Barbara Webb, UK
Janet Wiles, Australia
Myra S. Wilson, UK
Rachel Wood, UK
Florentin Woergoetter, Germany
Tom Ziemke, Sweden

Sponsoring Institutions

We are greatly indebted to our sponsors:

Table of Contents

Animat Approaches for Adaptive Behaviour

Perception and Motor Control

Action Selection and Behavioural Sequences

Navigation and Internal World Models

Learning and Adaptation

Evolution

Collective and Social

From Mirror Writing to Mirror Neurons

Michael A. Arbib

Computer Science, Neuroscience and the USC Brain Project
University of Southern California, Los Angeles, CA 90089-2520, USA
arbib@usc.edu

Abstract. The article offers a personal perspective on Simulation of Animal Behavior, starting with the inspiration of Norbert Wiener's 1948 Cybernetics for the publication of Brains, Machines, and Mathematics in 1964. This led to a range of simulations of the brains and behaviors of frogs (Rana computatrix), rats, monkeys and humans. Such work is paralleled by work in biologically-inspired robots, traceable back to Grey Walter's Machina speculatrix of 1953. Recent work includes detailed modeling of hand control, mirror neurons and sequencing as part of a program to determine "What the Macaque Brain Tells the Human Mind". The Mirror System Hypothesis for the evolution of the language-ready brain suggests a path for evolution of brain mechanisms atop the mirror system for grasping, with new processes supporting simple imitation, complex imitation, gesture, pantomime and finally protosign and protospeech. It is argued that this progression suggests the "dead end of the simple model" if we are to fully explore the lessons of Simulation of Animal Behavior for computational neuroscience and biologically-inspired robotics.

Keywords: cybernetics, Rana computatrix, schema theory, biologically-inspired robots, hand control, mirror neurons mirror system hypothesis, language evolution.

1 Mirror Writing

The allusion to "mirror writing" in the title is not to define a precursor of the study of mirror neurons, but simply to mark a major theme of SAB2010, a celebration of a most distinguished predecessor, Leonardo da Vinci, whose work grounded both art and technology in the careful study of animal form and mechanism. Jean-Arcady Meyer's talk on "Leonardo's automata" initiates this celebration, which culminates on the last day of the conference at Le Clos Luce, last home of Leonardo da Vinci. Leonardo's notebooks were not only written in a form of shorthand that he invented himself but also used "mirror writing", starting at the right side of the page and moving to the left. The talk will open discussion of the extent to which mirror neurons should be part of the next generation of animal-inspired animats whose control systems are inspired by the study of the brains or nervous systems of diverse animals. But there is much to talk about before that.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 1–12, 2010.

2 Fifty Years of Brains, Machines and Mathematics

My own interest in SAB themes started in the late 1950s when, until then a pure mathematician, I was captivated by the themes of Norbert Wiener's *Cybernetics, or Control and Communication in the Animal and the Machine* [1]. This book led me to the work of Warren McCulloch and Walter Pitts. Their 1943 characterization of neurons as threshold logic units [2] showed that a neural network could serve as the "brain" for any Turing machine [3]. Such ideas, plus much reading in automata theory and computability, led to my 1960 Honours Thesis in Mathematics at Sydney University on "Turing Machines, Finite Automata and Neural Nets", published a year later in the *Journal of the Association for Computing Machinery* [4]. Then the visual neurophysiologist Bill Levick introduced me to the 1959 paper "What the Frog's Eye Tells the Frog's Brain"[5] written by neurophysiologist Jerry Lettvin and neuroanatomist Humberto Maturana in collaboration with McCulloch & Pitts to provide neuroethological exploration of the ideas of their 1947 group-theoretic approach to pattern recognition[6]. This led me to an interest in neural networks not only as mathematical automata but as models of processes in real brains.

Inspired by all this, I moved to MIT in January of 1961. On summer vacation in 1962, I spent the winter term at the University of New South Wales back in Sydney, giving a course of lectures on *Brains, Machines and Mathematics* that were published in 1964 as my first book [7]. A brief review of each chapter will help us see what was in place almost 50 years ago, and what has changed since.

Chapter 1. Neural Nets, Finite Automata, and Turing Machines showed how any finite automaton could be simulated by a McCulloch-Pitts network. It then introduced Turing machines and the basics of recursive sets and regular and realizable events. It thus located neural networks within the domain of discrete mathematics.

Chapter 2. Structure and Randomness focused first on Lettvin et al.'s approach to the visual system of the frog, offering a brief comparison with the work of Hubel and Wiesel on the discovery of simple and complex cells in the visual cortex of cats [8]. The intriguing issue was why the ganglion cells of the frog's retina should already be computing features of apparent relevance to the frog's behavior (responsive to local cues for prey-like and predator-like stimuli) while the cat's primary visual cortex was extracting edge information of no direct behavioral relevance. The chapter then turned to an exposition of Rosenblatt's Perceptron [9] to introduce the theme of supervised learning in neural networks. A glaring omission was discussion of Hebb's formulation of cell assemblies and the unsupervised learning that formed them [10].

Chapter 3. The Correction of Errors in Communication and Computation focused on the challenge of understanding how reliable brains could be composed from unreliable neurons, and started with von Neumann's multiplexing scheme [11]. An exposition of Shannon's theory of reliable computation in the presence of noise [12] grounded the Cowan-Winograd theory of reliable automata, which took a neural network and made it more reliable by introducing redundancy in a fashion guided by any suitable code from communication theory.

Chapter 4. Cybernetics started with the study of feedback and oscillation that was central to Wiener's insights that mathematics developed for control systems could also help us understand biological function and its disorders, while the section on resonant frequencies in neural networks explored ideas of Peter Greene [13]. Further

topics included a brief look at prosthesis and homeostasis, the Pitts & McCulloch [6] model of gestalt and universals, and a brief look at links to artificial intelligence, including the hierarchical Test-Operate-Test-Exit units of Miller, Galanter and Pribram [14] and the work on machine learning of checkers by Samuel [15].

Finally, **Chapter 5. Gödel's Incompleteness Theorem** started with a tour of the foundations of mathematics then developed the notions of recursive and arithmetical logics as a basis for the proof of Kurt Gödel's incompleteness theorem [16] which shows that any *consistent* logic L from whose axioms one can recursively prove theorems about arithmetic must be incomplete – there will be true properties of the integers that cannot be deduced from L's axioms. The book closed by showing why this result does not bar machine intelligence – for surely an intelligent person can make mistakes (and thus would not be consistent) and learn from those mistakes, as should any machine that is potentially intelligent.

3 Simulation of Animal Behavior

In reviewing *Brains, Machines and Mathematics* in *Scientific American,* Jacob Bronowski complained that it did not provide a proper mathematics of the brain. Subsequent years have seen the development of detailed neural modeling of neurons based on compartmental modeling [17], dynamic systems, optimization principles, diverse learning theories and so on. But here I want to stress that my entry into the simulation of animal behavior came a few years later with Rich Didday in asking *What the Frog's Eye Tells the Frog,* inspired by the work of David Ingle [18] on the prey catching behavior of frogs. We developed perhaps the first Winner-Take-All model in providing a distributed neural model of prey selection [19]. This work opened up a series of investigations of *Rana computatrix,* the frog which computes, yielding eleven post-Didday Ph.D. theses, spanning such contributions as those of Rolando Lara [20], Donald House [21], DeLiang Wang [22], and Fernando Corbacho [23]. These papers introduced the major themes of computational neuroethology: linking behavior to interaction of specific neurons in specific brain regions of the animal under study. Our work did not culminate in a single overall model embedded in a robot, but the work of House did inspire Ron Arkin's first venture into biologically-inspired robotics [24].

However, the simulation of animal behavior does not always require neural verisimilitude. In my own work, this led to the development of schema theory. "Pure" schema theory studies schemas as dynamical, interacting systems which underlie mental and overt behavior. "Neural" schema theory uses data from neurophysiology, lesion studies and brain imaging to relate schemas to distributed neural mechanisms, with an initial schema model being reformulated in the light of these "sub-behavioral" data. Mary Hesse and I went one step further up the hierarchy – introducing "social" schema theory to relate "schemas in the head" to the collective representations of a society – whether a language, religion, ideology, myth, or scientific society [25].

An early example of schema-based interpretation for visual scene analysis is the VISIONS system [26]. When a new image is presented, low-level processes build a representation in the intermediate database – current estimates including contours and

surfaces tagged with features such as color, shape, and location. A Long Term Memory contains the schemas for scene interpretation organized in networks so that activation of certain schemas increases (cooperation) or decreases (competition) the confidence level for other schemas. For a given image, the activation of schema instances continues until the image is interpreted. However, attending to details or regions that had not yet been fully analyzed may lead to top-down commands to update the intermediate database. The Working Memory (WM) then provides a set of schema instances each linked to a certain region of the image, and provided with certain cross-linkages (e.g., a roof is part of a house) as well as certain parameters (e.g., the size, shape, location and color of the roof).

A touchstone for "not-so-neural" simulation of animal behavior – and the even more relaxed notion of "cognitive robotics", combining action-oriented perception [27] & embodied cognition – is Grey Walter's *Machina speculatrix* (whose name, perhaps subconsciously, inspired that of *Rana computatrix*), the famous 1953 robot that would "explore" until its battery ran low, at which time it would revisit its "hutch" to recharge [28]. Grey Walter's influence was itself recharged by the publication of Valentino Braitenberg's *Vehicles* in 1984 [29], and celebrated on the 50th anniversary of its publication with a conference held in Bristol– the proceedings were published as a 2003 theme issue of the Philosophical Transactions of the Royal Society, *Biologically inspired robotics*, compiled by R. I. Damper, and include several contributions by researchers familiar in SAB circles.

My group has a track record of simulating not only frogs but also mammals, starting with studies of depth perception [30] and cerebellum [31] and including study of rat navigation, yield the Taxon Affordance and World Graph Models [32, 33]. However, a crucial impetus to new research was provided by data on preshaping of hand movements by Jeannerod and Biguer [34] which led me to develop an influential schema-theoretic model of the reach-to-grasp [35]. The resultant misleading impression of expertise concerning hand movements led to an invitation to the IUPS Satellite Symposium on Hand Function and the Neocortex, Melbourne, 1983 which required a flurry of effort with Thea Iberall & Damian Lyons on coordinated control programs for movements of the hand [36, 37] to turn impression into reality.

At UMass, my group developed a new approach to how interacting brain regions support language – embedding language mechanisms in a framework of action-oriented perception [38, 39]. After I moved to USC in 1986, nothing much happened in my language research for more than 10 years. However, in 1990, Marc Jeannerod (Lyon, France), Giacomo Rizzolatti (Parma, Italy) & Hideo Sakata (Tokyo, Japan) invited me to join their HFSP project to explore mechanisms of hand-eye coordination in macaque and human. Work with Jeannerod and his colleagues led to a control systems recasting of optimization for arm movements and its extension to arm-hand coordination [40], and integrated modeling of the superior colliculus (frog tectum *redux*), the basal ganglia, and their interaction with parietal and frontal cortex in learning and controlling conditional and compound actions. The collaboration with Rizzolatti & Sakata led to the FARS (Fagg-Arbib-Rizzolatti-Sakata) model of parietal-premotor interactions in primate control of grasping [41]. During this collaboration, the Parma group discovered mirror neurons [42] and we developed the first model of mirror neurons, showing how they could be formed by learning [43].

All this established a program to determine "What the Macaque Brain Tells the Human Mind". The aim is to exploit more data on macaque brain regions which are possible homologues of human brain areas relevant to language, and add data on the connectivity of these areas in both human and macaque, to yield improved estimates of degrees of homology [44]. This program will be furthered by methods of Synthetic Brain Imaging to predict the results of human studies from models inspired by the findings of animal neurophysiology. [43, 45, 46]. The remaining sections chart some of our recent progress concerning the transition from action to language.

4 Modeling Hand Control, Mirror Neurons and Sequencing

The key idea of the FARS model [47] is that parietal area AIP does not "know" the identity of an object to be grasped, but can only extract affordances (opportunities for grasping for the object viewed as an unidentified solid); prefrontal cortex uses the IT (inferotemporal) identification of the object, in concert with task analysis and working memory, to help AIP select the appropriate action from its "menu". Neurons in an area of premotor cortex of the macaque monkey called F5 then instruct the primary motor cortex F1 (M1) to command the grasp.

A neuron is called a *mirror neuron* if its firing correlates with both execution of a specific action and observation of more-or-less related actions. Mirror neurons for different manual actions were first discovered in F5 [48] and later found in parietal cortex. The MNS (Mirror Neuron System) model [43] then showed how mirror neurons could be formed by a learning process: The training signal is provided by activation of canonical neurons which encode a grasp in the animal's repertoire. Training enables mirror neurons to learn to fire in response to appropriate patterns of hand movement relative to object affordances – thus becoming able to respond to the observed actions of others. A major goal of the proposed work is to extend MNS & its extension, MNS2 [49], to handle the compound actions involved in tool use – in some of which the hand is the end effector, while in others part of the tool is the end effector [50]. For example, the blade of a screwdriver must be matched against the affordance of the groove in the screw head, shifting the locus of visual attention, and modifying the interpretation of haptic feedback.

To model compound actions, we start from the observation that a sequence may result from *competitive queuing* [CQ; 51, 52]: At any time neurons representing different actions are activated with different priority signals; a WTA process then selects the action associated with the largest priority. The "winner" is then inhibited and the competition begins anew to choose the next action. By setting initial activation appropriately, one may be able to guarantee the execution of a specific sequence. We have recently developed a more powerful methodology, *augmented competitive queuing* (ACQ) [53]. A key difference between ACQ and classical CQ is that the activation level of motor program elements is dynamically computed in each "time step" rather than being completely specified before sequence execution. We define the *executability* of an action as a measure of how readily it can be executed based on available *affordances* – decreasing with the effort required, and 0 if the action is currently impossible. The *desirability* of each action represents the expected reinforcement that will follow if the action is executed in the current circumstances (an action desirable

when one is hungry may not be desirable when one is thirsty) and is learned via temporal difference [TD] learning [54]. We then define the (context-dependent) *priority* of an action A at any particular time t by the formula:

*priority(A, t) = executability(A, t) * desirability(A, t).*

The goal of opening a bottle requires a motor program with 2 distinct movement patterns – turning the cap, and pulling the cap off. If a reward is given for opening the bottle, TD learning would eventually leave *cap pulling* with a higher desirability value than *cap turning*. Until *cap pulling* is executable, *cap turning* would repeatedly be selected, unscrewing the cap. Once executable, *cap pulling*'s higher desirability value will dominate the ACQ network, it will be selected for execution, and the bottle will be opened. The dissociation of motor schema executability from desirability allows formation of dynamic motor sequences with goal-oriented, repetitive elements that can take advantage of serendipitous events and skip unnecessary steps.

A crucial feature of ACQ is *an expanded role for the Mirror System*. Where previous discussions of the mirror system have it encoding either the action intended by the agent, or the action of another as recognized by the agent, ACQ posits that during self-action, the mirror system can be activated in two ways: by corollary discharge which encodes the intended action, and by analysis of visual input so that recognition may encompass apparent actions as well as intended actions. Moreover, we add that the mirror system can signal when it observes that the intended action was not performed successfully. As a result, TD learning can update estimates of expected reinforcement (desirability) for (i) the intended action (specified by the output from the Actor) *unless it is blocked by a lack-of-success signal*, as well as (ii) any other actions that appear to have been performed. Similarly, estimates of whether or not an action was executed successfully are used to update estimates of executability for the intended action when performed for the currently perceived environmental state.

5 The Mirror System Hypothesis (MSH)

Macaque area F5 is homologous to human Brodmann's area 44, which is part of Broca's area, an area traditionally associated with the production of speech. Moreover, human brain imaging studies (e.g., [55]) show activation for both grasping and observation of grasping in or near Broca's area, though we do not have data on activity of individual neurons in this mirror system. But why might a mirror system for grasping be associated with an area traditionally associated with speech production? The answer starts with the observation that for deaf people, language may take the form of a signed language. We thus now associate Broca's area with language production as a multimodal performance rather than speech alone [56]. This led Rizzolatti and myself [57] to formulate the Mirror System Hypothesis – that mechanisms for the parity property of language (the receiver often recognizes what the sender means) evolved atop mirror systems for grasping. Chimpanzees typically take ten or more trials to learn to "imitate" a behavior, paying more attention to where the manipulated object was being directed than to the movements of the demonstrator [58]. Such imitation is laborious compared to the rapidity with which humans can imitate novel sequences – this is the contrast between "simple" imitation and "complex" imitation.

My hypothesized sequence [59] leading to protosign and beyond is that *complex imitation of praxic actions* grounded evolution of mechanisms for:

(1) *Pantomime*. Whereas imitation is the generic attempt to reproduce praxic movements performed by another, pantomime is communicative, intended to get the observer to think of a specific action, object or event.

(2) *Protosign* is a system of conventionalized gestures adopted by a community to exploit the semantic richness of pantomime while avoiding its ambiguities and cost. The claim is that *conventional* symbols emerged to expand and then replace the communicative capacities of pantomiming.

MSH can also be seen as a hypothesis about language mechanisms in the human brain (Figure 1). The core idea [60] is that, just as macaque mirror neurons are related to specific actions, so does the human brain contain mirror neurons related to specific words – but to words as articulatory actions, not to words as semantic units. Rather, semantics is provided by linkages to schema networks in the ventral stream and prefrontal cortex which support both perception and the planning of action. Supporting evidence comes from Hickok & Poeppel [61] who observe that speech perception diverges into a dorsal stream which maps sound onto articulatory-based representations and a ventral stream which maps sound onto meaning. Schema activity can control the execution of actions via the dorsal pathway; conversely, mirror system recognition of actions can update motor schemas in the schema assemblage (see the 2 connections at right of Figure 1).

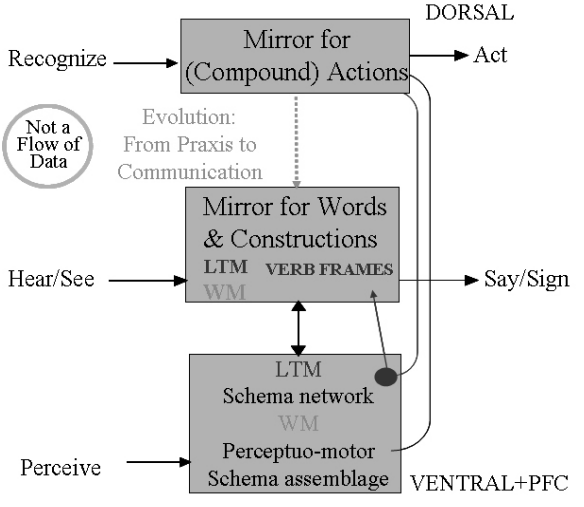


Fig. 1. This overview [60] shows that the mirror system for action recognition (shared by macaque & human) has an evolutionary relationship to the human mirror system for the phonological form of words and constructions but that these systems are not linked by a direct flow of data. Instead, the phonological form gains its meaning by being linked to an assemblage of perceptual and motor schemas which support both perception and the planning of action. As in the FARS model, the dorsal stream is responsible for setting the parameters of action while the ventral system is responsible for selection among possible actions.

The key data for MSH place a mirror system for grasping, and thus (inferentially) for language, in Broca's area, and so one might infer that the prime effect of damage to Broca's area would be the inability to pronounce words. However, although production is indeed effortful in Broca's aphasia, a major factor there is *agrammatism,* a pattern of syntactically defective speech that may range in severity from production only of one-word utterances, completely lacking in grammatical organization, to mildly 'telegraphic' speech. There has been much detailed and systematic description of the linguistic output in agrammatic speech, and cross linguistic comparisons have revealed that the features of agrammatism are not fixed, but are conditioned by the grammatical structure of the speaker's language.

6 SemRep – Capturing the Basic Elements and Relationships in a Scene or Episode

We sought a flexible framework within which to characterize the relation between form and function, but with the freedom to modify it as research proceeds, importing mechanisms from frameworks such as those cited above only when appropriate for our neurolinguistic modeling. Our choice is Construction Grammar (CG) [62], which defines a more or less language-specific set of *constructions* each of which combines *syntax* with *semantics*. CG respects the diversity of the grammatical systems of different languages, yet can encompass categories grounded in semantic and pragmatic prototypes as well as syntactic parameters.

CG neither supports nor refutes the Mirror System Hypothesis but will allow our modeling to assess data from experiments designed to test localization of components posited in different models of grammar. Kemmerer [e.g., 63] has made explicit the relevance of CG to neurolinguistics and has used it to present the major semantic properties of action verbs and to argue that the linguistic representation of action is grounded in the mirror neuron system. A key issue for us, then, is to define a semantics which includes action and has links to mirror systems as well as CG.

We are currently applying CG to the description of visual scenes. As background, recall the view on the non-linguistic representation of such scenes by assemblages of instances of visual schemas in the VISIONS system. Itti & Arbib [64] added the cumulative action of attention to VISIONS and outlined its extension to a dynamic visual environment. Arbib and Lee [65] introduced SemRep as a hierarchical graph-like semantic representation of a visual scene, an abstraction from the assemblages of schema instances generated in the VISIONS system WM, but with the crucial addition of actions and events extended in time but related to objects linked to specific regions of the spatial frame defined by vision. We are developing Template Construction Grammar (TCG) as a specific implementation of CG processing to support modeling of the relation between vision and language (for somewhat related efforts, see [67, 68]). Our key innovation is that the meaning of an utterance is given as a SemRep graph (with suitable extensions to be provided in further work). A SemRep may yield one or more sentences as TCG finds ways to "cover" the relevant portion of the given SemRep with a set of small subgraphs chosen so that a construction is available which expresses that subgraph in the given language. In production mode, the template acts to match constraints for selecting proper constructions by being superimposed on the

SemRep graph. The semantic constraint of each construction is considered to be encoded in the template since the template specifies concepts as well as the topology of a SemRep graph. Thus, constructions are applied recursively, starting with lexical constructions, which have no variables and then by applying higher-level constructions in such a way that slots are matched to the results of earlier application of constructions whose category matches that of the slot. In this way, the scheme for VISIONS may be lifted to a structure in which the Working Memory for the present state of applying constructions to the current SemRep provides the workspace for the operation of construction selection and attachment.

7 Impasse du Petit Modèle

In June of 2005, I gave the Closing Address at the *Ecole d'été Maths et Cerveau* at the Institut Henri Poincaré in Paris. In preparing the address, I had chosen the theme that, although different specific models may be appropriate for different circuits, subsystems or functionalities of the brain, no single model will suffice for the challenges of building an integrative computational approach appropriate to cognitive neuroscience. It thus came as an amazing coincidence to discover that the alleyway next to Hotel Jack's, where I had been given a room by the School's organizers, was called "Impasse du Petit Modèle", the very theme of my talk. Yet in asserting *the dead end of the little model*, I am flying in the face of the observation that much of the best science is done by focusing on a very narrow problem which nonetheless promises to shed light on a large range of topics. Nonetheless, further progress in the Simulation of Adaptive Behavior demands a "balancing act" between large scale models and focused research, as demonstrated by the "big model" of neurolinguistics inspired by the Mirror System Hypothesis, which can be used to provide a framework for a number of more focused efforts in computational neuroscience and biologically-inspired robotics.

References

1. Wiener, N.: Cybernetics: or Control and Communication in the Animal and the Machine. The Technology Press and John Wiley & Sons, New York (1948)
2. McCulloch, W.S., Pitts, W.H.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. 5, 115–133 (1943)
3. Turing, A.: On computable numbers, with an application to the Entscheidungsproblem. In: Proceedings of the London Mathematical Society, vol. 2(42), pp. 230–265 (1936)
4. Arbib, M.A.: Turing Machines, Finite Automata, and Neural Nets. J. Assoc. Computing Machinery 8, 467–475 (1961)
5. Lettvin, J.Y., et al.: What the frog's eye tells the frog brain. Proceedings of the IRE 47, 1940–1951 (1959)
6. Pitts, W.H., McCulloch, W.S.: How we know universals, the perception of auditory and visual forms. Bulletin of Mathematical Biophysics 9, 127–147 (1947)
7. Arbib, M.A.: Brains, Machines and Mathematics. McGraw-Hill, New York (1964)
8. Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular and functional architecture in the cat's visual cortex. Journal of Physiology (London) 160, 106–154 (1962)

9. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychol. Rev. 65, 386–408 (1958)
10. Hebb, D.O.: The Organization of Behavior. John Wiley & Sons, New York (1949)
11. von Neumann, J.: Probabilistic logics and the synthesis of reliable organisms from unreliable components. In: Shannon, C.E., McCarthy, J. (eds.) Automata Studies, pp. 43–98. Princeton University Press, Princeton (1956)
12. Shannon, C.E., Weaver, W.: The mathematical theory of communication. University of Illinois Press, Urbana (1949)
13. Greene, P.H.: On looking for neural networks and "cell assemblies" that underlie behavior: I. A mathematical model. Bulletin of Mathematical Biology 24, 247–275 (1962)
14. Miller, G.A., Galanter, E., Pribram, K.H.: Plans and the Structure of Behavior. Holt, Rinehart & Winston (1960)
15. Samuel, A.L.: Some Studies in Machine Learning Using the Game of Checkers. IBM. J. Res. and Dev. 3, 210–229 (1959)
16. Gödel, K.: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I. Monatshefte für Mathematik und Physik 38 (1931)
17. Rall, W.: Theoretical significance of dendritic trees for neuronal input-output relations. In: Reiss, R. (ed.) Neural Theory and Modelling, pp. 73–97. Stanford University Press, Stanford (1964)
18. Ingle, D.J.: Visual releasers of prey catching behaviour in frogs and toads. Brain, Behav., Evol. 1, 500–518 (1968)
19. Didday, R.L.: The Simulation and Modelling of Distributed Information Processing in the Frog Visual System. Stanford University, Stanford (1970)
20. Lara, R., Arbib, M.A.: A model of the neural mechanisms responsible for pattern recognition and stimulus specific habituation in toads. Biol. Cybern. 51, 223–237 (1985)
21. House, D.H.: Depth Perception in Frogs and Toads: A Study in Neural Computing. Lecture Notes in Biomathematics, vol. 80. Springer, Heidelberg (1989)
22. Wang, D., Arbib, M.A.: Modeling the dishabituation hierarchy: the role of the primordial hippocampus. Biol. Cybern. 67, 535–544 (1992)
23. Corbacho, F., et al.: Schema-based learning of adaptable and flexible prey-catching in anurans I. The basic architecture. Biol. Cybern. 93, 391–409 (2005)
24. Arkin, R.C.: Neuroscience in motion: the application of schema theory to mobile robotics. In: Ewert, J.-P., Arbib, M.A. (eds.) Visuomotor Coordination: Amphibians, Comparisons, Models, and Robots, pp. 649–671. Plenum Press, New York (1989)
25. Arbib, M.A., Hesse, M.B.: The Construction of Reality. Cambridge University Press, Cambridge (1986)
26. Draper, B.A., et al.: The schema system. International Journal of Computer Vision 2, 209–250 (1989)
27. Arbib, M.A.: The Metaphorical Brain: An Introduction to Cybernetics as Artificial Intelligence and Brain Theory. Wiley-Interscience, New York (1972)
28. Walter, W.G.: The Living Brain. Duckworth, London (1953)
29. Braitenberg, V.: Vehicles: Experiments in Synthetic Psychology. Bradford Books/The MIT Press (1984)
30. Dev, P.: Perception of Depth Surfaces in Random-dot Stereograms: A Neural Model. Int. J. Man-Machine Studies 7, 511–528 (1975)
31. Boylls, C.C.: Synergies and cerebellar function. In: Szentágothai, J., Arbib, M.A. (eds.) Conceptual Models of Neural Organization, pp. 157–165. The MIT Press, Cambridge (1975)

32. Lieblich, I., Arbib, M.A.: Multiple Representations of Space Underlying Behavior. The Behavioral and Brain Sciences 5, 627–659 (1982)
33. Guazzelli, A., et al.: Affordances, Motivation, and the World Graph Theory. Adaptive Behavior 6, 435–471 (1998)
34. Jeannerod, M., Biguer, B.: Visuomotor mechanisms in reaching within extra-personal space. In: Ingle, D.J., Mansfield, R.J.W., Goodale, M.A. (eds.) Advances in the Analysis of Visual Behavior, pp. 387–409. The MIT Press, Cambridge (1982)
35. Arbib, M.A.: Perceptual structures and distributed motor control. In: Brooks, V.B. (ed.) Handbook of Physiol-ogy — The Nervous System II. Motor Control, pp. 1449–1480. American Physiological Society, Bethesda (1981)
36. Iberall, T., Bingham, G., Arbib, M.A.: Opposition Space as a Structuring Concept for the Analysis of Skilled Hand Movements. In: Heuer, H., Fromm, C. (eds.) Generation and Modulation of Action Patterns, pp. 158–173. Springer, Heidelberg (1986)
37. Arbib, M.A., Iberall, T., Lyons, D.: Coordinated control programs for control of the hands. In: Goodwin, A.W., Darian-Smith, I. (eds.) Hand Function and the Neocortex, pp. 111–129. Springer, Berlin (1985)
38. Arbib, M.A., Caplan, D.: Neurolinguistics must be Computational. Behavioral and Brain Sciences 2, 449–483 (1979)
39. Arbib, M.A., Conklin, E.J., Hill, J.C.: From Schema Theory to Language. Oxford University Press, New York (1987)
40. Hoff, B., Arbib, M.A.: Models of Trajectory Formation and Temporal Interaction of Reach and Grasp. J. Mot. Behav. 25, 175–192 (1993)
41. Fagg, A., Arbib, M.: Modeling parietal-premotor interactions in primate control of grasping. Neural Netw. 11, 1277–1303 (1998)
42. Rizzolatti, G., et al.: Premotor cortex and the recognition of motor actions. Cognitive Brain Research 3, 131–141 (1996)
43. Oztop, E., Arbib, M.A.: Schema design and implementation of the grasp-related mirror neuron system. Biol. Cybern. 87, 116–140 (2002)
44. Arbib, M.A., Bota, M.: Language Evolution: Neural Homologies and Neuroinformatics. Neural Networks 16, 1237–1260 (2003)
45. Arbib, M.A., Fagg, A.H., Grafton, S.T.: Synthetic PET Imaging for Grasping: From Primate Neurophysiology to Human Behavior. In: Sommer, F.T., Wichert, A. (eds.) Exploratory Analysis and Data Modeling in Functional Neuroimaging, pp. 232–250. The MIT Press, Cambridge (2003)
46. Arbib, M.A., et al.: Synthetic PET: analyzing large-scale properties of neural networks. Human Brain Mapping 2, 225–233 (1995)
47. Fagg, A.H., Arbib, M.A.: Modeling parietal-premotor interactions in primate control of grasping. Neural Netw. 11, 1277–1303 (1998)
48. Gallese, V., et al.: Action recognition in the premotor cortex. Brain 119, 593–609 (1996)
49. Bonaiuto, J., Rosta, E., Arbib, M.A.: Extending the mirror neuron system model, I: Audible actions and invisible grasps. Biol. Cybern. 96, 9–38 (2007)
50. Arbib, M.A., et al.: Tool use and the distalization of the end-effector. Psychol. Res. 73, 441–462 (2009)
51. Houghton, G., Hartley, T.: Parallel Models of Serial Behavior: Lashley Revisited. Psyche. 2 (1995)
52. Rhodes, B.J., et al.: Learning and production of movement sequences: behavioral, neurophysiological, and modeling perspectives. Hum. Mov. Sci. 23, 699–746 (2004)
53. Bonaiuto, J., Arbib, M.A.: Extending the mirror neuron system model, II: What did I just do? A new role for mirror neurons. Biological Cybernetics 102, 341–359 (2010)

54. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Cambridge (1998)
55. Grafton, S.T., et al.: Localization of grasp representations in humans by positron emission tomography. 2. Observation compared with imagination. Exp. Brain Res. 112, 103–111 (1996)
56. Emmorey, K.: Language, Cognition, and the Brain: Insights from Sign Language Research. Lawrence Erlbaum and Associates, Mahwah (2002)
57. Rizzolatti, G., Arbib, M.A.: Language within our grasp. Trends Neurosci. 21, 188–194 (1998)
58. Myowa-Yamakoshi, M., Matsuzawa, T.: Factors influencing imitation of manipulatory actions in chimpanzees (Pan troglodytes). J. Comp. Psychol. 113, 128–136 (1999)
59. Arbib, M.A.: From Monkey-like Action Recognition to Human Language: An Evolutionary Framework for Neurolinguistics (with commentaries and author's response). Behavioral and Brain Sciences 28, 105–167 (2005)
60. Arbib, M.A.: Aphasia, apraxia and the evolution of the language-ready brain. Aphasiology 20, 1–30 (2006)
61. Hickok, G., Poeppel, D.: Dorsal and ventral streams: a framework for understanding aspects of the functional anatomy of language. Cognition 92, 67–99 (2004)
62. Croft, W., Cruse, D.A.: Cognitive Linguistics. Cambridge University Press, Cambridge (2005)
63. Kemmerer, D.: Action verbs, argument structure constructions, and the mirror neuron system. In: Arbib, M.A. (ed.) Action to Language via the Mirror Neuron System, pp. 347–373. Cambridge University Press, Cambridge (2006)
64. Itti, L., Arbib, M.A.: Attention and the minimal subscene. In: Arbib, M.A. (ed.) Action to Language via the Mirror Neuron System, pp. 289–346. Cambridge University Press, Cambridge (2009)
65. Arbib, M.A., Lee, J.: Describing visual scenes: Towards a neurolinguistics based on construction grammar. Brain Research 1225, 146–162 (2008)
66. Arbib, M.A., Lee, J.: Vision and Action in the Language-Ready Brain: From Mirror Neurons to SemRep. In: Mele, F., Ramella, G., Santillo, S., Ventriglia, F. (eds.) BVAI 2007. LNCS, vol. 4729, pp. 104–123. Springer, Heidelberg (2007)
67. De Beule, J., Steels, L.: Hierarchy in Fluid Construction Grammar. In: Furbach, U. (ed.) KI 2005. LNCS (LNAI), vol. 3698, pp. 1–15. Springer, Heidelberg (2005)
68. Bergen, B.K., Chang, N.: Embodied Construction Grammar in Simulation-Based Language Understanding. In: Östman, J.-O., Fried, M. (eds.) Construction grammar(s): Cognitive and Cross-Language Dimensions, pp. 147–190. John Benjamins, Amsterdam (2005)

How Virtual Machinery Can Bridge the "Explanatory Gap", in Natural and Artificial Systems

Aaron Sloman

School of Computer Science,
University of Birmingham, UK, B15 2TT
http://www.cs.bham.ac.uk/~axs

Abstract. We can now show in principle how evolution could have produced the "mysterious" aspects of consciousness if, like engineers in the last six or seven decades, it had to solve increasingly complex problems of representation and control by producing systems with increasingly abstract, but effective, mechanisms, including self-observation capabilities, implemented in non-physical virtual machines which, in turn, are implemented in lower level physical mechanisms. For this, evolution would have had to produce far more complex virtual machines than human engineers have so far managed, but the key idea might be the same. However it is not yet clear whether the biological virtual machines could have been implemented in the kind of discrete technology used in computers as we know them.

Keywords: Architecture, Body, Causation, Cognition, Consciousness, Darwin, Designer Stance, Evolution, Explanatory Gap, Huxley, Mind, Virtual Machinery.

1 Introduction: A Problem for Darwin

A problem that puzzled Darwin and fired up his critics, was how mental phenomena could fit into the theory of evolution by natural selection.

There was evidence for evolution of physical forms, including: fossil records showing gradual changes in skeletal structures, existing species that have been shown to adapt physical features and behaviours to meet changing circumstances, and artificially bred variants of animals and plants. Such evidence convinced Darwin and many of his contemporaries that random mutations and environmental selection pressures could, over time, produce radical changes. Despite gaps in the evidence, most scientists now seem to accept that the whole process, starting from complex molecules and leading to the existence of all the biological diversity now on earth, can be explained in roughly Darwinian terms – though there remain disagreements on some of the mechanisms, e.g. in [11].

However, since Darwin's time till the present day, many serious thinkers, including some of his leading supporters, have doubted that there is compelling evidence for the claim that *mental* functioning evolved in the same way, and

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 13–24, 2010.

some think there is no evidence that human minds, or other animal minds, could be products of Darwinian evolution. For some, that is because all the evidence available supports only the hypothesis that evolutionary mechanisms can produce *physical* changes of shape and detailed physiology and to some extent also physical behaviour, such as feeding behaviours, mating behaviours, caring for offspring, and ways of escaping from predators.

If it were clear how physical changes can produce mental changes, that could help to provide an account of how evolution could produce new mental phenomena, and the behaviours that seem to require specifically human mental processes, such as production and enjoyment of poetry, stories, music and dancing, and the advance of mathematics, science and technology. For it could be argued that evolution can produce mental phenomena by producing the required physical mechanisms. But many cannot conceive of physical matter producing mental processes and some even deny that it is possible at all, because of the huge gulf in kind between, on the one hand, behaviours of atoms, molecules, and larger structures composed of those, and, on the other hand, processes of experience, thought, pleasure, pain, and self-awareness.

Even one of Darwin's strongest supporters, T.H. Huxley, is widely reported to have written "How it is that anything so remarkable as a state of consciousness comes about as a result of irritating nervous tissue, is just as unaccountable as the appearance of the Djin when Aladdin rubbed his lamp"[1]. He was not alone. Romanes wrote in [18] "But we are totally in the dark as to the causal connection, if any, between such a state of turmoil in the ganglion and the occurrence of consciousness."(p75) (quoted in Whittaker's review [27]). Moreover, Alfred Wallace, co-inventor of the theory of evolution by natural selection, doubted that evolution could produce anything like *states of consciousness*.

This problem was later labelled the "explanatory gap". Individuals use different names for what it is that they are opposing to physical phenomena. Huxley and Romanes used "consciousness". Some use "sentience". Following Block [3], many now refer to "Phenomenal Consciousness" (PC) in contrast with "Access Consciousness" (AC), or, in the terminology of Chalmers [4], distinguish the so-called "Hard Problem" of consciousness from a (relatively) "Easy Problem". Such formulations presuppose a dichotomy: a binary divide between things that do and things that do not have the problematic extra feature over and above their physical features. Later we shall challenge the use of a binary division (as Thomas Whittaker did as long ago as 1884).

Debates about evolution of mind echoed and extended older philosophical discussions about the nature of mind and the relations between mind and body. Not only philosophers, but also psychologists, neuroscientists, physicists, anthropologists, biologists and more recently AI researchers, roboticists and cognitive scientists of various sorts have all been exercised about this. Responses to the questions about whether natural selection can produce mental competences and consciousness, and whether physical processes can produce mental processes,

[1] The source is alleged to be [10], though I could not find the words in an online version.

vary. The variations include: rejection of the problem as somehow due to a deep muddle [19], claiming that it is a real problem but lacking any solution that human minds can understand [13], offering a reformulation of the problem alleged to solve it [7], resurrecting the problem with a new label [14,3,4], proposing a philosophical or scientific research project to solve it [2,5], offering specific solutions that appeal to recent advances in physics or mathematics [15,20], assembling experimental and observational data about it [2], producing working computer models of various kinds [21], developing new technical philosophical concepts in the hope of clarifying it [12], and many more. In 1978, I proposed that the best way to make progress on the philosophical problems was to use new opportunities provided by the development of computers to investigate ways of designing working minds possibly starting with working fragments [22, Chapter 1]. This has been ignored by most philosophers and the majority of psychologists. Those who tried have generally underestimated the problems, expecting success too soon.

Nevertheless, much has been learnt, including a great deal about the diversity of the phenomena involving consciousness and other aspects of mentality. This includes studies of development of various mental competences from infancy, e.g. [9,17], various "disorders of consciousness" caused by brain-damage, physical or mental abuse, and the effects of drugs of various kinds, including local and global anaesthetics used medically.

2 Causes of Difficulty

Despite the vast amount in print, there does not seem to be any clear consensus that one theory of the relationship between mind and brain is right, or even along the right lines. I think there are two main reasons why philosophy has got stuck, and with it theoretical psychology and biology.

The first hurdle is the difficulty of identifying the problems in a form that genuinely addresses all the main concerns that have arisen in the history pointed to here. Part of the explanation for the difficulty is that there is no *one* problem: rather biological evolution had to solve *many* design problems in the long slow march from microbes to species clearly demonstrating mental competences. This is related to the claim in [24,26] that there is no one thing referred to by the noun "consciousness", because the concept of being conscious of something is highly polymorphic – it refers to many different types of phenomena, which vary according to which kind of thing is conscious and what kind of thing it is conscious of. For example being conscious of a pain in your leg is very different from being conscious of your growing unpopularity at work. Both are different from a fly escaping a swatter because it was conscious of the motion. In a biological context this diversity is to be expected: diversity is a pervasive aspect of products of evolution.

The second reason for lack of progress is that most researchers lack conceptual tools adequate for the task of formulating answers with explanatory power. I shall try to show that people working in computer science and software engineering have, over several decades, unintentionally provided new concepts and tools for thinking about, modelling and explaining some of the kinds of phenomena cited by objectors to Darwinism. The key idea is that all organisms use information:

living things are *informed control systems* – they use information in making control decisions, i.e. selecting between alternative possible actions, internal or internal. This idea is not new. But it is often ignored by people who ask how matter can produce or influence mind without asking how mind can influence matter and its motion, which it clearly does, e.g. as I type these words.

Early versions of this claim are in books by Craik and Wiener, [6,28] published in 1943 and 1948. But we have learnt much since then. In particular, whereas early information-based machines (e.g. Jacquard looms) used information to control *physical* actions, the information-processing machinery developed using computers has increasingly been concerned with acting *on information*, and acting *on abstract mechanisms for acting on information* (a possibility Ada Lovelace foresaw). Most of what computers now do is not describable in the language of physics: they run processes in *virtual machines* that are implemented in physical machines though what the virtual machines are and what they do cannot be fully described using the language of the physical sciences. For example, the concepts of "winning" and "losing", required to describe the operation of a computer-based chess program, refer not to physical processes but to more abstract interactions between competing agents [23].

3 Towards Better Conceptual Tools

The conceptual tools required for building, testing, debugging, extending, comparing and explaining such virtual machines were developed piecemeal over several decades by hardware and software engineers solving different problems. Portions of the story are in [8]. We need to understand what they have achieved and its relevance to information processing in organisms.

It is sometimes suggested that if you describe a computer as running a certain virtual machine, e.g. a chess playing virtual machine, you are merely hallucinating a pattern onto the physical changes occurring, like choosing to see a rectangular array of dots as composed of horizontal rows of dots, or of vertical columns of dots, implying that virtual machines exist only in the eye (or mind) of the beholder, and cannot *do* anything. But that ignores the *causal* interactions that occur within virtual machines and also between virtual and physical processes. Chess virtual machines really do consider the consequences of certain options and on that basis choose one of them in deciding what move to make next – which in turn causes further changes within the virtual machine and in the computer's memory and possibly also on the screen. For blind human users it may utter the coordinates of the move. The causation is not hallucinated.

We may fancy that one piece of shadow on a forest floor chases another, or that an arrow shaped shadow points at a patch of light, whereas in fact there is no chasing or pointing: the visible patterns are mere by-products of interactions between wind, leaves, branches and rays of sunlight. If an arrow-shaped shadow appears to point at a patch of light that is mere coincidence. In contrast, engineers have learnt how to make enduring, active patterns in computers that really do influence other patterns, which may themselves be influencing or referring to other patterns, or even to themselves. We depend on this in flight control systems.

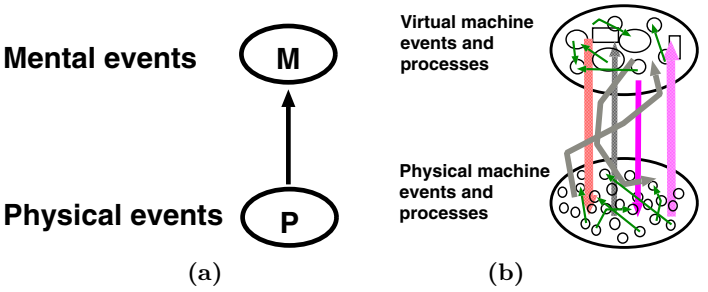


Fig. 1. In (a), illustrating epiphenomenalism, physical changes produce patterns visible to a viewer, which do not themselves interact causally, whereas (b) illustrates two-way interaction between physical mechanisms and mechanisms in virtual machines, which interact both with one another and with the physical substrate and environment

Concurrency of interactions is important, as we'll see, and does not match the common view of a computer as a Turing machine.

In Figure 1, (b) schematically indicates a physical system on which is implemented complex virtual machinery composed of enduring, interacting subsystems which can influence and be influenced by other patterns and physical structures and processes. The causal arrows go up and down as well as sideways. In modern computers, such enduring but changing virtual machinery co-exists with, and influences, underlying physical machinery, *which it helps to control,* even though the virtual machinery is all *fully implemented in deterministic physical machinery.* Rules running in a virtual machine can cause changes in the physical memory and attached devices (including screens, motors, speakers, and network links to computers). The use of "cause" there is justified because so much engineering has gone into making a complex collection of conditional statements true, including counter-factual conditionals about what would or would not have happened in various possible situations. A web of connected hardware, software, and firmware subsystems whose overall structure is very complex, and can even change rapidly between and during causal interactions, makes those conditional statements true, by ensuring that the connections are not mere coincidences. That includes not only causation between processes in virtual machinery, but also causation across levels, e.g. using mechanisms that ensure that a decision taken at a high level causes certain changes in physical memory, or causes appropriate signals to go to an attached device. All that uses very complex technology, unimaginable in Darwin's time. However, engineers can make mistakes, and bugs in the virtual machinery are detected and removed, usually by altering a textual specification that drives the creation of the web.

The technology supporting all that functionality includes (in no significant order): *memory management, paging, cacheing, interfaces of many kinds, interfacing protocols, protocol converters, device drivers, interrupt handlers, schedulers, privilege mechanisms, resource control mechanisms, file-management systems, interpreters, compilers, "run-time systems" for various programming languages, garbage collectors, varied types of data-structure and operations on*

them, debugging tools, pipes, sockets, shared memory systems, firewalls, virus checkers, security systems, network protocols, operating systems, application development systems, name-servers, and more. Concurrency does not require multiple CPUs, since enduring process records in memory allow a paused process to continue to influence running processes through the latter's memory accesses. However, insofar as a computer has sensors and other interfaces connected with the environment there will be many concurrent processes not wholly under the control of the computer, interacting partly as a result of external interrupts.

Some people find it hard to see how virtual machines can cause anything to happen because they fail to make a three-way distinction, between:

1. *Mathematical Models* (MMs), e.g. numbers, sets, grammars, proofs, ...
2. *Physical Machines* (PMs), including atoms, voltages, chemical processes, ...
3. *Running Virtual Machines* (RVMs), e.g. calculations, games, formatting, proving, checking spelling, handling email, self-monitoring, ...

MMs are static abstract structures, like proofs, and axiom systems that never do anything. Unfortunately some uses of "virtual machine" refer to MMs, e.g. "the Java virtual machine". These are abstract, inactive, mathematical entities, not RVMs, whereas PMs and RVMs are active and cause things to happen both internally and in external environments. Millions of computer users use RVMs every day, with little knowledge of what they are using. Different computer scientists and software engineers, know about different sub-sets, and the whole system is rarely described adequately. For instance, Pollock's mostly excellent [16], over-simplifies by frequent references to "the machine table", ignoring the layered implementations used by most application software.

The technology summarised above allows physical machines on our desks to support varying collections of non-physical machinery made up of various kinds of concurrently interacting components whose causal powers operate in parallel with the causal powers of underlying machines, and help to control those physical machines. However, the non-physical (virtual) machinery has *different levels of granularity* and *different kinds of functionality* from the physical machines. The coarser granularity is indispensable for processes of design, testing, debugging, and also for run-time self-monitoring and control, which would be impossible to specify at the level of individual transistors (because of explosive combinatorics, especially with time-sharing). We need to understand the variety of uses of virtual machinery, compared with physical information-processing machinery, including the importance of the coarser granularity, in order to understand the evolutionary pressures that could have produced biological (e.g. brain-based) virtual machines.

4 What Exactly Needs to Be Explained?

Thinkers are divided as to the kind of diversity of mental phenomena found in nature. Some believe that there is a major dichotomy separating conscious organisms and the rest. Others claim there are only differences of *degree*, with gradually increasing sophistication and complexity of mental phenomena

emerging from gradually increasing complexity of the physical substrate. However, claims regarding continuity (in the mathematical sense) in biological evolution are implausible (a) because all biological phenomena are implemented in chemical mechanisms (e.g. using DNA), and there are only *discrete* ways of transforming one molecular structure to another, and (b) the fact that evolutionary changes occur only across generations implies that between any two evolutionary stages there can be only a finite number of intervening stages: which rules out continuous change. Biological changes must be discontinuous, whether small or large. So, instead of seeking a single major dichotomy between conscious organisms and the rest, or hoping to find continuous variation, we need to understand a large collection of discontinuous design changes, with both small and large differences in structure, behaviour and functionality.

Specifying a good conceptual framework for studying that variety is not easy. Evolution produced widely varying physical structures, and also myriad behavioural capabilities and internal information-processing capabilities, with different subsets of capabilities shared between different species. Even humans differ widely. E.g. new-born infants appear to be unable to perceive, think about, or communicate about most of the things older humans can, and some older humans are also limited in their mental capacities by effects of genetic abnormality, injury, disease, or degeneration. There are also differences in mental functioning that come from cultural influences – e.g. whether people can think in Chinese, or whether they can understand harmonic progressions in music.

Whittaker's review of Romanes [27] asks whether mind-like features are present in *all* living things, raising the possibility "that the lowest animals, ... have the beginnings not only of sensibility but also of will and intelligence."[2] His use of "will" implied causal powers. In more neutral language: all organisms are informed control systems. In deploying stored energy, they select between alternatives (e.g. both external and internal behaviours) on the basis of available information. Whittaker also made an important point about the structure of the space of possible minds: "The development of mind is represented as proceeding only in a single line. .... Nothing is said as to the possibility that at the same level of general intelligence there may be essentially different mental types, ..." (page 294). In a note in *Mind* 1984, Romanes claimed Whittaker had misread his meaning, showing that he too assumed non-linear mental diversity.

That diversity is important when we discuss the evolution of mentality, when we try to design and build working models of mentality and when we try to explain the relationships between matter and mind. If many different things evolved, using many different kinds of working mechanism, then that diversity must be explained by any satisfactory answer to the question of how physical phenomena can produce mental phenomena. The variations in design produced by evolution will be responses to variations in pressures, needs and opportunities. We can summarise that by saying there are many different sets of requirements

[2] No doubt the idea has occurred to many people. I was unaware of Whittaker's work when I presented a similar idea in "What Has Life Got To Do With Mind? Or vice versa?": http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#maggiefest

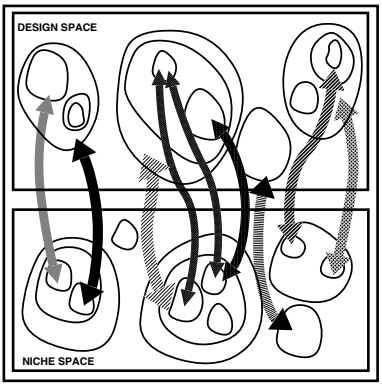


Fig. 2. The space of possible sets of requirements (niches), the space of possible designs, and the complex structural relationships between designs and requirements. Relationships between requirements and designs are often represented as numerical fitness functions. But numerical fitness values cannot do justice to the diversity of ways of satisfying or failing to satisfy a set of requirements. Compare descriptive consumer reports on products.

(i.e. different niches) as well as different designs – satisfying different sets of requirements, as indicated crudely in Figure 2.

We have rejected (a) continuous variation, (b) the use of a dichotomy, and (c) a linear arrangement of types of mentality. What alternatives remain? If we analyse environments in depth, we can systematically develop different combinations of requirements that can arise, for instance, requirements related to the spatial separation of different sources of food and drink, requirements based on different kinds of fixed and changing features of the environment (including obstacles, passage-ways, etc.), different kinds of materials available that can be manipulated to meet various needs, different sorts of food (e.g. food that tries to escape and food that doesn't) different sorts of predators, and different sorts of competitors for the same resources. These (and many other) differences in requirements[3] entail advantages and disadvantages in both physical design features, e.g. strength, speed of motion, camouflage, types of gripper, etc., and also designs for virtual machinery for processing information – e.g. *factual* information about the environment (including other agents), *control* information about what to do when, *meta-information* about information, and *meta-control information* about good and bad ways to acquire, process and use information. Long before human engineers found the need to develop virtual machinery, could biological evolution have responded to similar pressures, and produced solutions to those problems, which we do not yet understand? Seeking evidence will be difficult, but potentially enormously important.

Conceptual tools and engineering advances in the last half century have made a huge difference to our ability to think about these requirements and design options. But we have not yet developed a biologically adequate theory of types

[3] http://www.cs.bham.ac.uk/research/projects/cogaff/misc/creativity-boden.html

of virtual machinery. On the contrary, we are still a long way from that.[4] But we have made progress that was unimaginable by Darwin and his peers.

5 In Defense, and Explanation, of Qualia

Our task is to produce a new characterisation and explanation of the phenomena that led to views about contents of experience that (a) are private to the individual concerned, (b) have a character that is utterly distinct from physical structures and processes, (c) seem to be produced by physical and chemical processes in brains and environments perceived or acted on, yet seem to be of a kind that cannot be produced by physical processes.

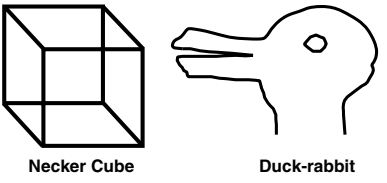


Fig. 3. Each of the two figures is ambiguous and flips between two very different views. (a) can be seen as a 3-D wire frame cube. For most people it flips between two different views of the cube, in which the 3-D locations, orientations and other relationships vary. In (b), the flip involves changes in body parts, the facing direction, and likely motion – requiring a very different ontology.

Despite deep confusions about consciousness and *qualia*, noted by Ryle, Dennett and others, it is clear that such things exist. Although hard to characterise and to identify in other individuals and other species, we need to study examples, to determine requirements for explanatory mechanisms. Their existence and some of their diversity can be demonstrated using the examples in Figure 3. Stare at each of them for a few minutes.[5] When your interpretation of a picture flips, only changes in you occur, apparently involving states and processes that are private to you, and somehow directly accessible by you. Describing carefully how the content changes when the interpretations flip, reveals that very different vocabularies are required, using a purely geometric ontology for (a) and an ontology of animal parts and capabilities in (b). A cube cannot be experienced as "looking to left or to right", whereas a duck, or rabbit can.

These and other phenomena can be used to demonstrate that there are mental states and processes involving mental entities within us even though they cannot be observed by opening up the skull, using the most powerful physical and chemical measuring instruments. How can biological evolution produce states within an organism that are closely tied to sensory input but can change spontaneously and which have semantic content referring to possible external

[4] See http://www.cs.bham.ac.uk/research/projects/cogaff/talks/
[5] See http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#cons09

entities describable only using a complex ontology? Until recently the status of such entities was highly problematic, but I claim that we can now see, at least in outline, how to explain their occurrence: the entities with semantic content occur in a virtual machine that rapidly produces various layers of interpretation of the sensor input using different ontologies, as proposed in [22, Ch 9].

We already know how to produce computing systems that can observe some of their own internal information processes by recording the existence and properties of abstract data-structures that occur, procedures that are followed, and difficulties and successes encountered. Very often the contents of such self-observations are not the *physical* states of the computer components but the components and contents of *virtual machines*. What now need to collect many examples of the types of qualia that might occur in a human-like robot and develop designs that could explain both the occurrence of those cases and their roles in information-processing systems of various kinds. E.g. one of many ways in which it can be useful to attend to your internal data-structures rather than the full structure of perceived entities, is that you can use your internal data-structures to drive a process of communication with others, perhaps using drawings and gestures, to show them what experiences they can expect if they visit certain terrain, for example. Some of the requirements for such mechanisms have been described in presentations on my web site[6], though most of them still leave many problems unsolved and much work to be done.

One of the important facts relating to the diversity of kinds of mind referred to by Whittaker is that not all organisms that have qualia know that they have them! We can separate the *occurrence* of mental contents in an organism from their *detection* by the organism, which requires additional architectural complexity to support self-observation and self-description mechanisms. Many organisms can (I suspect) create and use such entities without having the meta-semantic mechanisms required to detect and represent that fact, as humans do. It is very likely that the vast majority of organisms have very little self-observation capability, despite having conscious (but not self-conscious) contents, used to guide behaviour. We still need to understand why and how evolution produced those that are capable of introspection. For the others it can be said that they have and use potential contents of experience, but they do not experience them! However it is not likely to be a binary divide but a collection of sub-divisions with more or less functionality, depending on the species. I expect we shall need to experiment with a range of increasingly complicated working examples, using different kinds of mechanism, in order to understand better some of the questions to be asked about about mental phenomena in biological organisms. This is very close to Arbib's research programme described in [1].

6 What Next?

Long experience of philosophical debates shows that there are thinkers whose belief in an unbridgeable mind/body explanatory gap will be unshaken by all

[6] http://www.cs.bham.ac.uk/research/projects/cogaff/talks/

this. As argued in [26], some cases of opposition will be based on use of incoherent concepts (e.g. a concept of "phenomenal consciousness" *defined* to involve no causal or functional powers). One way to test these ideas, outlined in [25], is to use the ideas to specify a robot design that starts off with the cognitive capacities of a human infant and develops its own architecture in way that produces human-like growth in cognitive sophistication, including development of introspective capabilities that can be focused on experience of things like Figure 3, until it reaches the stage where it becomes puzzled as to how its own internal states and processes, detected at a virtual machine level, can exist in a physical world.

After reading about philosophy of mind, and having philosophical discussions with others, different individual robots that all start from the same design should be able to develop philosophical beliefs about the nature of the mind-body relationship. However, just as the same language learning capability in infant humans can lead to the use of very different languages, depending on the environment in which the infant grows up, so should the same starting design (robot genome) be able to produce "adult" robots whose philosophical views differ as widely as those of human philosophers, such as Renee Descartes, John Searle, Daniel Dennett and the author of this paper. This would not imply that philosophical beliefs about the nature of mind are merely cultural phenomena (like languages): some of them may be, while others are based both on personal (including robotic) experience, and deep scientific and technical knowledge, which other robots starting from the same initial design may never acquire.

If we, who have designed all the robots, can tell which one is right about how it works, and how its mental processes are related to physical processes in and around it, then that would demonstrate at least the possibility that humans with those theories are also right about how *they* work!

Alas: we are nowhere near being able to build such robots, since current achievements in AI vision, motor-control, concept-formation and other forms of learning, language understanding and use, motive-generation, decision-making, plan-formation, problem-solving, and many other areas are still (mostly) far inferior to those of humans. And if we omit the competences that appear unique to humans, current robots are still far inferior to other animals. No easy way to close those gaps is visible on the horizon. But there are many things to try.

Acknowledgements. A lecture at the University of Birmingham by Joe Cain on 13th October 2009 drew my attention to the extent and importance of scientific puzzlement regarding evolution of mental functions even among supporters of Darwin. I owe thanks to many colleagues, over many years, including Jeremy Wyatt and other members of the EU CogX Cognitive Robotics project.

References

1. Arbib, M.A.: Rana computatrix to Human Language: Towards a Computational Neuroethology of Language Evolution. Philosophical Transactions: Mathematical, Physical and Engineering Sciences 361(1811), 2345–2379 (2003)
2. Baars, B.J.: A cognitive Theory of Consciousness. CUP, Cambridge (1988)

3. Block, N.: On a confusion about the function of consciousness. Behavioral and Brain Sciences 18, 227–247 (1995)
4. Chalmers, D.J.: Facing Up to the Problem of Consciousness. Journal of Consciousness Studies 2(3), 200–219 (1995)
5. Chalmers, D.J.: The Conscious Mind: In Search of a Fundamental Theory. OUP, New York (1996)
6. Craik, K.: The Nature of Explanation. CUP, London (1943)
7. Dennett, D.C.: Consciousness Explained. Penguin, London (1991)
8. Dyson, G.B.: Darwin Among The Machines: The Evolution of Global Intelligence. Addison-Wesley, Reading (1997)
9. Gibson, E.J., Pick, A.D.: An Ecological Approach to Perceptual Learning and Development. OUP, New York (2000)
10. Huxley, T.H.: Lessons in Elementary Physiology. MacMillan & Co., New York (1866)
11. Jablonka, E., Lamb, M.J.: Evolution in Four Dimensions. MIT Press, Cambridge (2005)
12. Kim, J.: Mind in a Physical World. MIT Press, Cambridge (1998)
13. McGinn, C.: Consciousness and Its Objects. OUP, Oxford (2004)
14. Nagel, T.: What is it like to be a bat. In: Hofstadter, D., Dennett, D.C. (eds.) The Mind's I: Fantasies and Reflections on Self and Soul, pp. 391–403. Penguin (1981)
15. Penrose, R.: The Emperor's New Mind: Concerning Computers Minds and the Laws of Physics. OUP, Oxford (1989)
16. Pollock, J.L.: What Am I? Virtual machines and the mind/body problem. Philosophy and Phenomenological Research 76(2), 237–309 (2008), http://philsci-archive.pitt.edu/archive/00003341
17. Rochat, P.: The Infant's World. Harvard University Press, Cambridge (2001)
18. Romanes, G.J.: Mental evolution in animals, K. Paul, Trench, London (1883), http://www.archive.org/details/cihm_16907
19. Ryle, G.: The Concept of Mind. Hutchinson, London (1949)
20. Ryser, P.: Creative Choice: How the Mind Could Causally Affect the Brain. Journal of Consciousness Studies 16(2-3), 6–29 (2009)
21. Shanahan, M.P.: Consciousness, Emotion, & Imagination: A Brain-Inspired Architecture for Cognitive Robotics. In: AISB 2005 Symp. on Next Generation Approaches to Machine Consciousness, pp. 26–35 (2005)
22. Sloman, A.: The Computer Revolution in Philosophy. Harvester Press, Hassocks (1978), http://www.cs.bham.ac.uk/research/cogaff/crp
23. Sloman, A.: What Cognitive Scientists Need to Know about Virtual Machines. In: Taatgen, N.A., van Rijn, H. (eds.) Proc. 31st Ann. Conf. of the Cognitive Science Society, pp. 1210–1215. Cognitive Science Society, Austin (2009), http://www.cs.bham.ac.uk/research/projects/cogaff/09.html#901
24. Sloman, A.: Why the "hard" problem of consciousness is easy and the "easy" problem hard (And how to make progress) (2009), http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#cons09
25. Sloman, A.: An Alternative to Working on Machine Consciousness. Int. J. Machine Consciousness (2010), http://www.cs.bham.ac.uk/research/projects/cogaff/09.html#910
26. Sloman, A.: Phenomenal and Access Consciousness and the "Hard" Problem: A View from the Designer Stance. Int. J. of Machine Consciousness (2010), http://www.cs.bham.ac.uk/research/projects/cogaff/09.html#906
27. Whittaker, T.: Review of G.J.Romanes Mental evolution in animals. Mind 9(34), 291–295 (1884), http://mind.oxfordjournals.org
28. Wiener, N.: Cybernetics: or Control and Communication in the Animal and the Machine, 2nd edn. The MIT Press, Cambridge (1961)

Do Empirical Models of Robot-Environment Interaction Have a Meaning?

Ulrich Nehmzow[1], Phillip J. McKerrow[2], and Steve A. Billings[3]

[1] University of Ulster, UK
[2] University of Wollongong, Australia
[3] University of Sheffield, UK

Abstract. The "meaning" of an empirical model of a physical system such as a mobile robot is an ill-defined concept, and clearly it would strengthen any hypotheses based on empirical models if some formal model verification was possible.

In this paper, we present experiments on empirical modelling of mobile robot operation, in which the interactions of *Scitos G5* and *Magellan Pro* mobile robots with purposefully designed environments are measured and modelled through system identification. The experimental setups chosen were such that we could determine from theoretical considerations what the models should be.

The comparison between the actually obtained empirical models and the theoretically correct solutions demonstrates that, in the experiments conducted, the obtained empirical models are "correct".

1 Empirical Models of Physical Systems

1.1 Definitions

We discuss two types of models of mobile robot behaviour in this paper: theoretical and empirical. We derive the empirical models from measurements made by the robot's sensors while it is performing the behaviour that we wish to model. The empirical models are parameterised, numerical representations of a physical process that is used for guiding physical experiments, making predictions about the behaviour of the physical system under investigation, and to capture the laws that govern the observed behaviour. It is especially this last point — scientific theory — that we are interested in.

[1] argues that the utility of empirical models for *prediction* is limited. In robotics this certainly is true because of the usually chaotic nature of robot-environment interaction [3], which precludes long-term accurate predictions. However, in this paper we demonstrate that empirical models in robotics can help to understand the fundamental laws that govern robot-environment interaction, and demonstrate in four experiments that it is possible to obtain empirical models that reflect the — in this case known — true relationship *accurately*.

We derive the theoretical model from a geometric analysis of the robot's motion. A robot's behaviour is a dynamical system that changes the location of

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 25–37, 2010.

the robot over time. Both model types express the behaviour as a set of equations that establish relationships between sensed variables as the robot moves from one state to another over the course of time. We have chosen four experiments where the theoretical models describe the behaviour in terms of intrinsic properties of the path traversed, such as radius of curvature. If the models are true representations of the robot's behaviour, then the dynamic invariants of the models and the underlying system should be the same. Establishing that the empirical models match the theoretical models gives us confidence to use them as simulation models in visualisations of robot behaviour and as control models to calculate driving commands.

1.2 Model Verification and Validation

[2] have argued that a formal verification ("demonstration of truth") and validation ("demonstration of truth") of empirical models of natural systems is impossible. However, proof exists within a set of axioms, and a statement is considered to be proven if it is consistent with those assumptions.

There is the further dilemma that often more than one empirical model will produce the same output (non-uniqueness of models). When faced with two models that give the same results, we either have to demonstrate their equivalence (as in Section 2.1), demonstrate that one empirical model is a better match to the theoretical model than the other (as in Section 2.4), or choose the simplest and most transparent model. Narmax system identification (Section 1.6) attempts to do the latter when it derives a model from measured data.

1.3 Background: Numerical Modelling of Robot-Environment Interaction

We have shown elsewhere [4,5] that it is possible to express certain input-output relationships pertaining to a (mobile) robot's interaction with its environment in closed mathematical form (for instance sensor-motor couplings — in other words, robot control code — in the form of polynomials [7] [6]). These models contrast to models of the internal structure of the robot, such as for instance the one given in [8], where the mapping from wheel rotation to robot motion is modelled with an odometer calibration matrix plus a steering equation.

To illustrate the main point of this paper, we will look at the example of the wall-following behaviour of a *Magellan Pro* mobile robot. This robot was programmed by a trained roboticist to use its laser range readings over an arc of 180° ahead of it (see figure 1) to achieve a wall-following behaviour (figure 2).

We then obtained a very simple linear polynomial representation of the robot's turning speed ω as a function of two of its laser perceptions, using Armax system identification [12]. This polynomial model is given in equation 1.

$$\omega(t) = 0.119 \ldots - 0.135 * Laser_1(t)^{-1} + 0.258 * Laser_{90}(t)^{-1}. \quad (1)$$

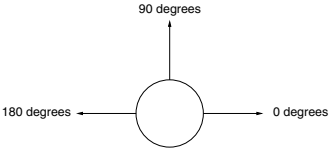


Fig. 1. The Magellan's laser sensor field of view spans 180°, at a resolution of 1°, with direct ahead being 90°

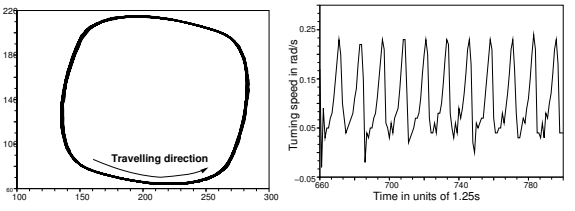


Fig. 2. The MAGELLAN's right-hand wall-following trajectory (left - units are in cm) and turning speed plotted against time (right)

1.4 Modelling Mobile Robot Behaviours

Mobile robots perform tasks that involve motion (velocity or position controlled). They perform those tasks within an environment and, so, their behaviour is influenced and effected by that environment. Thus, when achieving a given task, a robot's behaviour is considered to emerge from the interaction of the robot with the world.

The relationship between the task, the robot and the environment is a robot behaviour that describes the interaction between the robot and the environment as it achieves the task. Usually, more complex environments require more complex behaviours. Our hypothesis is that the interaction can be empirically modelled by a function that can be found by observing the robot's behaviour (figure 3).

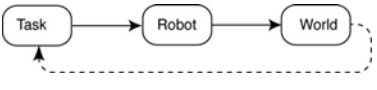


Fig. 3. Relationship between task, robot and environment

The command to carry out a task (figure 3) sets sets the robot in motion. As the robot moves, it interacts with the world, and that interaction results in changes to the robot's motion. After a sequence of motions and interactions the robot completes its task. The arrows in figure 3 indicate the direction of causality The commands to the robot result in its motion, a one-to-one mapping. The inverse process of trying to infer the commands from the motion is potentially a one-to-many process and is much harder to model.

Achieving the task, changes the state of the world. This is a forward process with a one-to-one mapping. Trying to infer the task from the change in the state of the world is an inverse process that may have a one-to-many mapping. Similarly, inferring a robot's location from its sensor data is a many-to-many mapping that we try to fuse into a many-to-one mapping in localisation and calibration [8].

1.5 Mobile Robot Action Behaviours

A mobile robot is a machine that moves in (x, y, θ, t) coordinates. Its motion can be decomposed into several basic behaviours:

1. Translation by a distance Δd in a straight line from point (x_1, y_1, t_1) to (x_2, y_2, t_2).
2. Rotation by an angle $\Delta\theta$ from heading (θ_1, t_1) to (θ_2, t_2).
3. Combined translation and rotation from point to point along a curved path.
4. Linear velocity (v_r, t_1) in the heading (θ_1, t_1).
5. Rotational velocity (ω_r, t_1) around the point (x, y, t_1).
6. Combined linear and angular velocity to follow a trajectory from $(x_1, y_1, \theta_1, t_1)$ to $(x_2, y_2, \theta_2, t_2)$.

Any motion of a mobile robot can be described by a combination of these action behaviours. Higher-level behaviours combine these action behaviours with perception behaviours to enable a mobile robot to interact with the world to perform a task. The design of the following experiments was guided by the aim of producing simple tasks where we:

1. know the correct theoretical model *a priori*,
2. use one action behaviour,
3. test either the action behaviour or control an action behaviour to achieve a perception behaviour, and
4. we modify one of the three components in the Task-Robot-World relationship in a known way and model the relationship between the other two.

In this way, it should be possible to reduce the system to its underlying components.

1.6 Experimental Setup

We used the following four scenarios of robot-environment interaction , in which the mathematical relationship between the variables under consideration was known *a priori* to investigate the question posed in section 1.3:

1. A mobile robot moves in a straight line along a sinusoidal wall, and we model the range reading of the wall-facing sensor as a function of position (Section 2.1).
2. A mobile robot follows a straight wall in a sinusoidal fashion, and again we model the range reading of the wall-facing sensor as a function of position (Section 2.2).

3. A mobile robot is placed at an unknown position in a circular environment, and we model the robot's position (x, y) as a function of the robot's sensory perception (Section 2.3).
4. A mobile robot turns at constant rate while moving forward at constant velocity, which results in a circular trajectory. We model the relationship between $x(t)$ and $y(t)$ (Section 2.4).

Robot and Environment. All experiments were conducted in a 100 m^2 circular robotics arena. The arena is equipped with a motion tracking system which gives the robot's position in 3D (ground truth).

The robot used in the first three experiments was a *Scitos G5* mobile robot (DAX). We used its Hokuyo laser range finder for our experiments.

The robot used in the final experiment was a *Magellan Pro* mobile robot. Like the *Scitos G5*, this robot is equipped with laser and sonar range sensors. We coarse coded the laser readings into 10 sectors (u_1 to u_{10}) by averaging 62 readings for each 24 degree intervals. The robot's two degrees of freedom used here were translational and rotational velocity.

Empirical models of the form given in equation 1 were obtained using the Narmax system identification process [9], [10], [11].

2 Robot Experiments

2.1 Experiment 1: Following a Sinusoidal Wall

In the first experiment we drove DAX in a straight line along a sinusoidal wall (Figure 4).

We then collected range data from the robot's wall-facing laser range sensor during 13 traversals of the route (figure 4). We removed the DC component (mean) from the logged data and subsampled it so that 10 sampling points per period remained, and modelled it using an Armax process [12].

Knowing that the wall is — as much as experimentally possible — sinusoidal, we expect to obtain a model of the form $d(t) \propto sin(t)$. The model we actually

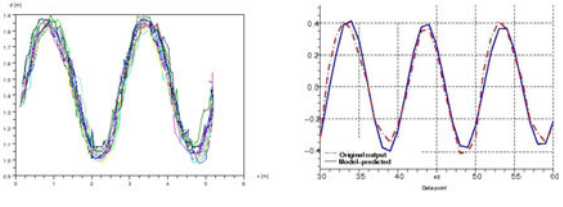


Fig. 4. Left: Experimental setup for the first experiment: the robot drives in a straight line along a sinusoidal wall. Middle: Range data logged in the first experiment. Right: Actually observed sensor reading and time series generated by equation 2 (experiment 1).

obtained is given in equation 2, and appears to have a different form at first sight.

$$d(t) = 0.9 * d(t-1) + 0.14 * d(t-2) - 0.7 * d(t-3), \tag{2}$$

where *d(t)* is the perceived distance to the wall at time *t*.

However, under the assumption that $d(t) = sin(t)$ and for a sufficiently high sampling rate (*i.e.* small Δt) equation 2 can be rewritten as shown in equation 3.

$$\begin{aligned} d(t) &= .9\, d(t-1) + .14\, d(t-2) - .7\, d(t-3) \\ &= .9\, sin(t-\Delta t) + .14\, sin(t-2\Delta t) - .7\, sin(t-3\Delta t) \\ &= .9\,(sin\, t cos\, \Delta t - cos\, t sin\, \Delta t) \\ &+.14\,(sin\, t cos\, 2\Delta t - cos\, t sin\, 2\Delta t) \\ &-.7(sin\, t cos\, 3\Delta t - cos\, t sin\, 3\Delta t) \\ &\approx .9\, sin\, t + .14\, sin\, t - .7\, sin\, t \\ &= 0.34\, sin\, t. \end{aligned} \tag{3}$$

Equation 3 therefore confirms our expectation that the empirical model reflects the *true* relationship between the modelled variables. However, it also illustrates that the output from the modelling process may not resemble the *a priori* model. In such cases, as in this example, it is unwise to immediately jump to the conclusion that the empirical model is not "correct". We may, as here, need to reformulate the equations first. When the empirical model (Equation 4) is used to predict the behaviour we get a close fit to the measured behaviour (Figure 4).

2.2 Experiment 2: Sinusoidally Following a Straight Wall

The second experiment is essentially a variant of experiment 1, this time however the environment is "simple" while the robot's behaviour is more "complex": the robot followed a straight wall, whilst its steering commands were governed by the mathematical sine function. This experimental setup is shown in figure 5.

We logged the range reading of the robot's laser range sensors during 9 passes along the straight wall. As the robot turned, the sensor pointing at the wall

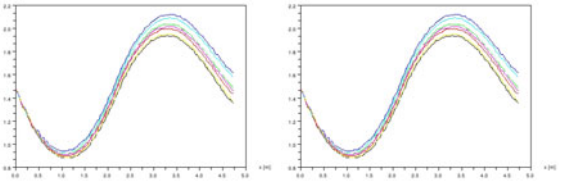


Fig. 5. Left: Experimental setup for the second experiment: the robot drives in a sinusoidal line along a straight wall. Middle: The robot's perceived distance to the wall in the second experiment. Right:Actual range data, and model-predicted range data (experiment 2.)

changed, so we selected the minimum range reading as the distance to the wall. The robot's perceived distance to the wall is shown in figure 5. As before, we modelled the perceived distance d as a function of t, expecting a relationship $d(t) \propto sin(t)$. The empirical model obtained is given in equation 4.

$$d(t) = 1.6 * d(t-1) - d(t-2). \tag{4}$$

Rewritten (equation 3) this becomes $d(t) \approx 0.6 * sin\, t$, again confirming our expectation. Figure 5 demonstrated the numerical agreement between model-predicted and actually observed range data.

2.3 Experiment 3: Self-localisation in a Circular Arena

In our third experiment we placed DAX in a circular arena (figure 6). Here, we were interested in modelling x and y (defined in figure 6) as a function of the robot's laser range readings. The theoretical solution, derived from figure 6, is given in equation 5.

$$\begin{aligned} 2y_c &= D_f + D_b \\ y_r + D_f &= y_c \\ y_r - D_b = -y_c &\rightarrow D_b - y_r = y_c \\ \rightarrow y_r + D_f = D_b - y_r &\rightarrow 2y_r = D_b - D_f \\ \rightarrow y_r &= 0.5 * D_b - 0.5 * D_f, \end{aligned} \tag{5}$$

with D_f and D_b being the range readings at the front and back of the robot respectively. Analogously follows

$$x_r = 0.5 * D_{left} - 0.5 * D_{right}. \tag{6}$$

Having logged *all* laser range readings $\boldsymbol{L}$ over the entire arc of 240° available, we determined the model of the robot's (x_r, y_r) position as a function of $\boldsymbol{L}$ given in equation 7. This model is very close to the ground truth established by the theoretical considerations of equations 5 and 6. The empirical model uses range readings $D_{93°}$ in terms of the angle of the laser beam to the front direction of the robot. So $D_{6°}$ in Equation 7 is equivalent to D_{right} in Equation 6.

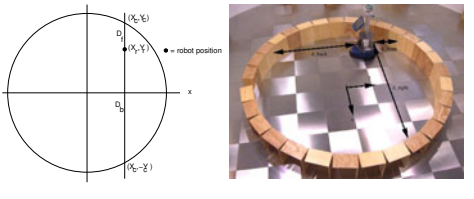


Fig. 6. Experiment 3: Definitions and Experimental Setup

$$x_r = 0.02m - 0.50 * D_{6^\circ} + 0.49 * D_{183^\circ} \qquad (7)$$
$$y_r = 0.02m - 0.51 * D_{93^\circ} + 0.49 * D_{273^\circ}.$$

2.4 Experiment 4

The previous experiments have demonstrated that in many cases system identification will result in empirical models that capture the *true* theoretical relationship between model inputs and outputs. The final experiment will confirm this observation, but show that model structure identification is an essential component of the modelling process, and that selecting an inappropriate structure may result in good approximations, but not *true* models. Modelling $y(t) = f(x(t))$ we establish that the correct empirical model can be found, but only if knowledge about the experimental setup is taken into account.

Experimental Setup. In this experiment a *Magellan Pro* robot moved with a constant translational velocity of 0.15 m/s, while constantly turning at an angular speed of 0.1 rad/s. The resulting trajectory of the robot, shown in figure 7, should be a sector of a circle.

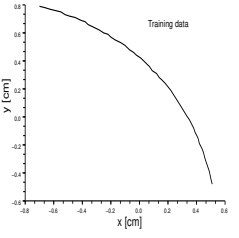


Fig. 7. Logged trajectory of the robot, moving at a constant forward speed of 15 cm/s while turning constantly at 1rad/s

Without any consideration of the physical properties of the experiment, the data shown in figure 7 can be modelled, and results in the model given in equation 8.

$$\begin{aligned} y(t) = 0.36 - 0.78x(t) - 0.05x(t-1) \\ -0.7x^2(t) - 0.08x^2(t-1) \\ +0.25y(t-1) - 0.37y^2(t-1). \end{aligned} \qquad (8)$$

This empirical model (equation 8) matches the data of figure 7 reasonably, although not particularly well (figure 8) — it is not the *true* model of this experiment!

In fact, it is obvious that the trajectory of a robot moving at a constant velocity whilst turning at a constant rate will be a circle. That is, the theoretical model is a circle with an origin and a radius. Taking this additional knowledge into account, a better (in the sense of truer) model can be obtained.

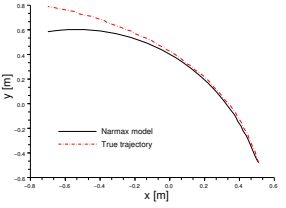


Fig. 8. The model given in equation 8 will explain the robot's trajectory to some degree, but it does not explain the *true* relationship between x and y (experiment 4)

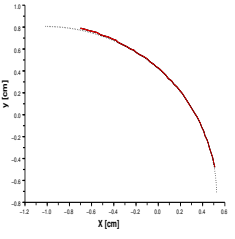


Fig. 9. The robot's trajectory (thick line) fits the circle $(x+1.02)^2+(y+0.74)^2 = 1.55^2$ (dotted line). Origin and radius of circle estimated by graphical interpolation.

Using a graphical method, we estimate that the robot's trajectory lies on a circle of $(x + 1.02)^2 + (y + e)^2 = 1.55^2$, see figure 9.

Exploiting this information, can obtain a model $(y + 0.74)^2 = f((x + 1.02)^2)$ (instead of $y = f(x)$), obtaining the model given in equation 9, which represents the correct relationship between x and y as given in figure 9.

$$\begin{aligned} (y + 0.74)^2 &= 2.39 - 0.999(x + 1.02)^2 \\ \rightarrow (y + 0.74)^2 &+ 0.999(x + 1.02)^2 = 1.55^2 \end{aligned} \tag{9}$$

To obtain the model given in equation 9 we estimated the origin of the circle graphically, which is imprecise, and in fact not necessary. Knowing by experimental consideration that the trajectory lies on a circle is sufficient to identify origin *and* radius of the circle correctly, using an Armax process.

Knowing that the trajectory shown in figure 7 lies on a circle segment, it obeys equation 10.

$$\begin{aligned} (x - c)^2 + (y - d)^2 &= r^2 \\ \rightarrow x^2 - 2cx + c^2 + y^2 - 2dy + d^2 &= r^2 \\ \rightarrow y = -\frac{r^2 - c^2 - d^2}{2d} &+ \frac{x^2}{2d} + \frac{y^2}{2d} + \frac{cx}{d} \end{aligned} \tag{10}$$

Identifying equation 10 as the appropriate Armax process $y = f(x, x^2, y^2)$ we obtain the empirical model given in equation 11

$$y = -1.39x - 0.64x^2 - 0.67y^2 + 0.54, \tag{11}$$

Equations 10 and 11 can be rewritten as shown in (12) to give the final empirical model in equation 13, which indeed agrees with our earlier graphical estimate (figure 9).

$$\frac{1}{2d} = -0.64 \, and \, \frac{1}{2d} = -0.67 \rightarrow d \approx -0.76, \tag{12}$$
$$\frac{c}{d} = -1.39 \rightarrow c = -1.06,$$
$$\frac{r^2 - 1.06^2 - 0.76^2}{2 * 0.76} = 0.54 \rightarrow r^2 = 2.52.$$

$$(x + 1.06)^2 + (y + 0.76)^2 = 1.59^2 \tag{13}$$

2.5 Wall-Following Revisited

Returning to the model of a Magellan's wall-following behaviour, introduced earlier in equation 1, we will now analyse that model in order to understand better how the wall-following behaviour actually works, and whether it is in fact a wall-following behaviour.

The model is given again in table 1, together with each term's sensitivity index $S_i = a_i \frac{\sigma_i}{\sigma_\omega}$, with a_i being each term's coefficient, and σ_i the standard deviation of the sensor signal used in term i. σ_ω is the standard deviation of the model output, the robot's turning speed.

Table 1. Model of wall-following behaviour and normalised sensitivity indices (SI) for each model term

Term	SI	Explanation
$+0.119$		Bias left turn
$-0.135 * \mathrm{Laser}_1(\mathrm{t})^{-1}$	40.4%	
$+0.258 * \mathrm{Laser}_{90}(\mathrm{t})^{-1}$	59.6%	Left turn near obstacle

There are a number of conclusions concerning the robot's wall-following behaviour we can draw from the model:

- The behaviour is not "symmetrical", such as for instance a balancing between attractive and repulsive forces. Instead, the robot will turn left at 0.119 rad/s in the absence of any sensor perception. Thus, the constant term indicates that the robot exhibits an anti-clockwise circle following behaviour.
- Although all 180 laser readings were used as input to the system identification step, only a few sensors from the right hemisphere of the robot were selected. This is in accordance with what one would expect: if following a right-hand wall, the only sensors that matter are towards the right and straight ahead.

– There are only two sensors needed to model the robot's apparent wall-following behaviour: the sensor straight to the front (Laser_{90}), and the sensor straight to the right (Laser_1). The negative value of term 2 (Laser_1) causes the robot to turn to the right, with velocity increasing the smaller the range to the wall. The third term causes the robot to turn left, with velocity increasing as the free space ahead decreases.
– Having thus identified the behaviour shown in figure 2, one can see that it does not actually implement a behaviour comprising a wall-seeking and an obstacle-avoiding behaviour, but instead implements a robot that turns always left, unless a compensating right-turn component results in a straight motion. This right-turn compensation term is surprising, in that it increases as the robot gets closer to a right-hand wall. However, a collision with the wall is avoided, because Laser_{90}, which looks straight ahead, will produce a left-turn component as it gets closer to obstacles. This intricate balance between left-turn bias and compensating right-turn and left-turn components explains why the trajectory in figure 2 is not parallel to the wall and follows a rounded square path even though the room is rectangular.
– Analysing the significance indices, one can see that both laser sensors are highly important, and needed to implement this behaviour.

3 Discussion

> "I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind." [13]

Empirical modelling of robot-environment interaction is an important tool for the roboticist, as is identifying the underlying dynamics, the veracity of the model being a key issue. Unlike opaque modelling methods, transparent modelling methods like Narmax are capable of identifying *true* input-output relationships [17,16], and are therefore particularly interesting for a formal analysis of robot-environment interaction. Most modelling methods are well able to allow predictions of system states, but fail to model the system's underlying dynamics. Often, all they do is fit measurements to a curve with a small mean squared error. Narmax methods have been shown to capture the dynamics of physical systems (qualitative validation, [14,15]), and ongoing work in our laboratories investigates this property with respect to mobile robot dynamics. Our experiments indicate that we can capture the *true* relationship between input and output variables, as well as an agreement between their numerical values.

Example Finally, we show that our modelling approach identifies functional components of a robot's behaviour, and allows us to identify the "true" underlying behaviours giving rise to the observed behaviour. In the case of the

wall-following robot (figure 2), for example, the observed trajectory suggests a wall-following behaviour. Analysis of the model (table 1) reveals that actually the robot performs a continuous left turn, which is compensated for by a right turn when no obstacles are present, resulting in a straight motion. The "wall-following" of figure 2, therefore, is not achieved by a balance between seeking and avoiding a wall, but by a continuous left turn which is corrected into a straight-line movement when there are no obstacles in front of the robot.

This example demonstrates that meaning can be assigned to the parameters of empirical models of robot behaviour. Also, it shows the importance of that understanding to analysing robot behaviours correctly.

Acknowledgements. The research presented in this paper was supported by the British Engineering and Physical Sciences Research Council under grant GR/S30955/01, we acknowledge this support gratefully. We also thank Christoph Weinrich of the Technical University Ilmenau, who carried out the experimental work reported in this paper, and Otar Akanyeti for his constructive comments on this work.

References

1. Oreskes, N.: The role of quantitative models in science. In: Canham, C.D., Cole, J.J., Lauenroth, W.K. (eds.) Models in Ecosystem Science. Princeton University Press, Princeton (2003)
2. Oreskes, N., Shrader-Frechette, K., Belitz, K.: Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences. Science 263 (1994)
3. Nehmzow, U., Walker, K.: The Behaviour of a Mobile Robot Is Chaotic. AISB Journal 1(4) (2003)
4. Nehmzow, U.: Scientific methods in mobile robotics. Springer, London (2006)
5. Nehmzow, U., McKerrow, P.J., Billings, S.A.: On the Meaning of Computer Models of Robot-Environment Interaction. In: Proc. Towards Autonomous Robotic Systems, TAROS (2007)
6. Iglesias, R., Kyriacou, T., Nehmzow, U., Billings, S.A.: Task Identification and Characterisation in Mobile Robotics through Non-Linear Modelling. Robotics and Autonomous Systems 55 (2007)
7. Iglesias, R., Kyriacou, T., Nehmzow, U., Billings, S.A.: Task Identification and Characterisation in Mobile Robotics. In: Proceedings of Towards Autonomous Robotic Systems (2004)
8. McKerrow, P.J., Ratner, D.: Calibrating a 4-wheel mobile robot. In: Proc. IROS IEEE/RSJ (2002)
9. Billings, S.A., Chen, S.: The determination of multi variable nonlinear models for dynamical systems. In: Leonidas, C. (ed.) Neural Network Systems, Techniques and Applications. Academic press, London (1998)
10. Korenberg, M., Billings, S.A., Liu, Y.P., McIlroy, P.J.: Orthogonal parameter estimation algorithm for non-linear stochastic systems. Int. J. Control 48 (1988)

11. Billings, S.A., Voon, W.S.F.: Correlation based model validity tests for non-linear models. Int. J. Control 44 (1986)
12. INRIA, Scilab 4.0. (2007), http://www.scilab.org
13. Kelvin, W.T.: Popular Lectures and Addresses, vol. 1. Macmillan, Basingstoke (1891)
14. Zheng, G.L., Billings, S.A.: Qualitative validation and generalisation in nonlinear systems identification. Control 72 (1999)
15. Haynes, B.R., Billings, S.A.: Global analysis and qualitative model validation in nonlinear system identification. Nonlinear Dynamics 15 (1994)
16. Swain, A.K., Billings, S.A., Stansby, P.K., Baker, M.: Accurate prediction of non-linear wave forces: Part 1 Fixed Cylinder. Mechanical Systems and signal Processing 12 (1998)
17. Zhu, D., Billings, S.A., Balikhin, M., Coca, D.: Data derived continuous time model for magnetospheric dynamics. Geophysical Research Letters 33 (2006)

Information Dynamics of Evolved Agents

Paul L. Williams[1] and Randall D. Beer[1,2]

[1] Cognitive Science Program
[2] School of Informatics and Computing,
Indiana University, Bloomington IN 47406, USA
{plw,rdbeer}@indiana.edu

Abstract. Information-theoretic techniques have received much recent attention as tools for the analysis of embodied agents. However, while techniques for quantifying static information structure are well-established, the application of information theory to the analysis of temporal behavior is still in its infancy. Here we formulate a novel information-theoretic approach for analyzing the dynamics of information flow in embodied systems. To demonstrate our approach, we apply it to analyze a previously evolved model of relational categorization. The results of this analysis demonstrate the unique strengths of our approach for exploring the detailed structure of information dynamics, and point towards a natural synergy between temporally-extended information theory and dynamical systems theory.

1 Introduction

From its origins in the study of communication networks, information theory has developed into a powerful general framework for the analysis of complex systems. In recent years, information theory has also been increasingly applied to the study of model brain-body-environment systems, as well as in neuroscience more broadly, as a means of exploring the mechanistic underpinnings of intelligent behavior. The core concepts of information theory are a set of general measures for quantifying the uncertainty associated with individual variables and for characterizing the nonlinear relationships between them [4]. Applied to brain-body-environment systems, these measures can be used to quantify relationships between sensory, neural and motor components, or between such components and features of the external world. For example, using a variety of robotic and simulation platforms, information-theoretic techniques have been used to investigate the ways in which embodied agents select and shape information from their environment [14,12]. Information-theoretic techniques have also been applied to uncover the latent structure of sensory and motor variables based on the time course of activity that they produce [12]. Finally, through a clever inversion of the evolutionary agent approach, information-theoretic techniques have also been used as fitness functions to evolve adaptive behavior, demonstrating that intelligent solutions often result when agents are selected simply to maximize informational structure [15,20,9].

For the most part, information-theoretic analyses of brain-body-environment systems have ignored the interesting temporal behavior of these systems, instead collapsing over time to apply static measures of information structure [10,19,22]. More recently, though, other studies have begun to apply information theory to analyze the temporal behavior

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 38–49, 2010.

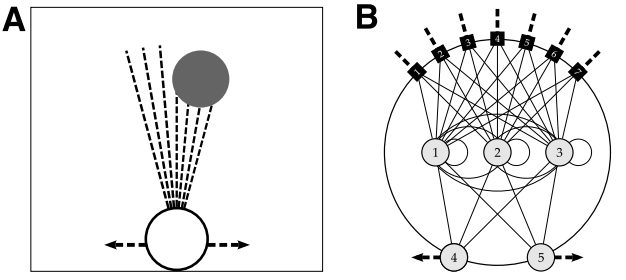


Fig. 1. Relational agent and environment. (A) The agent moves horizontally while circles fall towards it from above. (B) The agent is controlled by a 5-node (3 interneuron, 2 motor) CTRNN.

of these systems, characterizing the structure of interactions between components in terms of information flow [11,18,9]. However, work in this vein has thus far been of a very preliminary nature. For example, studies have characterized the average strength of interactions between components but not their detailed time course, and the behaviors studied so far have been limited to simple kinds of object tracking.

This paper aims to extend this line of work by formulating a novel toolset for analyzing the information dynamics of embodied systems, and by demonstrating its application to a previously developed model of relational categorization [23].

2 A Model of Relational Categorization

The relational categorization agent was developed as an extension of earlier studies of object discrimination [2], and uses the same model body and environment as these earlier studies. The agent moves back and forth along a horizontal line while circular objects fall toward it from above (Fig. 1A). Using an array of seven rays, the agent can sense the distance to each intersection between a ray and a falling object. The agent is controlled by a 5-node (3 interneuron, 2 motor) CTRNN (Fig. 1B) whose parameters were evolved so as to minimize the final horizontal separation for certain objects ("catch") and to maximize the separation for others ("avoid"). Specifically, on each trial, the agent is first presented with one circular object. Then, a second circular object is presented, differing in size from the first, and the agent's task is to catch this second object when it is smaller than the first and to avoid it when it is larger. Thus, the agent must classify the second object based on the relational category *smaller*. Overall, agents evolved to be highly proficient at this task, with the best agent in each evolutionary run attaining a fitness of 90% or higher. In previous analysis and in this paper, we consider only the best-evolved agent, which achieved an average performance of 99.83%. Behaviorally, this agent was found to passively observe the first falling object without moving, and then move to catch or avoid the second falling object as appropriate.

A dynamical analysis of this agent revealed several key features of its underlying neural mechanism. First, we examined how the size of the first circle is "stored" so that it can influence the agent's behavior on the second object. To determine this, we examined the state variables of the agent—its neural outputs and body position—at the time

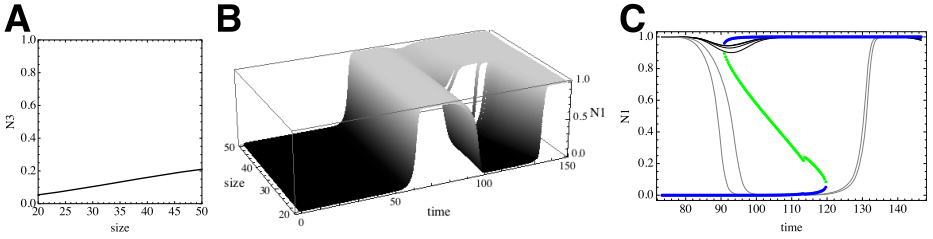


Fig. 2. A summary of previous dynamical analysis. (A) The state of N3 at the trial midpoint stores the size of the first object. (B) Trajectories of N1 for a range of second object sizes; whether or not N1 switches off determines the catch/avoid response. (C) A bistable region determines the behavior of N1. Trajectories of N1 are shown for different first object sizes along with the equilibrium points (EPs). Blue EPs are stable and green EPs are saddles.

when the first object is removed, and found that a single neuron (N3) correlates with the size of the object at this time (Fig. 2A). Then, we examined how the agent's response to the second object is triggered. We found another neuron (N1) whose switching on or off at a particular point in time determines whether the agent catches or avoids the second object (Fig. 2B). We next explored how these two features of the neural dynamics interface with one another. That is, we examined how the first object's size stored in N3 comes to determine whether or not the second object switches the agent into catch or avoidance behavior. We found that the dynamics of the evolved CTRNN is unistable when objects are far away or very close, but bistable when objects are at intermediate distances, with saddle-node bifurcations separating these unistable and bistable regions (Fig. 2C). Crucially, the timing and location of these bifurcations were found to correlate with the second circle's size. Depending on the state that the first object leaves N3 in, N1 is pulled toward the catch or avoid attractor after a saddle-node bifurcation occurs. Thus, the relational agent's behavior arises from an interaction between build-up of state during the first circle presentation and the timing and location of a saddle-node bifurcation in the CTRNN dynamics. It was a desire to better understand how multiple sources of information interact over time to produce this behavior that first led us to the information-theoretic approach described here.

3 Overview of Information Dynamics

The central idea of our information dynamics approach is to explore how information about particular stimulus features flows through a brain-body-environment system. For example, in our analysis of the relational agent, the stimulus features that we consider are the sizes of the first and second objects presented to the agent. Traditionally, information-theoretic analyses have focused on static measures of informational structure. In contrast, our approach is to unroll these static measures across time to explore how informational structure evolves over the course of behavior. In addition, by further unrolling across values of the stimulus feature, we are able to trace how information about particular stimuli flows through the system.

The first step in our analysis is to evaluate the agent's behavior for a uniformly distributed sample of the stimulus feature, recording the trajectories of all neural and bodily state variables for each stimulus presentation. From the values taken on by each state variable at each moment in time and the corresponding stimuli that produced them, we then estimate a joint probability distribution over values of the state variable and the stimulus feature. In other words, the stimulus feature is treated as a random variable (henceforth F), and each state variable of the agent system (its sensors, neurons, and body position) is treated as a random process—a time-indexed sequence of random variables—and we estimate joint distributions for F paired with each time-indexed random variable. To estimate these distributions, we use average shifted histograms, a kind of kernel density estimation, for their beneficial trade-off between computational and statistical efficiency, though other approaches could certainly be used [17].

The second step is to measure the amount of information that each state variable contains about the stimulus feature as a function of time. Using the standard concepts of information theory [4], the information that one random variable X contains about another random variable Y (and vice versa) is given by their *mutual information*:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}. \tag{1}$$

So, for instance, if V_t is a random variable corresponding to a particular state variable V at time t, then $I(F;V_t)$ gives the amount of information that V contains about the stimulus feature F at time t. We measure the mutual information $I(F;V_t)$ for each state variable V and time index t, thereby generating a *flow of information* for each state variable of the system.

To better understand the dynamic properties of these information flows, we next apply measures of information gain and loss to the flow for each state variable. Each of these measures derives from the formula for *conditional mutual information* between two random variables, X and Y, given a third random variable Z:

$$I(X;Y|Z) = \sum_{x \in X} \sum_{y \in Y} \sum_{z \in Z} p(x,y,z) \log_2 \frac{p(x,y|z)}{p(x|z)p(y|z)}. \tag{2}$$

The conditional mutual information quantifies the amount of information shared by X and Y when Z is known. Using this formula, the *information gain* for state variable V at time t is given by:

$$I_G(F,V,t) = I(F;V_t|V_{t-1}) \tag{3}$$

which, in words, corresponds to the new information that V contains about F at time t when the information that it already contained at $t-1$ is excluded. This measure of information gain can also be understood as a form of transfer entropy [16], where transfer is quantified from the stimulus feature to the state variable. By similar reasoning, a measure of *information loss* can also be attained:

$$I_L(F,V,t) = I(F;V_{t-1}|V_t). \tag{4}$$

In words, information loss for a state variable V is the amount of information that V contained about F in the preceding time step that it now lacks. Taken together,

information gain and loss specify how the information contained by a component changes from one moment to the next:

$$I(F;V_t) = I(F;V_{t-1}) - I_L(F,V,t) + I_G(F,V,t). \tag{5}$$

The primary benefit of treating information gain and loss separately is that, while loss of information about external features happens generically for components of a dissipative dynamical system, a gain in information signals the influence of another component (or, in the case of a sensor, direct influence from the stimulus) in transferring information to the specified component. Thus, identifying information gain provides a basis for characterizing how information is transferred between components of the system. Although the present study focuses solely on information flow in individual state variables, the transfer of information between state variables is a direction we plan to pursue in future work (see [8] for related ideas applied to cellular automata).

The final element of our approach explores the structure of information flow at a finer level of detail. The key insight here is that mutual information measures the *average* (expected) information that one variable provides about another, averaged over all outcomes for the two variables. Thus, by unrolling this average in various ways, it is possible to examine the specific relationships between combinations of outcomes that contribute to these averaged quantities. In neuroscience, for example, this idea has been applied to examine which specific neural responses are most informative about a stimulus ensemble [5], and which stimuli tend to evoke the most informative responses [3]. Using this idea, we define a measure of *specific information*:

$$I(F=f;V) = \sum_{v \in V} p(v|f) \left[\log_2 \frac{1}{p(f)} - \log_2 \frac{1}{p(f|v)} \right] \tag{6}$$

which quantifies the information that a state variable V provides about a particular stimulus f (e.g., a particular circle size). The quantity $\log_2 \frac{1}{p(f)}$ is called the surprise of f, so, translated literally, $I(F=f;V)$ is the average reduction in surprise of f resulting from knowledge of V (see also [3], where $I(F=f;V)$ is called the specific surprise). Using specific information, we expand our analysis of information flow to consider the information that each state variable V provides about each particular stimulus f at each time t. Similarly, applying specific information to expand our measures of information gain and loss, we explore how state variables gain and lose information about particular stimuli as a function of time.

4 Information Dynamics of Relational Categorization

To demonstrate our information dynamics approach, we next apply it to explore two questions about the relational categorization agent. First, we examine how the agent extracts information about first object size. Then, we explore how the agent integrates size information from the first and second objects. For all of the following results, the agent was evaluated on circle sizes in the range [20, 50] (the same used during evolution) sampled at an interval of 0.01. Probability distributions were estimated using

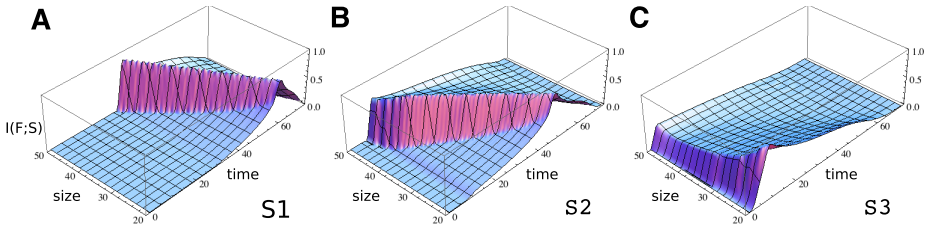


Fig. 3. Flow of information for sensors S1-S3. All plots throughout are normalized to [0,1].

average shifted histograms with 100 bins and 8 shifts, though the results reported were qualitatively robust over a broad range of discretization (20 to 200 bins).

We begin by examining what information about first object size (henceforth F) is available to the agent. In general, this amounts to examining the flow of information for F in each of the agent's seven sensors. However, since the agent uses a passive strategy and objects are always presented at the agent's midline, information from the sensors is bilaterally symmetric and thus we can simplify our analysis by considering the sensors on only one side. Plots of the information flow for sensors S1-S3 are shown in Fig. 3, where specific information is plotted as a function of object size and time. The flow of information for each sensor exhibits the same prominent feature: a high "ridge" of information that begins first for large sizes and travels across to successively smaller sizes. However, the time at which this ridge forms varies for the three sensors, beginning first for the innermost sensor (S3) and later for sensors further from the agent's midline.

What do these plots tell us about the information available to the agent? Recall that a high value for the specific information indicates that a particular object size is unsurprising given the state of the sensor at a certain point in time. This means that the sensor tends to take on distinct values for that object size, serving to distinguish that object size from differently sized objects. Thus, the high ridge of information informs us that the state of each sensor first distinguishes large objects, then objects successively smaller in size. Furthermore, examining the path of the object, the origin of these ridges becomes immediately clear: the peak value in information for each object size occurs at precisely the time when objects of that size first intersect the corresponding ray sensor. Since larger objects intersect earlier, the sensors first provide information about these sizes. Thus, the primary information available to the agent stems from the timing of when differently sized objects intersect each ray sensor, with this timing information varying for each of the three sensors.

The next step is to inspect how size information flows through other components of the brain-body-environment system. In general, this includes both the interneurons and the position of the agent's body; however, since the agent uses a passive strategy, we can simplify our analysis by considering only neural state variables. Each of the three interneurons shows a markedly different pattern of information flow (Fig. 4). First, consider the information contained by each neuron at time 70, which corresponds to the time when the first object is removed. Previously, we noted the primary role played by N3 in storing information about F, a fact that is also evident from the plots of information flow. When the object is removed, N3 contains a high amount of information about

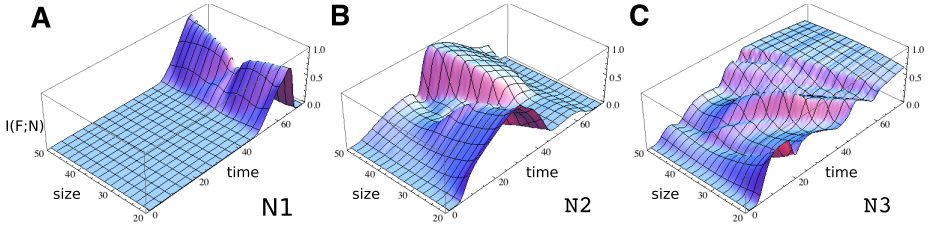


Fig. 4. Flow of information for neurons N1-N3

all object sizes, while N1 and N2 contain no information. However, crucially, while our previous observation about N3 was based on a single snapshot of the agent's state, the plots in Fig. 4 depict the entire temporally extended process that leads to this eventual state. Thus, by examining information flow, we can move beyond considering simply how information ends up stored, and instead explore the dynamic process that results in this storage. In particular, the plots in Fig. 4 make clear that, although size information ends up concentrated in N3, both N1 and N2 also contain size information at earlier times in the trial. N1 contains a high amount of information about small and large sizes late in the trial, while N2 contains information first about small and then about large sizes. Thus, contrary to the picture of N3 as the sole bearer of size information, these observations suggest that N1 and N2 may also play an important informational role, an idea that we return to momentarily.

Nonetheless, the most striking feature of these information flow plots is the gradual build-up of size information in N3 (Fig. 4C). In contrast with N1 and N2, which both gain and subsequently lose information, N3 continually accumulates information through a succession of information "waves". To better understand the source of these waves, we next examine the dynamics of information gain for N3 (Fig. 5). Comparing Figures 4C and 5 reveals that the waves of information flow translate to ridges of information gain. In particular, the plot of information gain shows a series of three prominent ridges, each traveling across from large to small sizes, as well as some other secondary ridge-like features. Importantly, each prominent ridge of information gain closely aligns with a corresponding ridge of information for one of the three sensors (Fig. 3). Recall that the latter are produced by objects intersecting the ray sensors at different times depending on their size. Thus, together these observations suggest a hypothesis for how the agent extracts size information: N3 primarily integrates information available from the sensors, stemming from the timing of when each sensor is broken by differently sized objects.

To test this hypothesis, we next explored the impact of removing information from each of the sensors, using an approach based on *informational lesioning* [6]. The basic idea of informational lesioning is to systematically diminish a component's ability to convey information while simultaneously minimizing its deviation from normal behavior. In this way, the informational contribution of a component can be quantified independently from other functional roles that the component might play. For each of the sensors, normal behavior is essentially the same: it remains off for a certain period before intersecting the object, then begins to increase monotonically, with the timing of

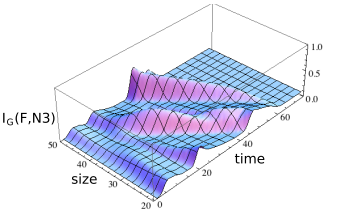


Fig. 5. Information gain for N3

this transition dependent on object size. Additionally, the rate at which sensor values increase, i.e., the curvature of the sensory trajectory, also varies with object size. Thus, size information actually comes from two features of the sensory input: the timing of the off/on transition, and the curvature of the subsequent trajectory. In order to isolate timing information, which our analysis predicts to be the most salient, we first removed information due to varying curvatures. This was done by replacing the curved portion of sensor trajectories with a single best-fit linear approximation formed separately for each sensor, so that trajectories for a given sensor differed only in the timing of their off/on transition. Performing this manipulation simultaneously on all sensors had essentially no impact on performance (99.67%), thus confirming that timing information is primarily what the agent uses. Next, to test the prediction that timing information from different sensors is integrated, we independently removed the timing information from each symmetric pair of sensors by setting their values according to the mean off/on transition time. Under this manipulation, performance dropped to 71.84% for S1/S7, 92.53% for S2/S6, and 96.93% for S3/S5. Thus, timing information clearly is integrated from the different sensor pairs, though with varying contributions made by each. Interestingly, the relative contribution from each pair correlates with the magnitude of information gain that each produces for N3 (height of ridges in Fig. 5). For example, the greatest impact on performance results from lesioning S1/S7, which also produce the largest information gain for N3. Thus, not only does information flow analysis yield a correct qualitative prediction for how information is extracted, but also points to some quantitative features of this process.

However, as alluded to earlier, this explanation does not provide the full story. In particular, the other two interneurons also contain size information at different times, and thus may also contribute to the information stored in N3. To explore this possibility, we performed similar lesion experiments for N1 and N2. Omitting details, the results were a minimal change in performance when N1 was lesioned (98.59%) but a considerable decrease when N2 was lesioned (92.27%). Thus, the results indicate that N2 also plays a significant informational role with respect to F. In particular, it is likely that N2 accounts for the secondary features in the plot of information gain for N3 (Fig. 5), though further testing is needed to bear this out. However, in general the results make clear that, contrary to the view of one neuron capturing size information, information is in fact distributed both spatially—across different components of the system—and temporally—with different components carrying information at various times.

Let us now examine how the agent integrates information about first and second object size (abbreviated S). Rather than an exhaustive informational analysis, our primary

interest here will be to compare with previous dynamical findings; thus, we will consider only N1, which we know plays a critical role in triggering the agent's decision. To understand the information flow for each feature, we found it most useful to consider different fixed values of the other stimulus, instead of averaging over them. Thus, we first examined the flow of information for F with different fixed values of S. An example is shown in Fig. 6A for $S = 35$, with qualitatively similar results for other values of S. In examining these plots, we can ignore values occurring after time 110, which happen after the catch/avoid decision is made, and also the narrow band of high values occurring around size 35, which correspond to values of F too similar to S for the agent to discriminate. Thus, the predominant feature is a rapid increase in information about all sizes, followed by a sudden collapse just prior to the catch/avoid decision. For different values of S, this same feature is also present, but with the timing of the collapse occurring earlier or later depending on S. Moreover, examining the flow of information about S for different fixed values of F (Fig. 6B), we find essentially the same pattern. N1 quickly gains information about S, then loses it all just before the response is made. Thus, somewhat strangely, although we know that N1 plays a crucial role in driving the agent's response, N1 suddenly loses all information about first and second object size just prior to the decision being made.

Of course, this result is not actually strange when we consider that what matters for categorization is not the size of either object individually, but rather their relative size. To investigate relative size information, we can define a binary random variable R corresponding to whether or not the second object is smaller than the first. The flow of information for R is shown in Fig. 6C for several different values of S. As expected, these plots show a rapid increase in relative size information for N1, reflecting the neural precursor to the agent's behavioral response. In addition, the slope of this increase can be seen to vary for different values of S, corresponding to the varying time of the agent's decision for different second object sizes. The timing of this increase in relative size information also coincides with the collapse in information about F and S, so that collectively these features signal the integration of size information in N1. Interestingly, this timing also aligns with the timing of the underlying bifurcation that occurs to split the trajectories into catch and avoid bundles [23]. Thus, we can observe an agreement between the dynamical and informational accounts, with each providing a different view of how the behavioral response is produced. Dynamically, this response is carried out through the timing of a bifurcation relative to a previous build-up of state, while informationally the response is captured by a sudden increase in size information for both objects, followed by a collapse in information about individual object size and a corresponding rise in information about relative size.

5 Future Directions

Current work is aimed at extending the analytical techniques described here in several ways. First, we are developing tools to characterize how information is transferred between components of a brain-body-environment system. While the techniques described above consider the flow of information in individual state variables with the rest of the system treated as an undifferentiated information source, these tools will

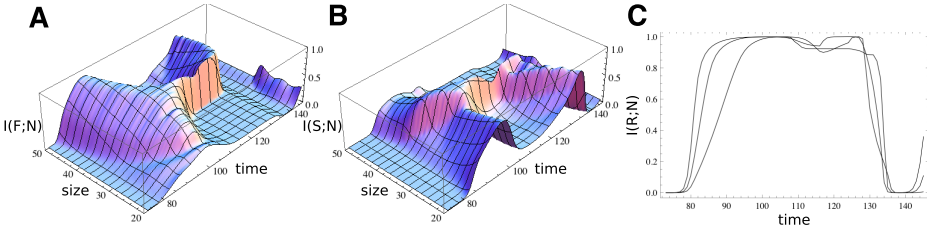


Fig. 6. Information flow in N1 during the second half of the trial

allow us to understand the interactions between state variables that produce their respective flows. Second, we are working on techniques to explore how information may be distributed across multiple components of a system through synergistic or redundant encodings. As with measures of information transfer, these techniques are designed to move beyond the flow of information in individual components to begin to understand the significance of their interactions. Finally, we are currently extending our analytical toolset to include additional interventional methods. Interventional methods are needed to supplement purely observational ones in order to distinguish information transfer from spurious correlations [13,1]. The informational lesioning experiments described above provide one preliminary example along these lines.

One of the primary strengths of information-theoretic techniques is that they extend naturally to situated and embodied aspects of behavior. Techniques for analyzing information dynamics, demonstrated above for sensory and neural variables, apply just as readily to bodily and environmental state variables, and thus can be used to investigate interactions that span the brain-body and body-environment boundaries. In future work, we plan to apply these techniques to analyze agents that exhibit interestingly embodied and extended solutions to cognitive tasks. For example, although embodiment did not play a significant role in the relational agent's behavior due to its passive strategy, another previously evolved agent was found to actively scan objects before categorizing them [2]. We plan to investigate what role this scanning behavior may play in allowing the agent to elicit and structure information from its environment. As a second example, another agent evolved for the relational categorization task was found to use its position in the environment, rather than its neural state, to store information about first object size. Thus, this agent exhibits a simple form of information offloading, where aspects of cognitive processing are shifted to the body or environment. Using information dynamics, we plan to examine the process of extraction that results in this stored information, similar to the analysis presented here though with the caveat that extraction and storage in this case spread across the brain-body and body-environment divides.

Finally, a significant consequence of extending information-theoretic techniques to the temporal domain is that it opens up the possibility of exploring the relationship between informational and dynamical approaches to embodied systems. Although the analysis presented here was preliminary in nature, it already points to some promising possibilities along these lines. For example, the two approaches were found to provide distinct yet compatible accounts for how the agent makes its categorical discrimination regarding relative object size. In informational terms, this was manifested by a sudden

gain and then loss in information about both first and second object size in N1, followed by rapid gain in information about relative size, while dynamically this was explained by an appropriately timed bifurcation in the underlying dynamics. In addition, one can begin to see how the two approaches might complement one another. Using informational techniques, it was natural to explore how the agent extracts size information, a question that would be difficult to address or even formulate in dynamical terms. Conversely, the bifurcation identified via dynamical analysis helps to explain *why* the information flow in N1 exhibits a sudden collapse in information about particular sizes and increase in information about relative size.

More generally, it is our view that informational and dynamical approaches are likely to provide complementary insights, as a result of the unique perspectives and strengths that each approach affords. Dynamical tools are especially well-suited for characterizing the long-term behavior of systems, the stability of this behavior in response to perturbations, and the changes in long-term behavior that systems undergo when aspects of their structure are modified. In contrast, informational tools offer a view of how the behavior of a system is realized in terms of the specific interactions between its components, and may be especially useful for characterizing the non-autonomous and transient aspects of behavior that are tied to specific external features. As well as these differences, informational and dynamical ideas also share deep similarities, and it will undoubtedly be informative to explore this common ground as it relates to properties of brain-body-environment systems. For example, information gain and loss are closely related to the divergence or convergence of trajectories in phase space [7], which are characterized dynamically by the Lyapunov exponents [21]. In general, the divergence of trajectories leads to an increase in information, while convergence leads to its loss. Similarly, the limit sets and basin boundaries of a system also relate directly to its properties of convergence and divergence, and thus are likely to play a significant role in shaping the flow of information. However, the true test for all of these ideas will be to apply both dynamical and informational techniques to analyze concrete systems and to compare and contrast the resulting insights that each provides. The analysis presented here can be viewed as an initial step in this direction. Ultimately, such work may hopefully begin to reconcile the dynamical and information theoretic perspectives on intelligent agents which has generated so much controversy in recent years.

References

1. Ay, N., Polani, D.: Information flows in causal networks. Advances in Complex Systems 11(1), 17 (2008)
2. Beer, R.D.: The dynamics of active categorical perception in an evolved model agent. Adaptive Behavior 11(4), 209–243 (2003)
3. Butts, D.A.: How much information is associated with a particular stimulus? Network 14, 177–187 (2003)
4. Cover, T.M., Thomas, J.A.: Elements of Information Theory, 2nd edn. Wiley-Interscience, New York (2006)
5. DeWeese, M.R., Meister, M.: How to measure the information gained from one symbol. Network 10, 325–340 (1999)
6. Keinan, A.: Controlled analysis of neurocontrollers with informational lesioning. Philosophical Transactions of the Royal Society A 361(1811), 2123–2144 (2003)

7. Liang, X.S., Kleeman, R.: A rigorous formalism of information transfer between dynamical system components. II. Continuous flow. Physica D 227(2), 173–182 (2007)
8. Lizer, J.T., Prokopenko, M., Zomaya, A.: Local information transfer as a spatiotemporal filter for complex systems. Physical Review E 77, 26110 (2008)
9. Lizier, J.T., Prokopenko, M., Tanev, I., Zomaya, A.Y.: Emergence of glider-like structures in a modular robotic system. Proceedings of ALife XI, 366–373 (2008)
10. Lungarella, M., Pegors, T., Bulwinkle, D., Sporns, O.: Methods for quantifying the informational structure of sensory and motor data. Neuroinformatics 3(3), 243–262 (2005)
11. Lungarella, M., Sporns, O.: Mapping information flow in sensorimotor networks. PLoS Computational Biology 2(10), 1301–1312 (2006)
12. Nehaniv, C.L., Mirza, N.A., Olsson, L.: Development via information self-structuring of sensorimotor experience. In: Lungarella, M., Iida, F., Bongard, J.C., Pfeifer, R. (eds.) 50 Years of Aritficial Intelligence. LNCS (LNAI), vol. 4850, pp. 87–98. Springer, Heidelberg (2007)
13. Pearl, J.: Causality: Models, Reasoning, and Inference. Cambridge University Press, Cambridge (2000)
14. Polani, D., Sporns, O., Lungarella, M.: How information and embodiment shape intelligent information processing. In: Lungarella, M., Iida, F., Bongard, J.C., Pfeifer, R. (eds.) 50 Years of Aritficial Intelligence. LNCS (LNAI), vol. 4850, pp. 99–111. Springer, Heidelberg (2007)
15. Prokopenko, M., Gerasimov, V., Tanev, I.: Evolving spatiotemporal coordination in a modular robotic system. In: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J.C.T., Marocco, D., Meyer, J.-A., Miglino, O., Parisi, D. (eds.) SAB 2006. LNCS (LNAI), vol. 4095, pp. 548–559. Springer, Heidelberg (2006)
16. Schreiber, T.: Measuring information transfer. Physical Review Letters 85(2), 461–464 (2000)
17. Scott, D.W.: Multivariate Density Estimation: Theory, Practice, and Visualization. Wiley-Interscience, Hoboken (1992)
18. Seth, A.: Causal networks in simulated neural systems. Cognitive Neurodynamics 2(1), 49–64 (2008)
19. Seth, A., Edelman, G.: Environment and behavior influence the complexity of evolved neural networks. Adaptive Behavior 12, 5–20 (2004)
20. Sporns, O., Lungarella, M.: Evolving coordinated behavior by maximizing information structure. ALifeX, 323–329 (2006)
21. Strogatz, S.H.: Nonlinear Dynamics and Chaos. Addison-Wesley, Reading (1994)
22. Tarapore, D., Lungarella, M., Gómez, G.: Quantifying patterns of agent–environment interaction. Robotics and Autonomous Systems 54(2), 150–158 (2006)
23. Williams, P.L., Beer, R.D., Gasser, M.: An embodied dynamical approach to relational categorization. In: CogSci 2008, pp. 223–228 (2008)

Taming the Beast: Guided Self-organization of Behavior in Autonomous Robots

Georg Martius[1,2,3] and J. Michael Herrmann[1,2,4]

[1] Bernstein Center for Computational Neuroscience Göttingen
[2] Institute for Nonlinear Dynamics, University of Göttingen
[3] Max Planck Institute for Dynamics and Self-Organization
[1−3]Bunsenstraße 10, 37073 Göttingen, Germany
[4] University of Edinburgh, School of Informatics, IPAB
10 Crichton Street, Edinburgh EH8 9AB, U.K.
georg@nld.ds.mpg.de, michael.herrmann@ed.ac.uk

Abstract. Self-organizing processes are crucial for the development of living beings. Practical applications in robots may benefit from the self-organization of behavior, e.g. for the increased fault tolerance and enhanced flexibility provided that external goals can also be achieved. We present several methods for the guidance of self-organizing control by externally prescribed criteria. We show that the degree of self-organized exploravity of the robot can be regulated and that problem-specific error functions, hints, or abstract symbolic descriptions of a goal can be reconciled with the continuous robot dynamics.

1 Introduction

Intrinsically motivated but non-trivial behavior is an important prerequisite for autonomous robot development. Self-organization of robot control is a promising approach, where the resulting behavior is characterized by on-going exploration or by a refinement of those behavioral traits that can be called natural for a specific robot in a particular environment [1,2]. Animals, including humans, acquire their behavioral repertoire in a similar way, behavioral elements are developed autonomously and are further refined during the whole life span. Nevertheless, modulatory effects on the self-organizing behavior can be imposed as well by the environment. Animals can learn by imitation or by downright teaching from superior fellows. Furthermore, behavior is subject to the dictate of drives that are partly intrinsic and partly external to the agent. Finally, humans derive goals for their own behavior from rational reasoning.

Incentives for behavioral adaptation is an interesting subject for study in behavioral science where the interference of such higher forms of learning with the underlying self-organization does not seem to be a problem. In robotics, however, the situation is different. Although promising examples exist [1,3,4], self-organization of behavior is still a field of active exploration. Further questions such as the interaction of learning by self-organization and learning by supervision or by external reinforcement are just starting to gain scientific interest.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 50–61, 2010.

Usually, goal-oriented behavior is achieved by directly optimizing the parameters of a control program such that the goal is approached more closely. The learning system must receive information about whether or not the behavior actually approaches the goal. This information may be available via a reward signal in reinforcement learning or by a fitness function in evolutionary algorithms. We will allow for different types of goal-related information when aiming at a combination of self-organizing control with external drives. For this combination the term *guided self-organization* (GSO) was proposed [5,6]. In this general perspective, GSO is the combination of goal-oriented learning and developmental self-organization. Each of the two learning paradigms bring about their particular benefits and GSO aims at combining them in an optimal manner. Self-organizing systems tend to have a high tolerance against failures and degrade gracefully, which is an advantage that should not be given up when developing systems aiming to achieve tasks in practical applications. Although being interested in the wider context, we will be dealing in this particular study with a specific approach to self-organizing control, namely homeokinetic learning [7].

What can we expect from a *guided homeokinetic controller*? It has been shown earlier that a variety of behaviors can emerged from the principle of homeokinesis [1,2]. This process of self-organization selects certain elements from the space of action sequences such that a set of behaviors is realized. The emerging behaviors show a coherent sensorimotor dynamics of the particular robot in its environment. The goal is now to shape the self-organization process to produce desired or preferred behaviors within a short time. Part of the idea is to channel the exploration of the homeokinetic controller around certain behaviors, such that control modes can be found which match the given robotic task.

In the present paper, we will discuss three mechanisms of guidance. The first one uses online reward signals to shape the emerging behaviors and is briefly discussed in Section 3. A second mechanism for guiding consists in the incorporation of supervised learning e. g. by specific nominal motor commands that we call *teaching signals* (Section 4). Using distal learning [8] we study the utilization of teaching signals in terms of sensor values in Section 5. In Section 6 we propose a third mechanism that allows for the specification of mutual motor teaching. The latter two are presented here for the first time.

2 Self-organized Closed Loop Control

Self-organizing control for autonomous robots can be achieved by establishing an intrinsic drive towards behavioral activity as described by the homeokinetic principle [7], for details cf. [1,2].

The dynamical evolution of the sensor values $x \in \mathbb{R}^n$ of the robot is described by

$$x_{t+1} = \psi(x_t) = M(x_t, y_t, \mathcal{A}) + \xi_{t+1}. \tag{1}$$

where M is the internal predictive model that maps the sensations x and the actions $y \in \mathbb{R}^m$ to the predicted sensory inputs, $\mathcal{A}$ is a set of parameters and ξ

is the mismatch between the predicted and the actually observed sensor values. In this study, the internal model M is implemented as a linear neural network:

$$M(x_t, y_t, \mathcal{A}) = Ay_t + Sx_t + b, \tag{2}$$

where $\mathcal{A} = (A, S, b)$. The actions y are generated by a controller function

$$y_t = K(x_t, C, h) = g(Cx_t + h) \tag{3}$$

where $g(\cdot)$ is a componentwise sigmoidal function, we use $g_i(z) = \tanh(z_i)$, C is a weight matrix and h is a bias vector.

The parameters $\mathcal{A}$ of the model are adapted online to minimize the prediction error $\|\xi\|^2$ (Eq. 1) via gradient descent. However, the minimization is ambiguous with respect to A and S because y is a function of x, see (3). In contrast to our earlier approach [5], we introduce a bias into the model learning in order to capture the essential part of the mapping by the matrix A. This is achieved by the adaptation of A based on a prediction error that is obtained for a discounted S term, i. e.

$$\Delta A = \epsilon_A (\xi_{t+1} + \delta S x_t) y_t^\top, \tag{4}$$
$$\Delta S = \epsilon_A \xi_{t+1} x_t^\top, \tag{5}$$

where a small value of $\delta = 0.001$ fully serves the purpose and $\epsilon_A = 0.1$ is a learning rate.

If the parameters of the controller (C, h) are also adapted by the minimization of the prediction error $\|\xi\|^2$ then stable but typically trivial behaviors are achieved. The robot may get trapped in any state with $\xi = 0$ which happens prevalently when it is doing nothing. There are, however, specific cases where such a principle can be successfully applied: If the drive for activity is provided from outside or brought about by e. g. evolution [9], or if a homeostatic rule is applied to, for instance, the neural activity [10,11]. The homeokinetic paradigm [7,1] instead suggests to use the so-called *postdiction error*. This error is the mismatch

$$v_t = x_t - \hat{x}_t \tag{6}$$

between true sensor values x_t and reconstructed sensor values $\hat{x}_t$ that are defined using Eq. 1 as

$$\hat{x}_t = \psi^{-1}(x_{t+1}) \tag{7}$$

assuming that ψ is invertible. If $\hat{x}_t$ (rather than x_t) had been actually observed then by definition the best possible prediction based on the present model M (1) would have been made. The error functional minimizing the postdiction error v_t is called *time-loop error* (TLE) and can be approximated by

$$E_{TLE} = \|v_t\|^2 = \xi_{t+1}^\top \left(L_t L_t^\top\right)^{-1} \xi_{t+1}, \tag{8}$$

where $L_{t,ij} = \frac{\partial \psi(x_t)_i}{\partial x_{t,j}}$ is the Jacobian matrix of ψ at time t. Thus another important feature of this error quantity becomes evident: The minimization of v entails

the minimization of the inverse Jacobian. This in turn means that small eigenvalues of L are increased. Thus the controller performs stabilization in inverted time, i. e. destabilization in forward time. This eliminates the trivial fixed points (in sensor space) and enables spontaneous symmetry breaking phenomena. The reader might wonder why the system does not start to behave chaotically or reach uncontrollable oscillations. The reason is that the destabilization is limited by the nonlinearities $g(\cdot)$ and that the TLE is invariant to oscillation frequencies as discussed in [12]. Intuitively, the homeokinesis can be understood as the drive to sustain a non-trivial behavior that can be predicted by the internal model. Since the internal model is very simple smooth behaviors are preferred. Fig. 1 illustrates how the homeokinetic controller is connected to a robot.

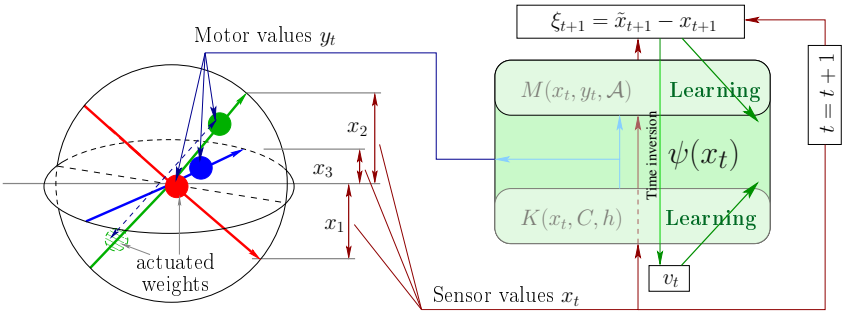


Fig. 1. The Homeokinetic controller connected to the SPHERICAL robot in the sensorimotor loop. The SPHERICAL robot is driven by weights that are moved along the axes by actuator and is equipped with axis-orientation sensors (x_i). The homeokinetic controller consists of the controller function K and the predictor M, both together form ψ (Eq. 1). The TLE is obtained by propagating ξ_{t+1} through ψ in inverted time.

The TLE (8) can be minimized by gradient descent which gives rise to a parameter dynamics that evolves simultaneously with the state dynamics:

$$x_{t+1} = \psi(x_t) + \xi_{t+1}, \tag{9}$$

$$C_{t+1} = C_t - \epsilon_C \frac{\partial}{\partial C} E_{TLE} \quad \text{and} \quad h_{t+1} = h_t - \epsilon_h \frac{\partial}{\partial h} E_{TLE}, \tag{10}$$

where $\epsilon_C = \epsilon_h = 0.1$ is chosen for the learning rate. We use a fast synaptic dynamic for the learning of the controller and the model such that the system adapts quickly. Assuming sensory noise, the TLE is never zero nor has a vanishing gradient such that the rule (10) produces an itinerant trajectory in the parameter space, i. e. the robot traverses a sequence of behaviors that are determined by the interaction with the environment. These behaviors are, however, waxing and waning and their time span and transitions are hard to predict.

Let us consider as a first example a robot with two wheels that is equipped with wheel velocity sensors. In the beginning the robot rests, but after a short time it autonomously starts to drive forward and backward and to turn. If a

wall is encountered such that the wheels stop the robot will immediately stop the motors and eventually drive in the free direction. A more complex example for the self-organization of *natural* behaviors is provided by the SPHERICAL robot (Fig. 1) which starts to roll around different internal axes as we will see below. Furthermore, high-dimensional systems such as snake- or chain-like robots, quadrupeds, and wheeled robots have been successfully controlled [13]. It is of particular interest that the control algorithm induces a preference for movements with a high degree of coordination among the various degrees of freedom. All the robotic implementations demonstrate the emergence of play-like behavior, which are characterized by coordinated whole body movements seemingly without a specific goal. The coordination among the various degrees of freedom arises from their physical coupling that is extracted and enhanced by the controller, because each motor neuron is adapted to be sensitive to coherent changes in all degrees of freedom due to Eq. 10. In this paper we will propose a mechanism to guide the self-organizing behaviors towards desired behaviors.

3 Guided Self-organizing Control

How can we guide the learning dynamics such that a given goal is realized by the self-organizing process? One option is to modify the lifetime of the transient behaviors depending on a given reward signal. For this purpose we can explicitly modify the frequencies of occurrence of different behaviors and obtain more of a desired and less of an undesired behavior. The prediction error ξ occurs as a factor in the learning rule (8), i. e. the lifetime of well predictable behavior is extended such that the original TLE already contains a reward for predictability in this formalism. When applying this method to the SPHERICAL robot (Fig. 1) we can, for example, achieve fast locomotion by rewarding high velocity and obtain curved driving and spinning modes when rewarding rotational velocity around the upwards axis, see [5] for more details.

A second and more stringent form of guidance will be studied in the present paper. We will formulate the problem in terms of problem-specific error functions (PSEF) that indicate an external goal by minimal values. A trivial example of such an error function is the difference between externally defined and actually executed motor actions. This is a standard control problem which, however, becomes hard if the explorative dynamics is to be preserved.

Guided self-organization (GSO) focuses on this interplay between the explorative dynamics implied by homeokinetic learning and the additional drives. The challenge in the combination of a self-organizing system with external goals becomes clear when recalling the characteristics of a self-organizing system. One important feature is the spontaneous breaking of symmetries of the system. This is a prerequisite for spontaneous pattern formation and is usually achieved by self-amplification, i.e. small noisy perturbations cause the system to choose one of several symmetric options while the intrinsic dynamics then causes the system to settle into this asymmetric state. A nonlinear stabilization of the self-amplification forms another ingredient of self-organization. These two conditions

which we will call our working regime, are to be met for a successful guidance of a self-organizing system. There are a number of ways to guide the homeokinetic controller which we will discuss in the following.

4 Guidance by Teaching

First we will describe how the problem-specific error functions (PSEF) can be integrated and then we will consider a few examples. Recall that the adaptation of the controller parameters is done by performing a gradient descent on the time-loop error. The PSEF must depend functionally on the controller parameters in order to allow the same procedure. Unfortunately, the simple sum of both gradients is likely to steer the system out of its working regime and we cannot easily identify a fixed weighting between the two gradients that would satisfy an adequate pursuit of the goal and maintaining explorativity. One reason is that the nonlinearities (cf. Eq. 3) in the TLE cause the gradient to vary over orders of magnitude. A solution to this problem can be obtained by scaling the gradient of the PSEF according to the Jacobian matrix of the sensorimotor loop such that both gradients become compatible. It turns out that this transformation can be obtained using the natural gradient with the Jacobian matrix of the sensorimotor loop as a metric. The update for the controller parameters C is now given by

$$\frac{1}{\epsilon_C} \Delta C_t = -\frac{\partial E_{TLE}}{\partial C} - \gamma \frac{\partial E_G}{\partial C} \left(L_t L_t^\top \right)^{-1}, \tag{11}$$

where E_G is the PSEF and $\gamma > 0$ is the guidance factor deciding the strength of the guidance. For $\gamma = 0$ there is no guidance and we obtain the unmodified dynamics, cf. (10).

For clarity we will start with a very simple goal, namely we want a robot to follow predefined motor actions called *teaching signals* in addition to the homeokinetic behavior. We can define the PSEF as the mismatch η_t^G between motor teaching signals y_t^G and the actual motor values, thus

$$E_G = \|\eta_t^G\|^2 = \|y_t^G - y_t\|^2. \tag{12}$$

Since y_t is functionally dependent on the controller parameters (3), the gradient descent can be performed, i.e. the derivative reads $\frac{\partial E_G}{\partial C_{ij}} = -\eta_{t,i}^G \, g_i' \, x_{t,j}$, where $g_i' = \tanh' \left(\sum_{j=1}^n C_{ij} x_{t,j} + h_i \right)$. A similarly motivated approach is homeotaxis [14], where an action error is added to the TLE as well, however the error was minimized in one step, and not along its gradient.

An evaluation of the guidance mechanism has been performed using the TwoWheeled robot, which was simulated in our realistic robot simulator LpzRobots [15]. The motor values determine the nominal wheel velocities and the sensor values report the actual wheel velocities of both wheels. We provided to both motors the same oscillating teaching signal. The resulting behavior is a mixture between the taught behavior and self-organized dynamics depending the value of γ. For $\gamma = 0.01$ the teaching signals are followed most of the time but

with occasional exploratory interruptions, especially when the teaching signals have a small absolute value. In this case the system is closer to the bifurcation point where the two stable fixed points for forward and backward motion meet. These interruptions cause the robot, for example, to move in curved fashion instead of strictly driving in a straight line as the teaching signals dictate. The exploration around the teaching signals might be useful in general to find modes which are better predictable or more active.

5 Sensor Teaching and Distal Learning

Let us now transfer the motor teaching paradigm to sensor teaching signals. This is a useful way of teaching because desired proprioceptive sensor values can be more easily obtained than motor values, for instance by passively moving the robot or parts of the robot. This kind of teaching is also commonly used when humans learn a new skill, e. g. think of a tennis trainer that teaches a new stroke by moving the arm and the racket of the learner. Thus, a series of nominal sensations can be acquired that can serve as teaching signals. Setups where the desired outputs are provided in a different domain than the actual controller outputs are called *distal learning* [8]. Usually a forward model is learned that maps actions to sensations (or more generally to the space of the desired output signals). Then the mismatch between a desired and actual sensation can be back-propagated to obtain the required change of action. The back-propagation can also be done using an inversion of the forward model which we have already at hand, see Eqs. 1 and 7. The idea is actually very simple, namely calculating motor teaching signals from sensor teaching signal using the inverted model by solving $x_t^D = M(x_{t-1}, y_{t-1}^G, \mathcal{A})$ w.r.t. y^G, cf. Eq. 2, which can in turn be inserted into Eq. 12. Afterwards we apply the motor teaching mechanism (Section 4).

The potential of this method will become more obvious in the following more complex example. We use a simulated robot named the SPHERICAL which is of relatively simple shape, but involves a complicated control problem, see Fig. 1. We will consider the goal of restricting the movements of the robot to rotations around one of its axes. The robot is actuated by three internal weights that are movable along orthogonal axes. Thus a single change in the positions of the weights results in a change of the center of mass of the robots and thus in a certain rolling movement. Control has to take into account strong inertia effects and a non-trivial map between motor actions and body movements. Let us first consider the behavior without guidance ($\gamma = 0$). From a resting initial situation, the rule (10) induces an increasing sensitivity by noise amplification until a coherent physical movements develop. Shortly afterwards a regular rolling behavior is executed which breaks down infrequently to give way for different movement patterns. In particular the rolling modes around one of the internal axes are seen to occur preferably, see Fig. 2(a,b). This modes are characterized by small sensor values for the rotation axis whereas the remaining two sensor values oscillate.

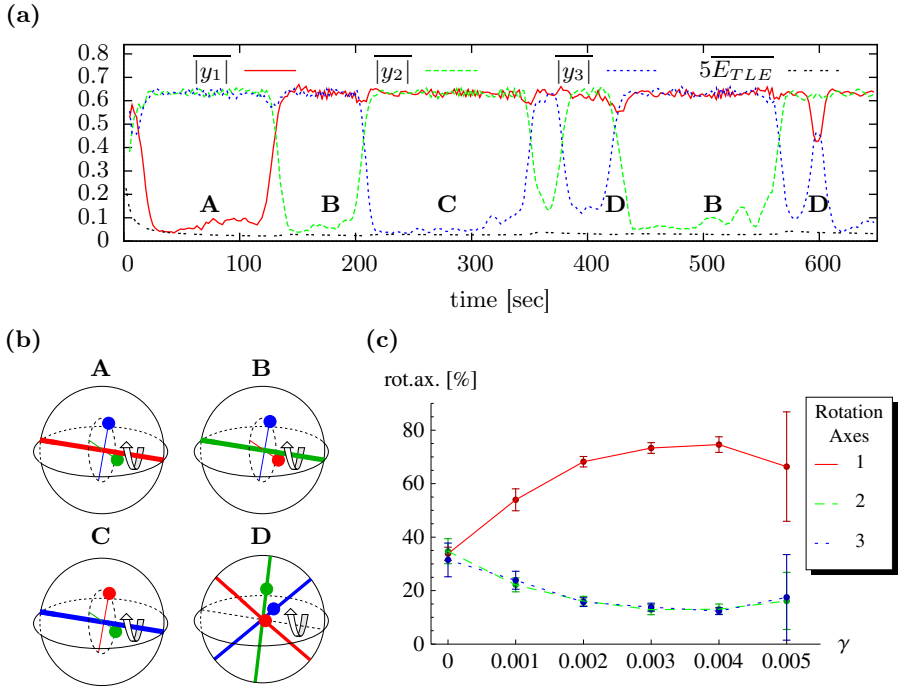


Fig. 2. The SPHERICAL robot without guidance explores its behavioral options. With guidance it prefers a specific axis of rotation. **(a)** Amplitudes of the motor value oscillations ($y_{1\ldots3}$) and the TLE (E_{TLE}) averaged over 10 sec (scaled for visibility) without guidance ($\gamma = 0$). Corresponding behaviors are indicated with letters **A-D**. **(b)** Sketch of four typical behaviors (**A-D**), namely the rolling mode around the three internal axis (**A-C**) and around any other axis (**D**); **(c)** Behavior for the distal learning task. The percentage of rotation around each of the internal axes is shown for different values of the guidance factor γ (no teaching for $\gamma = 0$). The rotation around the red (first) axis is clearly preferred for non-zero γ (mean and standard deviation are plotted for 10 runs each of a duration of 60 min).

In order to guide the robot into the rotation around the first axis we use a distal teaching signal where the first component is zero and the remaining two components contain the current sensor values such that they do not generate any learning signal (i.e. the mismatch is zero). The teaching signal vector is formally $x_t^G = (0 \;\; x_{t,2} \;\; x_{t,3})^\top$, where $x_{t,1\ldots3}$ are the sensor values at time t. As a descriptive measure of the behavior, we used the index of the internal axis around which the highest rotational velocity was measured at each moment of time. Figure 2(c) displays for different values of the guidance factor (γ) and for each of the axes the percentage of time it was the major axis of rotation. Without guidance there is no preferred axis of rotation as expected. With distal learning the robot shows a significant preference for a rotation around the first axis up to 75 %. For overly strong teaching, a large variance in the performance occurs. This is caused by a too strong influence of the teaching

signal on the learning dynamics. Remember that the rolling modes can emerge due to the fine regulation of the sensorimotor loop to the working regime of the homeokinetic controller, which cannot be maintained for large values of γ. We may ask why is it not possible to force the controller to stay in the rotational mode around the first axis? When the robot is in this rotational mode the teaching signal is negligible. However, the controller's drive to be sensitive will increase the influence of the first sensor such that the mode becomes unstable again.

To summarize, the SPHERICAL robot with the homeokinetic controller can be guided to move mostly by rotation around one particular axis, by specifying the constancy of a single sensor as a teaching signal.

6 Guidance by Cross-Motor Teaching

Finally we will propose a guidance mechanism with internal teaching signals. As an example we want to influence the controller to prefer a mirror-symmetry in the motor patterns. This can be achieved by using the motor value of one motor as the teaching signal for another motor and vice versa. For two motors, this can be expressed as: $y^G_{t,1} = y_{t,2}$ and $y^G_{t,2} = y_{t,1}$, where y^G_t is again the nominal motor value vector, see Eqs. 11 and 12. This self-supervised teaching induces soft constraints which reduce the effective dimension of the sensorimotor dynamics and thus guide the self-organization along a sub-space of the original control problem.

Let us consider the TWOWHEELED robot again and suppose the robot should move mostly straight, not get stuck at obstacles or in corners and cover substantial parts of its environment. We will see that all this can be achieved by a simple guidance of the homeokinetic controller where both motors are mutually teaching each other.

For experimental evaluation we placed the robot in an environment cluttered with obstacles and performed many trials for different values of the guidance factor. In order to quantify the influence of the guidance we recorded the trajectory, the linear velocity, and the angular velocity of the robot. We expect an increase in linear velocity because the robot is to move straight instead of turning. For the same reason the angular velocity should be lowered. In Fig. 3 the behavioral quantification and a sample trajectory are plotted. Additionally the relative area coverage is shown, which reflects how much more area of the environment was covered by the robot with guidance compared to freely moving robot. As expected, the robot shows a distinct decrease in mean turning velocity and a higher area coverage with increasing values of the guidance factor. Note that the robot is still performing turns and drives both backwards and forwards and that it does not get stuck at the walls, as seen in the trajectory in Fig. 3(b), such as sensitivity (exploration) and predictability (exploitation) remain.

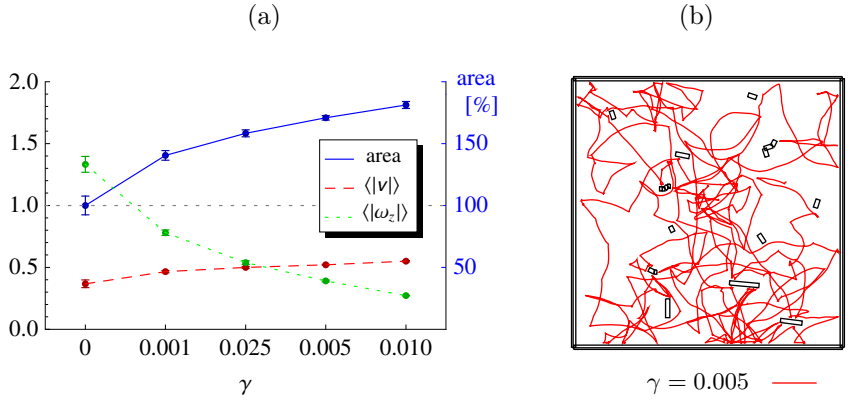


Fig. 3. Behavior of the TwoWheeled robot when guided to move preferably straight. **(a)** Mean and standard deviation (of 5 runs each 20 min) of the area coverage (**area**), the average velocity $\langle|\boldsymbol{v}|\rangle$, and the average angular velocity $\langle|\omega_z|\rangle$ for different values of the guidance factor γ. Area coverage (box counting method) is given in percent of the case without guidance ($\gamma = 0$) (**right axis**). The robot is driving straighter and its trajectory covers more area for larger γ; **(b)** An example trajectory of the robot with $\gamma = 0.005$.

7 Discussion

We have presented here two new methods for guiding self-organizing behavior that are based on teaching signals. Desired motor patterns were specified by means of an error function that was integrated into the learning dynamics. The strength of guidance can be conveniently adjusted. Because teaching information is often given in the sensor space whereas learning is performed in the motor representation, a transformation is necessary which is obtained from the adaptive internal world model. The feasibility of both approaches was demonstrated by robotic experiments.

We introduced cross-motor teachings in order to be able to specify relations between different motor channels. If it is known or desired that certain degrees of freedom of a robot should move in a coherent way, e. g. symmetrical or anti-symmetrical, then these relation can be injected as soft constraints that reduce the effective dimensionality of the system. As an example, the TwoWheeled robot showed that by enforcing the symmetry between the left and right wheel the behavior changes qualitatively to straight motion.

The exploratory character of the controller is nevertheless retained and helps to find a behavioral mode even if the specification of the motor couplings is partially contradictory. The resulting behaviors are not enforced by the algorithm. For example the TwoWheeled robot can choose freely between driving forward or backward whereas in direct teaching the direction of driving is obviously dictated by an external teacher. Furthermore, it is evident that the robot remains sensitive to small perturbations and continues to explore its environment.

Guided self-organization using cross-motor teachings shares some properties with other approaches to autonomous robot control such as evolutionary algorithms [16] and reinforcement learning (RL) [17]. Evolutionary algorithms can optimize the parameters of the controller and are able to produce the same behaviors as we found in this study, cf. [18,19,20]. A critical experiment would investigate high-dimensional systems that cannot be decomposed into identical components.

A further difference is that self-organizing control is merely modulated by guidance, whereas evolutionary algorithms tend to converge to a static control structure. RL uses discrete actions or a parametric representation of the action space. In either case, high-dimensional systems will cause slow convergence. Preliminary experiments with a chain-like robot (cf. [13]) show a clear advantage of cross-motor teaching in comparison to generic RL although similar relations among the actions in RL compensate part of this drawback. Natural actor-critics [21] may bring a further improvement of the RL control, but natural gradients can also be incorporated here. A decisive advantage of cross-motor teaching may be that goal-directed behaviors emerge within the self-organization of the dynamics from a symbolic description of the problem and do not need continuous training data such as in imitation learning [22].

It is, however, clearly an interesting option to adapt cross-motor teaching to an imitation learning scenario. Although delayed rewards are still non-trivial for continuous domains, RL can cope with them in principle, while the guidance with rewards [5] requires instantaneous rewards.

Acknowledgment. The project was supported by the BMBF grant #01GQ0432.

References

1. Der, R., Hesse, F., Martius, G.: Rocking stamper and jumping snake from a dynamical system approach to artificial life. Adapt. Beh. 14, 105–115 (2006)
2. Hesse, F., Martius, G., Der, R., Herrmann, J.M.: A sensor-based learning algorithm for the self-organization of robot behavior. Algorithms 2(1), 398–409 (2009)
3. Stefano, N.: Behaviour as a complex adaptive system: On the role of self-organization in the development of individual and collective behaviour. ComplexUs 2(3-4), 195–203 (2006)
4. Tani, J.: Learning to generate articulated behavior through the bottom-up and the top-down interaction processes. Neural Networks 16(1), 11–23 (2003)
5. Martius, G., Herrmann, J.M., Der, R.: Guided self-organisation for autonomous robot development. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 766–775. Springer, Heidelberg (2007)
6. Prokopenko, M.: Guided self-organization. HFSP Journal 3(5), 287–289 (2009)
7. Der, R.: Self-organized acquisition of situated behavior. Theory in Biosciences 120, 179–187 (2001)
8. Jordan, M.I., Rumelhart, D.E.: Forward models: Supervised learning with a distal teacher. Cognitive Science 16(3), 307–354 (1992)
9. Nolfi, S., Floreano, D.: Learning and evolution. Auton. Robots 7(1), 89–113 (1999)

10. Di Paolo, E.: Organismically-inspired robotics: Homeostatic adaptation and natural teleology beyond the closed sensorimotor loop. In: Murase, K., Asakura, T. (eds.) Dyn. Systems Approach to Embodiment and Sociality, pp. 19–42 (2003)
11. Williams, H.: Homeostatic plasticity in recurrent neural networks. In: Schaal, S., Ispeert, A. (eds.) From Animals to Animats: Proc. 8th Intl. Conf. on Simulation of Adaptive Behavior, vol. 8. MIT Press, Cambridge (2004)
12. Der, R., Martius, G.: From motor babbling to purposive actions: Emerging self-exploration in a dynamical systems approach to early robot development. In: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J.C.T., Marocco, D., Meyer, J.-A., Miglino, O., Parisi, D. (eds.) SAB 2006. LNCS (LNAI), vol. 4095, pp. 406–421. Springer, Heidelberg (2006)
13. Der, R., Martius, G., Hesse, F., Güttler, F.: Videos of self-organized behavior in autonomous robots (2009), http://robot.informatik.uni-leipzig.de/videos
14. Prokopenko, M., Zeman, A., Li, R.: Homeotaxis: Coordination with persistent timeloops. In: Asada, M., Hallam, J.C.T., Meyer, J.-A., Tani, J. (eds.) SAB 2008. LNCS (LNAI), vol. 5040, pp. 403–414. Springer, Heidelberg (2008)
15. Martius, G., Hesse, F., Güttler, F., Der, R.: LpzRobots: A free and powerful robot simulator (2009), http://robot.informatik.uni-leipzig.de/software
16. Nolfi, S., Floreano, D.: Evolutionary Robotics. In: The Biology, Intelligence, and Technology of Self-organizing Machines. MIT Press, Cambridge (2001); 1st Print (2000), 2nd Print (2001)
17. Sutton, R.S., Barto, A.G.: Reinforcement learning: Past, present and future. In: SEAL, pp. 195–197 (1998)
18. de Margerie, E., Mouret, J.B., Doncieux, S., Meyer, J.A.: Artificial evolution of the morphology and kinematics in a flapping-wing mini UAV. Bioinspiration and Biomimetics 2, 65–82 (2007)
19. Ijspeert, A.J., Hallam, J., Willshaw, D.: Evolving Swimming Controllers for a Simulated Lamprey with Inspiration from Neurobiology. Adaptive Behavior 7(2), 151–172 (1999)
20. Mazzapioda, M.G., Nolfi, S.: Synchronization and gait adaptation in evolving hexapod robots. In: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J.C.T., Marocco, D., Meyer, J.-A., Miglino, O., Parisi, D. (eds.) SAB 2006. LNCS (LNAI), vol. 4095, pp. 113–125. Springer, Heidelberg (2006)
21. Peters, J., Vijayakumar, S., Schaal, S.: Natural Actor-Critic. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 280–291. Springer, Heidelberg (2005)
22. Peters, J., Schaal, S.: Natural Actor-Critic. Neurocomputing 71(7-9), 1180–1190 (2008)

Slime Mold Inspired Chemical Sounding

R. Andrew Russell

Intelligent Robotics Research Centre
Monash University, Clayton, VIC3800, Australia
andy.russell@eng.monash.edu.au

Abstract. This paper describes the biological motivation, design and testing of a novel obstacle sensor system based on the chemical sounding mechanism observed in slime molds. The sensor system emits a volatile sounding chemical and adjacent obstacles reduce the dispersion of the chemical. The resulting increase in chemical concentration is detected and interpreted as indicating an obstacle. To demonstrate the capabilities of the sensor it was mounted on an omnidirectional mobile base. Using direct sensor/actuator connection appropriate for a fungus-like organism the robot was able to demonstrate simple obstacle avoidance.

Keywords: Slime mold, biomimetics, chemical sounding.

1 Introduction

In robotics research it is widely recognised that biology and particularly animals provide useful inspiration in terms of mechanisms, algorithms and techniques [1]. However, there are valuable lessons to be learnt from plants, fungi and similar organisms. These organisms can sense chemicals, light clues, touch, gravity, temperature, humidity and even magnetism [2]. They can also respond to sensory stimuli in a number of different ways, including by moving. Some movements involved in catching animals for food, dispersing seeds, defending against attack and cross-pollination require quite rapid responses. However, plants, fungi and slime molds have only a rudimentary nervous system at best and so their 'processing' of sensory signals and patterns of resulting behaviour are naturally limited. Because the structure and life-style of these organisms are so different from animals it could be expected that they will have evolved some unique capabilities.

Although quite primitive in terms of evolution, slime molds are of interest for roboticists. They have been used as a model for self-assembly [3] and their ability to perform computations has also been investigated [4]. One of the other interesting capabilities of slime molds is their ability to detect and respond to obstacles using chemical sounding. This paper describes a project to implement chemical sounding in a robotic system. There is interest in building ever smaller robots and currently popular robotic sensors including vision, ultrasonic and laser scanners may prove difficult to miniaturise for use on these robots. For these kinds of systems novel forms of sensing, including chemical sounding may find an application. The project

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 62–71, 2010.

also investigated robot structures and direct sensor/actuator coupling appropriate for modelling chemical sounding in slime molds.

2 Chemical Sounding

Sensing methods can be classified as either active or passive techniques [5]. In passive sensing such as mammalian vision the receiver only collects the sensory stimuli that originate elsewhere. Active sensing involves the sensor system emitting some form of energy or material and then receiving it back again after it has been modified by the environment (bat echo location would be an example). The form of modification allows the receiver to infer something about its external environment. In chemical sounding volatile chemicals are released, modified by the surrounding environment and then detected.

2.1 Chemical Sounding in Nature

The slime mold *Phycomyces* is a type of organism that grows in the form of thin filaments. Reproduction starts with the growth of sporangiophores (fruiting bodies) that propagate perpendicular to the substrate. A sphere containing spores develops on the end of the sporangiphore and the spores are released from the tip. In order to distribute the spores they must be released into air currents beyond the thin layer of stagnant air close to the substrate surface. It is hypothesised that the sporangiophores emit a chemical signal and use the increased concentration caused by obstacles to grow away from any nearby surfaces and clear of stagnant air [6].

2.2 Chemical Sounding in Nature

The slime mold relies on diffusion to transport its sounding chemical. This will work in a reasonable time frame for distances of the order of 1mm. Diffusion of chemical from a point source is governed by the following equation [7]:

$$I(r,t) = \frac{F}{4\pi Dr} erfc\left(\frac{r}{(4Dt)^{\frac{1}{2}}}\right) \tag{1}$$

where:

D = diffusion constant (m^2/s) for the specific chemical (typical value is $2x10^{-5}$ in air),
t = time,
r = radius, and
F = continuous point source intensity.

If it takes 1 second for the chemical concentration to reach a particular value 10mm from the source then approximately the same concentration will be observed at 40mm after 60 seconds, 80mm after an hour and 95mm after one day. From this it appears that diffusion could not be used in robotic experiments because it is too slow and also influenced by natural airflow fluctuations at the scale of easily constructed robots

(about 100mm). For this reason in this experiment the sounding chemical was actively dispersed using a small cooling fan. In order to test the viability of chemical sounding at the scale of easily manufactured robots the prototype sensor illustrated in Fig. 1 was constructed.

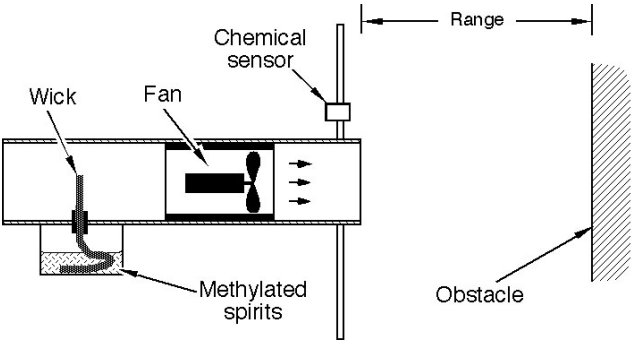


Fig. 1. The prototype chemical sounding sensor

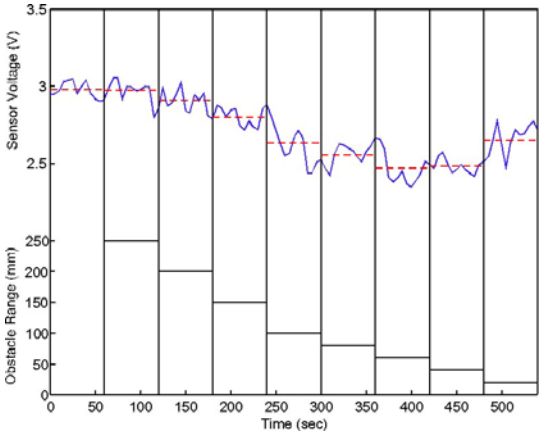


Fig. 2. Sounding sensor response to a 10cm by 10cm target at varying distances from the sensor. During the first 60 seconds no target was present. The dashed lines show the average sensor response over each 60 second period where the target distance was held constant.

As shown in Fig. 2 the results from the prototype sensor indicate that chemical sounding can be used to detect adjacent obstacles.

3 The Slime Mold Robot

When *Phycomyces* is growing its sporangiphore (which employs chemical sounding) in many ways it behaves like a plant and is rooted in one place. However, it was

decided to implement the chemical sounding system on a mobile robot in order to clearly demonstrate the action of the sensor and control system.

The structure of an organism/robot and the way it interacts with its environment can have a strong bearing on its ability to perform tasks, and therefore the apparent 'intelligence' of the creature. In the case of walking machines it has been pointed out by Blickhan, et al. [8] that creating self-stable mechanisms may reduce the computational burden and control effort required. Similar ideas in a more general problem-solving context are discussed by Barras [9]. Therefore, to investigate the capabilities of chemical sounding as it occurs in slime molds it is necessary to match the actuation and sensor/actuator coupling implemented on the robot with the biological correlate. Because the slime mold has many of the characteristics of a plant it is instructive to consider the characteristics of plant tropisms.

A list of the characteristics of plant tropisms adapted from Hart [10]:

a) movement is initiated by a strongly directional stimulus

b) a stimulus is usually detected by a specific receptor and this evokes a specific movement via a direct pathway

c) the response involves a change of organ orientation and the direction of which is related to the direction of the stimulus

d) a tropic growth response usually involves stimulation and inhibition and often occurs in regions of the organ other than those directly receiving the stimulus

e) there is usually a non-linear relationship between the level of the stimuli and the extent of the response, and often some form of sensory adaptation to the stimulus.

To provide the robot with an overall directional motivation it was made sensitive to light so that it could be attracted towards sources of light. This could be equated to growth of the slime mold's sporangiphore. In addition, the robot was equipped with a chemical sounding sensor that produced an avoidance response to avoid collisions. The light and obstacle stimuli are both directional as required by point (a) above. It is assumed that the sporangiphore is equally sensitive to obstacles at any point around its axis and could respond by bending in any direction. For this reason the robot was built with omnidirectional sensing and movement capabilities. Direct connections were made between sensors and actuators (point (b) above) and the direction of movement was related to the sensing direction of the sensor (point (c)). Sensors and actuators were physically separated (point (d) above) and the introduction of sensor thresholds resulted in a non-linear relationship between stimuli and response (point (e)). The major elements of the resulting robot are discussed below.

3.1 Control Algorithm

To make the robot modular and extensible it was formed from a number of identical facets. Each facet contains the full set of sensors (light and chemical sounding) and a wheel drive unit. Control consists of direct sensor/actuator connections as follows:

- If a facet detects a light intensity greater than the minimum of all facets then the associated motor drives forward a distance proportional to light intensity and this moves the facet towards the light source.
- If a facet detects an obstacle the motor drive (or two drives if there are an odd number of facets) opposite the facet is driven forwards and this moves the robot away from the obstacle. The distance moved is about twice the maximum distance that the robot moves in response to the light sensor.

Because of the relatively slow response time of the chemical sounding sensor the robot control program consisted of a repeated sequence of sense/move operations. In order to implement omnidirectional sensing and movement using the concept of facets the sensors and actuators were implemented as follows.

3.2 Light Sensing

Matched Light Dependent Resistors (LDRs) were used to detect the light intensity falling on each of the facets of the robot. The receptive field of an LDR is close to 180˚. A diffuser made from a section of ping-pong ball extended the receptive field beyond 180˚ (Fig. 3). The prototype robot was implemented with 3 facets angled at 0˚, 120˚ and 240˚. Therefore, even a point source of light can be detected by two sensors at more than half of the possible headings around the robot. A more widely dispersed light source, such as the table lamp used in these experiments, would usually be detected by two sensors. Sensing by two sensors allows the robot to move towards the light by a comparatively smooth zigzag path.

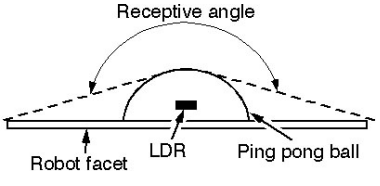


Fig. 3. Receptive field of the light sensor

3.3 Chemical Sounding

Each of the robot's facets also incorporates a chemical sounding sensor. A single chemical source (methylated spirits container with a wick) and fan feeds methylated spirits/air mixture into a central plenum. A vent on each facet releases a stream of chemical/air from the plenum and build-up of methylated spirits vapour is monitored by a TGS2600 gas sensor (Taguchi tin oxide gas sensor manufactured by Figaro Engineering Inc.).

3.4 Omnidirectional Wheel Drive System

To allow direct connection between each sensing facet of the robot and the motor drive system it was necessary for the robot to be able to move in the direction of each

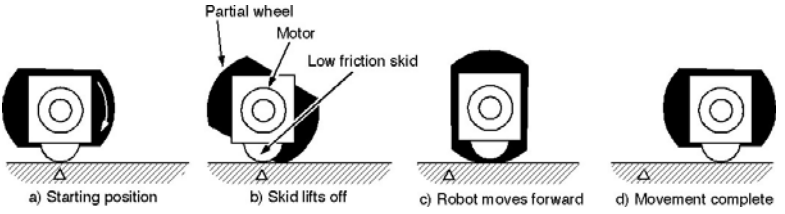


Fig. 4. Motor/wheel unit for the intermittent omnidirectional drive

facet. For simple plant-like actuation it was desirable to eliminate the necessity for the robot to turn. This required an omnidirectional drive system and a simple and inexpensive system was developed for this project.

Each facet of the robot is associated with a drive unit (Fig. 4) consisting of a geared motor (200:1) directly coupled to a rubber wheel. The wheel incorporates two flat areas and when the wheel rolls onto these areas the wheel unit is lowered onto a low friction skid. A microswitch ensures that the wheel unit always stops in the low-friction position shown in Fig. 4a. When balanced on the skid one of the other wheel units can pull the robot in its direction. Thus, with three facet/motor units the robot can move at three headings spaced 120˚ apart, or at a heading in between by combining the action of two motor units activated sequentially. The prototype robot (Fig. 5) consists of three identical facets. Sensor reading and motor control is coordinated by an Arduino Pro Mini microcontroller (http://arduino.cc/en/Main/ArduinoBoardProMini/) with a small amount of support electronics. The major source of power consumption for the slime mold robot is the heaters in the three TGS2600 chemical sensors which together draw 126mA at 5V (630mW). The microcontroller and support electronics consumes an additional 42mA (210mW) and when running the drive motors draw 15mA (75mW).

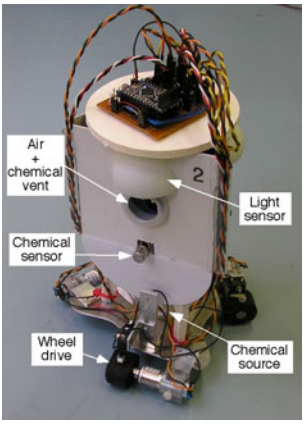


Fig. 5. The chemical sounding robot

4 Results

The robot responds to two aspects of its environment. Initially, the light seeking and obstacle avoidance systems were tested in isolation. When these systems were shown to be functioning correctly they were combined.

4.1 Phototaxis

In order to test the light seeking capabilities of the robot, it was started about 680mm from a table lamp. Different starting positions and a number of starting orientations were tested. In each case the robot successfully made its way towards the light. Three recorded trajectories of the robot are shown in Fig.6. In this case the robot was released with three different starting orientations and from two different starting locations.

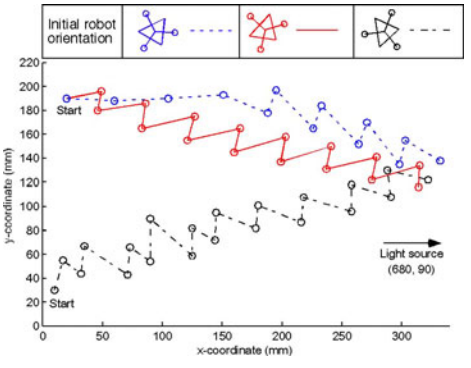


Fig. 6. Robot response to a light stimulus with no obstacles present

4.2 Chemotaxis

When deployed on the mobile robot for obstacle avoidance each chemical sensor is sampled 30 times at one second intervals. These samples are then averaged to give the sensor output. On start up in an area free of obstacles an initial group of sensor reading is taken. To avoid spurious responses a threshold equivalent to a change of sensor output of 0.24V with respect to the initial value is set. A sensor change of greater than 0.24V is taken as evidence of an obstacle. With this threshold the obstacle sensor can detect a 160mm by 140mm rectangular obstacle at a range of 100mm from the sensor.

4.3 Combined Response

In the first test of the robot response with both the light and chemical sounding sensors active the experimental area was set up as shown in Fig. 7. The robot started at location (0mm, 0mm). A table lamp served as the light source and it was located at (800mm, 200mm). The light illuminated two of the robot's facets (facets 2 and 3). After the chemical sounding sensor had stabilised and established sensor readings corresponding to 'no obstacles present' an obstacle 40mm by 140mm by 160mm made of polystyrene foam was positioned adjacent to the robot. When the robot was

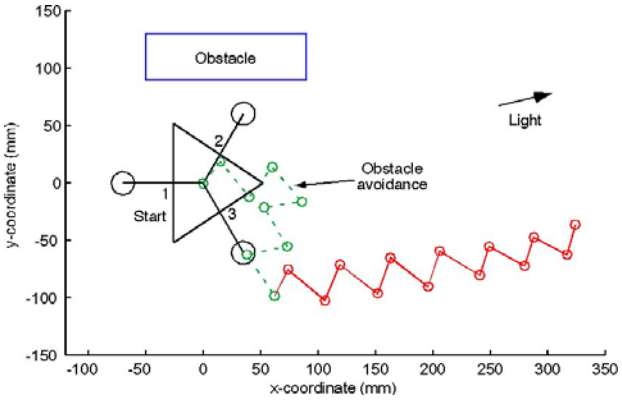


Fig. 7. Starting close to an obstacle the robot moves clear before continuing towards the light

enabled the chemical sounding sensors on facets 1 and 2 detected the obstacle and the robot manoeuvred away from the obstacle. Once clear of the obstacle the robot homed-in on the light. In this test the time response of the sensor was less critical because the time required to set up the experiment allowed the chemical concentration around the robot to accommodate to the presence of the obstacle.

As a final test the polystyrene foam obstacle was positioned between the robot and the lamp so that the robot would collide with it if it moved directly towards the light. Fig. 8 shows that initially the robot headed towards the light. After 4 sense/movement cycles the sensor detected the obstacle and the robot took an avoiding action (indicated by the dashed line). Once clear of the obstacle the robot then continued to track towards the light source. In this experiment, because of the movement of the robot, the chemical sounding sensor had less time to accommodate to the presence of the

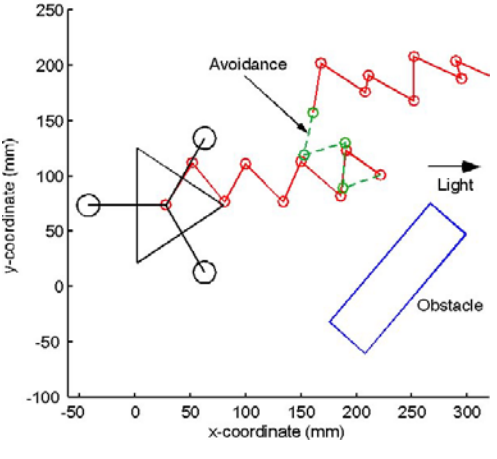


Fig. 8. Avoiding an obstacle while attracted to a light source

obstacle. However, the robot successfully detected and avoided the obstacle. It should be noted that due to imperfections in its construction there was a slight tendency for the robot to rotate counter clockwise as it moved.

5 Conclusions

This project has demonstrated the implementation of biologically inspired chemical sounding in a robot system. The chemical sounding sensor system can detect obstacles at a range of 10cm. Simple direct connection between sensors and actuators together with broadly omnidirectional sensing and actuation mimic the associated capabilities of the slime mold *Phycomyces.* Using the chemical sounding sensor the robot is able to demonstrate simple obstacle detection and avoidance. Because of the direct sensor/actuator connection obstacle avoidance is very unsophisticated. The avoidance system cannot deal with situations where attractive and repulsive signals balance or environments with complex obstacles fields where the robot can become trapped. For a robot that is 20cm high and 15cm diameter the sensing range of 10cm is adequate. However, the sensor response time of 30 seconds would be too slow for many mobile robot applications. The chemical sounding sensor also works best in environments where there is very little air movement.

In spite of the current limitations of the chemical sounding sensor, in the future there may be a niche for this mode of robot sensing. The potential dimensions of robots could range from the size of a bacterium to that of an elephant and working environments may include underwater, underground and terrestrial. Within this spectrum many different modes of sensing will be required. Currently, the majority of mobile robots use computer vision, ultrasonics or laser range finders as their main navigational sensors. These sensing modes are only viable for a limited spectrum of robot applications. For example, as robot dimensions are reduced below 1cm or the surroundings become turbid they will be less able to function effectively. In these situations other sensing modes such as chemical sounding may be more viable. At this early stage of development the sensor response time is quite long. However, with further refinement, particularly involving improvements to the chemical sensors, this time can be reduced.

References

1. Pfeifer, R., Lungarella, M., Iida, F.: Self-organization, embodiment, and biologically inspired robotics. Science 318, 1088–1093 (2007)
2. Simons, P.: The Action Plants: Movement and Nervous Behaviour in Plants. Blackwell, Oxford (1992)
3. Ishiguro, A., Shimizu, M., Kawakatsu, T.: A Modular Robot That Exhibits Amoebic Locomotion. Robotics and Autonomous Systems 54, 641–650 (2006)
4. Adamatzky, A.D., Jones, D.: Towards *Physarum* Robots: Computing and Manipulating on Water Surface. Journal of Bionic Engineering 5, 348–357 (2008)
5. Middelhoek, S., Hoogerwerf, A.C.: Classifying solid-state sensors: the sensor effect cube. Sensors and Actuators 10, 1–8 (1986)

6. Cohen, R.J., Jan, Y.N., Matricon, J., Delbrück, M.: Avoidance response, house response and wind responses of the sporangiophore of *Phycomyces*. The Journal of General Physiology 66, 67–95 (1975)
7. Dusenbery, D.B.: Sensory Ecology. W.H. Freeman and Company, New York (1992)
8. Blickham, R., Seyfarth, A., Geyer, H., Grimmer, S., Wagner, H., Günther, M.: Intelligent mechanics. Philosophical Transactions of the Royal Society A 365, 199–220 (2007)
9. Barras, C.: Hints of intelligence in dumb drop of oil. New Scientist 205(2744), 8–9 (2010)
10. Hart, J.W.: Plant Tropisms and other Growth Movements. Unwin Hyman Ltd., London (1990)

A Conserved Network for Control of Arthropod Exteroceptive Optical Flow Reflexes during Locomotion

Daniel Blustein and Joseph Ayers

Marine Science Center, Northeastern University, Nahant, MA 01908, USA
dblustein@husky.neu.edu

Abstract. We have developed an exteroceptive reflex network of sensory interneurons and command neurons that simulates arthropod optical reflexes based on current ethological and neurophysiological models. The simple neural network was instantiated in software with discrete-time map-based neurons and synapses and can mediate four forms of optomotor reflexes in arthropods: (1) translational responses to pure translational optic flow; (2) rotational responses to pure angular rotation and (3) combinations of translation and rotation that occur during obstacle avoidance and (4) yaw. Simple neural networks are well suited for controlling robots and can be used to test neurophysiological hypotheses, particularly related to sensory fusion in arthropods.

1 Introduction

Animals make use of optomotor responses that integrate the optical flow input of their moving surroundings into their behavioral output. The role of optical flow information is a key research thrust within both biology and robotics. The biologist Jacques Loeb introduced the tropism theory of animal conduct which describes the behavior of symmetrical organisms driven by bilateral sensors [1]. Braitenberg [2] promoted the importance of decussating fibers: sensory information that inputs contralaterally into the motor system of an organism. Using simple vehicles Braitenberg illustrated how layered decussating and ipsilaterally projecting sensors were capable of mediating most positive and negative taxic behavior [3].

In walking arthropods such as crustaceans, the ethological and neurophysiological bases for optomotor responses have been well studied. Crabs exposed to the angular rotation generated by a rotating striped circular arena turn in place in an attempt to stabilize their surroundings [4]. Lobsters placed on treadmills within moving striped bars walk forward when exposed to rearward translational optical flow and backward with anterior translational flow. The walking direction and speed of the lobster depends on the respective characteristics of the optical flow [5]. Combinations of translational and angular flow elicit yawing movements to correct for disturbances during locomotion. We have previously shown that these responses can be mediated by a simple decussating network [6].

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 72–81, 2010.

This model can explain optomotor responses in flying arthropods including the fly which generates torque in the rotational direction of an imposed optical flow stimulus in each axis [7,8]. Other observations from flying insects however are unexplained by the current model and serve the reformation of the model.

Visually-mediated obstacle avoidance behaviors in flying arthropods including the centering response in bees [9], saccades in flies [10], and correctional steering maneuvers in locusts [11] are unsupported by the previously proposed model. A centering response occurs in bees as they fly through a tunnel: they slow as the tunnel narrows or if optical flow is experimentally increased [12]. This occurs as optical flow is balanced bilaterally and suggests the presence of high threshold lateral flow detectors that mediate obstacle avoidance. In a saccade, a fly turns away from a region of high optical flow or visual expansion [13]. Locust lobula giant movement detector (LGMD) neurons responsive to looming objects are thought to utilize edge-detection by way of lateral inhibition [14] to elicit an avoidance response via the descending contralateral movement detector interneuron (DCMD) [15]. Even though the avoidance of areas of high optical flow and of looming objects are distinct mechanisms that allow for successful locomotion, they represent functionally interrelated components of optomotor reflex networks. During forward locomotion, for example, a looming object will also present an area of increased translational optical flow. We have chosen to incorporate high threshold optical flow detectors into our network to account for the described behavioral observations (Fig. 1).

As in lobsters [5], the rate of optical flow effects the level of motor output in flying arthropods. Changes in wing beat frequency in tethered locusts [16] and flight speed adjustments in free flying bees [17] are both modulated by optical flow. However, there are notable differences between flying and walking arthropods at the effector level [18]. Backward walking represents a change in the pattern of coordination relative to forward walking; direction of flight does not share this characteristic. Transitions between forward and backward flight in insects occur through changes in the tilt of the wing stroke plane [19]. This difference is illustrated in locusts as progressive (ie. front to back) fields of optical flow initiate forward flight but regressive flow fields inhibit flight entirely [16]. Another difference between walking and flying arthropods is in the biomechanics of steering. Walking arthropods steer by independently controlling left and right sets of walking legs. Flight control though, is the result of the input of visual interneurons onto various elements of the motor system including the head, wings, abdomen, and halteres [20]. Production of yaw torque in flies is nearly passive as the wing's high angle of attack on one side creates high drag and a turn to that side [19,21]. There is also evidence for differences in wingstroke amplitude underlying yaw control [13]. Regardless of these differences in the nature of the effectors, the conserved features of walking and flying arthropod systems can be used to inform robotic implementation of optical processing using simple neural networks.

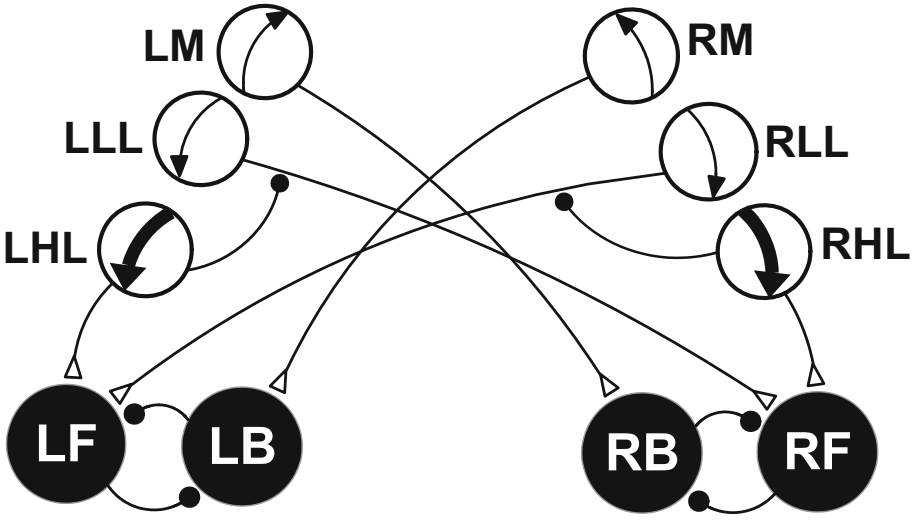


Fig. 1. *Proposed optical flow neural network for arthropods.* White circles are sensory neurons receiving optical flow information from the eye with arrows denoting directional sensitivity. Sensory neuron signals are passed through synapses (lines) to the command neurons (black circles). Lines with open triangle ends represent excitatory connections; filled circles ends, inhibitory connections. Optical flow sensory interneurons: **LM**, left medial; **LLL**, left low threshold lateral; **LHL**, left high threshold lateral; **RM**, right medial; **RLL**, right low threshold lateral; **RHL**, right high threshold lateral. Command neurons: **LF**, left forward; **LB**, left backward; **RF**, right forward; **RB**, right backward.

2 Methods

We are developing neural circuit based controllers for both walking and flying biomimetic robots and plan to use optical flow sensors to modulate behavioral output. Our goal in the present experiments is to evaluate a neuronal circuit controller which will use optical flow to control both walking and flight. Neurophysiological studies have identified direction-sensitive sensory neurons in the optic nerve of crayfish [22] and in the visual systems of various flying insects (in locust: [23]; in honeybee: [24]; in butterfly: [25]; in hawk moth: [26]; in fly: [27]). The sensor we will use for transduction of optical flow is an analog optical flow chip.[1] In the descending fibers of the crayfish, Bowerman and Larimer [28] discovered command neurons for various stereotyped behaviors including forward and backward locomotion. In the robot controllers we use similar commands to activate and modulate the central pattern generators (CPGs). While distinct commands for locomotory direction are compatible with terrestrial robots, we are evaluating commands for wing plane alterations to control the direction of flight in flying robots.

[1] Centeye, Inc, Washington, DC.

Ayers and Rulkov [6] presented a hypothetical neural network based on decussation to account for translational and rotational components of optical flow responses. Since this network lacks a mechanism for obstacle avoidance, we have added high threshold lateral flow detectors inspired by Srinivasan et al. [12] to mediate this function (Fig. 1).

Our proposed neural network is implemented as a simulation in LabVIEW[2] using a discrete time map-based (DTM) model for each component neuron and synapse [29]. The individual neurons and synapses are instantiated as LabVIEW Virtual Instruments and connected by LabVIEW "wire" in the G Programming language. A complete explanation of the neuron and synapse equations and their software implementation has been previously described [30]. The DTM mathematical model is well suited for robotic implementation because it uses computationally-efficient difference equations that mimic neural dynamics rather than processor-intensive differential equations that model ionic conductances. It can approximate a variety of neural integrative processes including tonic firing, bursting, intrinsic silence and chaotic activity as well as synaptic integration and dynamics.

3 Results

The neural simulation was run with a variety of different optical flow milieus. Under purely translational optical flow propagating from tail to head in the medial direction, the corresponding optical flow sensory interneurons fire and excite both backward commands (Fig. 2). This output matches the observed behavioral response under conditions of medial optical flow resulting from a rear to front translational optical flow stimulus [5].

In a lobster exposed to purely angular optical flow in the clockwise direction, medial left and low threshold lateral right sensory interneurons fire initiating forward walking on the left side and backward walking on the right. This causes the animal to rotate clockwise in place (Fig. 3) as described by Bethe [4].

By integrating angular and translational optical flow, animals are able to avoid obstacles and maintain a desired course. If a forward-flying bee approaches an obstacle on its right side (Fig. 4A), the high rate of optical flow will cause the corresponding high-threshold sensory interneuron to fire (Fig. 4C). This excites the ipsilateral forward command while simultaneously partially inhibiting the connection between the low-level lateral detector and the contralateral forward command resulting in an avoidance turn while maintaining forward flight (Fig. 4).

Wind and water currents present additional optical flow situations that arthropods encounter. When a forward walking lobster is pushed off course by surge, a yawing response is observed (Fig. 5, left panel). With surge coming from the left side, the lobster experiences translational plus angular optical flow on the ipsilateral side and translational minus angular flow on the side contralateral to the surge. This results in a compensatory turn into the surge in order to maintain the

[2] Version 2009, National Instruments, Austin, TX.

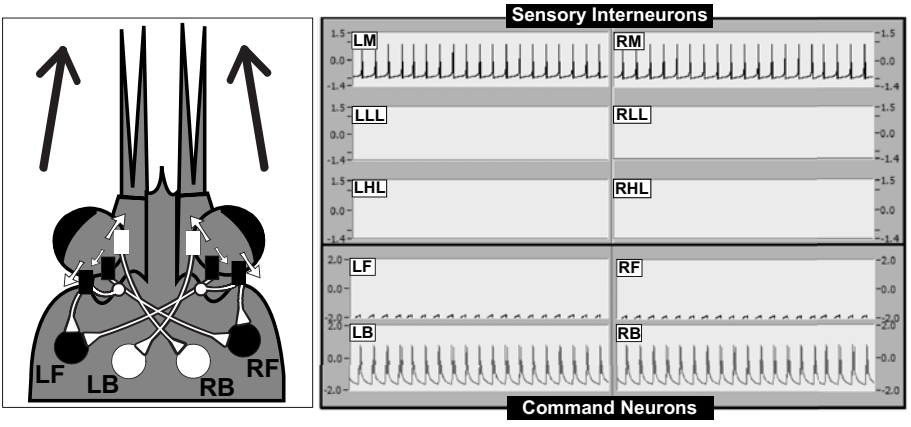


Fig. 2. *Medial translational flow initiates backward walking.* **Left panel.** Representation of optical flow neural network. White rectangles and circles represent active sensory interneurons and walking commands, respectively. Black arrows mark direction of optical flow and white arrows denote directional sensitivity of sensory neurons. For clarity, reciprocal inhibitory connections between opposing walking commands have been omitted in this and subsequent figures. **Right panel.** Neural activity of sensory interneurons and walking commands. Neural network and abbreviations are described in Fig. 1.

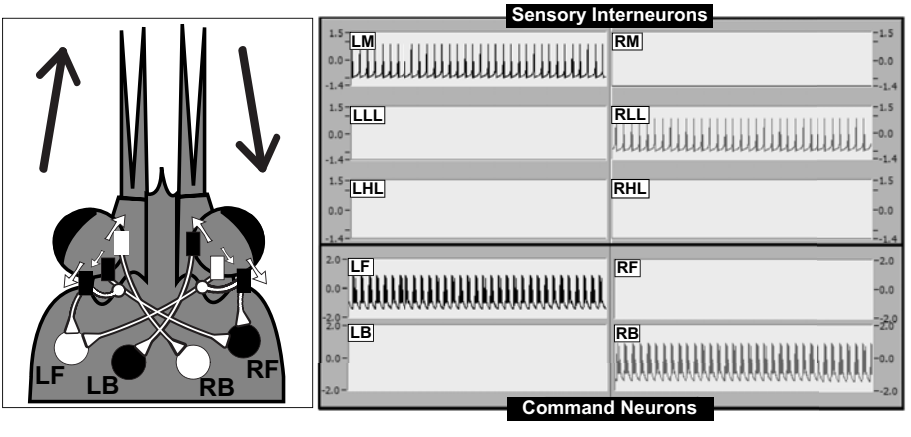


Fig. 3. *Clockwise angular optical flow initiates rotation in place.* **Left panel.** Representation of optical flow neural network. White rectangles and circles represent active sensory interneurons and walking commands, respectively. Black arrows mark direction of optical flow and white arrows denote directional sensitivity of sensory neurons. **Right panel.** Neural activity of sensory interneurons and walking commands. Forward walking on left and backward walking on right elicits clockwise rotation. Neural network and abbreviations are described in Fig. 1.

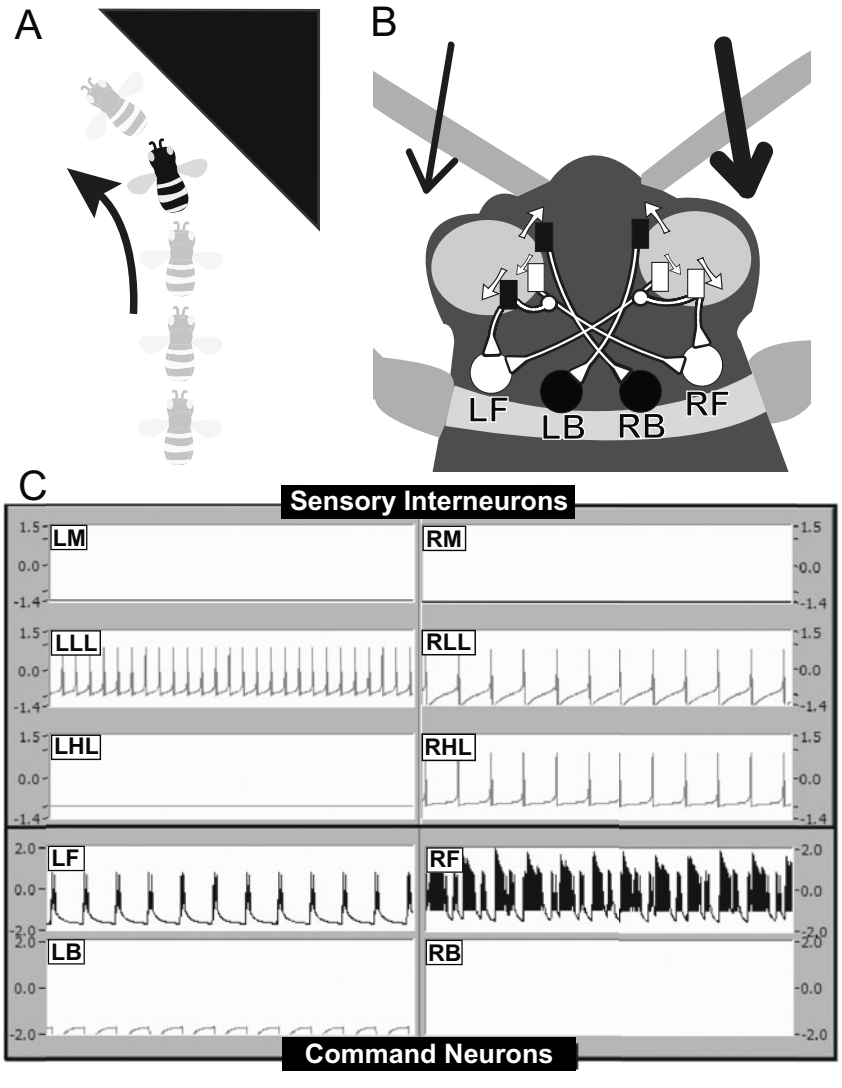


Fig. 4. *Optical flow-mediated obstacle avoidance.* **A.** Flight path of honeybee with a right side obstacle. Opaque bee marks point of activation of high threshold lateral detector and initiation of avoidance turn. **B.** Representation of optical flow neural network at location of opaque bee in **A**. White rectangles and circles represent active sensory interneurons and flight commands, respectively. Black arrows mark direction of optical flow, arrow thickness depicts relative magnitude of flow rate. White arrows denote directional sensitivity of sensory neurons. **C.** Neural activity of sensory interneurons and flight attitude commands represented in **B**. A high rate of optical flow on the right side activates RHL which excites RF and inhibits RLL. The activity of LF drops due to decreased excitatory input from RLL. Since the firing rate of RF becomes greater than that of LF at the location of the opaque bee in **A**, an avoidance turn is initiated. Neural network and abbreviations are described in Fig. 1.

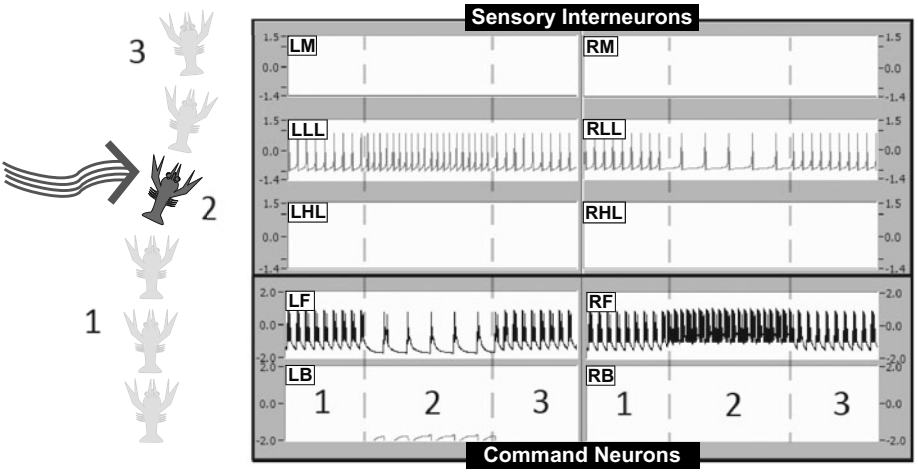


Fig. 5. *Optical flow-mediated yawing.* **Left panel.** Walking path of lobster submitted to strong surge from the left. **Right panel.** Neural activity of sensory interneurons and walking commands. Numbers correspond to different phases of the response. At 1 the lobster is walking forward and there is steady lateral optical flow. The lobster encounters surge at 2 which results in the yaw response. After the lobster's compensatory turn, forward walking resumes at 3. Abbreviations are described in Fig. 1.

intended direction of locomotion. The neural activity of this reflex is outlined in the right panel of Fig. 5.

4 Discussion

We have demonstrated that a simple decussating network between unidirectional optical flow sensory neurons and motor commands can mediate four forms of optomotor reflexes in arthropods: (1) translational responses to pure translational optic flow; (2) rotational responses to pure angular rotation and (3) combinations of translation and rotation that occur during obstacle avoidance and (4) yaw. As the sensory interneurons are operating in their linear range where discharge frequency is proportional to the rate of optical flow, the responses are proportional and symmetric. Assuming that in flying insects the commands would regulate the pitch angle of wing beats and that in walking arthropods they would act on motor commands for walking in the different directions, this is a conserved model.

Of the neural networks we constructed for this simulation, we present only the most stable and robust system that accurately resolved sensory inputs and behavioral outputs. Nevertheless, there are several components of the model that can be enhanced. The resolution between translational and angular optical flow in biology is unclear. Locusts seem to separate the rotatory and translatory components of optical flow but *Drosophila* appear to sum the sensory inputs

[31]. Tests of the network in a robotic platform could help inform this distinction. Additionally, optical flow-mediated odometry could be incorporated into the network as observed in bees [12]. Robotic implementation of the expanded neural networks could help reveal mechanisms for such visually-mediated odometry. It is important to note that there may be different mechanisms present in walking arthropods as crabs have been shown to perform odometry through proprioceptive feedback rather than visual processing [32].

The described optical flow neural network has particular application in the field of robotics. Biomimicry-driven development of robotic control systems, rather than engineering from the ground up, shows particular promise in the creation of adaptively behaving machines. Robots built with neurons will be more like their animal analogues and will present stable and flexible behavioral output as observed in nature. To date biology has inspired robotics tremendously, specifically in optical flow driven control. 'Bee-bot' uses the centering error described by Srinivasan between bilateral low resolution sensors to steer itself between obstacles [33]. The Closed Quarter Aerial Robot employs insect flight principles to maneuver around obstacles and to land safely [34]. The demonstrated success of biologically-inspired robots is promising but can be extended by incorporating modeled neurons and synapses into the control architecture. We hypothesize that by replacing algorithmic processing with neurons and synapses, robots will output more robust and adaptive behavior. The inherent variability of neurons and our capability to model chaotic neural activity will translate to a more varied behavioral output that will create new solutions to unfamiliar environmental challenges.

The next step in the development of biomimetic robots is resolving the question of sensory fusion: how do fused sensory modalities contribute to the basic behavioral commands? There are a variety of sensory inputs that contribute to reflexive behavioral outputs in arthropods. Sherman and Dickinson [20] showed that the visual system of flies was more sensitive to optical flow when rotated at low angular velocities but with increased flow speeds the input from this system decreased and mechanosensory input from the halteres predominated. Mechanosensory wind inputs may serve to sustain forward flight in flies by counteracting the repulsive effects of visual expansion due to self-motion [35]. Integration of multi-modal inputs is not just limited to flying insects though. Statocyst-driven unidirectional rotation sensors have been found in the optic nerve of crayfish that are active without optical input [36]. Further, gravity-mediated righting, chemotaxis, escape reflexes, and antennal rheotaxis all integrate to produce coherent locomotion in walking arthropods.

We can implement these different exteroceptive reflexes and layer the respective neural networks to create a biologically-faithful nervous system for a biomimetic robot. The performance of the robot can inform our biological understanding as well. As an example, rheotaxic antennal reflexes are important for the stabilization and directed locomotion of lobsters in turbulent marine environments. However, the integration of antennal and optic inputs has not been well studied and presents an exciting opportunity for the application of this model.

By comparing the behavior of a robot controlled by our neural network with that of a behaving organism, the contribution of different sensory modalities to the nervous system can be resolved.

References

1. Loeb, J.: Forced Movements, Tropisms, and Animal Conduct. J.B. Lippincott, Philadelphia (1918)
2. Braitenberg, V.: Taxis, Kinesis and Decussation. Prog. Brain Res. 17, 210–222 (1978)
3. Braitenberg, V.: Vehicles: experiments in synthetic psychology. MIT Press, Cambridge (1986)
4. Bethe, A.: A comparative study of the functions of the central nervous system of arthropods. J. Comp. Neurol. 8, 232–238 (1898)
5. Davis, W.J., Ayers, J.A.: Locomotion: Control by positive-feedback optokinetic responses. Science 177, 183–185 (1972)
6. Ayers, J., Rulkov, N.F.: Controlling Biomimetic Underwater Robots with Electronic Nervous Systems. In: Kato, N., Kamimura, S. (eds.) Bio-mechanisms of Animals in Swimming and Flying, pp. 295–306. Springer, Tokyo (2007)
7. Götz, K.: Flight control in *Drosophila* by visual perception of motion. Biol. Cybernetics 4, 199–208 (1968)
8. Blondeau, J., Heisenberg, M.: The 3-dimensional optomotor torque system of *Drosophila melanogaster*. J. Comp. Physiol. 145, 321–329 (1982)
9. Srinivasan, M.W., Zhang, S.W.: Visual motor computations in insects. Annu. Rev. Neurosci. 27, 679–696 (2004)
10. Frye, M., Dickinson, M.: Closing the loop between neurobiology and flight behavior in *Drosophila*. Curr. Opin. Neurobiol. 14, 729–736 (2004)
11. Rowell, C.: Mechanisms of flight steering in locusts. Experientia 44, 389–395 (1988)
12. Srinivasan, M.V., Zhang, S.W., Lehrer, M., Collett, T.S.: Honeybee navigation en route to the goal: visual flight control and odometry. J. Exp. Biol. 199, 237–244 (1996)
13. Dickinson, M.: The initiation and control of rapid flight maneuvers in fruit flies. Integr. Comp. Biol. 45, 274–281 (2005)
14. Rind, F., Bramwell, D.: Neural network based on the input organization of an identified neuron signaling impending collision. J. Neurophysiol. 75, 967–985 (1996)
15. O'Shea, M., Rowell, C., Williams, J.: The anatomy of a locust visual interneurone; the descending contralateral movement detector. J. Exp. Biol. 60, 1–12 (1974)
16. Baader, A., Schäfer, M., Rowell, C.: The perception of the visual flow field by flying locusts: a behavioural and neuronal analysis. J. Exp. Biol. 165, 137–160 (1992)
17. Baird, E., Srinivasan, M., Zhang, S., Cowling, A.: Visual control of flight speed in honeybees. J. Exp. Biol. 208, 3895–3905 (2005)
18. Wendler, G.: The organization of insect locomotion systems and computer-based flight control in the tobacco hawkmoth *Manduca sexta*. In: Ayers, J., Davis, J., Rudolph, A. (eds.) Neurotechnology for Biomimetic Robots, pp. 451–468. MIT Press, Cambridge (2002)
19. Ellington, C.: The novel aerodynamics of insect flight: applications to micro-air vehicles. J. Exp. Biol. 202, 3439–3448 (1999)
20. Sherman, A., Dickinson, M.: A comparison of visual and haltere-mediated equilibrium reflexes in the fruit fly *Drosophila melanogaster*. J. Exp. Biol. 206, 295–302 (2003)

21. Ristroph, L., Bergou, A., Ristroph, G., Coumes, K., Berman, G., Guckenheimer, J., Wang, Z., Cohen, I.: Discovering the flight autostabilizer of fruit flies by inducing aerial stumbles. Proc. Natl. Acad. Sci. USA 107, 4820–4824 (2010)
22. Wiersma, C., Yamaguchi, T.: Integration of visual stimuli by the crayfish central nervous system. J. Exp. Biol. 47, 409–431 (1967)
23. Rind, F.: Identification of directionally selective motion-detecting neurones in the locust lobula and their synaptic connections with an identified descending neurone. J. Exp. Biol. 149, 21–43 (1990)
24. Ibbotson, M.: Wide-field motion-sensitive neurons tuned to horizontal movement in the honeybee, *Apis mellifera*. J. Comp. Physiol. 168, 91–102 (1991)
25. Ibbotson, M., Maddess, T., DuBois, R.: A system of insect neurons sensitive to horizontal and vertical image motion connects the medula and midbrain. J. Comp. Physiol. 169, 355–367 (1991)
26. Kern, R.: Visual position stabilization in the hummingbird hawk moth, *Macroglossum stellatarum L.* II. Electrophysiological analysis of neurons sensitive to wide-field image motion. J. Comp. Physiol. 182, 239–249 (1998)
27. Joesch, M., Plett, J., Borst, A., Reiff, D.F.: Response properties of motion-sensitive visual interneurons in the lobula plate of *Drosophila melanogaster*. Curr. Biol. 18, 368–374 (2008)
28. Bowerman, R., Larimer, J.: Command fibres in the circumesophageal connectives of crayfish, II. Phasic fibres. J. Exp. Biol. 60, 119–134 (1974)
29. Rulkov, N.F.: Modeling of spiking-bursting neural behavior using two-dimensional map. Phys. Rev. E 65, 041922 (2002)
30. Ayers, J., Rulkov, N.F., Knudsen, D., Kim, Y.B., Volkovskii, A., Selverston, A.: Controlling underwater robots with electronic nervous systems. Appl. Bionics Biomech. 7, 57–67 (2010)
31. Preiss, R., Spork, P.: How locusts separate pattern flow into its rotatory and translatory components (*Orthoptera: Acrididae*). J. Insect Behavior 8, 763–779 (1995)
32. Layne, J., Barnes, W.J., Duncan, L.: Mechanisms of homing in the fiddler crab *Uca rapax*: 2. Information sources and frame of reference for a path integration system. J. Exp. Biol. 206, 4425–4442 (2003)
33. Coombs, D., Roberts, K.: "Bee-bot": using peripheral optical flow to avoid obstacles. In: Proc. of the SPIE, vol. 1825, pp. 714–721 (1992)
34. Green, W., Oh, P., Barrows, G.: Flying insect inspired vision for autonomous aerial robot maneuvers in near-Earth environments. In: Proceedings of IEEE International Conference of Robotics and Automation, pp. 2347–2352. IEEE Press, New Orleans (2004)
35. Budick, S., Reiser, M., Dickinson, M.: The role of visual and mechanosensory cues in structuring forward flight in *Drosophila melanogaster*. J. Exp. Biol. 210, 4092–4103 (2007)
36. Wiersma, C., Fiore, L.: Unidirectional rotation neurones in the optomotor system of the crab, *Carcinus*. J. Exp. Biol. 54, 507–513 (1971)

Modifying Directionality through Auditory System Scaling in a Robotic Lizard

Danish Shaikh[1], John Hallam[1], and Jakob Christensen-Dalsgaard[2]

[1] Mærsk Mc-Kinney Møller Institute for Production Technology, University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark
{danish,john}@mmmi.sdu.dk
[2] Institute of Biology, University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark
jcd@biology.sdu.dk

Abstract. The peripheral auditory system of a lizard is strongly directional. This directionality is created by acoustical coupling of the two eardrums and is strongly dependent on characteristics of the middle ear, such as interaural distance, resonance frequency of the middle ear cavity and of the tympanum. Therefore, directionality should be strongly influenced by their scaling. In the present study, we have exploited an FPGA–based mobile robot based on a model of the lizard ear to investigate the influence of scaling on the directional response, in terms of the robot's performance in a phonotaxis task. The results clearly indicate that the model's frequency response scales proportionally with the model parameters.

1 Introduction

The peripheral auditory system of lizards [1,2], such as *Mabuya macularia* and *Gecko gecko*, is amazingly directional. The directionality is generated by acoustical coupling of the two eardrums, created by very efficient transmission of sound through internal pathways in the head. This transmission is strongly influenced by the properties of these internal pathways, as well as by head size and is therefore frequency dependent. Since the properties of the internal pathways scale with size of the lizard it is important to understand how scaling changes the directional characteristics of the system.

The system is nominally symmetrical with respect to the median plane and relatively simple in design. It is smaller in size than the wavelength of the sounds that lizards react to and consequently the incident sound waves diffract around the animal's head and body. Therefore the sound pressure at the two ears is essentially the same. However, the phase difference between the sound waves arriving at either side contains valuable information regarding the direction from which the sound appears to originate relative to the animal, and it is this small phase difference cue that is converted by the auditory system into a relatively larger difference in the perceived amplitude of the sound on either side. As shown

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 82–92, 2010.

in Fig. 1(a), the auditory system is composed of a tympanum (TM) or eardrum on each side of the head. Wide internal tubes called Eustachian tubes (ET) connect the tympani to each other through the central mouth cavity. Thus, for example, when external sound waves impress upon the right ear, they cause vibration of the right tympanum, and this in turn produces an internal sound pressure on inside of the right tympanum. This sound pressure generates internal sound waves which travel via the Eustachian tubes, through the mouth cavity, to the left side and affect the sound pressure on the inside of the left tympanum. Since there are external sound waves impressing upon the left tympanum as well, the motion of the tympanum is the result of the superposition of external and internal pressures. The resulting motion of the left tympanum is due to the difference between instantaneous sound pressures on either side, which depends on the relative phases of the external sound pressure at the two ears. This process occurs simultaneously in the opposite direction as well, i.e. from left to right. This acoustical coupling [3] transforms the lizard's auditory system into a pressure difference receiver with the highest directionality reported for any vertebrate [4].

Pressure difference receiver ears have been quite widely studied both theoretically and experimentally. They occur in lizards [4], frogs [1,3], birds [5] and crickets [6,7]. Zhang et al. have implemented a model of the lizard's peripheral auditory system as a set of coupled digital filters and used it in step-control of a mobile robot in a phonotaxis task [8]. This work has been extended by Shaikh et al. [9,10] to the continuous control of a mobile robot based on a Braitenberg vehicle [11]. In this paper we present a re-implementation of the ear model and investigate its performance in a phonotaxis task, with a Field Programmable Gate Array (FPGA) based mobile robot platform.

Zhang et al.'s model is matched to a nominal ear separation of 13 mm. It can be hypothesised that any scaling of the ear separation and the ear model parameters should result in a corresponding scaling in the frequency response as well. Consistently scaled models may be expected to perform equivalently while inconsistent scaling of ear separation and model parameters may result in loss of directional acuity. We test this hypothesis by scaling the ear model parameters to match three sample peak frequencies (i.e. frequencies for which the model has peak response) taken from the low, mid and high frequency ranges with respect to the ear model, and then observing the performance of the robot in the phonotaxis task for various degrees of ear separation.

The remainder of this paper is organized as follows. In Sect. 2 we present a simple theoretical model [12] of the lizard auditory system and discuss the hypothesis of auditory system scaling versus directionality in more detail. The robotic implementation and the experimental setup are described in Sect. 3. In Sect. 4 we describe the experiments performed with the robot, present the performance results obtained and subsequently discuss and justify the same. Finally, we conclude this paper in Sect. 5.

2 Theoretical Background

2.1 Theoretical Model of the Lizard Peripheral Auditory System

Figure 1(b) shows the equivalent electrical circuit model [13] of the pressure difference receiver ear model. Voltages V_1 and V_2 model sound pressures P_1 and P_2 at the left and right ear respectively. Currents i_1 and i_2 model the left and right tympanic motion in response to the sound pressures acting upon them. Impedance Z_r models the total effect of tympanal mass and stiffness and the Eustachian tubes connecting the tympani to the central cavity, while Z_v models the central cavity itself. Z_r is the same for both left and right sides since, for the sake of simplicity, the ear structure is assumed to have left–right symmetry. Voltage V_3 represents the sound pressure generated in the central cavity due to the interaction of the sound pressures experienced from the left and right side. This causes current i_3 to flow through the impedance Z_v, modelling the movement of sound waves inside the central cavity as the pressure inside it varies. All impedances are complex numbers with frequency-dependent values.

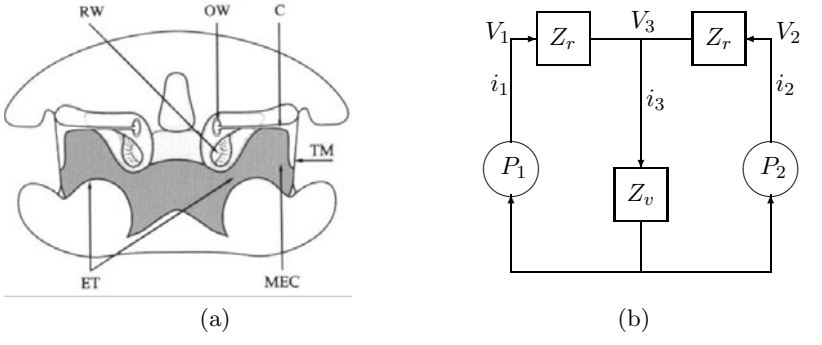


Fig. 1. Peripheral auditory system of a lizard. (a) Lizard ear structure [8], showing the tympanal membrane TM, the Eustachian tubes ET, the middle ear cavity MEC, the cochlea C, the round window RW and the oval window OW. (b) Lumped-parameter model of lizard ears. Voltages V_1, V_2 and V_3 model sound pressures at the left, right and central cavity respectively, while currents i_1, i_2 and i_3 model the tympanic motion in response to the sound pressures.

The physical amplitude of the external sound waves is the same at both the tympani due to diffraction effects mentioned in Sect. 1, and thus V_1 and V_2 differ only in terms of their phase. On the other hand, i_1 and i_2 differ in both phase and amplitude, because of the interaction between the left and right internal sound pressures and this difference represents the proximity of either of the ears to the sound source. We can represent this difference mathematically [8] as

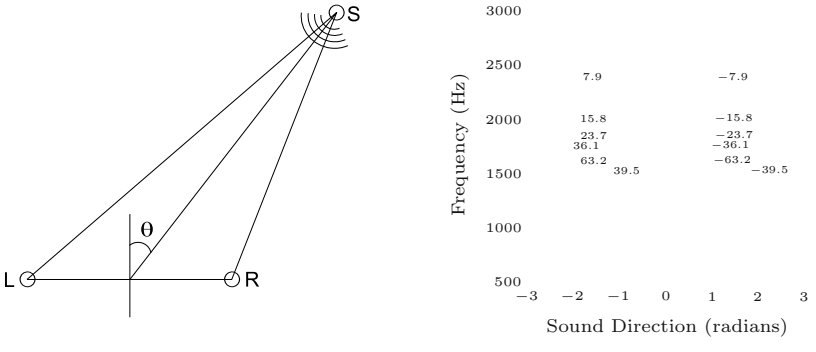
$$\left|\frac{i_1}{i_2}\right| = \left|\frac{G_\mathrm{I} \cdot V_1 + G_\mathrm{C} \cdot V_2}{G_\mathrm{C} \cdot V_1 + G_\mathrm{I} \cdot V_2}\right| = \left|\frac{G_\mathrm{I} + G_\mathrm{C} \cdot \frac{V_2}{V_1}}{G_\mathrm{C} + G_\mathrm{I} \cdot \frac{V_2}{V_1}}\right| , \tag{1}$$

$$\text{where} \quad G_{\mathrm{I}} = \frac{Z_{\mathrm{r}} + Z_{\mathrm{v}}}{Z_{\mathrm{r}}(Z_{\mathrm{r}} + 2Z_{\mathrm{v}})} \quad \text{and} \quad G_{\mathrm{C}} = -\frac{Z_{\mathrm{v}}}{Z_{\mathrm{r}}(Z_{\mathrm{r}} + 2Z_{\mathrm{v}})} \ .$$

Frequency-dependent gains G_{I} and G_{C} model the effect of sound pressure on the motion of the ipsilateral and contralateral tympani respectively. In digital signal processing terminology these gains are analogue filters and their coefficients have been experimentally determined, by taking measurements of the tympanic membrane vibrations via laser vibrometry, by Christensen-Dalsgaard and Manley [4]. Expressing i_1 and i_2 in dB, (1) can be rewritten as

$$i_{\mathrm{ratio}} = 20\,(\log|i_1| - \log|i_2|)\ \ \mathrm{dB}\ . \tag{2}$$

A plot of the current ratio given by (2) is shown in Fig. 2(b) for different frequencies and radial positions θ of the sound source S with respect to the left (L) and right (R) ears (refer to Fig. 2(a)). The model responds well to a wide range of frequencies (about 1000 Hz–2200 Hz), with a peak response around 1500 Hz.



(a) (b)

Fig. 2. (a) Sound source placement with respect to ears, taken and redrawn from [8]. (b) Current ratio i_{ratio} plot from (2) in dB. Positive values indicate $|i_1| > |i_2|$ and negative values indicate vice versa. The model shows strong directionality over a wide range of frequencies.

2.2 Auditory System Scaling and Frequency Response

There have been extensive studies of hearing mechanisms and sound localization abilities of a number of species of animals [14]. Early results however, have been obtained via behavioural experiments and therefore show behavioural response of the animal, produced by neural processes, to auditory stimuli. Later, techniques such as laser doppler vibrometry have been employed with animals to study the underlying biophysical mechanisms, and have allowed researchers to build accurate artificial models of their auditory systems. The focus there is on

what frequencies can animals *detect* and *localize* rather than their behavioural *response* to them. Traditional explanations rely on physical factors such as head size and the acoustical physics involved. The two physical cues available for sound localization are the interaural time difference (ITD) and interaural level or intensity difference (ILD or IID). Both of these cues are higher in magnitude for animals with larger heads (and consequently greater separation between the ears), because a) sound requires more time to travel between the ears, thereby generating a larger ITD cue, and b) the larger the head, the lower the frequencies that it can block, thereby creating a sound intensity difference between the ears. This would suggest that having smaller heads implies the ability to hear higher frequencies as compared with larger heads. This has been confirmed for mammalian species [15].

In the case of lizards, the only cue available is the ITD for the reasons outlined in Sect. 1. The ear model parameters are also matched to a nominal ear separation of approximately 13 mm, resulting in strong directionality over 1000 Hz–2200 Hz with a peak at about 1500 Hz. This leads to a two-fold hypothesis. First, for a given ear model, matched to a given ear separation, if we physically vary the ear separation, the directionality should a) improve if the ear separation is greater than the matched one, as long as separation is less than half the wavelength and b) degrade if the ear separation is less than the matched one. This is because greater ear separation implies greater ITD cues while smaller ear separation implies smaller ITD cues, relative to the ITD cues available at the matched ear separation. Second, scaling the ear model by a given factor should result in a proportional scaling of the frequency response of the model as well. This is because, by scaling the ear model, we are matching it to a proportionally different ear separation, and thereby to a proportionally different magnitude of the binaural ITD cues.

3 Robotic Implementation

The physical robot (refer to Fig. 3(a)) consists of an Xilinx XtremeDSP Starter Platform FPGA board with the lizard ear model implemented on it as digital filters using Verilog Hardware Description Language (VHDL), mounted on a mobile platform assembled with electro-mechanical parts from the Robotics Starter Kit from Digilent Inc., USA. The sole processing unit on the board is the Xilinx Spartan®-3A DSP 1800A, an FPGA chip optimized for DSP applications. The main reasons for choosing this particular platform were its generous logic gate count (1.8 million gates) and flexibility. Two omnidirectional microphones (model FG-23329-P07 from Knowles Electronics, USA) are mounted at the front of the robot on worm gears, allowing the separation to be changed with millimeter precision between 3 mm to 120 mm. The voltage signals from these microphones are pre-amplified, digitised and fed into the FPGA, where they are processed by the lizard ear model and left and right output power is computed. These power values are fed into the decision model. The decision model is basically a set of *if–then* control rules, namely, 1) *if* sound is coming from left *then*

(a) The robotic platform. (b) The experimental setup in the arena.

Fig. 3. Robotic implementation. The full arena is 3.17 m×2.34 m in size.

turn left on-the-spot at full speed, 2) *if* sound is coming from right *then* turn right on-the-spot at full speed, and 3) *if* sound is coming from front *then* go forward with a fixed speed. The speed values of the motors varied from 0-255, which determine the power applied to the motor (0 means 0 % power, motor is not running) and 255 means 100 % power, motor is running at full speed).

The experimental setup (refer to Fig. 3(b)) is similar to the one used in [8]. Two standard audio loudspeakers placed 2 m apart served as continuous single tone sound sources. The robot started from a fixed starting point 2 m behind the mid-point between the loudspeakers, facing straight forward, and was allowed to move autonomously within the test arena boundaries until it a) hit the loudspeaker, b) moved behind the loudspeaker or c) moved outside the arena boundaries. During the movement of the robot, its position in terms of (x,y) coordinates was recorded via an overhead infrared camera system, which tracked an infrared LED (Light Emitting Diode) on top and in the center of the robot's body.

4 Experiments and Results

We are interested in exploring our hypothesis, presented in Sect. 2.2, that the frequency response of the ear model should scale in proportion to the ear separation and parameters of the ear model. The following experiments were performed to test this hypothesis.

First, we set the ear model parameters (i.e. the filter coefficients) to their default values (for the normal 13 mm ear separation), so that the ear model was matched to the default peak frequency of 1500 Hz (i.e. the peak response was at 1500 Hz). Then we varied the separation between the microphones (henceforth referred to as the ear separation) using 3 mm, 6.23 mm, 13 mm, 36 mm and 100 mm, labelled as *min*, *small*, *normal*, *large* and *max* respectively. These values represent a uniform change on a logarithmic scale in the ear separation with respect to the normal 13 mm ear separation. For each of the 5 ear separation

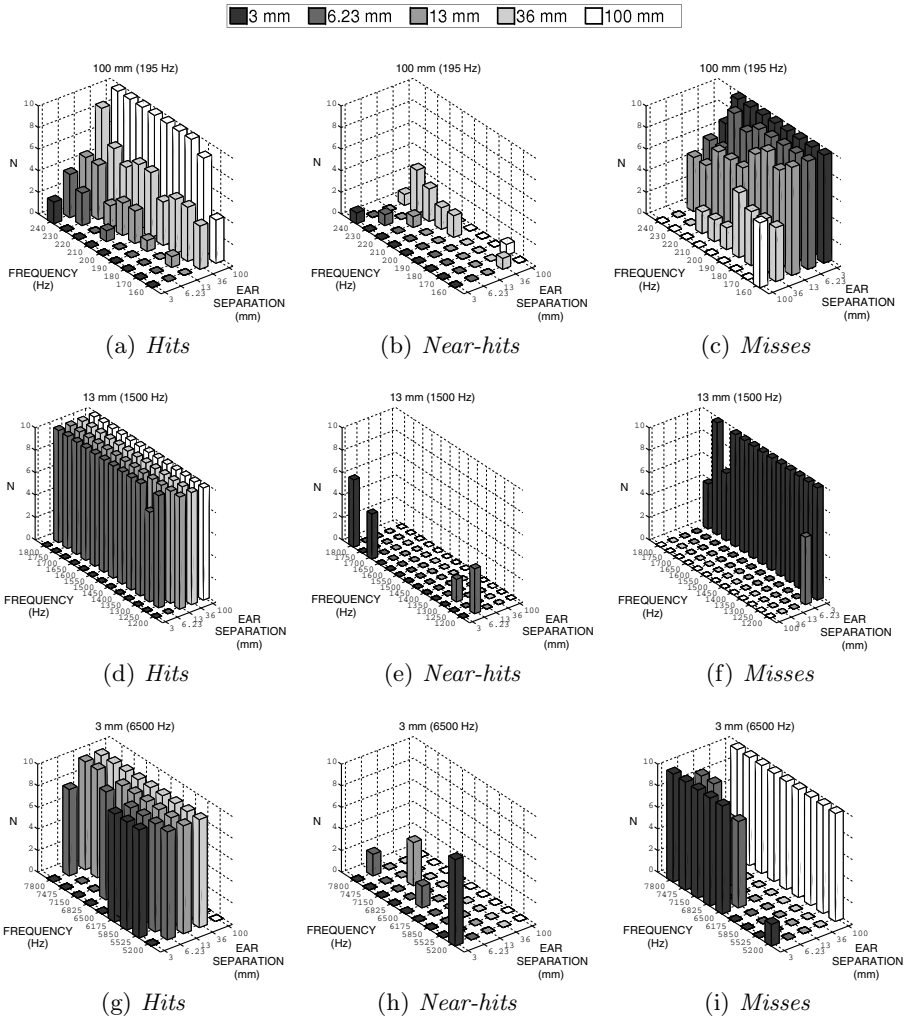
values, 13 sets of experiments were performed in the frequency range of 1200 Hz to 1800 Hz in steps of 50 Hz, within the 1000 Hz–2200 Hz range of the ear model. In each set, one of the loudspeakers continuously emitted a tone of the selected frequency, and the path of the robot was tracked from the starting point until the one of the three finishing conditions was met. This was done 10 times in total per set, 5 trials with the left loudspeaker activated and 5 trials with the right loudspeaker activated. In order to minimise any "memory effects", we randomised the experiments by a) alternating between the left and right sides, b) randomising the order of selection of the frequencies and c) randomising the order of selection of the ear separations. Second, we scaled the ear model parameters to match it to the *max* ear separation of 100 mm, so that the peak response of the model was at $1500\,\text{Hz} \times \left(\frac{13\,\text{mm}}{100\,\text{mm}}\right) = 195\,\text{Hz}$ (rounded up to 200 Hz for purely aesthetics reasons). For each of the 5 ear separations, we repeated the experiments described above for the frequency range of 160 Hz–240 Hz in steps of 10 Hz, resulting in 7 sets of experiments per ear separation. Finally, we scaled the ear model parameters to match it to the *min* ear separation of 3 mm, so that the peak response of the model was at $1500\,\text{Hz} \times \left(\frac{13\,\text{mm}}{3\,\text{mm}}\right) = 6500\,\text{Hz}$. For each of the 5 ear separations, we repeated the experiments described above for the frequency range of 5200 Hz–7800 Hz in steps of 325 Hz, resulting again in 7 sets of experiments per ear separation. The final outcome of each trial was classified as either a *hit* (the robot hits the loudspeaker), a *near hit* (the robot passes within a circle of radius 20 cm around the loudspeaker) or a *miss* (the robot stays outside the circle). In all three cases above, the total number of hits, near hits and misses were individually summed over the 10 trials, for each sound frequency and ear separation combination. Figure 4 illustrates these results.

4.1 Trajectory Directness

In order to determine the performance of the robot in terms of its trajectory, we use a "directness" statistic given by (3) defined in [8]. It measures the average heading of a given robot trajectory vector from the starting point to the loudspeaker. A given trajectory is divided into n vectors ($n = 13$ in our case), each of length l. For each vector, the heading θ relative to the position of the loudspeaker is calculated. Then these are averaged over the total number of vectors, and we get the average heading. This procedure is repeated for all trajectories.

$$\boldsymbol{v}_{\text{avg}} = \frac{1}{\sum_{i=1}^{n} l_i} \left(\sum_{i=1}^{n} l_i \cos\theta_i, \sum_{i=1}^{n} l_i \sin\theta_i \right) . \tag{3}$$

Keeping in mind the large volume of data generated in this manner, a more visually comparative method of presenting the data is needed. This is achieved with Fig. 5, depicting the angular error θ_{err} in degrees between the average headings and the ideal heading of 0°.



(a) *Hits* (b) *Near-hits* (c) *Misses*

(d) *Hits* (e) *Near-hits* (f) *Misses*

(g) *Hits* (h) *Near-hits* (i) *Misses*

Fig. 4. Total number of *hits*, *near-hits* and *misses* by the robot for each combination of sound frequency and ear separation. The ear model parameters are matched to 100 mm in Fig. (a)–(c), to 13 mm in Fig. (d)–(f) and to 3 mm in Fig. (g)–(i). Note that Fig. (c) and Fig. (f) have reversed ear separation axes for best visibility of all data points.

4.2 Discussion

We can see that the robot shows strong directionality for all three ear model scaling factors, consistent with our hypothesis that scaling the ear model parameters results in a directly proportional scaling in the frequency response. In terms of the number of hits and misses, we see in Fig. 4 that in almost all cases the robot performs better when the actual ear separation is greater than the matched ear separation, and worse when the actual ear separation is less than the matched one. In terms of the average angular error in trajectory heading, again we see

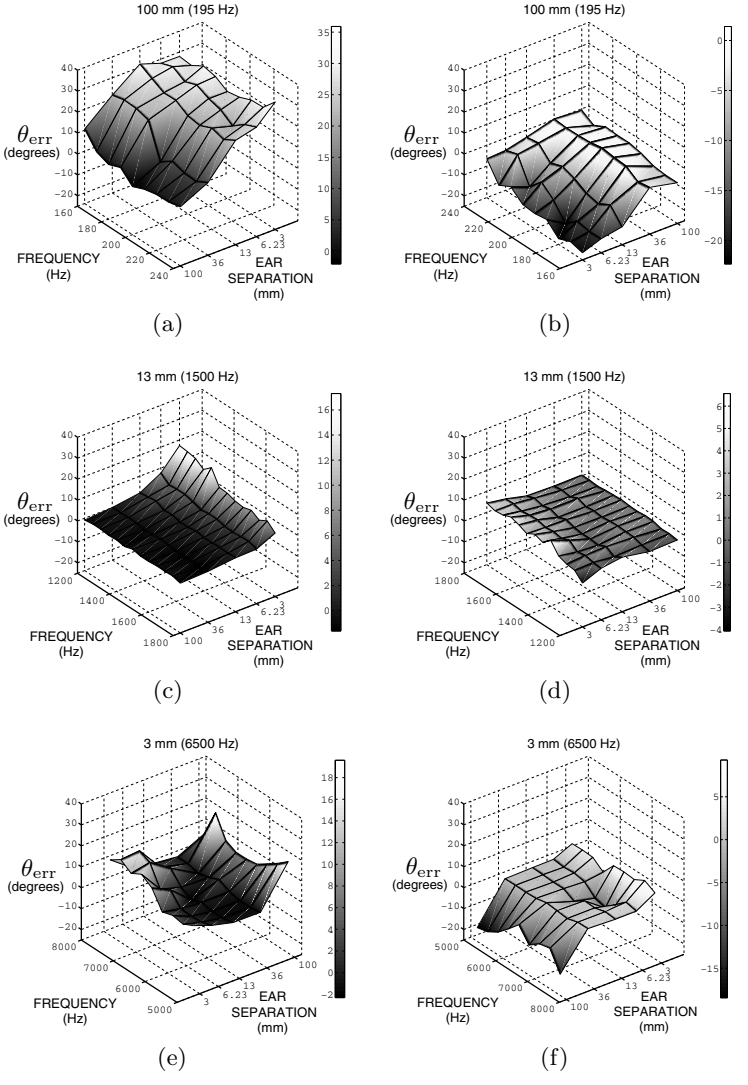


Fig. 5. Average angular error θ_{err} in the trajectory heading with respect to the ideal heading of 0°. Figures (a), (c) and (e) correspond to the tests performed with the left speaker activated and Fig. (b), (d) and (f) correspond to the tests performed with the right speaker activated. Note that the x and y axes have been rotated for best visibility of the entire error surface gradient.

in Fig. 5 that the error decreases when actual ear separation is greater than the matched ear separation, and increases when the actual ear separation is less than the matched one. These results confirm our hypothesis that greater ITD cues improve directionality. But when the ear model is matched to 3 mm ear separation (6500 Hz peak frequency) and the actual separation is 100 mm, there

are no hits (Fig. 4(g), 4(h) and 4(i)), and the angular error is maximum (Fig. 5(e) and 5(f)); the directionality degrades strongly. This is explained by the fact that the wavelength λ of a 6500 Hz sound wave is $\frac{340\,\mathrm{ms}^{-1}}{6500} \approx 52\,\mathrm{mm}$, and the 100 mm ear separation is about twice this wavelength. The phase difference between the sound waves incident from either side, at 90° from the center of the robot, is zero for ear separation of $\frac{k\lambda}{2}$, for all $k \in \mathbb{N}$. Therefore for this 100 mm ear separation, the phase difference between the sound waves at either side varies periodically with angle of incidence and is negligible for sounds incident at 90° from the center of the robot. This results in the apparent sound source direction being correctly related to the direction for relatively few incidence angles. Consequently, the decision model is unable to steer the robot to the source, as confirmed by the error surface plots.

5 Conclusions

We have presented a mobile robot implementation of a lizard ear model, and investigated its behaviour in a phonotaxis task. We experimentally tested the hypothesis that the frequency response scales proportionally with the ear model parameters and the ear separation, by scaling the parameters to match various peak frequencies, for various ear separations and then comparing the performance of the robot.

In this paper, we assumed that the physical presence of the head does not affect the directionality. In the future, mounting a physical 3D model of a real lizard's "head" on the robot, with the microphones located at its ear cavities, and then investigating the phonotaxis performance could prove to be interesting. It might provide some information about whether the directionality of a lizard's peripheral auditory system is influenced by its head, and if so, the manner in which it is influenced.

References

1. Christensen-Dalsgaard, J.: Directional hearing in nonmammalian tetrapods. In: Popper, A.N., Fay, R.R. (eds.) Sound Source Localization. Springer Handbook of Auditory Research, vol. 25, pp. 67–123. Springer, New York (2005)
2. Wever, E.G.: The Reptile Ear: Its Structure and Function. Princeton University Press, Princeton (1978)
3. Christensen-Dalsgaard, J., Manley, G.A.: Acoustical coupling of lizard eardrums. JARO-Journal of the Association for Research in Otolaryngology 9(4), 407–416 (2008)
4. Christensen-Dalsgaard, J., Manley, G.A.: Directionality of the lizard ear. Journal of Experimental Biology 208(6), 1209–1217 (2005)
5. Klump, G.M.: Sound localization in birds. In: Dooling, R.J., Fay, R.R., Popper, A.N. (eds.) Comparative Hearing: Birds and Reptiles. Springer Handbook of Auditory Research, vol. 13, pp. 249–307. Springer, Heidelberg (2000)
6. Michelsen, A., Popov, A., Lewis, B.: Physics of directional hearing in the cricket *Gryllus bimaculatus*. Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology 175(2), 153–164 (1994)

7. Michelsen, A.: Biophysics of sound localization in insects. In: Hoy, R.R., Popper, A.N., Fay, R.R. (eds.) Comparative Hearing: Insects. Springer Handbook of Auditory Research, vol. 10, pp. 18–62. Springer, Heidelberg (1998)
8. Zhang, L., Hallam, J., Christensen-Dalsgaard, J.: Modelling the peripheral auditory system of lizards. In: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J.C.T., Marocco, D., Meyer, J.-A., Miglino, O., Parisi, D. (eds.) SAB 2006. LNCS (LNAI), vol. 4095, pp. 65–76. Springer, Heidelberg (2006)
9. Shaikh, D., Hallam, J., Christensen-Dalsgaard, J., Zhang, L.: A braitenberg lizard: Continuous phonotaxis with a lizard ear model. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009. LNCS, vol. 5602, pp. 439–448. Springer, Heidelberg (2009)
10. Shaikh, D., Hallam, J., Christensen-Dalsgaard, J.: Control of a braitenberg lizard in a phonotaxis task with decision models. In: Kyriacou, T., Nehmzow, U., Melhuish, C., Witkowski, M., (eds.): Technical Report Series: Proceedings of Towards Autonomous Robotic Systems, University of Ulster, Intelligent Systems Research Centre, University of Ulster, pp. 48–54 (2009)
11. Braitenberg, V.: Vehicles: Experiments in Synthetic Psychology. MIT Press, Cambridge (1984)
12. Fletcher, N.H.: Acoustic Systems in Biology. Oxford University Press, USA (1992)
13. Fletcher, N.H., Thwaites, S.: Physical models for the analysis of acoustical systems in biology. Quarterly Reviews of Biophysics 12(1), 25–65 (1979)
14. Heffner, H.E., Heffner, R.S.: Hearing. Comparative Psychology: A Handbook, pp. 290–303 (1998)
15. Heffner, H.E., Heffner, R.S.: High-frequency hearing. The Senses: A Comprehensive Reference. Audition 3, 55–60 (2008)

SCRATCHbot: Active Tactile Sensing in a Whiskered Mobile Robot

Martin J. Pearson[1], Ben Mitchinson[2], Jason Welsby[1], Tony Pipe[1], and Tony J. Prescott[2]

[1] Bristol Robotics Lab, Bristol, UK
martin.pearson@brl.ac.uk
[2] Active Touch Laboratory, University of Sheffield, UK

Abstract. The rodent vibrissal (whisker) system is one of the most widely investigated model sensory systems in neuroscience owing to its discrete organisation from the sensory apparatus (the whisker shaft) all the way to the sensory cortex, its ease of manipulation, and its presence in common laboratory animals. Neurobiology shows us that the brain nuclei and circuits that process vibrissal touch signals, and that control the positioning and movement of the whiskers, form a neural architecture that is a good model of how the mammalian brain, in general, coordinates sensing with action. In this paper we describe SCRATCHbot, a biomimetic robot based on the rat whisker system, and show how this robot is providing insight into the operation of neural systems underlying vibrissal control, and is helping us to understand the active sensing strategies that animals employ in order to boost the quality and quantity of information provided by their sensory organs.

1 Introduction

In order to cope with nocturnal or poorly-lit environments mammals have evolved a range of non-visual sensory capacities many of which have not been successfully replicated in robots. One such capacity is the tactile hair (vibrissal) sensory system [1] [2]. Tactile hairs are found in all mammals, except for man, and are highly developed in many rodent species (such as rats and mice) and in a variety of aquatic mammals such as seal, walruses, and manatees. Research interest has centred on the facial vibrissae, or whiskers, and our own research has focused on the vibrissae of murid rodents such as rats and mice.

Our approach to this system begins with neuroethology, wherein we study vibrissal neural systems holistically, including the observation of natural behaviour as well as comparative and evolutionary data, and leading to computational models. We then expose these models to the complexities of real-world operation, and the demands of functional robotics, revealing shortcomings that do not manifest in simulation. This engineering process feeds back, raising questions that are not raised (or addressed) by current biological data, and guiding us in the design of future biological experiments. Along the way, we hope to show that whiskers can be a useful robotic sensory system.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 93–103, 2010.

Below, we briefly review the ethology of rat whisking behaviour, including results from behavioural experiments conducted in our own laboratory. We then go on to describe our current whiskered robotic platform, SCRATCHbot (the robot name is derived from the acronym Spatial Cognition and Representation through Active TouCH). In this paper we focus on the active sensing and whisker control aspects of the SCRATCHbot platform in comparison to the whisking behaviour of rodents. Some of our recent work on tactile sensing and discrimination using artificial whiskers is described in a companion paper [3], and therefore is not discussed in any detail here. Our work builds on, and was inspired by, a large number of previous research efforts in robotic tactile sensing systems that we have recently reviewed in [1]. SCRATCHbot also replaces and improves on our own earlier whiskered robot, Whiskerbot [6], which was simpler in both mechanical and control terms.

2 Neuroethology of Rat Whisking Behaviour

The whiskers of murid rodents are of two types. The 'macrovibrissae' form two regular grids of longer whiskers (approximately 30 per side) emerging from the 'mystacial pads' on either side of the snout and can be moved back and forth in a behaviour known as 'whisking'. The 'microvibrissae' are shorter, non-actuated whiskers, that are distributed over the front and underside of the snout in a less regular pattern. Rats generally whisk their macrovibrissae when they are exploring an environment or attempting most forms of tactile discrimination. In neurobiological investigations of this system, studies of neural responses to 'passive' whisker deflection are, therefore, beginning to give way to studies of more natural 'active' deflection where moving whiskers encounter objects in the world. These studies show that whisker motion plays a key role in signal formation within the brain, hence one of our main goals in developing SCRATCHbot is to get better insight into the effects of this *active sensing* control on the signals processed by the vibrissal system.

Macrovibrissal movements are driven by a complex musculature. This includes intrinsic muscles within the pad, that allow for some individual control of whisker motion, and extrinsic muscles that move all of the whiskers in the pad together or that alter the relative positions of the whiskers by changing the shape of the pad. The principal, and first-described, component of whisker motion is the anterior-posterior (AP) movement of all macrovibrissae together. A smaller synchronised up-down component to this motion has been identified (that is, a typical whisk is reminiscent of a 'rowing' action), as has a torsional rotation of the shaft during the whisk cycle. Furthermore, the whisker columns move at somewhat different speeds during the AP sweeps with the net effect that the angular separation, or spread, between the whiskers varies significantly within each whisk cycle. Finally, the whiskers do not always move in concert on the two sides of the face, and the mystacial pad moves substantially during whisking. Nonetheless, AP motion of all whiskers together describes a large proportion of overall whisker motion [5].

Whisking motor patterns vary substantially with behavioural circumstance, but discernible 'bouts' of more-or-less periodic whisking at 6-10Hz interspersed by periods of inactivity, are typical. Whisk frequency tends to be relatively constant within a bout but other kinematic parameters can vary substantially. The strongest observed external influence is whisker-environment contact. For instance, a unilateral unexpected whisker-environment contact generally leads to suppression of protraction ipsilaterally (i.e. on the side the contact was made) and to increased protraction amplitude contralaterally [4] [5] (see figure 1 left). We have hypothesised that this is the outcome of a control policy we term 'Minimal Impingement, Maximal Contact' (MIMC), which tends to maximise the count of whisker-environment contacts, whilst keeping the depth of those contacts within a managed range to maintain signal quality. A further observation [5] that spread between whisker columns is reduced during environmental contact is consistent with this policy, with rearward, non-contacting, whiskers brought forward to meet an ipsilateral obstruction. In addition to these asymmetries, a temporary loss of bilateral synchrony in whisker movements is often observed following a unilateral contact [4], whilst repeated contacts with the environment can lead to longer periods of desychronization (unpublished results from our laboratory).

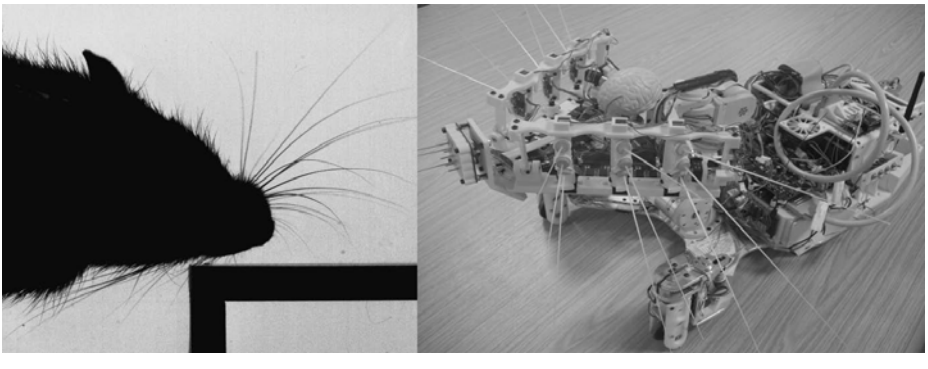


Fig. 1. Natural and artificial vibrissal systems. *Left.* Frame from a high-speed video sequence recorded in our laboratory, showing an exploring rat with whiskers at the maximum protraction phase of the whisk cycle. *Right.* The SCRATCHbot robot platform has two 3x3 arrays of actuated whiskers and a single central array of non-actuated microvibrissae. Control uses biomimetic algorithms based on reverse-engineering of rat neural systems for vibrissal sensory processing and actuation in order to generate life-like active sensing behaviours.

Psychophysical and behavioural experiments (see [8] for review) show that, using only the data gathered by their macrovibrissae, rats can locate objects accurately in space, perform fine textural discriminations, judge gap widths, and that both macro- and micro- vibrissae are required for effective prey capture. However, a reasonable hypothesis is that macrovibrissae are primarily used for

locating objects, and then microvibrissae are brought to bear for close investigation. As a consequence of these findings, and from inspecting many in-house video recordings of rats exploring environments, we consider the 'orient' behaviour, in which a rat positions its head so that the front of its snout is brought to bear on its apparent focus of attention, to be a key component of active sensing. Indeed, orienting should perhaps be considered as the primary active sensing strategy employed by the animal, with repetitive whisker motion (whisking) adding a second component that allows better exploration of space, increased frequency of contact, and more precise control over the nature of those contacts. If we allow that the body must also be moved if the rat is to orient its snout to locations a little distance away, then we could consider that locomotion of a rat in a novel environment may be largely the consequence of a stream of orients to one location after another. That is, the rat shifts its focus of attention and the head, whiskers, and body follow. Thus, we might consider orienting to constitute the foundation of exploratory behaviour in general, and therefore to be a prerequisite for effective active sensing in any whiskered entity, animal or robot.

After orienting, the animal will often keep its snout near to an attended object for a few whisks in order to investigate it more closely using the sensory equipment around the snout. This activity can be complex, and is thus less easy to describe, but we often see an investigative behaviour we refer to as 'dabbing', whereby the microvibrissae are lightly touched or brushed against the object in synchrony with macrovibrissal protractions. The result is that tactile information is obtained at high spatial density towards the centre of the dab, through the microvibrissal array, whilst, within the same narrow time window, surrounding surfaces are sampled in a sparser fashion by the macrovibrissae.

3 Toward a Robot Model of Vibrissal Active Sensing

Based on the experimental data reviewed above, we have identified a number of specific co-ordinated motor actions as pre-requisites for effective active vibrissal sensing. Next we briefly describe the robot hardware and software architecture that we have developed to support this active sensing control.

3.1 Hardware

The SCRATCHbot platform (see figure 1 right) is built from 3 main components: a head, on to which the whisker arrays are mounted; a neck, that allows the head to be moved in 3d and independently from the body; and a body that carries the computing resources, locomotion systems, and power supply.

The head is designed to carry six independent columns of three 'macrovibrissae', with each column driven in a single axis (anterior-posterior) by a small dc motor and gearbox. These columns are arranged into two arrays of three, projecting from opposing sides of the head chassis, and coupled for a second axis of rotation (array tilt). A third, non-actuated, 'microvibrissal' array of nine short whiskers is mounted between the bi-lateral active arrays. The cross-sections of

the macrovibrissae are tapered toward the tip and their lengths (160 - 220mm) are approximately four times larger than the long whiskers of a typical adult rat. To measure deflections of the whisker shaft caused by environmental contact a small magnet is bonded to the base of each whisker and a Hall effect sensor used to sample the displacements of the magnet in two directions. To maintain the pose of each whisker, and to return it to its resting angle after deflection, the whisker base is mounted into a polyurethane rubber plug. The non-actuated whiskers (microvibrissae) have the same transduction technology and polymer return mechanism but are shorter (80mm) and mounted into a single polyurethane casting. Dedicated microcontrollers are used to sample the different whisker arrays and to control the rotation of the whisker columns. The neck component enables the head to be moved with three degrees of freedom: elevation, pitch and yaw, each axis actuated by a brush-less dc motor and harmonic drive gearbox, and controlled by seperate micro-controllers. The robot chassis is a single sheet of aluminium onto which three independently-controlled motor drive units and the neck are mounted. The central computing resources consist of a PC-104+ reconfigurable computing platform, composed of a single board Computer and a closely coupled array of FPGAs for hardware accelerated processing.

3.2 Processing Architecture

The control architecture implemented on the robot takes inspiration from the neural pathways identified in the rat whisker sensory system [1] [2]. Neural structures such as the trigeminal sensory complex, superior colliculus and basal ganglia are modelled and developed in software, at various levels of modelling abstraction, and integrated into a unified system for testing using the BRain And Head Modelling System (BRAHMS) execution framework [7]. To allow independent development of robot hardware and software neural models, a platform simulator has been written which can be inserted into the BRAHMS system in place of the robot interface. Figure 2 is a block diagram of the components that make up the current processing architecture. At the bottom left is the interface to the hardware consisting of the sensors (x, y, θ) and actuators (whiskers, wheels, neck). The remainder of the architecture can be described as an inner loop (blue arrow) mediating whisking pattern modulation, and a middle loop (green arrow) mediating the *orient to point of contact* behaviour described below. Higher loops (orange arrow) model cortical and hippocampal systems, for such competences as object discrimination and spatial mapping, are the subject of current work in our laboratory.

4 Active Sensing Behaviour

In this section we describe some of the active sensing control strategies that we have so far implemented and tested on the robot platform. For details of recent and ongoing work on tactile sensory processing please see [11] [3].

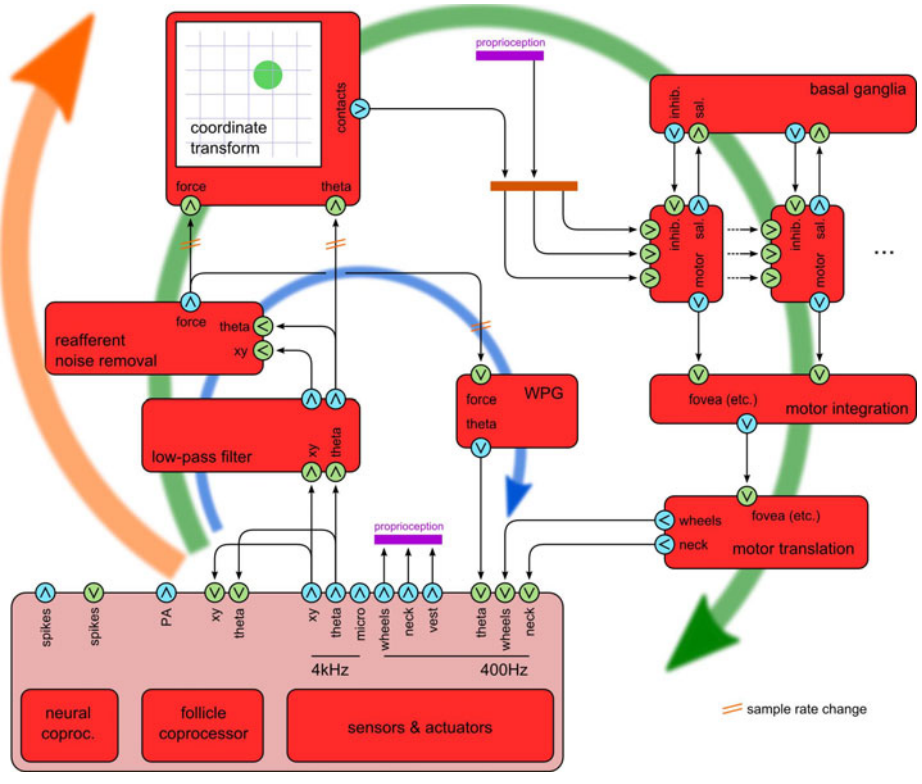


Fig. 2. Overview of control architecture of SCRATCHbot. Arrows indicate control loops within our model whisker sensory system that correspond to current understanding of the real sensory system. See text for details.

4.1 Feedback Modulated Whisking Pattern Generation

The Whisking Pattern Generator (WPG) in SCRATCHbot is a model of the central pattern generator present, though not yet located, in the rat brain and whose activity underlies the rhythmic whisker motions observed in the behaving animal. Typical rat whisking, as described above, can be broadly described as modulated periodic oscillations. The simplest possible model is a single oscillator generating the angular position of each separate whisker through a gain. In such a model, the whiskers would be constrained to move synchronously (all whiskers in phase), symmetrically (whisking on the two sides having the same profile), and periodically (each whisking cycle is identical). However, as previously noted, although bilaterally synchonized and symmetric movements are sometimes observed in the animal, this simple model is inadequate to describe rat whisking generated under natural circumstance of exploration and object contact. Therefore, in our robotic models, we gradually relax these constraints by testing more complex WPG models in order to investigate the impact of different modulation strategies. For instance, in our original whisking robot (Whiskerbot) we

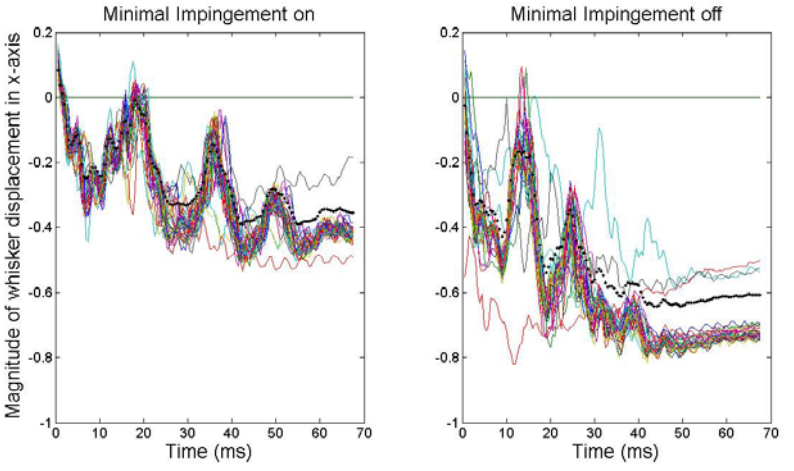


Fig. 3. Comparing the x-component of single whisker deflections sampled in a 70ms time window immediately following contact during bouts of 3Hz active whisking of SCRATCHbot. *Left)* traces from 40 contacts with the Minimal Impingement feedback control strategy active, *Right)* 40 contacts with MI switched off. The black dotted trace in each plot is the mean signal across all samples in each set. Magnitude of deflection has been normalised to the 10-bit ADC sample range of the whisker, i.e., ±512.

used a single WPG to generate a base whisking signal and derived movement patterns for each whisker using this signal. This WPG model was used to test the likely consequences of a Minimal Impingement (MI) control strategy on the whisker deflection signals processed in the brain. As noted previously, our own behavioural observations in animals had indicated that whiskers rapidly cease to protract following contact with an object during exploration. We hypothesized that this result implied a control strategy that sought to minimize the extent to which whiskers were allowed to bend against surfaces. To implement MI in our robot control architecture the total activity across all whisker deflection cells on one side of the face was fed back to suppress activity in the ipsilateral WPG relay. This has the desired effect that protraction ceased rapidly after contact. With MI enabled, the signals were also cleaner and more closely matched those observed in the animal [6]. In previous work [11] it was found that the most useful phase in the whisk cycle for extracting textural information was immediately following contact with the surface. Similarly, it has also been shown that radial distance estimation (contact point along the length of a flexible beam) can be determined by observing the frequencies of oscillation just after impact [13]. Therefore, we focus here on this phase of the whisk cycle to intuit any advantages that the proposed MI approach may afford an artificial system or, by inference, suggest why the rat might adopt such strategy. Figure 3 compares the x-component of deflection during a 70ms time window immediately following contacts sampled from a single whisker during bouts of 3Hz whisking with the MI feedback control switched on and off. The experimental setup is shown in

figure 4, with the data sampled from the middle whisker of the front column with contact incidence determined off-line using a simple thresholding function applied to each sample set. As expected the range of deflection signals with the MI feedback active is significantly less than without, which invites an improvement in sensory resolution by simply scaling the sample range. By comparing the average standard deviation across each data set (normalised to the sample range) it is also clear that with the MI strategy active the variation in response profiles between subsequent contacts is constrained (0.1727 versus 0.2329 without MI active). Such repeatability between contacts will inevitably improve the confidence of classification algorithms applied to extract information such as radial distance to contact or surface texture (an example of which is currently being evaluated [3]).

Predictable variation in whisker spread (the angular separation between the whiskers) was noted previously as a characteristic of animals that are exploring surfaces [5]. To investigate the possible causes of this variability we extended the modulation options of SCRATCHbot's WPG by implementing a separate relay

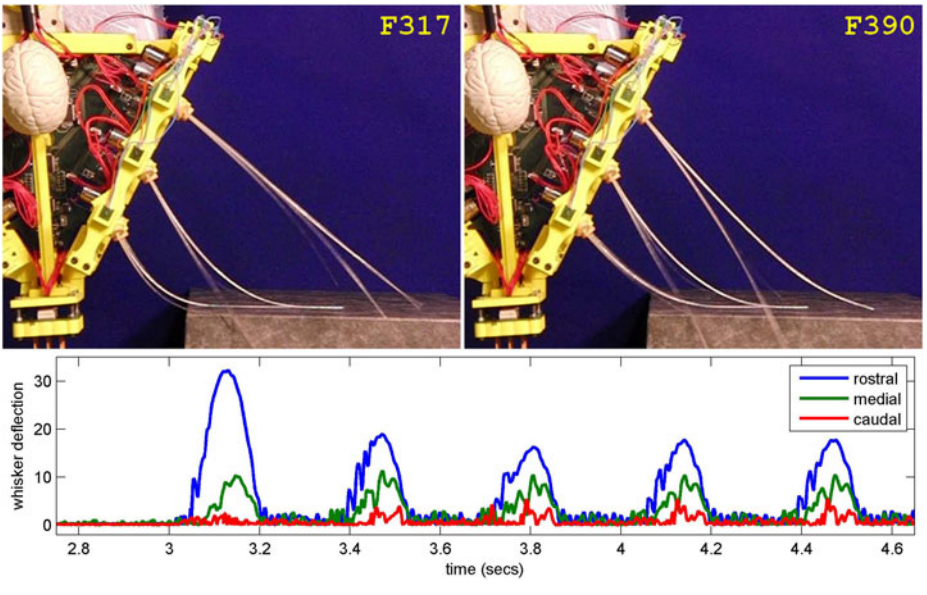


Fig. 4. Investigating the impact of per-column MIMC on whisking patterns. For this experiment we fixed the robot head in a position facing a stationary 'wall'. The first whisk against the wall is unmodulated (left upper panel), and shows that the more rostral column whiskers are heavily deflected, whilst the most caudal column whiskers do not touch the wall at all. On the second whisk, the MIMC modulation has taken effect (right upper panel), the rostral whiskers are less protracted and thus are deflected less heavily, and the caudal whiskers are brought forward (i.e. whisker 'spread' is reduced) and generate contacts with the wall. The effect of this modulation on the contact signals collected can be seen in the lower panel – across the three columns, there is a tendency towards normalisation of contact depth.

for each column (rather than having just one for each side of the head). Whisker-environment contact excites all of these relays, whilst suppressing only those relays driving the whiskers that contacted the environment. The result is that, in addition to the per-side MIMC elicited in Whiskerbot, more rearward whiskers move more rapidly than they would otherwise, and are thus brought forward to meet a contacted obstacle. The net result is a reduction in inter-column spread following contact as seen in the animal. Another way of putting this, is that, by implementing MIMC at the per-column level, 'control' of whisker spread appears as an automatic consequence of this general active sensing strategy - the whiskers are brought forward to meet the environment wherever possible, whilst being restrained from bending too far against it. In experiments with SCRATCHbot platform (see Figure 4) we have been able to demonstrate that per-column MIMC is effective in cleaning up contacts on multiple whiskers and in generating more contacts than would otherwise have occurred.

4.2 Orienting to Point of Contact

The tendency of rats to direct their snout and micro-vibrissal array toward unexpected macrovibrissal contacts was chosen as a second behaviour suitable for investigation by physical modelling. For this purpose our control system implements the hypothesis that a region of the mammalian brain known as the superior colliculus (SC) is used by the rat to control orienting to tactile stimuli [9]. A model SC was designed, implemented in software, and integrated into the BRAHMS processing framework for demonstration on SCRATCHbot.

There is no evidence of proprioception in the whisker musculature, instead, angle cells innervating the follicle are thought to provide the information necessary to transform deflections of moving whiskers into an appropriate head-centred reference frame. Our robotic model therefore integrates whisker deflection information (from the Hall effect sensors) with shaft encoding of whisker column angles in order to map environmental contacts onto a 3-D representation of the space surrounding the robot's head. The most salient contact point then primes a request for an appropriate series of orienting motor commands that move the tip of the snout to that position. The request to perform orienting competes with other salient behaviours for control of the motor plant. This competition is resolved using an action selection mechanism modelled on a group of brain structures known as the basal ganglia [10].

A key task for the motor system is to generate control signals for the wheels and neck that achieve the desired snout movement; this takes place in the Motor Translation Layer of our control architecture. Conventional robotic approaches to this problem (e.g. potential-field or sampling-based) can be expensive to solve, and are not generally bioplausible. We use, instead, an algorithm we call 'Snake', which takes a bio-inspired approach, causing free (uncontrolled) nodes of the mechanics adjacent to a controlled node (the snout, in this case) to follow it. Thus, actuators are 'recruited' to contribute to the movement in a distal-first pattern, as has been seen in the animal during fictive orienting, and more massy central nodes tend to be moved less than lightweight peripheral

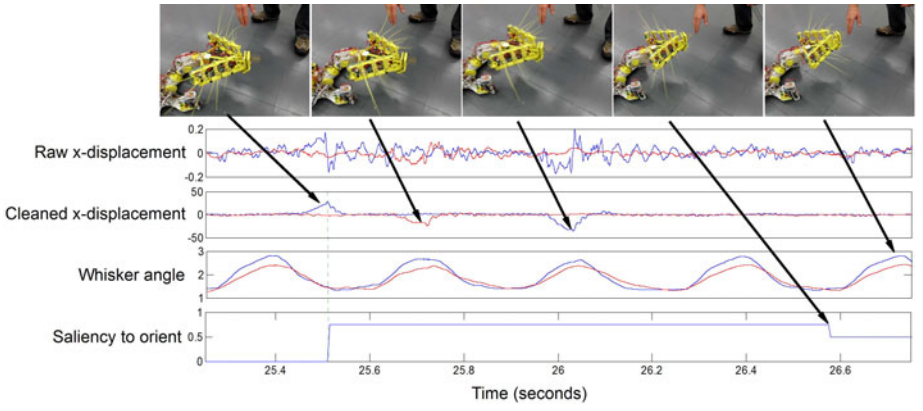


Fig. 5. Frames taken during an orient towards a point of whisker contact, along with plots of data recorded during the run. The upper plot shows the unfiltered x-component output from two whiskers on the left side of the robot (normalised to the maximum calibration range). The second plot has the re-afferent component of these signals removed, greatly improving the signal-to-noise ratio. The third plot displays the current angular position of the rear 2 whisker columns, π radians indicating straight ahead. The lower plot shows the saliency of the *orient* behaviour. Frame 1: contact on whisker 5 (Blue trace) during retraction leads to an increase in saliency of *orient*; this action is selected, and the orient begins. Additional contacts during the orient (frames 2 and 3) are ignored. In frame 4 the snout arrives at the initial point of whisker contact, completing the orient, and saliency is reduced. In frame 5 the micro-vibrissae are used for fine inspection of the contacted feature.

nodes. This algorithm results in motion that appears quite natural to the human observer.

Figure 5 shows video stills from a typical robot experiment demonstrating the orient to contact response. Implementing this task for our whiskered robot provided insight into some additional problems that the rat must also encounter and has overcome through the mechanisms of evolution and neural plasticity. Specifically, it was evident that there is a significant noise component in the whisker deflection signals that is due to self-motion (i.e. caused by the whisking and head movements) and can cause the robot to make orients to *ghost* objects that are not actually present. This motivated us to look for brain structures that might function to remove this noise, a prime candidate being the cerebellum. Interestingly, the cerebellar-inspired algorithms that we have implemented to successfully remove this re-afferent noise [12] essentially learn the dynamics of each whisker as it is moved. Therefore, if a whisker shaft were to be damaged or replaced, the new dynamics would be acquired and integrated into the control system without the need for manual calibration. This tolerance to damage of individual whiskers and the gradual degradation in performance afforded by an array-based system could provide significant advantages to platforms operating over long periods or in remote environments.

The future work plan for the SCRATCHbot platform is to investigate techniques and algorithms applicable to the field of Spatial Localisation And Mapping (SLAM) using active whisker touch, and the enhancement of the superior colliculus and cerebellar models toward predictive prey pursuit behaviour and tactile saliency mapping. Both of these developments move the bio-mimetic sensory technology of active whisker based touch closer toward mobile robotic applications such as search and rescue, remote sensing or underwater exploration.

References

1. Prescott, T.J., Pearson, M.J., Mitchinson, B., Sullivan, J., Pipe, A.G.: Whisking with robots: From rat vibrissae to biomimetic technology for vibrissal active touch. IEEE Robotics and Automation Magazine 16, 42–50 (2009)
2. Kleinfeld, D., Ahissar, E., Diamond, M.E.: Active sensation: insights from the rodent vibrissa sensorimotor system. Current Opinion in Neurobiology 16, 435–444 (2006)
3. Evans, M., Fox, C.W., Pearson, M.J., Prescott, T.J.: Tactile discrimination using template classifiers: Towards a model of feature extraction in mammalian vibrissal systems. Submitted to SAB 2010 (2010)
4. Mitchinson, B., Martin, C.J., Grant, R.A., Prescott, T.J.: Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. Proceedings of the Royal Society, B 274, 1035–1041 (2007)
5. Grant, R.A., Mitchinson, B., Fox, C., Prescott, T.J.: Active touch sensing in the rat: Anticipatory and regulatory control of whisker movements during surface exploration. Journal of Neurophysiology 101, 862–874 (2009)
6. Pearson, M.J., Pipe, A.G., Melhuish, C., Mitchinson, B., Prescott, T.J.: Whiskerbot: A Robotic Active Touch System Modelled on the Rat Whisker Sensory system. Adaptive Behavior 15, 223–340 (2007)
7. Mitchinson, B., Chan, T., Chambers, J., Pearson, M.J., Humphries, M.D., Fox, C.W., Gurney, K., Prescott, T.J.: BRAHMS: Novel middleware for integrated systems computation. Advanced Engineering Informatics 24, 49–61 (2010)
8. Prescott, T.J., Grant, R.A., Mitchinson, B.: Vibrissal behaviour and function. Scholarpedia (in press)
9. Helmelt, M.E., Keller, A.: Superior sensation: superior colliculus participation in rat vibrissa system. BMC Neuroscience 8, 12 (2007)
10. Prescott, T.J., Gonzalez, F.M., Gurney, K., Humphries, M.D., Redgrave, P.: The robot basal ganglia: Behavior and intrinsic processing. Neural Networks 19, 31–61 (2007)
11. Fox, C.W., Mitchinson, B., Pearson, M., Pipe, A.G., Prescott, T.J.: Contact type dependency of texture classification in a whiskered mobile robot. Autonomous Robots 26, 223–239 (2009)
12. Anderson, S.R., Pearson, M.J., Pipe, A.G., Prescott, T.J., Dean, P., Porrill, J.: Enhanced detection of robot whisker contacts by adaptive cancellation of self-generated signals. IEEE Transactions on Robotics (in Press)
13. Ueno, N., Svinin, M.M., Kaneko, M.: Dynamic Contact Sensing by Flexible Beam. IEEE Transactions on Mechantronics 3(4), 254–264 (1998)

Toward a Spiking-Neuron Model of the Oculomotor System

Jan Morén[1], Tomohiro Shibata[2], and Kenji Doya[2,3]

[1] Kyoto University, Kyoto, Japan
jan.moren@gmail.com
[2] Nara Institute of Science and Technology, Nara, Japan
[3] Okinawa Institute of Science and Technology, Okinawa, Japan

Abstract. We present a physiologically plausible spiking neuron-level model of the superior colliculus as part of the saccade-generating visual system. Two major features of the area are the bursting behavior of its output neurons that drive eye movements, and the spreading neuron activation in the intermediate layer during a saccade. We show that the bursting activity profile that drives the main sequence behavior of saccadic eye movements can be generated by a combination of NMDA and cholinergic receptors driven by a local circuit. We also show how the long-range spreading activation can occur, and propose that the functional role for this mechanism is to track the general activity level and trigger a system-wide reset at the end of a saccade.

1 Introduction

We are designing a system-scale spiking-neuron model of the visually guided saccadic system. Our aim is to understand the neural basis of a complete perception-action loop, and to explore the functional significance of neurophysiological features in the studied system. We currently focus on modeling the superior colliculus, the point of sensorimotor integration in the early visuo-motor system. We implement bursting neurons with an NMDA (membrane voltage-dependent) synapse-augmented spiking neuron model and a realistic local circuit, and propose a mechanism and a functional role for the spreading activation behavior observed in the intermediate layers of the superior colliculus in vitro [1].

The broad outlines of the early visuomotor pathways and the saccade mechanism are becoming well understood. System-scale modeling has advanced from simple single-step models to fairly detailed models of individual sensorimotor structures that take physiological and mechanical features of the visuomotor system into account [2]. While these models are becoming detailed and multifaceted, they are still at the computational level in nature and do not try to model these areas at a neurological level. A large-scale spiking neuron model allows us to bring together such models with neurophysiological data and answer questions about the neuron-level implementation of the observed functionality.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 104–113, 2010.

1.1 Neurophysiology

The retina performs initial processing of visual inputs and projects the result to the superficial superior colliculus (SC) directly as well as through the lateral geniculate nucleus. Visual input also reaches the area from cortical visual areas. The superficial SC in turn relays the data to the intermediate SC (SGI) directly and through the parabigeminalis. These steps evaluate the low-level short-term saliency of the incoming visual features. Auditory and somatosensory areas produce sensory data for the SGI in a similar manner.

SGI representation of saccade targets is retinotropic, and the bursting activity of its output neurons is relayed to pons and midbrain areas in the brainstem that in turn generate horizontal and vertical saccade eye muscle activity respectively. The Frontal Eye Fields (FEF) in the cortex projects top-down saccade information through the SC and to the brainstem directly; this redundancy allows either FEF or the SC to generate saccades independently of the other. The cerebellum regulates and tunes activity in both brainstem areas and the SC. See Sparks [3] for an overview.

The Superior Colliculus. The superficial SC consists of two interconnected layers and two major cell types – wide-field and narrow-field receptive cells – that project to the intermediate areas. The intermediate and deep SC consists of cells with multiple types of activity, although they are morphologically similar. Regular spiking neurons form burst and buildup cells, while separate populations form inhibitory interneurons within the SGI, as well as in the deep SC [4]. The burst cells form the major output of the SC.

A major regulatory input comes from the Substantia Nigra pars reticulata (SNpr) in the basal ganglia that tonically inhibits the SC [5]. The SGI outputs to the brainstem areas where it drives the saccade-generating circuitry; to the contralateral SC; and projects back to the FEF via the thalamus to form a saccade feedback circuit for cortical visual systems.

One focus of this paper is the burst neurons in the intermediate layer. Burst neurons act as regular spiking neurons when stimulated directly but exhibit bursting behavior when they receive inputs via the superficial layers [6]. They have NMDA as well as cholinergic inputs [7], and it is likely that the membrane potential-dependent characteristics of the NMDA receptor is a factor in this behavior [8].

Burst neuron activity is self-limited through local circuits in the SGI, but they do not exhibit the spike number constancy found in behaving animals [8]. This suggests that the spike number constancy is an effect of an extrinsic feedback loop from the saccade motor-related areas such as the central mesencephalic reticular formation (cMRF) [9], and that the feedback acts to counteract a local inhibitory mechanism in the intermediate and deep SC [10].

We are also interested in the poorly understood phenomena of spreading activation in the SGI. There is considerable evidence for asymmetric large-scale spreading activation occurring in the SGI during saccades in many animals [1]. But while connections in the superficial SC are long-range, both excitatory and

inhibitory intraconnections within the intermediate SC seem to be spatially limited to 500μm or less [11]. The spreading actiation can thus not be a direct effect of long-range activation within the intermediate areas. The asymmetry may arise from the complex-logarithmic mapping from the retina to the SC [12] but that still leaves open the question both of the functional significance of the spreading activation and of the mechanism by which is it achieved.

Collicular Models. Retinotropic SC activation was once thought to indicate saccade amplitude and direction only. The overall activity in the intermediate SC was treated as a form of vector averaging where the average position of activity in the SC corresponded to to a particular saccade, and the activity of the neurons determined the speed but not the magnitude of the saccade or movement profile. The SC itself was considered upstream of any saccade feedback control, though later models have involved the SC directly in the feedback loop. Some models have sought to incorporate the spreading activity in the intermediate SC by treating it as a moving hill of activity that encodes eye position or movement error, but this has since largely been refuted at least in the case of primates [13].

Current models posit that the SC is part of the saccadic feedback loop, and that each burst neuron spike encodes a movement segment with a direction given by the position of the neuron in the collicular map, and the magnitude given by the weight of the connection to the brainstem areas [14]. The activity is thus not encoding a vector, but is driving a movement trajectory directly, and the resulting trajectory is the sum of bursting activity in the SC during the saccade. This movement is inhibited through downstream feedback limiting the saccade magnitude [15].

Experimental data supporting this view shows that a saccade to the same point will always elicit the same total number of spikes from the neurons involved, and an on-line disturbance will be fed back to the intermediate SC and adjust the firing rate to produce a constant number of spikes [16]. A review of models of systems involved in saccade generation is found in Girard and Berthoz [2].

2 The Superior Colliculus Model Design

We consider the known neurophysiological features on one hand, and behavioral and theoretical functional models on the other, to build a set of spiking neuron-level models of the visual areas related to saccade generation. Our overall system organization is summarized in figure 1(a), and we show an overview of the superior colliculus model that is the focus of this paper in figure 1(b).

The superficial SC is topmost in Figure 1(b). For this study we have two neuron types – wide-field, with weak but broad activation, and narrow-field with focused, strong activation – with no interconnections. They project to the quasivisual neurons, and to the buildup neuron layer.

Quasivisual (QV) neurons receive inputs from the superficial SC and from cortical and other sensory areas, and act as one of the two major input layers for the intermediate SC. They project on to the buildup neuron layer, and also to

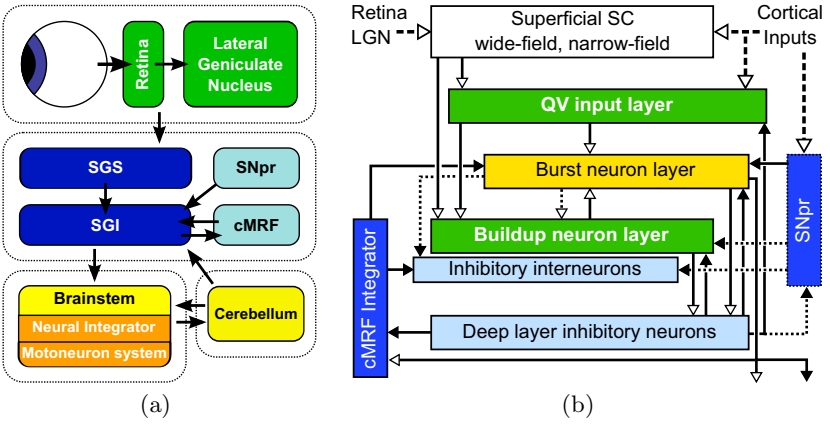


Fig. 1. (a): The principal components in the early saccade vision system. The retina and lateral geniculate nucleus does early vision processing; the SC and related areas integrate sensory data to generate motor commands; the brainstem systems regulate muscle activity; the cerebellum tunes the other systems over time. (b): Block diagram of the superior colliculus model. White triangles are excitatory connections and black triangles are inhibitory. Dashed connections are external inputs. Dotted connections are not used in this paper. QV: Quasivisual; SNpr: Substantia nigra pars reticulata; cMRF: Central mesencephalic reticular formation.

the burst neuron layer through NMDA synapses to give them their characteristic bursting behavior.

Buildup neurons receive inputs from superficial SC and QV neurons, and project on to the burst neurons. They have mutual short-range excitatory connections, but are reciprocally inhibited by interneurons that limit activity spread.

Burst neurons activate strongly only when receiving inputs from both quasivisual and buildup layers. The resulting burst primarily excites ipsilateral eye muscle motoneuron systems, and inhibits the contralateral SC and motoneuron system.

The burst is also projected to an area in the cMRF where burst neuron activity is integrated at the time scale of 100ms. This integrated signal projects back to inhibit the burst neuron layer and the buildup inhibitory interneurons. The net effect is to limit the total number of spikes generated in the burst neuron layer, and - through the differential activation strength of central versus peripheral burst neurons - create a rapid but controlled drop off in the spike rate. This directly generates the saccade main-sequence velocity profile.

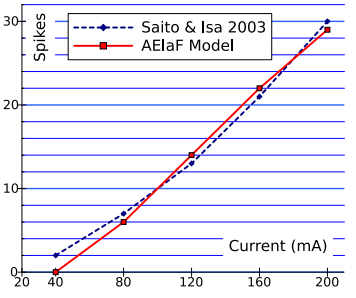
The SNpr (not implemented in this paper) acts as a latch. It tonically inhibits output-related structures in the intermediate SC, but the inhibition is released when inhibited by cortical signals through secondary structures, and triggers the currently prepared saccade. Inhibition is reestablished through the inhibitory burst from deep layer interneurons below.

Deep layer interneurons are locally interconnected and get inputs from burst and buildup neurons. They are near silent until the inputs reach a critical level,

AEIaF Neuron Params

C =	62 pF	V_t =	-47 mV
g_l =	4 nS	V_r =	-65 mV
E_l =	-65 mV	Δ_t =	2.0 mV
Adaptation			
a =	0.2 nS	τ_w =	20 ms
b =	30 pA		
Synapses			
E_e =	0.0 mV	E_i =	-75.0 mV
W_e =	0.72 nS	W_i =	0.04 nS
τ_e =	0.2 ms	τ_i =	3.0 ms
E_n =	0.0 mV	V_{max} =	-43.6 mV
W_n =	1.2 nS	V_{min} =	-60 mV
τ_n =	3.0 ms	g =	3.0

(a) neuron parameters



(b) neuron model and measured data

Fig. 2. (a): Parameters used for the AEIaF neuron for the simulation model. (b): 400 ms spiking response to input currents by the AEIaF neuron with model parameters specified on left; and by SC regular spiking neuron in the rat (from Saito and Isa, [8]).

at which point they fire a burst of inhibitory activity that resets burst layers, buildup layers and associated structures like the cMRF.

Spreading Activation. One focus of this model is the asymmetric spreading activation in the SGI. We assume that the spreading occurs among buildup neurons, while burst neuron activity remains confined to a restricted area around the stimulus origin. Nakahara et. al. [12] posit that the spreading asymmetry is a side effect of the logarithmic mapping from early vision areas. We approximate the asymmetry with rostrally shifted wide-field efferent projections and buildup neuron short-range interconnections. The Inhibitory interneurons restrict buildup-neuron activity, and their inhibition allows buildup neuron activity to spread. But what is the functional significance of this activity?

We propose that the spreading activation loosely keeps track of overall system activity to act as a local shutdown mechanism. Buildup neuron activity increases along with burst neurons, and eventually triggers the deep layer inhibitory neurons that in turn inhibits collicular areas, resets the cMRF integrator and reestablishes inhibitory control from the SN.

2.1 Neuron Model

SGI burst neuron behavior has been shown to be mediated through NMDA receptors in combination with local interconnections. We need a neuron model that is complex enough to reproduce non-linear membrane potential-dependent synapse properties, and that can be fitted roughly to the behavior patterns of SC neurons. On the other hand, a large-scale realistic neuronal simulation is resource intensive, and any unneeded complexity will greatly increase development and simulation time without any concomitant improvement of the results.

We use an implementation of the Adaptive Exponential Integrate and Fire (AEIaF) neuron model of Brette and Gerstner [17] as implemented in the NEST

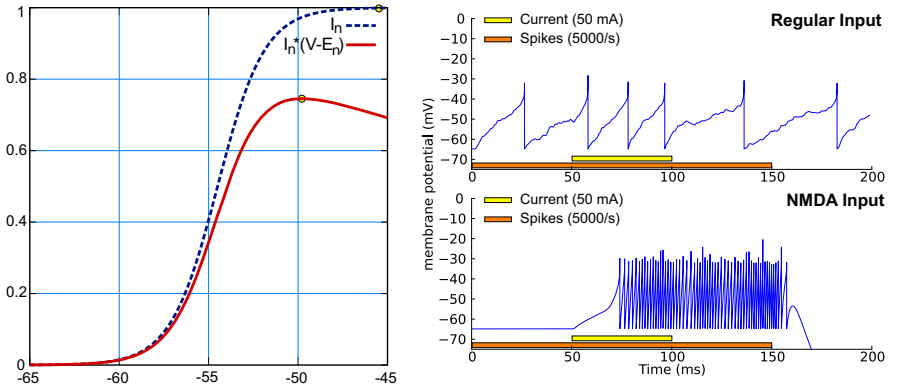


Fig. 3. (a): Activation of the NMDA synapse as a function of membrane potential. Dotted line shows the sigmoid activation profile. Solid line shows the resulting conductance-based synapse activation. (b): model neuron response to a spike train input to a regular synapse (top) and NMDA synapse (bottom). A 50 mA input current is added between 50 ms and 100 ms. Model parameters in figure 2(a).

simulator [18], and extend it with a membrane voltage-dependent NMDA synapse. The AEIaF model is a conductance-based integrate-and-fire model with an exponential soft spiking threshold rather than a hard threshold, and with a second state variable that recreates membrane potential- and spike-adaptation effects. With this model we can deploy model neurons with parameters and behavior similar to corresponding real neurons.

$$C\frac{dV}{dt} = -g_l(V - E_l) + g_l\Delta_t e^{\frac{V-V_t}{\Delta_t}} + I_e(V - E_e) + I_i(V - E_i) + I_n(V - E_n) + I + w, \tag{1}$$

where C is the membrane potential, g_l is the leak conductance, E_l is the resting potential, and the exponential term creates a soft spiking threshold around V_t with softness determined by Δ_t. I_e, I_i and I_n are the excitatory, inhibitory and NMDA synaptic inputs respectively, with synaptic reversal potentials E_*.

The synaptic inputs are shaped by an alpha function with time constant τ:

$$I = \frac{t}{\tau}e^{\frac{1-t}{\tau}} \tag{2}$$

w is an adaptation current with time constant τ_w and sub-threshold adaptation level set by a:

$$\tau_w\frac{dw}{dt} = a(V - E_l) - w \tag{3}$$

A spike event is triggered when the membrane potential diverges due to the exponential term; in practice a spike is triggered when V reaches a sufficiently

large value such as 0 mv. When a spike occurs the membrane potential is reset to V_r and a spike adaptation b is added to the adaptation current w:

$$V > V_{large} : \begin{cases} V = V_r \\ w = w + b \end{cases} \tag{4}$$

NMDA synapses are glutaminergic receptors sensitive to membrane potential. They have a pivotal role in Hebbian associative learning but here we are focused only on their non-linear voltage-dependent response. We model an NMDA synapse as a sigmoid function with gain g where $I_n \to 0$ at voltage $V_{min}, I_n \to W_n$ at voltage V_{max}:

$$I_n = \frac{W_n}{1 + e^{-4g(\frac{V - V_{min}}{V_{max} - V_{min}} - \frac{1}{2})}} \tag{5}$$

The activation function is depicted as a dotted line in figure 3(a). When multiplied by the voltage-dependent conductance $I_n(V - E_n)$, the resulting activation function will asymptotically approach the conductance gradient resulting in a lower, shifted peak activation as shown by the solid curve.

The response to regular and to NMDA input is shown in figure 3(b), top and bottom. The NMDA synapse is unresponsive to input when near the resting potential. When the membrane potential is raised through a direct current injection or cholinergic synapse input the NDMA synapse becomes responsive. The activation is self-sustaining, and needs no further secondary input. This mechanism forms one part of the burst neuron behavior.

2.2 Spatial Model

The current model focuses on the local network within the superior colliculus. It is implemented in NEST [18], a simulator for large-scale spiking neuron networks, using the AEIaF neuron from the previous section. To ease visualization, each neuron type is laid out in its own 3x3mm spatial layer; in a biological system several of these types are physically intermingled. The superficial, QV and buildup layers are 40x40 elements or 1600 neurons each. The burst and deep layer interneurons are 20x20 elements for 400 neurons and a relative density of 25%; it is estimated to be 15-30% in biological systems. The total number of model elements is 8850 neurons, on the order of 1/20 of the primate SC.

The interconnections have a gaussian weight distribution and 1ms delay unless otherwise specified. Input → wide-field, QV → build neurons and the build → build intraconnection have their connection weight center shifted rostrally to approximate the effect of asymmetric interconnections. The parameters are listed in table 1.

The model behavior is illustrated in figure 4. The superficial SC layers (not shown) projects a constant input at $\langle 2.0, 0.0 \rangle$, and the burst neurons are inhibited for the first 50ms. The quasivisual and buildup neurons activate from superficial input. Buildup neuron activity spread is limited by interneuron negative feedback. When burst neuron inhibition is removed, the NMDA input from QV neurons and cholinergic input from buildup neurons trigger a burst.

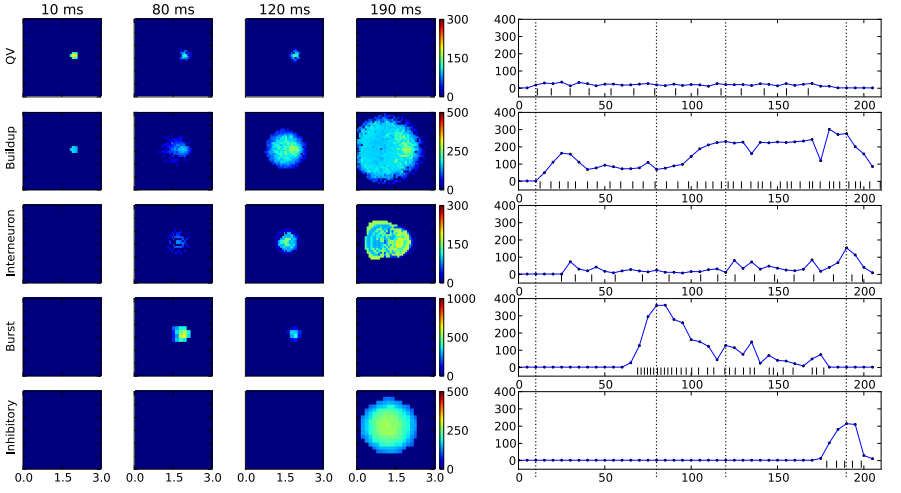


Fig. 4. Intermediate SC model activity in response to a steady input. From top to bottom: Quasivisual neuron layer (QV); buildup neuron layer; inhibitory interneurons; burst neuron layer; and deep layer inhibitory neurons. Superficial SC and synthetic cMRF integrator not shown. Burst neuron layer inhibited for the first 50ms.
On the left, 10ms averaged spatial activity in spikes/s of each area at 10ms (stimulus onset), 80ms (burst neuron peak), 120 ms (buildup neuron spreading activity) and 190 ms (deep layer inhibition). On the right, averaged activity traces of neurons in a 300μm radius around the stimulation center, with the spike train of each center neuron at the bottom. Dotted lines mark the 10, 80, 120 and 190ms time points. Note the buildup neuron spreading activation over time on the left and gamma-shaped burst neuron activity on the right.

Burst neuron activity is integrated in the cMRF (implemented as a set of 50 stochastic spike-summing units). The integrator linearly inhibits the burst neuron layer. The combination of bursting and gradual inhibition gives us the characteristic gamma-shaped burst neuron activity profile seen on the fourth row, right.

Table 1. Model interconnection parameters. r: projection radius; var: gaussian variance; w: weight (in terms of synaptic conductance); k: proportion of pairwise connections; τ_i: excitatory synapse time constant. nr: narrow field neuron; inh: deep layer inhibitory neuron; IntN: inhibitory interneurons; INT: cMRF integrator.

Model parameters				Model parameters			
wide→QV	r : 0.2	v : –	w : $1.2W_e$	Build→IntN	r : 0.6	v : 0.4	w : W_e
wide→build	r : 0.2	v : –	w : $1.2W_e$	IntN→Build	r : 0.6	v : 0.4	w : W_i τ_i : 1.5
nr→QV	r : 0.2	v : –	w : $1.2W_e$	Burst→Inh	r : 0.5	v : 0.1	w : $0.5W_e$ k : 0.1
nr→build	r : 0.2	v : –	w : $1.2W_e$	Inh→Inh	r : 0.5	v : 0.4	w : W_e
QV→Build	r : 0.5	v : 0.2	w : W_e	Inh→Burst	r : 0.5	v : 0.3	w : $0.5W_i$ k : 0.2
QV→Burst	r : 0.5	v : 0.2	w : $2W_n$	Inh→QV	r : 0.5	v : 0.3	w : $2W_i$
Build→Build	r : 0.5	v : 0.6	w : $2W_e$ k : 0.7	Burst→INT			w : 1.0
Build→Burst	r : 0.5	v : 0.6	w : $0.5W_e$	INT→Burst			w : −0.04
Build→Inh	r : 1.5	v : 1.0	w : W_e k : 0.25	INT→IntN			w : −0.01
				INT→Inhib			w : −0.04

The integrator also inhibits the buildup interneurons to allow buildup neuron activity to spread. As the interconnections are asymmetrical the spreading is lateromedial and rostral but only weakly caudal, as seen in the second row, left. This activation spread gradually activates the deep-layer inhibitory neurons and triggers an inhibitory burst that silences intermediate layer activity and resets the integrator.

3 Discussion

The saccadic visual system is the most direct visual perception-action loop in the mammal. The superior colliculus is a major component of this system and has a direct, functional role in directing saccadic eye movements. CS output to the brainstem motoneuron systems comes from a population of bursting cells that directly drive the saccade. It has recently been shown that this burst is not intrinsic but generated by a local circuit mediated by NMDA receptors. The precise circuit and the way in which it interacts with NMDA receptors has not been clearly elucidated. We implemented a straightforward NMDA receptor-like nonlinear input in a spiking neuron model. When interconnected in a spatial model consistent with known neurophysiological data, the burst neuron population exhibits the expected gamma-shaped bursting behavior.

A distinctive and puzzling feature of the intermediate superior colliculus is the presence of spreading activation around the time of a saccade. There have been several suggested reasons for this behavior, none of which seem to fit observed data in the primate. We propose that the behavior acts as a local activity tracker that triggers a subsystem-wide shutdown in order to avoid tonic activation. Interconnections within the intermediate SC may be strictly local, so we show how this spreading activation can occur through local interactions between buildup and inhibitory interneurons, regulated by an inhibitory input from the cMRF.

We are developing the superficial SC in a similar manner. This model is being connected with an eye motoneuron system model also in development, and with a retinal model to form a complete perception-action loop. This integrated system is expected to let us model neural-level interaction based on real-world interactions.

Acknowledgements. The authors gratefully acknowledge the Brain and Neural Systems Team, Computational Science Research Program, RIKEN for the support.

References

1. Choi, W.Y., Guitton, D.: Firing patterns in superior colliculus of Head-Unrestrained monkey during normal and perturbed gaze saccades reveal Short-Latency feedback and a sluggish rostral shift in activity. J. Neurosci. 29(22), 7166–7180 (2009)

2. Girard, B., Berthoz, A.: From brainstem to cortex: Computational models of saccade generation circuitry. Progress in Neurobiology 77(4), 215–251 (2005)
3. Sparks, D.L.: The brainstem control of saccadic eye movements. Nature Reviews Neuroscience 3(12), 952–964 (2002)
4. Zhu, J.J., Lo, F.: Recurrent inhibitory circuitry in the deep layers of the rabbit superior colliculus. The Journal of Physiology 523(3), 731–740 (2000)
5. Hikosaka, O., Takikawa, Y., Kawagoe, R.: Role of the basal ganglia in the control of purposive saccadic eye movements. Physiol. Rev. 80(3), 953–978 (2000)
6. Saito, Y., Isa, T.: Organization of interlaminar interactions in the rat superior colliculus. J. Neurophysiol. 93(5), 2898–2907 (2005)
7. Aizawa, H., Kobayashi, Y., Yamamoto, M., Isa, T.: Injection of nicotine into the superior colliculus facilitates occurrence of express saccades in monkeys. J. Neurophysiol. 82(3), 1642–1646 (1999)
8. Saito, Y., Isa, T.: Local excitatory network and NMDA receptor activation generate a synchronous and bursting command from the superior colliculus. J. Neurosci. 23(13), 5854–5864 (2003)
9. Cromer, J.A., Waitzman, D.M.: Comparison of Saccade-Associated neuronal activity in the primate central mesencephalic and paramedian pontine reticular formations. J. Neurophysiol. 98(2), 835–850 (2007)
10. Zhou, L., Warren, S., May, P.: The feedback circuit connecting the central mesencephalic reticular formation and the superior colliculus in the macaque monkey: tectal connections. Experimental Brain Research 189(4), 485–496 (2008)
11. Isa, T., Hall, W.C.: Exploring the superior colliculus in vitro. J. Neurophysiol. 102(5), 2581–2593 (2009)
12. Nakahara, H., Morita, K., Wurtz, R.H., Optican, L.M.: Saccade-Related spread of activity across superior colliculus arise from asymmetry of internal connections. J. Neurophysiol. 96(2), 765–774 (2006)
13. Soetedjo, R., Kaneko, C.R.S., Fuchs, A.F.: Evidence against a moving hill in the superior colliculus during saccadic eye movements in the monkey. J. Neurophysiol. 87(6), 2778–2789 (2002)
14. van Opstal, A., Goossens, H.: Linear ensemble-coding in midbrain superior colliculus specifies the saccade kinematics. Biological Cybernetics 98(6), 561–577 (2008)
15. Soetedjo, R., Kaneko, C.R.S., Fuchs, A.F.: Evidence that the superior colliculus participates in the feedback control of saccadic eye movements. J. Neurophysiol. 87(2), 679–695 (2002)
16. Goossens, H., Opstal, A.J.V.: Blink-Perturbed saccades in monkey. II. superior colliculus activity. J. Neurophysiol. 83(6), 3430–3452 (2000)
17. Brette, R., Gerstner, W.: Adaptive exponential Integrate-and-Fire model as an effective description of neuronal activity. J. Neurophysiol. 94(5), 3637–3642 (2005)
18. Gewaltig, M.O., Diesmann, M.: Nest (neural simulation tool). Scholarpedia 2(4), 1430 (2007)

An Integrated Neuromimetic Model of the Saccadic Eye Movements for the Psikharpax Robot

Steve N'Guyen[1,2], Patrick Pirim[2], Jean-Arcady Meyer[1], and Benoît Girard[1]

[1] Institut des Systèmes Intelligents et de Robotique
Université Pierre et Marie Curie-Paris 6, CNRS UMR 7222
4, place Jussieu
75252 Paris Cedex 05 - France
[2] Brain Vision Systems, 75013 Paris - France
steve.nguyen@isir.upmc.fr

Abstract. We propose an integrated model of the saccadic circuitry involved in target selection and motor command. It includes the Superior Colliculus and the Basal Ganglia in both cortical and subcortical loops. This model has spatial and feature-based learning capabilities which are demonstrated on various saccade tasks on a robotic platform. Results show that it is possible to learn to select saccades based on spatial information, feature-based information and combinations of both, without the necessity to explicitly pre-define eye-movement strategies.

1 Introduction

For living organisms, the ability to filter out the complex noisy sensory environment in order to focus attention on relevant events only is crucial. As this work contributes to the Psikharpax project [1] – which aims at designing a bio-inspired rat-like robot – we designed a robotic neuromimetic system capable of triggering gaze orientation movement toward salient stimuli. The combination of both Superior Colliculus (SC) and Basal Ganglia (BG) is known to be fundamental for this capability [2]. Dominey & Arbib [3,4,5] designed a now classical system level model of the saccadic circuitry, which provided a global explanation of the role and interactions of the implied brain regions. Nevertheless, some of their design choices are now outdated given the accumulated neurobiological data concerning this circuit. The present work thus proposes a new model of the saccadic circuitry including the integration of more up-to-date SC and BG models. This is the first neuromimetic model of saccadic circuitry with target selection learning capabilities implemented on a robotic platform. This model will be tested on various tasks, demonstrating its capability to behave correctly and to learn to associate spatial or feature-based cues to a reward.

Neurobiology of the circuit. The brain saccadic circuitry involves a number of cortical and subcortical areas, organized in two main pathways (Fig. 1, left). In

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 114–125, 2010.

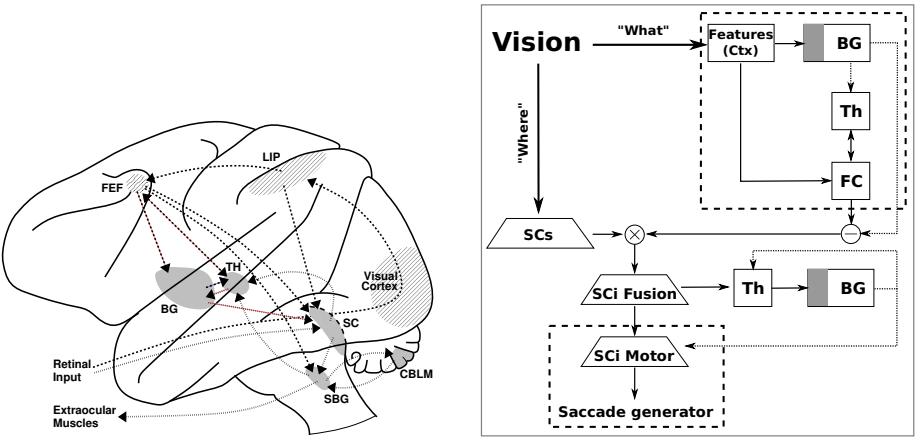


Fig. 1. (Left) Saccadic circuitry in the macaque monkey, subcortical pathway in dotted, cortical pathway in hatched. BG: basal ganglia; CBLM: cerebellum; FEF: frontal eye fields; LIP: lateral intraparietal cortex; SBG: saccade burst generator; SC: superior colliculus; TH: thalamus. (Right) Structure of the model , SCi : intermediate layers of the SC; SCs: superficial layer of the SC; lower dashed box: [6] SC model, upper-right dashed box: [7] CBTC model.

the subcortical one, retinal input projects directly to the superficial layers of the superior colliculus (SC), whose deep layers then projects to the saccade burst generator (SBG), which drives the extraocular muscles. Two derivations are added to this basic circuit: a superior colliculus-thalamus-basal ganglia-superior colliculus (STBS) loop [8], which is probably involved in target selection, and the superior colliculus-cerebellum-saccade burst generator circuit, probably involved in the calibration of the system. The cortical pathway goes from the retina through the cortical visual areas, to the lateral intraparietal cortex (LIP) and the frontal eye fields (FEF); LIP and FEF then project to the subcortical pathway through the SC (minor FEF-SBG projections also exist). Finally, a cortico-baso-thalamo-cortical loop (CBTC) affects selection processes in the cortical pathway.

The visual cortex, the LIP, the FEF and the SC are organized in layers of retinotopic maps representing the visual field. In the rat, the SC maps topology seem to be linear [9], i.e. position of targets are encoded in a retinotopic space with a linear mapping (this mapping is logarithmic in primate). As the SBG is divided in four circuits – respectively responsible for the upward, downward, leftward and rightward movements – the spatial encoding of the maps has to be decomposed in burst signals, the so-called spatio-temporal transformation.

In the present work we propose a model of the saccadic circuitry including the SC and the BG interacting through different loops. We then test this model on a robotic platform in various saccade learning tasks involving spatial and feature-based information. Finally, the proposed model is discussed with regard to previous system level models of the saccadic circuitry.

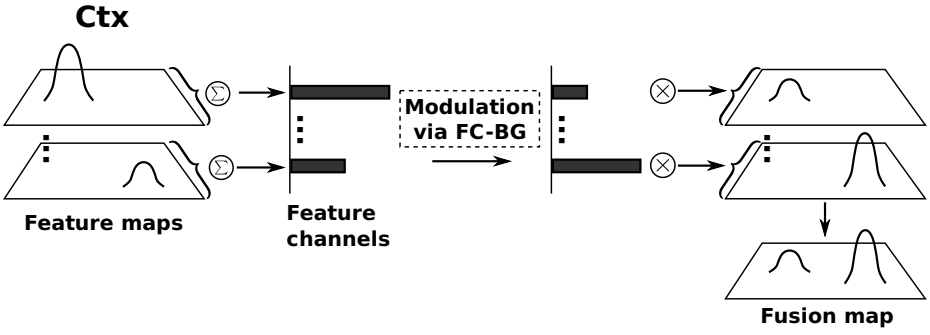


Fig. 2. Feature to location transformation. Assuming that each feature channel (e.g. each color) is modulating the whole activity of a feature map (e.g. color map), each perceived object is represented by a gaussian activity which location is retinotopically preserved. The activity amplitude represents the feature amplitude (e.g. quantity of a given color in the object).

2 Material and Methods

2.1 Model

Our model (Fig. 1, right) selects the target of the upcoming saccade based on its location in the visual field and on its features; it can learn which locations and features to favor, using temporal-difference learning (TD). It is then able to generate the saccadic orders driving the motors.

The previously evoked problem of transformation from the topological encoding of the selected target position in the colliculus output (Motor map on Fig. 1, right) into a two dimensional output command is solved using the superior colliculus model proposed in [6], simplified to use linear maps of the visual field. This model is fed with a map merging both location and feature information (Fusion map) – on which the most salient target is selected using a model of the subcortical basal ganglia circuit – and is based on the basal ganglia model proposed in [7], in which the cortical components have been removed and the position of the thalamus changed. According to neuroanatomy of the STBS loop, the target location information is provided by the visual apparatus of the robot (see Sect. 2.2) and corresponds to the direct projections of the retina onto the SC superficial layers. The visual feature detection is fed into the cortical part of our model, where the selection of the relevant features is operated by a CBTC loop using the [7] model[1].

Concerning visual features, the model's inputs are a family of maps, each of them encodes in retinotopic coordinates the activity of neurons responding to a specific feature only (Fig. 2, left). The selection of the visual features to favor in the CBTC is based on feature channels receiving the sum of the activity of the feature maps (Fig. 2, middle). On the output, the interaction with both FC and BG creates new modulated channels (FC tends to amplify channels and BG

[1] Available in Python on ModelDB http://senselab.med.yale.edu/modeldb/

tends to disinhibit only the maximum channel). The resulting channel amplitude is the FC activity minus GPi activity, a strong channel will be amplified by FC and fully disinhibited by BG, where a weak channel will be slightly amplified by FC but inhibited by BG. Here, the learning capability allows the biasing of BG disinhibition toward the rewarded features. Finally, normalized feature maps are multiplied by these channel values in order to obtain modulated feature maps. The global Fusion map which represents saliency (Fig. 2, bottom right) is obtained by summing all the modulated feature maps and feeds the previously described STBS loop.

The parameters of both BG models were adapted as to operate correctly with the number of channels required by our implementation (720 in STBS and 16 in CBTC, see below).

The strength of the projections of the inputs of the basal ganglia circuits (in gray on Fig. 1, right) is learnt using the basic implementation of the neuromimetic actor-critic algorithm proposed in [10]. This allows the system to learn that a location (resp. a feature) is more rewarded than another one, and thus to bias the selection process when multiple targets are perceived.

In both STBS and CBTC loops, reinforcement learning is computed as follows (cf. Fig. 3 (right)):
We first compute the TD-error δ

$$\delta = R_t + (\gamma \times V_t) - V_{t-1} \quad \text{with} \quad V_t = W_{Critic} \cdot Input_t$$

V_t being the estimated value function at time t, W_{Critic} the Critic input weights, $Input_t$ the input matrix and γ the discount factor, $\gamma = 0.7$ for all. We then update the Critic weights using eligibility traces:

$$W_{Critic} \leftarrow W_{Critic} + \eta \times \delta \times E_{Critic} \quad \text{with} \quad E_{Critic} \leftarrow \alpha \times E_{Critic} + Input_{t-1}$$

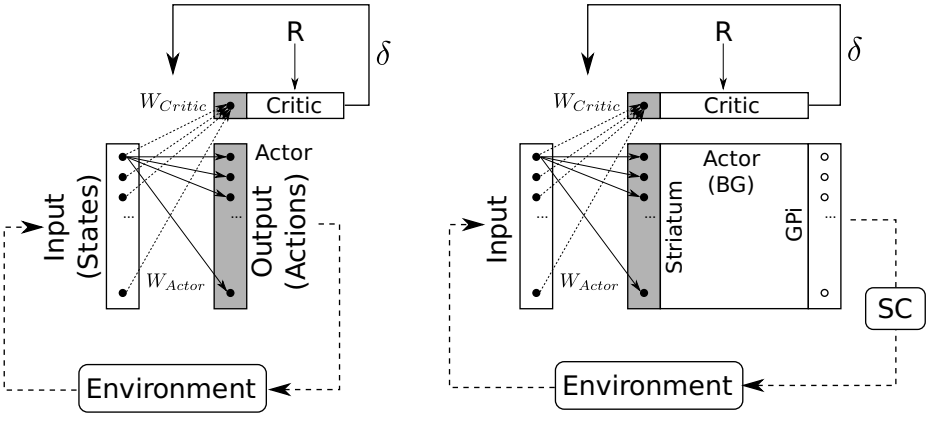


Fig. 3. (Left) Classical Actor-Critic. (Right) Implementation of the Actor-Critic learning in the Basal Ganglia model. Eligibility traces, acting as memory, are not represented

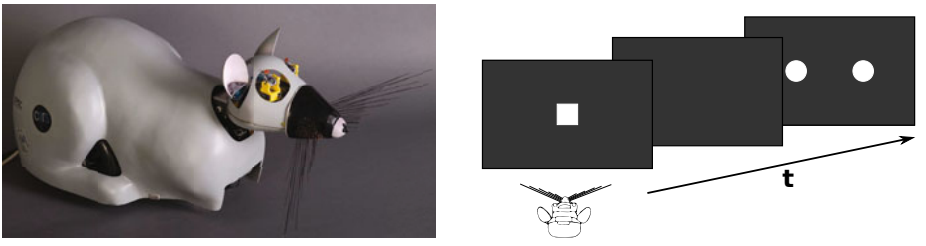


Fig. 4. (Left) Picture of the Psikharpax robotic platform.(Right) Schematic of the experimental protocol. Example for a fixation cue followed by a black screen and then 2 targets.

η is the learning rate and is fixed to 0.0005 and α is the eligibility trace decay factor fixed to 0.9 (these parameter values are identical in both BG circuits). The Critic's weights vector size is N, the size of the $Input$ vector, so connexions are "all-to-one". We then compute the Action vector, which is the weighted input:

$$A_t = W_{Actor} \cdot Input_t$$

Actor's weights are updated following:

$$W_{Actor} \leftarrow W_{Actor} + \eta \times \delta \times E_{Actor} \quad \text{with} \quad E_{Actor} \leftarrow \alpha \times E_{Actor} + Input_{t-1} \otimes A'_{t-1}$$

$$\text{and} \quad A'_{t-1} = 1 - \text{GPi}_{t-1}$$

Actor's weights size is $N \times N$, connexions are "all-to-all". So here, compared to "traditional" reinforcement learning (cf. Fig. 3 (left)), we can see that the "States" are inputs to be selected, "Actions" are weighted inputs according to the reward bias. The Basal Ganglia performs the selection on this weighted input and then the GPi project back to the SC, eventually triggering an action.

2.2 Experimental Settings

Experiments were conducted using our robotic platform Psikharpax (cf. Fig. 4, left). This platform is equipped with 2 miniatures VGA cameras (Field of view of approximately 60 ° horizontally and 45 ° vertically) each mounted on a pan-tilt mechanism driven by servomotors. Visual processing was real time computed using 2 BVS BIPS[2] processors. This system provides objects' position and various features such as movement (speed and direction), color, saturation, luminance and edges (with edges orientations and corners) but for this experiment we only used position and color.

The visual input related to target positions is a retinotopic map composed of 120×23 lPDS neurons [7] with a linear geometry representing a sensory field of 240 ° by 45 ° (notice that the total sensory field is wider than the field of view for future usage). All the SC maps in the model have the same size, geometry

[2] http://www.bvs-tech.com

and neurons model. Each unit of the Striatum part of the BG (in the STBS loop) receives input from 4 SC units. And reciprocally the GPi projects back to 4 SC units, trying to mimic the "funnel" property of SC-BG connectivity [11]. Hence the BG in the STBS loop is composed of 60×12 (720) channels. In the non-spatial loop (CBTC), colors are decomposed in 16 values, thus the corresponding BG entails 16 channels.

The behavior of the system was tested in a number of tasks by putting the robot's head in front of a 100cm diameter wide LCD screen at a distance of 65cm (Fig. 4, right) on which we simply displayed colored fixation and targets cues over a black background according to experimental protocols classically used with animals [12].

3 Results

3.1 Basic Behavior

We first tested the operation of the system by reproducing basic behaviors described in [12], namely the "saccade task", the "fixation task" and the "overlap task". These tasks where learned using a conditioning paradigm according to which the system received a positive reward when it succeeded and a negative

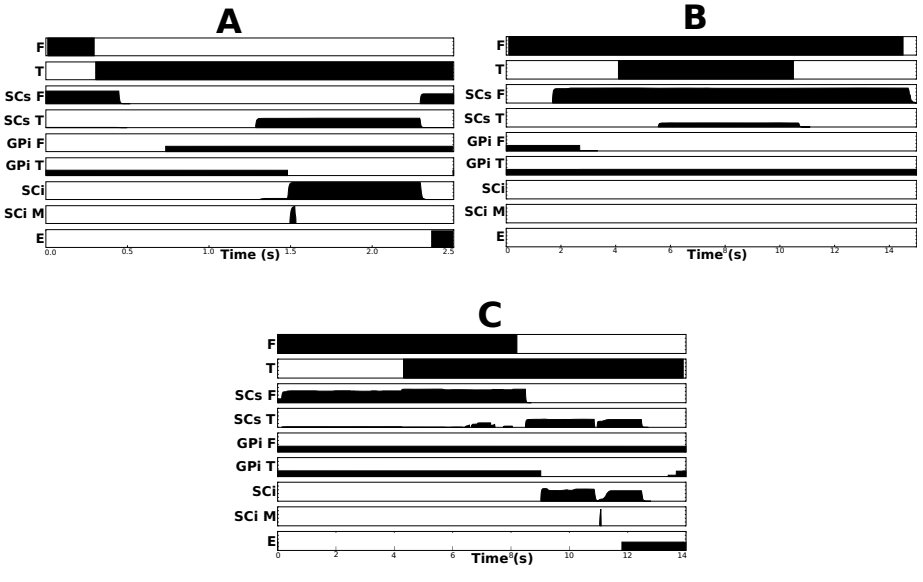


Fig. 5. (A) Saccade task. (B) Fixation task. (C) Overlap task. System activity in F: fixation cue. T: target cue. SCs F: SCs unit at the center of the field of view. SCs T: SCs unit at the target position. GPi F: GPi unit corresponding to SCs F. GPi T: GPi unit corresponding to SCs T. SCi: SCi unit corresponding to SCs T. SCi M: SCi unit on the motor layer corresponding to SCi. E: eye position.

one when it failed. One can notice that, as our model doesn't contain any working memory, we cannot reproduce the "delayed task". Results are summarized in figure 5.

- Saccade task: The basic saccade behavior is well reproduced as, when the target appears, the corresponding GPi unit disinhibits the SCi Motor area and then triggers a saccade.
- Fixation task: The system succeeds in learning to continuously fixate the fixation cue while ignoring the distractor. The activity of the GPi at the location of the distractor shows no disinhibition, preventing any activity on the motor layer of the SCi and thus no saccade.
- Overlap task: Here again the system successfully learned to perform this task. The GPi unit corresponding to the target starts to disinhibit only when the fixation cue disappears. The resulting saccade is not triggered when the target appears (target+fixation) but when the fixation disappears (target only).

3.2 Elaborated Behavior

As the system behaves correctly on simple minimal tasks, we now proceed to some more complex experiments.

We previously described our model as composed of 2 separated loops. The STBS loop dealing with spatial knowledge and the CBTC loop dealing with non-spatial knowledge. As both loops are provided with independent learning capabilities, it should be possible to learn to associate a specific place or a specific color to a reward.

Spatial reward: The corresponding procedure calls upon a black screen step (2.5s) followed by a fixation step (4s) during which a central cue (red square) appears until the robot fixates it. Then the fixation cue disappears and 2 targets appear (two colored disks, one actual target and one distractor) for 6s maximum until a saccade is done. Then the procedure restarts with the black screen and so on. A positive reward is given if a saccade to the desired location is done (always left or always right) and a negative one if a saccade to the wrong location is done. Averaging saccades[3] are not rewarded. Color of targets are randomly alternated (blue or green) between trials. Fig. 6A shows the mean results obtained from 10 runs of 100 trials each. We can see that the learning starts to correctly bias the behavior from around the 50^{th} trial, and stabilizes with a mean performance of 80%. We can also notice that the learned behavior doesn't fully block other behaviors, as a low level of both wrong saccades and averaging saccades still remains which is largely due to perception variability.

[3] Both targets are selected simultaneously and provoke a saccade to the center of gravity. This behavior is observed in animals.

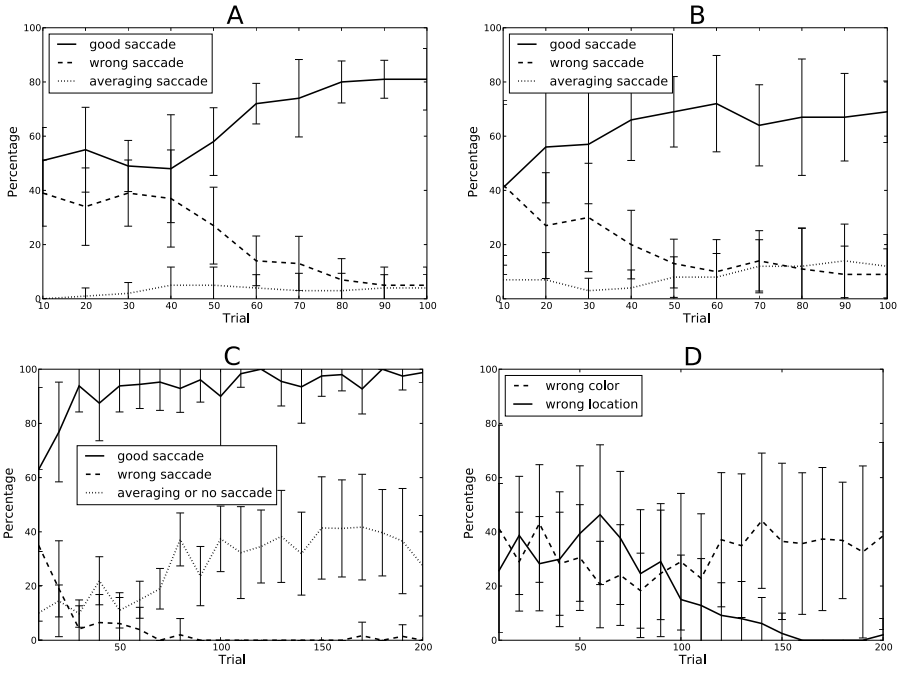


Fig. 6. A: Results of the "spatial reward" task. B: results of the "color reward" task. C: Performance of the "spatial/color" task. D: details of types of errors in task "spatial/color" for negative cases.

Non-spatial reward: The procedure for this task is the same than the preceding one, but we now reward a color based saccade (always green or always blue with randomly alternating position between trials). Here again, the system successfully learned the desired behavior quite similarly to the "spatial reward" task (cf. Fig. 6B). However, the mean performance here is slightly lower and variability higher. This can be explained by the competition between spatial and non-spatial learning. Observed behavior shows that spatial learning seems to have a stronger impact on behavior, so that having the correct colored target consecutively located at the same place will trigger spatial learning which will degrade further performance. Indeed, as the fusion map influenced by the CTBC loop projects to the SBTS loop, the spatial learning has got the "final word" and thus is more directly involved in behavior.

Combination of spatial and non-spatial: Here we combine the 2 preceding tasks by rewarding only a specific color at a specific location. This experiment seems more difficult as, in this case, spatial and non spatial learning are competing, but performance of good saccade in positive case – i.e. when a good saccade is possible – rises quickly to near 95% and, after 100 trials, wrong saccades (saccade to both wrong location and wrong color) almost never appear again (cf. Fig. 6C). So, here, the difficulty is more to learn not to saccade when a negative

case appears – i.e. when no good saccade is possible – as the percentage of correct behavior in this case does not exceed 40%. Fig. 6D shows the details of the type of error occurring in negative cases. After 100 trials, saccades to the wrong location fall to near zero. Thus, at the end of the learning, most of the errors are saccades to the right place but when the wrong color appears. This result is in accordance with the "color task" where we observed a dominance of the spatial loop (SBTS).

4 Discussion

We proposed an integrated model of Superior Colliculus and Basal Ganglia based on recent models [6,7]. To the best of our knowledge, this is the first system level neuromimetic model of the saccadic circuitry implemented in a robot, that includes subcortical selection of spatial information (in a STBS loop) and cortical selection of features (in a CBTC loop) with learning capabilities.

Moreover we demonstrated basic functioning capabilities on classical target selection tasks. Results have shown to be efficient although the proposed system is purely reactive, i.e. doesn't involve any working memory.

We also observed that in our model the spatial learning has a slightly stronger impact on behavior than the non-spatial one, because it is more directly involved in motor commands. Indeed, we can see in Figure 1 (right) that the spatial learning occurs at a lower level, i.e. nearer the output motor map, and thus has the "final word" on selection. Even if this property should be adjusted by appropriate weighting, the question whether animals can learn more easily a spatial reward than a non-spatial one has to be addressed, as this is what the current version of our model predicts.

The basic learning algorithm used here proved to be sufficient when the system had to learn simple sensorimotor associations with easily discriminable stimuli and without long behavioral sequences leading to the reward. Naturally, to deal with more complex situations, the learning algorithm must be improved [13].

Moreover, following computational considerations, we decided to separate the STBS and the CBTC learning capabilities. This solution has proved to be efficient, even if our model clearly lacks a cortical spatial selection loop, involving the FEF, which would allow cognitive control on top of our reactive model and would probably affect the spatial predominance.

Contrary to previous work, our model does not explicitly contain any exploration behavior module. Indeed at the beginning of the learning task no bias already exists between targets and thus a systematic averaging saccade should be done, but inherent noise in "real world" sometime provokes varying delay in target detection or variations in color perception value. Explicit exploration, which can be view as noise in selection is here replaced by implicit exploration, i.e. natural noise in environment. So here, the robotic implementation naturally solves the exploration problem allowing for a more parsimonious model. This solution is not said to be the best one, but has proven to be sufficient for the considered tasks.

4.1 Related Work

The seminal model of Dominey & Arbib [3,4,5] has memory and sequence learning capabilities that we have not replicated yet, and is thus more complete with regards to these points. Nevertheless, it is outdated on at least two aspects. First, while they integrate BG models in cortical loops only, we took into account the now clearly identified STBS loop, which can operate faster than the cortical circuitry. Second, their basal ganglia model is oversimplified: it is based on the direct/indirect interpretation of the BG connectivity [14], from which it keeps the direct pathway only. Consequently, simultaneously presented targets do not interact in the BG circuitry, preventing the resolution of conflicts. Their SC motor layer thus requires an ad hoc winner-takes-all mechanism. Our BG model, using most of the known BG connectivity, solves such conflicts.

More recently, Brown et al. [15] proposed a model including a CBTC loop dedicated to saccade strategy selection, and a STBS loop dedicated to target selection. Here again they include working memory mechanisms that we have not yet included. Their strategies specify whether the saccade will be based on the fixation point, the target position or target features. We showed here that such an explicit strategy selection is not necessary for the basic tasks they tested and that we implemented, our cortical feature selection circuit modulating the subcortical location selection circuit is sufficient. Moreover, their STBS does not stand on its own as it is strictly driven by cortical inputs, while our decorticated system is still able to learn and generate location- or fixation-based saccades. The details of their BG circuitry also suffer from limitations, discussed in details in [16].

Finally, Chambers et al. [11] proposed a model integrating both the subcortical and cortical pathways, where a single BG model dedicated to location-based selection integrates FEF and SC inputs. Using the various positive feedback loops of this circuitry, they show that manipulating the level of dopamine in their BG model generates reaction time and saccade size modifications reminiscent of Parkinson's disease patient behavior. Unfortunately, they rely on a single thalamus module, receiving both SC and FEF inputs, while it is clearly established that CBTC loops run through the ventral anterior and ventral lateral nuclei and the STBS loops through the pulvinar and intralaminar nuclei.

4.2 Future Work

It has been explained that, unlike previous works, the proposed model doesn't contain any working memory and thus cannot reproduce memory related tasks. Working memory related activity has been demonstrated in quasi-visual cells in the FEF, the LIP, the SC and the BG. Many models of these cells have been proposed (e.g. [17,18]) Therefore, we will add such a capability in future work.

Moreover, the SC is known to integrate vision, audition and tactile information [19]. In this work we only used visual information in order to test our model, but as our robotic platform is also equipped with auditory [20] and tactile [21] capabilities, we intend to extend it to these modalities.

Acknowledgements

This work has been funded by the EC Integrated Project ICEA (Integrating Cognition, Emotion and Autonomy), IST-027819-IP.

References

1. Meyer, J.-A., Guillot, A., Girard, B., Khamassi, M., Pirim, P., Berthoz, A.: The Psikharpax project: Towards building an artificial rat. Robotics and Autonomous Systems 50(4), 211–223 (2005)
2. Moschovakis, A., Scudder, C., Highstein, S.: The microscopic anatomy and physiology of the mammalian saccadic system. Prog. Neurobiol. 50, 133–254 (1996)
3. Dominey, P.F., Arbib, M.A.: A cortico-subcortical model for generation of spatially accurate sequential saccades. Cereb. Cortex 2, 153–175 (1992)
4. Dominey, P., Arbib, M., Joseph, J.-P.: A model of corticostriatal plasticity for learning oculomotor associations and sequences. J. Cogn. Neurosci. 7, 311–336 (1995)
5. Dominey, P.F.: Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. Biol. Cybern. 73, 265–274 (1995)
6. Tabareau, N., Bennequin, D., Berthoz, A., Slotine, J.-J., Girard, B.: Geometry of the superior colliculus mapping and efficient oculomotor computation. Biol. Cybern. 97(4), 279–292 (2007)
7. Girard, B., Tabareau, N., Pham, Q., Berthoz, A., Slotine, J.-J.: Where neuroscience and dynamic system theory meet autonomous robotics: a contracting basal ganglia model for action selection. Neural. Netw. 21(4), 628–641 (2008)
8. McHaffie, J., Stanford, T., Stein, B., Coizet, V., Redgrave, P.: Subcortical loops through the basal ganglia. Trends Neurosci. 28(8), 401–407 (2005)
9. Siminoff, R., Schwassmann, H., Kruger, L.: An electrophysiological study of the visual projection to the superior colliculus of the rat. J. Comp. Neurol. 127, 435–444 (1966)
10. Montague, P.R., Dayan, P., Sejnowski, T.J.: A framework for mesencephalic dopamine systems based on predictive hebbian learning. J. Neurosci. 16(5), 1936–1947 (1996)
11. Chambers, J.M., Gurney, K., Humphries, M., Prescott, A.: Mechanisms of choice in the primate brain: a quick look at positive feedback. In: Bryson, J., Prescott, T., Seth, A. (eds.) Modelling Natural Action Selection: Proceedings of an International Workshop, pp. 45–52. AISB Press, Brighton (2005)
12. Hikosaka, O., Sakamoto, M., Usui, S.: Functional properties of monkey caudate neurons. I. Activities related to saccadic eye movements. J. Neurophysiol. 61(4), 780–798 (1989)
13. Khamassi, M., Lachèze, L., Girard, B., Berthoz, A., Guillot, A.: Actor-critic models of reinforcement learning in the basal ganglia: From natural to artificial rats. Adaptive Behavior 13(2), 131–148 (2005)
14. Albin, R.L., Young, A.B., Penney, J.B.: The functional anatomy of disorders of the basal ganglia. Trends Neurosci. 18(2), 63–64 (1995)
15. Brown, J., Bullock, D., Grossberg, S.: How laminar frontal cortex and basal ganglia circuits interact to control planned and reactive saccades. Neural. Netw. 17(4), 471–510 (2004)

16. Girard, B., Berthoz, A.: From brainstem to cortex: Computational models of saccade generation circuitry. Prog. Neurobiol. 77, 215–251 (2005)
17. Droulez, J., Berthoz, A.: A neural network model of sensoritopic maps with predictive short-term memory properties. Proc. Natl. Acad. Sci. 88, 9653–9657 (1991)
18. Mitchell, J.F., Zipser, D.: Sequential memory-guided saccades and target selection: a neural model of the frontal eye fields. Vision Res. 43, 2669–2695 (2003)
19. Meredith, M.A., Stein, B.E.: Visual, auditory, and somatosensory convergence on cells in superior colliculus results in multisensory integration. J. Neurophysiol. 56(3), 640–662 (1986)
20. Bernard, M., N'Guyen, S., Pirim, P., Gas, B., Meyer, J.-A.: Phonotaxis behavior in the artificial rat psikharpax. In: International Symposium on Robotics and Intelligent Sensors, IRIS 2010, Nagoya, Japan (2010)
21. N'Guyen, S., Pirim, P., Meyer, J.A.: Tactile texture discrimination in the robot-rat psikharpax. In: BIOSIGNALS 2010, Third International Conference on Bio-Inspired Systems and Signal Processing, Valencia, Spain (2010)

Reconstructing the Acoustic Signal of a Sound Source: What Did the Bat Say?

Francesco Guarato[1], John Hallam[1], and Dieter Vanderelst[2]

[1] Mærsk McKinney Møller Institute, University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark
{fgu,john}@mmmi.sdu.dk
[2] Active Perception Lab, University of Antwerp, Koningstraat 8, B-2000 Antwerp, Belgium
Dieter.Vanderelst@ua.ac.be

Abstract. When attempting to model and understand bat biosonar behaviour, it would be very useful to know exactly what calls the bat emits, that is, what it really says, in the course of its exploration of the world. Calls could be recorded by miniature radio microphone, but such systems are complex and not all bats are sufficiently strong to carry one. In this paper we describe a technique for reconstructing the actual emitted signal of a bat using recordings collected by an array of remote microphones. The theory of the technique is described, experimental results with laboratory-recorded data (for which ground truth is available) are presented, and the performance of the method is discussed.

Keywords: signal, reconstruction, bat, behaviour, robotics, sensor.

1 Introduction

Echo-locating bats are among the most sophisticated users of sonar sensing systems, prompting engineers to try to discover how they adapt their emitted calls for the various tasks in which they engage. Unfortunately, it is usually not possible to record the call as actually emitted by the bat. First, the acoustic energy in the call is not emitted uniformly in all directions. The shape of the bat's head interacts with the sound field emitted by the bat's mouth and/or nostrils, and the resulting pattern of interference means that certain frequencies are radiated more strongly in some directions than in others.

If the call can be recorded locally, using a small telemetric microphone for example [6], a reasonable approximation of the emitted call can be obtained — although the microphone cannot be placed directly in front of the bat's mouth without interfering with normal behaviour: it usually sits on top of the head, between the ears. Also, not all bats can carry such a microphone: either the system must be miniaturised, or the bat must be sufficiently large.

On the other hand, bat calls are often recorded using arrays of distant microphones arranged around the bat's workspace, capturing a variety of different recordings of the bat's vocalisations. Unfortunately, propagation of ultrasound

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 126–135, 2010.

through air results in substantial frequency-dependent attenuation of the signal, and such recordings are therefore filtered versions of the emitted call with the filtering depending on the direction from emitter to microphone combined with the effect of the path between the two.

Clearly, in principle, if one knows the directivity of a sound source — that is the relative intensity of sounds emitted in differing directions — and the positions of remote microphones with respect to that source, it should be possible to compute the filtering effects and invert them to recover the signal emitted by the source. This is the key idea of the technique described in this paper.

Bat positions associated with calls can be recovered by processing microphone recordings to determine differences in arrival time of the signals. These differences are sufficient to determine the position of the source, given at least four microphones in general positions. [1] and [2] describe the basic technique and ways to improve it. Bat emission directivity can be computed from knowledge of the actual shape of the bat's head, using acoustic simulation techniques [7], [8]. Microphone positions must also be known, but these are relatively easy to obtain.

Thus, we wish to reconstruct the signal emitted by a broadband directional sound source of known directivity, using recordings from a number of microphones whose positions in a global reference frame are known. We assume that the position of the sound source is known, but its orientation with respect to the global frame is not, and must be estimated from the recorded signals.

Orientation estimation, using the technique outlined below, has already been tested through simulations on PC, [3], and experiments performed in a laboratory environment with a Polaroid transducer as sound source, [4] and [5], using a setup illustrated in Fig. 1. In both cases, orientation estimation was found to be very accurate even in noisy conditions, as the error between estimated and real source orientation was at most than 6° in the worst case. Data from the same experimental setup is used in this paper to assess the performance of the signal reconstruction method.

Fig. 1. Experiment: Polaroid transducer on tip of robot arm in front of 13 microphones (white circles). Axes X, Y, Z define world reference frame.

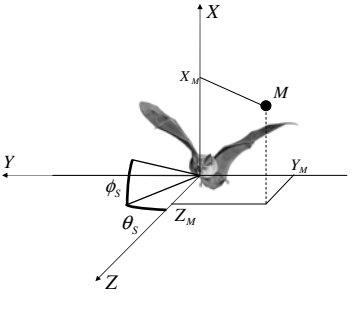


Fig. 2. Bat orientation gived by azimuth and elevation angles (θ_S, ϕ_S); microphone M position with respect to it by (X_M, Y_M, Z_M)

The paper is structured as follows. Section 2 gives a mathematical formulation of the problem and its solution, section 3 contains experiment results and their discussion with particular attention to method limitations, while section 4 summarizes the work described herein and illustrates the future strategies to be considered for improving its performances.

2 Reconstructing Source Signals

2.1 Problem Setting

A sound source with known directionality is placed at a fixed position in three-dimensional space and produces an acoustic signal having definite amplitude at a sufficient number of frequencies. Several microphones record the signal at different locations around the source; source and microphone positions in a world reference frame are known. In such a situation, a *source reference frame* centered at the source and aligned with the source directivity can be defined, and the microphone positions can be referred to it. Then, given the source directivity, the problem is to reconstruct the signal produced by source as it is a few millimeters in front of the source, so that the reconstructed signal is not filtered throughout distance and source directivity.

Fig. 2 illustrates the reference frame in which the position of microphone M is representd by the vector of Cartesian coordinates (X_M, Y_M, Z_M). The true source orientation we look for is given by the pair of azimuth and elevation angles (θ_S, ϕ_S) which relate the source reference frame to the global frame.

2.2 Analytical Solution

Let $D(f, \underline{v})$ be the function describing the source directivity, that is, the relative amplitude (and phase) of an acoustic signal at frequency f measured to position $\underline{v} = (x, y, z)$, the vector describing a microphone position in three-dimensional space with respect to the source reference frame. Let (θ_S, ϕ_S) be the azimuth and elevation angles defining the orientation of the source in three-dimensional space. From source directivity, source orientation and microphone positions with respect to the source, it is in theory possible to calculate the amplitude of the acoustic signal recorded by each microphone at each frequency for any orientation the source can assume. The prediction of the amplitude of the acoustic signal received by microphone m at frequency f is

$$g_{mf} = e_f D(f, R_S(\underline{v}_m)) = e_f D(f, \underline{v}_m^S) \ , \tag{1}$$

where $\underline{v}_m^S$ represents the position of microphone m rotated by R_S, that is by the two angles (θ_S, ϕ_S) in order to transform from the global to the source reference frame. The term e_f is a proportionality factor depending on frequency and represents the unknown spectrum of the call given by unknown intrinsic properties of the source (e.g. a bat) itself. Eq. 1 holds for all M microphones and all F frequencies present in the acoustic signal. R_S, which expresses the

relationship between the source reference frame and the global reference frame, is the rotation corresponding to the source orientation we look for.

The estimate of the true rotation R_S should take into account the difference between the calculated amplitude corresponding to the general rotation R and the true measured amplitude g_{mf} at microphone m

$$g_{mf} - e_f(R)D(f, R(\underline{v}_m)) \ , \tag{2}$$

for all microphones, for all frequencies. An error function $E(e_f, R)$ is built up from (2), see [3]-[5], as function of rotation R the source assumes. Function E is linear in the unknown call spectrum e_f and is minimized for a given rotation when e_f has the following expression:

$$e_f(R) = \frac{\sum_{m=1}^{M} \hat{g}_{mf} D(f, R(\underline{v}_m))}{\sum_{m=1}^{M} \left[D(f, R(\underline{v}_m))\right]^2} \ . \tag{3}$$

Expression (3) for e_f can be substituted into $E(e_f, R)$ so that it depends only on the variable R, see [3]-[5]. The true rotation of the source should correspond to the rotation minimizing $E(R)$, that is,

$$\widehat{R}_S = \arg\min_R E(R) \ , \tag{4}$$

being the estimation of the source orientation in terms of its azimuth and elevation angles $(\hat{\theta}_S, \hat{\phi}_S)$.

The spectrum of the signal produced by the source, e_f, can in theory be reconstructed by substituting the estimated orientation $\widehat{R}_S$ as R in (3) and calculating the amplitude of the signal emitted by source for each frequency of the signal. The time-domain signal is obtained by inverse Fourier transform

$$r_t = F^{-1}\left[e_f\right] \ . \tag{5}$$

Expression (5) can be implemented as a linear combination of filtered microphone-recorded signals, thus. Let

$$\Gamma_{mf} = \frac{D(f, R(\underline{v}_m))}{\sum_m D^2(f, R(\underline{v}_m))} \quad \text{and} \quad \gamma_{mt} = F^{-1}\left[\Gamma_{mf}\right] \ , \ \forall f, \ \forall m \ , \tag{6}$$

in the time domain, representing the estimated inverse filter from microphone m to the composite reconstructed signal. The component of the reconstruction provided by microphone m can be computed convolving recordings g_{mt} and γ_{mt}

$$\rho_{mt} = g_{mt} * \gamma_{mt} = g_{mt} * F^{-1}\left[\frac{D(f, R(\underline{v}_m))}{\sum_m D^2(f, R(\underline{v}_m))}\right] \ , \ \forall\, m = 1, \ldots, M \ . \tag{7}$$

At this point, by superposition (addition) of ρ_{mt} terms (one for each microphone), the final reconstructed signal is

$$r_t = \sum_{m=1}^{M} \rho_{mt} \ . \tag{8}$$

Note that in this computation both the amplitude and the relative phase of each microphone signal is needed.

3 Experiments

The experiments described in this section make use of a Polaroid transducer as sound source whose acoustic signal we aim to reconstruct according to (5)-(8). Data collected in these experiments were previously used for estimating transducer orientations using (2)-(4), see [4]-[5]. In the following, knowledge of transducer orientation is used as input for reconstructing its signal. The following results were obtained by processing a set of 1500 samples, corresponding to the duration of the signal emitted by the transducer, of microphone recordings. The first sample of each portion was determined analytically in the recordings by considering the distance of each microphone from transducer, and so the time delay between emission and reception, or manually by inspection of the recorded signal traces.

The Polaroid transducer sound source used in the experiments has a directivity quite well approximated by a piston in an infinite baffle,

$$D_T(f, \underline{v}) = \frac{1}{d} \cdot 2 \cdot \frac{|J_1(ka \sin \psi)|}{|ka \sin \psi|} \quad , \tag{9}$$

where the source reference frame is chosen aligned with the piston axis, $d = \sqrt{x^2 + y^2 + z^2}$ is the distance of the listener whose position with respect to source is given by $\underline{v} = (x, y, z)$, $k = 2\pi f/c$ with c being the velocity of sound in the air is the wavenumber of the emitted signal, a is the diameter of the transducer (26mm for the Polaroid) and $\psi = \arccos(\cos\phi \cos\theta)$, θ and ϕ being azimuth and elevation angles defining the orientation of vector $\underline{v}$ in three-dimensional space with respect to the transducer axis. The directivity of (9) is further corrected for the frequency-dependent absorbtion of ultrasound by air.

Using the Polaroid transducer in a laboratory setting means that ground truth information about the transducer position, orientation and emitted call spectrum is readily available.

3.1 Experimental Setting

A 13 microphone array was set in a laboratory environment for recording the acoustic signal produced by a Polaroid transducer placed in front of it, see Fig. 1. Microphone and transducer positions are shown in Table 1 with respect to the global reference frame of Fig. 1. The transducer was oriented as desired using a robot arm and produced an acoustic signal that was recorded by microphones.

In the processing of the signals collected in the experiments, temperature value and percentage of humidity were both taken into account, as causes affecting the the amplitude of an acoustic signal spreading out into the environment. Results presented below were obtained using averaged data from 50 calls of the Polaroid transducer collected at each position and orientation.

3.2 Results

Fig. 3 is the original acoustic signal, we aim to reconstruct, at a distance of 4mm from the transducer. It is a down-swept frequency modulated chirp in the

Table 1. Experiments. Positions (cm): x, y and z Cartesian coordinates.

Microphone	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	Transducer
X	76	96	131	131	131	131	95	74	131	131	131	131	131	47
Y	22	22	15	25	45	76	15	15	12	32	52	62	82	42
Z	0	0	6	6	6	6	0	0	26	26	26	26	26	27

range [30kHz, 90kHz], modelled on the typical call of the bat *Myotis daubentonii*. Fig. 4 is the reconstruction associated with $(0°, -20°)$ orientation. The time domain signal presents a small notch around sample 1000: this affects the Fourier transform at low frequencies, as its modulus reveals a smaller amplitude at low frequencies than the one of Fig. 3.

Orientation $(20°, -20°)$ gives the reconstruction of Fig. 5. Several significant notches are present. If the sample at which the recording of each microphone starts is manually chosen by checking the time domain recordings, the reconstruction is that of Fig. 6: notches disappear and modulus of the Fourier transform presents values higher than in the original signal at low frequencies.

Transducer orientation $(20°, 0°)$ gives the reconstruction of Fig. 7: it presents notches, even they are not as significative as the ones in Fig. 5. If starting sample of each recording is manually chosen, reconstruction becomes that of Fig. 8.

Fig. 9 shows the differences between the modulus of the Fourier transform of the reconstructed signal and the modulus of the original one in two cases: orientation $(0°, -20°)$ and orientation $(20°, -20°)$. In the latter, reconstruction with both analytically and manually chosen starting samples of the recorded signals are presented.

3.3 Discussion

Results in the previous section show signal reconstructions whose accuracy depends on transducer orientation and on precision in choosing the first sample of transducer's signal in each microphone recording. Indeed, the sample at which each microphone recording is supposed to start was calculated through the usual analytical formula considering the distance between transducer and each microphone. Reconstruction corresponding to orientation $(0°, -20°)$, Fig. 4, is similar to the true signal and Fig. 9 (A) shows an error between the Fourier transforms whose values are smaller than 0.4 in the low frequency range, otherwise less than 0.1. The biggest value appears in the low frequency range and corresponds to the notch in the time-domain signal of Fig. 4. Fig. 5 depicts a signal very different from the original one, as all notches are visible in Fig. 9 (B). Manually choosing first samples of transducer's signal in microphone recordings gives a reconstruction, Fig. 6, where notches are not present and the difference in Fig. 9 (C) is much reduced. However, manual selection of starting samples when transducer orientation is $(20°, 0°)$, Fig.s 7, 8, does not make difference because the beginning of each recording was hard to determine even manually.

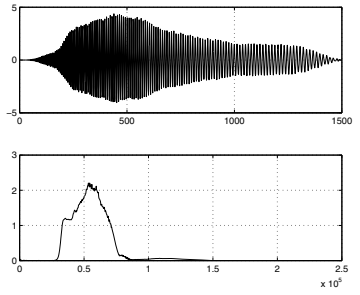


Fig. 3. Original signal and the modulus of its Fourier transform

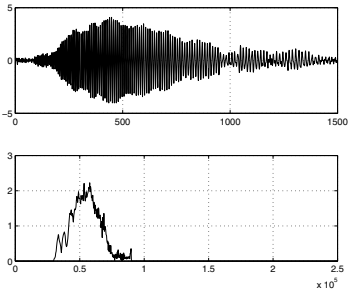


Fig. 4. Reconstruction for transducer orientation $(0°, -20°)$

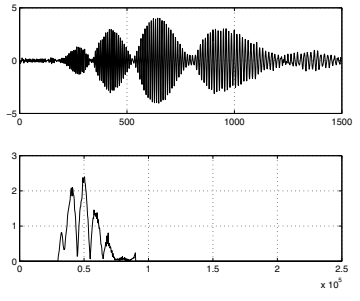


Fig. 5. Reconstruction for transducer orientation $(20°, -20°)$. First samples analytically calculated.

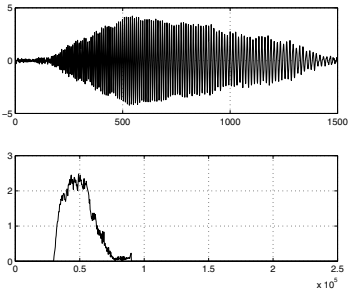


Fig. 6. Reconstruction for transducer orientation $(20°, -20°)$. First samples manually chosen.

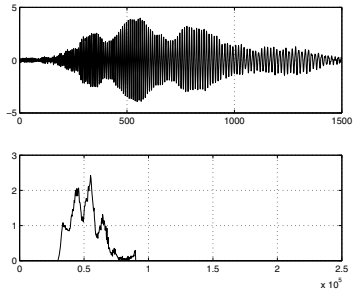


Fig. 7. Reconstruction for transducer orientation $(20°, 0°)$. First samples analytically calculated.

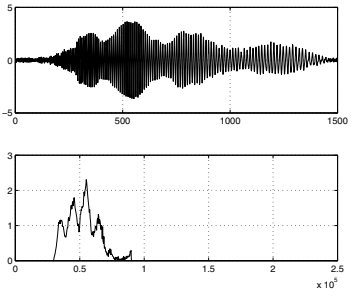


Fig. 8. Reconstruction for transducer orientation $(20°, 0°)$. First samples manually chosen.

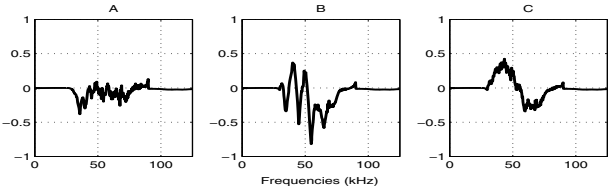


Fig. 9. Difference between modulus of Fourier transforms: (A) orientation $(0°, -20°)$; (B) orientation $(20°, -20°)$; (C) orientation $(20°, -20°)$ samples manually chosen

Manual choice of the sample at which each microphone recording starts guarantees better reconstructions as it is generally more accurate than the calculation of the same samples based on the distance between the microphone and the transducer. To see why this is so, consider that the reconstructed signal is a superposition of signal estimates from the various microphones. Equation (3) can, because the Fourier transform is a linear operation, also be written as

$$r_t = \sum_{m=1}^{M} F^{-1}\left[\frac{g_{mf} D(f, \widehat{R}(\underline{v}_m))}{\sum_{m=1}^{M}\left[D(f, \widehat{R}(\underline{v}_m))\right]^2}\right] , \tag{10}$$

that is, r_t is expressed as superposition of inverse Fourier transforms, one providing the estimated signal contribution for each microphone. An incorrect selection of the sample at which each microphone starts recording the transducer signal is equivalent to applying a delay to the corresponding component; the phase of its Fourier transform is changed and the microphone's contribution in (10) may interfere destructively with others resulting in notches in the final reconstructed signal, like the case of Fig. 5. The separation of the notches in the frequency domain is related to the relative delays of the signals being combined. The same argument has been proposed to account for how bats may distinguish closely spaced reflections from an object [10,11].

Nevertheless, manually choosing the starting samples is not enough for our purposes, see Figs. 7-8 where manual selection of samples does not guarantee a good reconstruction. This may be due to the combination of microphone positions and transducer orientation, such that at high frequencies (which occur at the start of the chirp) some microphones receive a zero- or low-amplitude signal, as their positions correspond to a null in transducer directivity. Hence, manual selection of the signal starting point identifies a sample located late with respect to the true start and the incorrect alignment of the signal provokes a reconstructed signal different from the original one. This problem could be overcome by using a better technique for determining true signal start in the recordings.

Fourier transforms depicted in section 3.2 were all normalized with respect to the highest value of the modulus of the original signal Fourier transform. Differences in Fig. 9 were calculated considering the normalized functions.

4 Conclusions and Future Work

This paper presented a method for reconstructing the acoustic signal produced by a directional sound source provided that its signal is recorded through a set of microphones whose positions with respect to source are known as well as source directivity. The method estimates source orientation relative to the global (microphone) reference frame, which is needed as an intermediate in the signal reconstruction. A mathematical formulation of the method was given and experiments in a laboratory environment for testing it in ordinary conditions were described. Some reconstructions of the original signal produced by a Polaroid transducer were chosen to be shown: the first one resembles the original signal, the second needs manual selection of the sample when transducer's signal starts to be recorded in order to return a better approximation of the original signal whereas the third one does not improve even with manual selection.

Section 3.3 points out how reconstruction strongly depends on the right selection of the samples in the recordings that should correspond to the start of signal reception in each microphone. In particular, the case of Figs. 5-6 clearly shows how a correct selection of these samples leads to a signal much more similar to the original one, though in the case depicted in Figs. 7-8 even a careful manual selection of the samples does not greatly improve the result. Mathematical formulation of the problem is correct, but developing a strategy for correctly selecting the initial samples in microphone recordings is needed.

Results presented in this paper are to be considered as a first application of the method in order to check which problems need to be solved before applying it to investigating the behaviour of real bats. It is important to say that this is an intermediate step: in the paper we focused on the lacks and defects of the method and suggested what we think are the causes, in order to show the next research to perform. As a future subject, a strategy for automatically selecting the starting sample of each recording is needed. In [9] a method based on using the reflected intensity distribution to discriminate time delays between three or more closely spaced object echoes is proposed. Another strategy to try is the spectrogram correlation and transformation model (SCAT) receiver [11] that has been used for modelling the high accuracy of FM bats in discriminating arrival times of different overlapping echoes. Both of these possible solutions take the bat receiver system as an example of accuracy. These methods could be used for estimating the exact samples in microphone recordings where the transducer signal starts by incrementally accumulating microphone contributions into the reconstruction, at each step checking to see whether the signal appears to contain "multiple echoes" (the effect of adding an incorrectly-delayed signal being equivalent to that generated when receiving multiple echoes from close spatial positions).

The ultimate goal for the technique presented in this paper is application to real bats whose calls have been recorded through a microphone array, for extracting two acoustic behavioural features: bat head orientation when emitting a call and the call itself. Knowledge of the emission pattern of the bat is also needed. Reliability of the method when reconstructing bat calls could be verified by comparing the reconstructed signal with one recorded through a telemike-like

telemetry system [6] mounted on the head of the bat. The bat of primary interest for that work, *N. leporinus*, is large enough to carry a telemetry system. *Noctilio* directivity could be recovered through acoustic simulation, see [7], provided that a 3D model of its head is available, though still a first evaluation of the performance of the method might be pursued by using an approximate transducer directivity such as (9).

Aknowledgments

The work described in this paper was funded by the EC via the FP7 project ChiRoPing, IST contract 215370. Facilities were provided by the Universities of Southern Denmark and of Antwerp.

References

1. Spiesberger, J.L.: Probability distributions for locations of calling animals, receivers, sound speeds, winds, and data from travel time differences. J. Acoust. Soc. Am. 118(3), 1790–1800 (2005)
2. Spiesberger, J.L., Wahlberg, M.: Probability density functions for hyperbolic and isodiachronic locations. J. Acoust. Soc. Am. 112(6), 3046–3052 (2002)
3. Guarato, F., Hallam, J.: Determining sound source orientation from source directivity and multi-microphone recordings. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009. LNCS, vol. 5602, pp. 429–438. Springer, Heidelberg (2009)
4. Guarato, F., Hallam, J., Vanderelst, D.: Determining Sound Source Orientation from Analytical Source Directivity and Real Multi-Microphone Recordings. In: IEEE Advanced Technologies for Enhanced Quality of Life, pp. 106–111 (2009)
5. Guarato, F., Hallam, J., Vanderelst, D.: Determining sound source orientation from source directivity and real multi-microphone recordings: experimental comparisons using both analytical and measured directivity models. In: Proceedings of TAROS (2009) (in press)
6. Riquimaroux, H.: Measurement of biosonar signals of echolocating bat during flight by a telemetry system. J. Acoust. Soc. Am. 117(4), 2526 (2005)
7. Otani, M., Ise, S.: Fast calculation system specialized for head-related transfer function based on boundary element method. J. Acoust. Soc. Am. 119, 2589–2598 (2006)
8. Otani, M., Ise, S.: A fast calculation method of the head-related transfer functions for multiple source points based on the boundary element method. Acoust. Sci. and Tech. 24, 259–266 (2009)
9. Matsuo, I., Kunugiyama, K., Yano, M.: An echolocation model for range discrimination of multiple closely spaced objects: Transformation of spectrogram into the reflected intensity distribution. J. Acoust. Soc. Am. 115(2), 920–928 (2004)
10. Matsuo, I., Yano, M.: An echolocation model for the restoration of an acoustic image from a single-emission echo. J. Acoust. Soc. Am. 116(6), 3782–3788 (2004)
11. Peremans, H., Hallam, J.: The spectrogram correlation and transformation receiver, revisited. J. Acoust. Soc. Am. 104(2), 1101–1110 (1998)

Simulating the Morphological Feasibility of Adaptive Beamforming in Bats

Dieter Vanderelst*, Fons De Mey, and Herbert Peremans

Active Perception Lab,
University Antwerp
{Dieter.Vanderelst,Fons.Demey,Herbert.Peremans}@ua.ac.be

Abstract. It has been suggested that it is advantageous for bats to adapt their emission beam pattern depending on the situation. Hartley [9] has proposed that bats could steer the direction in which they emit most energy by controlling the phase relationship between the sound emerging from both nostrils. In this paper, we evaluate based on simulations, whether such an adaptive mechanism would be viable in FM bats given their specialized facial morphology. We find that these bats could indeed relocate the center of their emission beam pattern using a phased array mechanism. Furthermore, we list two ways in which this would help bats localizing target objects.

1 Introduction

All bats of the suborder Microchiroptera use biosonar as a means of navigating and hunting in dark and complex environments [8]. These animals emit short ultrasonic pulses and derive information about the environment and prey from the returning echoes. A number of studies have revealed that bats use their sonar systems adaptively. They have been found to change the bandwidth, the frequency and the duration of their emissions to better suit the task requirements [24,22,3]. In addition to actively controlling the properties of individual emissions, bats are also known to adapt the properties of the emitted sequence of calls [23,13] to different stages of the same task e.g., systematic changes in the inter pulse interval when homing in on prey. Social constraints on top of task constraints can also induce bats to alter the properties of their emissions. It has recently been shown that individual bats hunting in a group shift the frequencies of their emission to avoid jamming [25,3] or stop echolocating altogether [2].

So far, one type of adaptation has received little attention. Bats could assign more energy to a part of space (a direction) that is of particular interest to them by either moving their heads or by actively shaping the radiation pattern of their emissions (i.e. their emission beam pattern).

* D.V. and F.D.M. were supported by the European Commission (Cilia and Chiroping projects). *M. microtis* specimen provided by I. Geipel and E. Kalko, University Ulm. All datasets will be made publicly available on the website of the Chiroping Project (See www.chiroping.org for details).

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 136–145, 2010.

In an experiment, the bat *Eptesicus fuscus* was found to actively probe the boundaries of a passage through an obstacle [22]. This indicates that bats select what parts of the environment they ensonify. In addition, evidence suggests that bats actively shape their emission beam pattern. Surlykke and colleagues [24] suggested that *Myotis daubentonii* can vary the width of its emission beam pattern. According to these authors, this bat focuses its emitted energy to be able to detect targets from a larger distance if needed. *Rhinolophus ferrumequinum* has been found to vary it emission beam pattern considerably despite fixing the head-aim [17]. This last finding has let Hartley [9] to suggest a mechanism by which CF-FM bats calling through their nose could alter the shape of their emission beam pattern. He proposed that these bats could control the horizontal position of the acoustical axis by actively varying the phase relationship between the sound waves emerging from both nostrils. CF-FM bats emit calls that mainly consist of a long narrow band component of constant frequency. The proposal of Hartley [9] was inspired by existing radar technology, i.e. Phased Array Beam Steering. Indeed, varying the phase difference between the nostrils would enable the bat to steer the direction of the mainlobe of the emissions in the fashion of a Phased Array [19]. So far, the results of Schnitzler and Grinnell [17] are the only evidence that indicates inter call variation in the emission beam pattern in a bat species emitting through its nose. Interestingly, Schnitzler and Grinnell [17] have not elicited calls from (partially) sedated bats by external brain stimulation as is typically done (e.g. [10,18,21]). Instead, their test subjects were awake. Therefore, the animals in their experiments might have been more in control of their vocalizations than is usually the case.

Here we numerically evaluate whether the suggested beam steering mechanisms could be of practical use to bats. Hartley [9] has evaluated his proposal only by considering the beam emission pattern of two baffled pistons representing the nostrils of a bat. However, in bats the nostrils are typically embedded in a facial morphology that is assumed to have evolved to support the beamforming of the bat. Most bats that call through their nose are equipped with a prominent noseleaf [14]. In the few species studied, these structures have been shown to play an important role in directing the emitted pulses [17,18,10]. Removing or altering the noseleaf results in a broadened or changed emission beam respectively. The noseleaf, as well as the rest of the facial morphology, could constrain the directionality of the emission beam pattern and reduce the range over which the beam can be relocated. Furthermore, in his original proposal, Hartley [9] did not evaluate how the spatial sensitivity of a bats complete sonar system (i.e. both hearing and emission) is altered by beamsteering. We want to evaluate how relocating the emission beam changes the sensitivity of the complete sonar system.

Several authors have reported that bats can move their noseleaf back and forth [10,18,26]. Hence, as suggested by these authors, noseleaf deformation could be an additional mechanism by which these animals could actively alter their emission beam patterns. Therefore, we also test whether, controlling

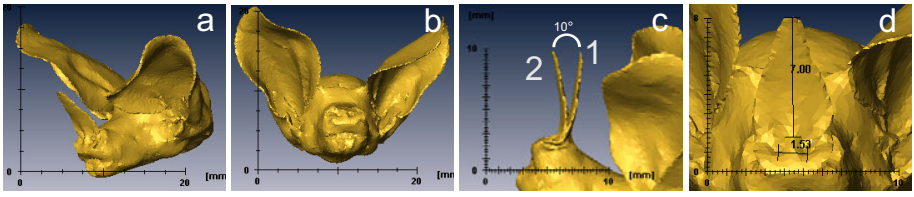


Fig. 1. Views of the 3D model of *M. microtis*: (a) perspective view of the model; (b) Model with noseleaf removed ; (c) Illustration of the bending of the noseleaf: (1) original position of the noseleaf; (2) bent position of the noseleaf; (d) Detail and measurements of the noseleaf.

the horizontal position of the emission beam pattern would be compatible with control over the vertical position due to noseleaf movement.

We test Hartely's [9] hypothesis by simulating both the emission beam pattern and the spatial hearing sensitivity of the neotropical bat *Micronycteris microtis*. This bat emits broadband pulses containing several harmonics between 50 and 160 kHz [7] making its noseleaf, measuring about 7 mm in length, quite large with respect to both the wavelengths in the emission and the bat's body dimensions (length: 35-51 mm, [16]).

2 Methods

The complete head of a *Micronycteris microtis* specimen (see figure 1) was scanned using a Skyscan 1076 Microtomography machine at a resolution of 35 μm. After reconstruction a 3D mesh model was extracted using standard biomedical software. This initial model was semi-automatically simplified and smoothed (see [4] for details of the procedure).

The final model was subjected to an acoustic 3D boundary element simulation [15]. This simulation technique has been validated against measurements of the spatial hearing pattern for the bat *Phyllostomus discolor* [4]. Virtual receivers were placed on an imaginary sphere with a radius of 1 m centered around the bat. Virtual point sources were placed at the location of the nostrils to simulate the emission beam pattern The simulations were run for different versions of the bat model: the intact model, a model from which the noseleaf is removed (figure 1b) and a model in which the noseleaf is bent forward about 10 degrees (figure 1c). Based on the spectrum of the calls of *M. microtis* reported by Geipel [7], we simulated the emission patterns from 50 kHz to 160 kHz in steps of 500 Hertz. Emission beam patterns are simulated with a spatial resolution of 2.5 degrees in azimuth (θ) and elevation (ϕ) over a range of -90 to +90 degrees (the resolution is given by the position of the virtual receivers).

The spatial hearing sensitivity was simulated by placing five virtual point sources in the right ear canal [4]. The same virtual receivers as for simulating the emission pattern were used. We can use sources instead of receivers to simulate the spatial hearing sensitivity of the bat due the reciprocity principle [19].

We report on the simulated sound pressure picked up by the virtual receivers averaged across the five sources.

Below, we report on the normalized acoustic energy radiated or received per unit solid angle in each direction as given by [19],

$$E_{\theta,\phi} = 10 \cdot \log 10 \left[\sum_{f} \frac{P^2_{f,\theta,\phi}}{\mathrm{Max}_{f,\theta,\phi}\, P^2_{f,\theta,\phi}} \right] \tag{1}$$

with $P_{f,\theta,\phi}$ denoting the magnitude of the pressure and f the frequency. In this equation, $\mathrm{Max}_{f,\theta,\phi}\, P^2_{f,\theta,\phi}$ denotes the maximum magnitude across all azimuth and elevation positions for each frequency. The simulated emission beam pattern is independently normalized for each frequency such that

$$\sum_{\theta,\phi} P^2_{f,\theta,\phi} \cos\phi \Delta\theta \Delta\phi = 1. \tag{2}$$

Hence, the results presented below assume that the bat emits calls with a flat spectrum in the range 50 to 160 kHz. Additionally, we assume that all radiated energy stays within the frontal hemisphere.

We conjecture that, because *M. microtis* uses broadband signals, azimuthal beam steering is most easily achieved by differentially varying the path lengths of the sound channels. Anatomical findings support the hypothesis that bats can deform the nasal tract [6]. Furthermore, from our CT data of *M. microtis* we found that the complex cavities leading up to both nostrils are bifurcated for about 10 mm (see figure 2). This distance seems long enough for the bats to be able to impose a functionally relevant path length difference between the sound channels leading up to the left and the right nostril. In the current report, we investigate the effect of a path length difference up to 1 mm. It should be noted that, to achieve this, the bat needs only be capable of creating an opposite change of $\mp$ 0.5 mm in the length of each of the paths leading to the right and the left nostril.

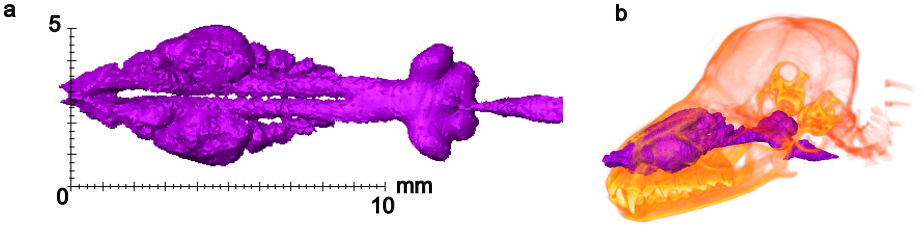


Fig. 2. (a) Top view of a 3D model of the nasal and paranasal cavities of the scanned specimen of *M. microtis*. (b) The same 3D model embedded in a rendering of the skull for reference. All measures in millimeter.

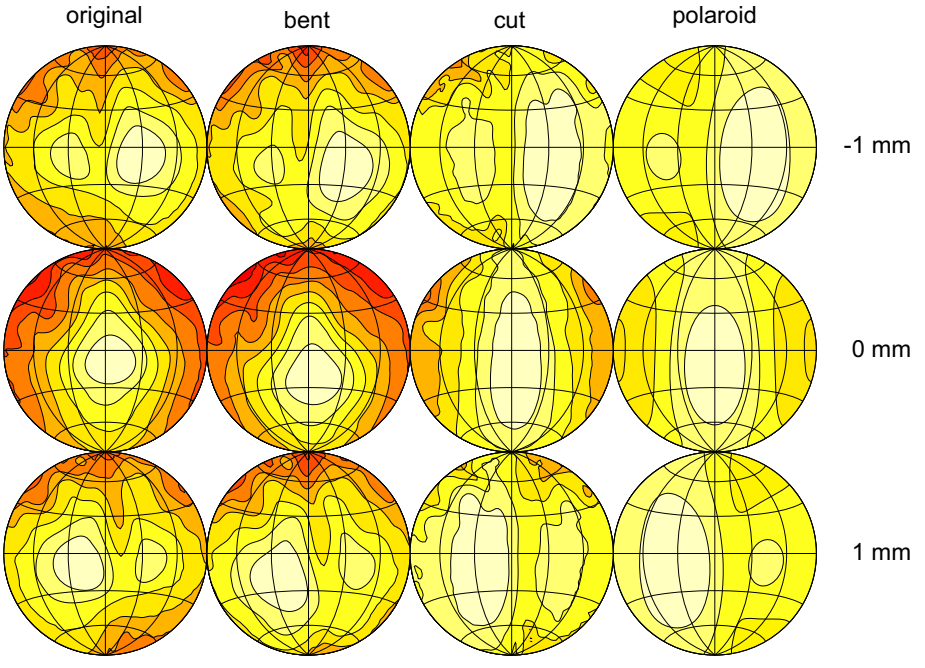


Fig. 3. The emission beam pattern for the 3 bat models and the baffled pistons as a function of three imposed path length differences (-1 mm, 0 mm and 1 mm).The patterns are in dB and normalized (max. = 0). The contours are spaced 3 dB apart. Orig.: Original model, Bent: Model with bent noseleaf, Cut: Model without noseleaf, Pol.: Baffled pistons.

3 Results

We first discuss the effects of the introduced path length differences on the simulated emission beam pattern of *M. microtis*. In a subsequent section we discuss the effect on the combined spatial sensitivity of the bat sonar system (emission beam pattern convolved with the hearing spatial sensitivity).

3.1 Emission Beam Pattern

We calculated the emission beam patterns for the different 3D models by imposing a difference in path length varying between -1 mm and 1 mm in steps of 0.25 mm (-1 mm corresponding with the path length leading up to the left nostril being 1 mm longer than that of the right nostril). The resulting emission patterns are illustrated in figure 3. For each imposed path length difference we extracted the centroid of the main lobe in the emission pattern of the different models. The main lobe is defined as the region within the -3 dB contour and its centroid gives a stable estimate of the direction in which the most energy is radiated [4,5,1].

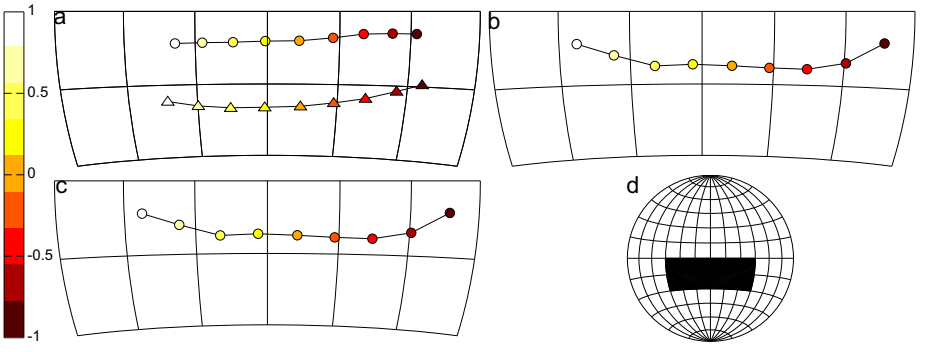


Fig. 4. (a) The location of the centroid for the 3D bat model as a function of imposed path length difference between the nostrils. The circles depict the data for the original model while the triangles indicate the data for the model with the bent noseleaf. (b) Idem, but for the model without the noseleaf. (c) Idem, but for the baffled pistons placed in the same configuration as the nostrils of the bat specimen. (d) This figure indicates which part of the frontal hemisphere is plotted in figures a-c.

Imposing a path length difference of 1 mm between the sound channels leading up to the left and the right nostril causes the main lobe of the emission beam pattern to shift in azimuth over a range of approximately 50 degrees (figure 4a). Bending the noseleaf 10 degrees forward shifts the mainlobe to a new position 13 degrees lower in elevation. Furthermore, the results show that both mechanisms can be combined to steer the sonar beam in both azimuth and elevation. Indeed, imposing the same path length differences between the nostrils with the noseleaf bent down yields about the same sweep of the main lobe in azimuth than for the model with the noseleaf in the upright position (figure 4a).

The simulation approach used here allows to investigate the limitations imposed on beam steering by the facial morphology of *M. microtis*. For that purpose, a simulation of a model with the noseleaf removed (see figure 1b) is included. The results in figure 3 show that the radiation pattern of the bat model without the noseleaf is significantly elongated along the elevation dimension showing that the noseleaf's role is mostly, as expected from its shape, focusing the energy radiated along the elevation dimension. However, the results in figure 4b show that the beam steering is not enhanced by removal of the noseleaf.

We also tested whether the facial morphology, apart from the noseleaf, of *M. microtis* imposes any limits on the beam steering. To this end, the emission beam pattern of two baffled pistons [19] was evaluated. The diameter and the orientation of the pistons were chosen to provide an optimal fit to the radiation pattern of both nostrils. The nostrils turned out to be best fitted by pistons with a diameter of 1 mm and an orientation of 0 degrees in azimuth and -13 degrees in elevation (fitting data and procedure omitted due to space constraints). The pistons were spaced 1.53 mm apart in the horizontal direction. This is the distance between the nostrils of the *M. microtis* specimen used to create the model.

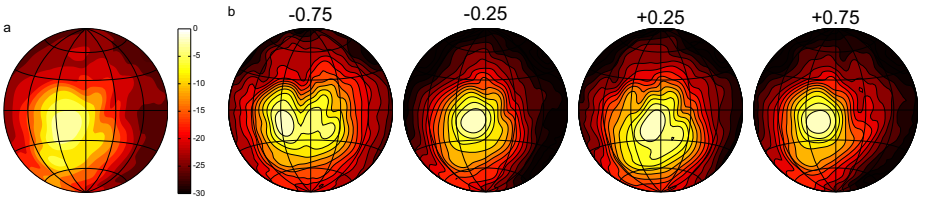


Fig. 5. (a) The spatial sensitivity of the right ear of the 3D bat model. The values are in dB and the contourlines are spaced 3 dB apart. (b) The combined spatial sensitivity of the sonar systems' right ear as function of the imposed path length difference.

For a sinusoidal excitation, the magnitude of the far-field pressure produced by the baffled piston emitters [19] is calculated using the equation derived by [20].

Figures 3 and 4c show that both the extent of the sweep of the combined piston model as well as the overall shape of the radiation patterns correspond quite well with the bat model with the removed noseleaf.

The results presented so far indicate that the morphology of the emission system of the bat allows for phased array beam steering. Moreover, the facial morphology does not constrain this ability as the azimuthal sweep for the original model had the same extent as that of two pistons placed in the same configuration as the nostrils.

3.2 Combined Spatial Sensitivity

The spatial hearing sensitivity of the right ear of the model is displayed in figure 5a. We convolved the hearing sensitivity of the left ear of the model with the emission beam pattern to arrive at the combined spatial sensitivity of the complete sonar system. The combined spatial sensitivity of the system for the nine path length differences is plotted in 5b. From these plots, it can be seen that the direction of highest sensitivity shifts with the introduced path length difference.

3.3 Gained Localization Performance

In the previous paragraphs we discussed how the simulated emission beam pattern and the complete spatial sensitivity of the bat model changed as a function of the imposed path length difference. Here we briefly indicate how these changes could support better localization by the bat.

By sweeping the center of the emission beam pattern through a number of known azimuth positions, the bat could locate an object by noting the imposed path length difference that returns the strongest echo [9]. Indeed, as illustrated in figure 6a, for a wide range of azimuths, the path length difference resulting in the strongest echo is related to the azimuthal position of the target. Therefore, relocating the main beam of the emission pattern allows for a simple azimuth determination in the range -40 to +40 degrees.

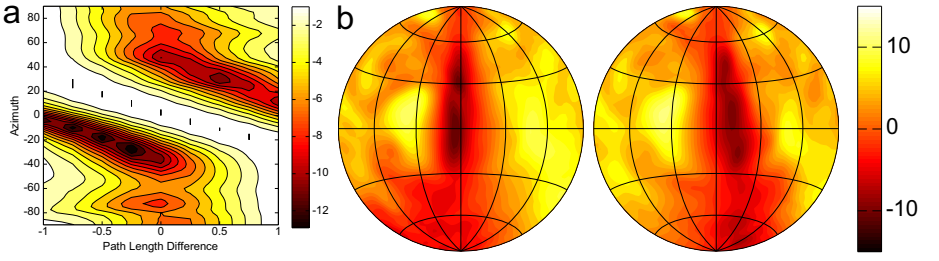


Fig. 6. (a) Energy in the received echo as a function of imposed path length difference and azimuthal target position for 0 degrees elevation. (b) The gain in energy as a function of space by imposing a path length difference between the nostrils of -1 and +1 mm.

Sweeping the emission beam, also allows the bat to assign energy to a region of space for which its ears are marginally sensitive. As illustrated in figure 6b, imposing a path length difference of -1 or +1 increases the energy assigned to certain peripheral regions by up to 10 dB. This might allow the bat to exploit the spectral localization cues generated in the periphery by the filtering of the ears that are not usable when the echo is too weak.

4 Conclusion

Phased Array beam steering has been used in sonar applications for a long time. Both medical and industrial applications make use of phased array principles. Exactly 20 years ago, Hartley [9] must have been the first to propose that animals, in casu bats, might steer their sound emissions by introducing a phase shift between two sites of sound production, i.e. the left and the right nostril. We investigated whether the facial morphology of a bat allows for this. From the results presented here we conclude that, theoretically at least, *M. microtis* is able to use both path length differences and noseleaf movements to shift the location of the main lobe of its spatial sensitivity. Steering the emission beam by phase differences in addition to head movements might have evolved as a energetically efficient and fast way of relocating the focus of attention (akin to eye movements in mammals). Moreover, being able to shift the emission beam without moving the head could be interesting for flying and swimming animals when fluid dynamics do not allow for large head movements.

FM bats typically emit rather short pulses. For example, *M. microtis* emits pulse of less than 1 ms. Therefore, it is probably impossible for these animals to deform their noseleaf or change the path a length differences between the nostrils during the course of a call. However, the bat could relocate the main beam of its spatial sensitivity from call to call. When positioning the main at different azimuthal positions, a simple search for the loudest echo yields a stable cue about the azimuthal location of a target. Additionally, bats could allocate more energy to the periphery and thus exploit the spectral cues generated by the ears in this part of the frontal hemisphere. However, these are but two

simple ways in which a movable main beam could be of use to a bat. It is possible that other uses of a movable mainlobe might provide the bat with more interesting cues about the location or the identity of targets. To investigate this, a theoretical framework must be employed that allows to quantify the contribution of the alterations to the spatial sensitivity of the bat to the localization and identification performance. Yovel and colleagues [27] have recently shown that *Rousettus aegyptiacus* alternates the direction in which it emits most energy when tracking a target. Moreover, they were showed that this strategy makes sense based on information-theoretic calculations. However, in this paper we content ourselves by confirming that the mechanism proposed by Hartley [9] is indeed allowed for the by facial morphology of the bat species under study. Indeed, our CT data indicates that the tracts leading up to the nostrils might be long enough for the proposed deformation to take place. Thus far, the evidence indicating that bats use a mechanism akin to phased array beam steering is very sparse but we hope that our analysis leads to experiments that verify directly whether bats do actually use these strategies or not.

Recently, bottlenose dolphins (*Tursiops truncatus*) have been found to be able to steer their sonar beams over an angle of 20 degrees without moving their heads [12]. The mechanism behind this might be akin to that of a phased array as Beluga whales (*Delphinapterus leucas*) were found to have two sites to produce echolocation clicks that can be controlled independently [11]. If dolphins and/or bats would turn out to be steering their beams using Phased Arrays this would be another example illustrating that Nature has little to learn from engineers. Indeed, technology thought up for our animats (i.e. industrial and medical appliances) has often existed for millions of years in animals.

References

1. Carlile, S.: The auditory periphery of the ferret. I: Directional response properties and the pattern of interaural level differences. The Journal of the Acoustical Society of America 88, 2180 (1990)
2. Chiu, C., Xian, W., Moss, C.F.: Flying in silence: Echolocating bats cease vocalizing to avoid sonar jamming. Proceedings of the National Academy of Sciences of the United States of America 105(35), 13116–13121 (2008)
3. Chiu, C., Xian, W., Moss, C.F.: Adaptive echolocation behavior in bats for the analysis of auditory scenes. Journal of Experimental Biology 212(9), 1392 (2009)
4. De Mey, F., Reijniers, J., Peremans, H., Otani, M., Firzlaff, U.: Simulated head related transfer function of the phyllostomid bat *Phyllostomus discolor*. Journal of the Acoustical Society of America 124(4), 2123–2132 (2008)
5. Firzlaff, U., Schuller, G.: Spectral directionality of the external ear of the lesser spear-nosed bat, *Phyllostomus discolor*. Hearing Research 181, 27–39 (2003)
6. Göbbel, L.: Morphology of the external nose in hipposideros diadema and lavia frons with comments on its diversity and evolution among leaf-nosed microchiroptera. Cells Tissues Organs 170(1), 39–60 (2002)
7. Geipel, I.: Prey detection of the neotropical leaf-nosed bat mycronycteris microtis in panama. Master's thesis, Georg-August-Universitat, Gottingen, Germany (2008)

8. Griffin, D.R.: Listening in the dark; the acoustic orientation of bats and men. Yale University Press, New Haven (1958)
9. Hartley, D.J.: Phased-array beam scanning as a possible aid to horizontal localization in horseshoe bats. Journal of the Acoustical Society of America 88(6), 2889–2891 (1990)
10. Hartley, D.J., Suthers, R.A.: The sound emission pattern and the acoustical role of the noseleaf in the echolocating bat, Carollia perspicillata. Journal of the Acoustical Society of America 82(6), 1892–1900 (1987)
11. Lammers, M.O., Castellote, M.: The beluga whale produces two pulses to form its sonar signal. Biology Letters 5(3), 297 (2009)
12. Moore, P.W., Dankiewicz, L.A., Houser, D.S.: Beamwidth control and angular target detection in an echolocating bottlenose dolphin (Tursiops truncatus). The Journal of the Acoustical Society of America 124, 3324 (2008)
13. Moss, C.F., Surlykke, A.: Auditory scene analysis by echolocation in bats. The Journal of the Acoustical Society of America 110, 2207 (2001)
14. Nowak, R.M., Walker, E.P., Kunz, T.H., Pierson, E.D.: Walker's bats of the world. Johns Hopkins Univ. Pr. (1994)
15. Otani, M., Ise, S.: Fast calculation system specialized for head-related transfer function based on boundary element method. Journal of the Acoustical Society of America 119(5), 2589–3598 (2006)
16. Reid, F.: A field guide to the mammals of Central America & Southeast Mexico. Oxford University Press, USA (1997)
17. Schnitzler, H.-U., Grinnell, A.D.: Directional sensitivity of echolocation in the horseshoe bat, *Rhinolophus ferrumequinem*. i. directionality of sound emission. Journal of Comparative Physiology A 116, 51–61 (1977)
18. Shimozawa, T., Suga, N., Hendler, P., Schuetze, S.: Directional sensitivity of echolocation system in bats producing frequency-modulated signals. Journal of Experimental Biology 60(1), 53–69 (1974)
19. Skolnik, M.I.: Introduction to radar systems, 590 p. McGraw Hill Book Co., New York (1980)
20. Strother, G.K., Mogus, M.: Acoustical beam patterns for bats: Some theoretical considerations. The Journal of the Acoustical Society of America 48, 1430 (1970)
21. Suga, N.: Cortical computational maps for auditory imaging. Neural Networks 3(1), 3–21 (1990)
22. Surlykke, A., Ghose, K., Moss, C.F.: Acoustic scanning of natural scenes by echolocation in the big brown bat, Eptesicus fuscus. Journal of Experimental Biology 212(7), 1011 (2009)
23. Surlykke, A., Moss, C.F.: Echolocation behavior of big brown bats, Eptesicus fuscus, in the field and the laboratory. The Journal of the Acoustical Society of America 108, 2419 (2000)
24. Surlykke, A., Pedersen, S.B., Jakobsen, L.: Echolocating bats emit a highly directional sonar sound beam in the field. Proceedings of the Royal Society B: Biological Sciences 276(1658), 853–860 (2009)
25. Ulanovsky, N., Fenton, M.B., Tsoar, A., Korine, C.: Dynamics of jamming avoidance in echolocating bats. Proceedings of the Royal Society B: Biological Sciences 271(1547), 1467–1475 (2004)
26. Weinbeer, M., Kalko, E.K.V.: Ecological niche and phylogeny: the highly complex echolocation behavior of the trawling long-legged bat, Macrophyllum macrophyllum. Behavioral Ecology and Sociobiology 61(9), 1337–1348 (2007)
27. Yovel, Y., Falk, B., Moss, C.F., Ulanovsky, N.: Optimal Localization by Pointing Off Axis. Science 327(5966), 701 (2010)

On the Influence of Sensor Morphology on Vergence

Harold Martinez, Hidenobu Sumioka, Max Lungarella, and Rolf Pfeifer

Artificial Intelligence Laboratory, Department of Informatics, University of Zurich, Andreasstrasse 15, 8050 Zurich, Switzerland
{martinez,sumioka,lunga,pfeifer}@ifi.uzh.ch
http://ailab.ifi.uzh.ch/

Abstract. In the field of developmental robotics, a lot of attention has been devoted to algorithms that allow agents to build up skills through sensorimotor interaction. Such interaction is largely affected by the agent's morphology, that is, its shape, limb articulation, as well as the position and density of sensors on its body surface. Despite its importance, the impact of morphology on behavior has not been systematically addressed. In this paper, we take inspiration from the human vision system, and demonstrate using a binocular active vision platform why sensor morphology in combination with other properties of the body, are essential conditions to achieve coordinated visual behavior (here, vergence). Specifically, to evaluate the effect of sensor morphology on behavior, we present an information-theoretic analysis quantifying the statistical regularities induced through sensorimotor interaction. Our results show that only for an adequate sensor morphology, vergence increases the amount of information structure in the sensorimotor loop.

Keywords: Embodied cognition, visual development, sensor morphology, information structure.

1 Introduction

In nature, living organisms are embodied and embedded in their ecological niches. Their neural structures have evolved to sample and process sensor inputs to create adaptive neural representations, and to select and control motor outputs to position their bodies or to impose changes on the environment [1]. Such sensorimotor activity involves a dynamic reciprocal coupling between organism and environment known as embodiment [2]. The implications of embodiment are far reaching and go beyond the mere interaction between a body and the environment in which it is embedded, to include also as the information-theoretic interrelations among the sensory system, the body, the environment, and the controller. Embodiment is understood as a fundamental aspect to develop cognitive capabilities because it enables a continuous flow of information between sensors, neural units, and effectors. The pattern of information flow defines complex sensorimotor networks, consisting of structured relations and dependencies among sensor, neural, and motor variables. This information structure, such as correlations, redundancies, and

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 146–155, 2010.

invariances in the sensorimotor loop makes learning, prediction, action selection, adaptability and developmental process possible [1],[3], [4].

Some algorithms employed to bootstrap the development of skills [5], [6], [7] are designed to restrict the action selection (repertoire) in order to increase predictability of the sensorimotor loop. In these cases, the objective function that drives the development of the agent is some quantitative measure of the agent's sensorimotor interaction (e.g. information gain, transfer entropy, the prediction error of the next sensor input, and the improvement in the prediction in the sensor input). Generally, in these mathematical frameworks, embodiment is simplified to the interaction with the environment.

In the application of the developmental algorithms there are some limitations, such as the number of sensor inputs, degrees of freedom (DOF), and convergence time among others. We claim that because of the embodiment, the sensor morphology and the robot body should be taken into account in order to exploit statistical dependencies and causal relations in the sensorimotor loop. Therefore appropriate sensor morphology could be the mechanism not only to decrease the convergence time, but also to sense information flow which increases the predictability, limiting the action space naturally.

In the first months of life, a child is able to develop sensorimotor competencies almost from scratch [8]. Behaviors such as tracking, saccadic movements and fixation start to develop at the beginning of a child's life and are mature after about three months [9], [10]. The development of behaviors like vergence could be explained as the result of the increment in predictability among actions and sensor inputs.

In this paper, we provide an information theoretical analysis that shows why the sensor morphology, and the sensorimotor coupling could bootstrap the development of vergence. The latter behavior increases the causality among actions and sensors, hence increasing the predictability of the future sensor stimulation, and enabling the agent to develop a model of the environment. In order to measure how much the agent can predict given specific sensor morphology we used transfer entropy as a measure of causality [11].

This paper is organized as follows. First, we describe the robot head platform used for our experiment, the sensor morphology, and the causality measure employed to quantify the results in the experiment. Then, we present the experiment and the related results. Before concluding the paper, we discuss our results and some of their implications for theories of infant development.

2 Materials and Methods

2.1 Robot

Our experimental testbed was the iCub robot head [12]. The iCub is an open humanoid platform, developed in the context of the RobotCub project, to promote studies in cognitive systems and embodied cognition. In contrast with other humanoid robots as QRIO, ASIMO, HOAP-2, the iCub robot head has 6 DOF

Fig. 1. iCub robot head

(Fig. 1) in order to emulate behaviors like vergence, smooth pursuit, and saccades, typical of the vision system. Both eyes can pan independently, and the common tilt movement is actuated by a belt system placed between the cameras. 3 DOF are used to control the neck of the head, while the other 3 DOF are used to control the cameras. Our experiments were conducted controlling just the latter 3 DOF. The neck of the robot was immobile during all the procedure. The image delivered by each camera has a resolution of 640x480 at 30 fps.

2.2 Sensor Morphology

The human vision system has to interpret a 3D world from 2D projections, and in this process the ocular movements play an important role. These motions are not an innate feature, but are developed through a prolonged interaction with the environment. Moreover, abilities such as stereopsis (depth perception from binocular vision that exploits parallax disparities) are a result of this development in the first months of life [13], [14].

The question is what mechanism drives this process, and what could be the contribution of the morphology of the eyes and the ocular muscles. In order to address this matter, we implemented a set of biologically plausible information processing mechanisms in the iCub head. Based on the results from Nothdurf (1990) [15], who showed how neurons respond to simple features such as intensity contrast, color, orientation, and motion, color was the main feature used in our experiments. These features define the pre-attentive visual cues [16]. In addition, the human vision is capable of binocular fusion; i.e. a single image is seen although each eye has a different image of the environment [17]. In our implementation we applied the average of both cameras to create the binocular single image. Another important aspect is foveation. Our eye has, in its center, a greater number of receptors than in the periphery. This was modeled with the log-polar transform, which changes the coordinate system from Cartesian (x,y) to the logarithm of the magnitude and the angle:

$$\rho(x, y) = M \cdot log(\sqrt{(x^2 + y^2)}) \; . \tag{1}$$

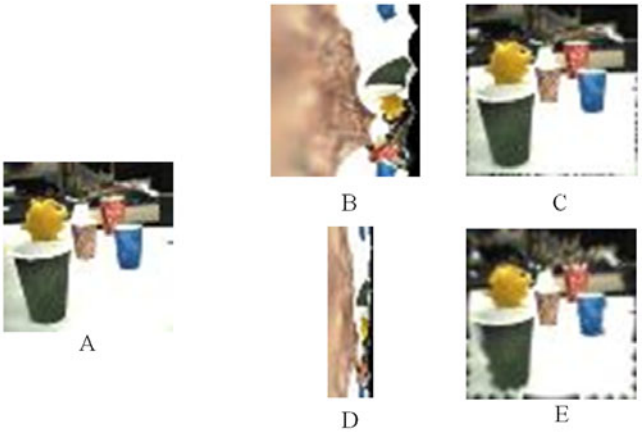


Fig. 2. Log-polar transform of 60x60 image. (A) Raw image. (B) Log-polar transform of A with M = 40. (C) Inverse log-polar transform from B. (D) Log-polar transform of A with M=12. (E) Inverse log-polar transform from D. Notice that the inverse transform is the reconstruction of the image with fewer pixels in peripheral area.

$$\varphi = arctan(\frac{y}{x}) \, . \tag{2}$$

Where x and y are the coordinates of the pixel in the picture, ρ is the logarithm of the magnitude and φ is the angle. The parameter M was used to increase or decrease the number of pixels used in the log-polar transform (Fig. 2). In our experiments, these aspects (color, foveation and image composition from the two cameras) were used to find out whether the vergence behavior increases information structure.

2.3 Information Metric

In order to present how the causality among the variables (actuators and sensors) relies on the morphology and specific behaviors, we used the transfer entropy [11]. This measure was selected to compare the results of the experiments, due to its capacity to find the nonlinear statistical dependencies which can be used to understand why a specific behavior could yield better causal relations among the data.

Originally, transfer entropy was introduced to identify the directed flow or transfer of information (also referred to as "causal dependency") between time series [11]. Given two time series X and Y, transfer entropy essentially quantifies the deviation from the generalized Markov property: $p(x_{t+1} \mid x_t) = p(x_{t+1} \mid x_t, y_t)$, where p denotes the transition probability. If this deviation is small, then Y does not have relevance on the transition probabilities of system X. Otherwise, if the deviation is large, then the assumption of a Markov process is not valid, The deviation of the assumption can be quantified by the transfer entropy, formulated as the Kullback-Leibler entropy:

$$T(Y \rightarrow X) = \sum_{X_{t+1}} \sum_{X_t} \sum_{Y_t} p(x_{t+1}, x_t, y_t) log(\frac{p(x_{t+1}|x_t, y_t)}{p(x_{t+1}|x_t)}) \ . \tag{3}$$

Where the sums are over all amplitude states, and the index $T(Y \rightarrow X)$ indicates the influence of Y on X. The transfer entropy is explicitly nonsymmetrical under the exchange of X and Y — a similar expression exists for $T(X \rightarrow Y)$ — and can thus be used to detect the directed exchange of information (e.g., information flow, or causal dependency) between two systems. As a special case of the conditional Kullback-Leibler entropy, transfer entropy is non-negative, any information flow between the two systems resulting in $T > 0$. In the absence of information flow, i.e., if the state of system Y has no influence on the transition probabilities of system X, or if X and Y are completely synchronized, $T(Y \rightarrow X) = 0$ bit.

2.4 Data Analysis

All numerical computations for data analysis were carried out in Matlab (Mathworks, Natick, MA), and were performed for data samples of 12,300 time steps. The resolution of the cameras was reduced to 60x60 pixels to facilitate the calculations. We used gray scale images to reduce computational costs for analyzing causal relations among sensor and motor variables. Given that the proposed sensor morphology is defined by the binocular single image and the foveation, we can still evaluate the effect of our proposed sensor for vergence. In order to calculate the transfer entropy between the images and the actions, we first generated a causality measure for each pixel, which was the sum of transfer entropy between each DOF and the pixel (Eq. 4). The causality of the image then was measured as the average causality of all the pixels (Eq. 5)

$$T_{p_j} = \sum_{E_i} T(E_i \rightarrow p_j) \tag{4}$$

$$T_I = \frac{\sum_{p_j} T_{p_j}}{|p|} \ , \tag{5}$$

where E_i is the i^{th} DOF time series, p_j is the j^{th} pixel time series, T_{p_j} is the causality induced by the 3DOF to the j^{th} pixel. T_I is the average causality in the frame averaging all the causality measured in each pixel. To calculate transfer entropy, time series were discretized to 8 states (3 bits) and joint probabilities and conditional probabilities were estimated using the naive histogram technique, that is, as normalized histograms. Temporal delays in [-25, 25] time steps across time series were introduced by shifting one time series relative to the other, thus allowing the evaluation of causal relationships across variable time offsets. Delayed causality was potentially introduced by the discrete nature of the updating of the control architecture and by the temporal persistence of sensor and motor states.

3 Experiment

In this experiment we compare different sensor morphologies and controllers in a fixed task. First, we tested different morphologies to find out which one could reduce the number of inputs to the system. Second, we tested different controllers to see how the sensor morphology restricts the space of coordinated behavior in terms of predictability.

In the setup we place the robot in front of four different cups (Fig. 3A). The objects were distributed in the field of view to force the robot to change the value of the 3 DOF of the cameras. The robot had to look at all of them in a predefined sequence. We used the color based tracker to change the attention of the robot to 4 different objects. In order to measure the influence of sensor morphology on vergence, we developed three different controllers: (1) the left camera performed random movements while the right one followed the sequence; (2) a controller that allowed parallel motions of the left and right camera; and (3) a controller that forced the vergence with both cameras to focus the object. We expect that the control quality (behavior) can affect the predictability, that is the possibility to explain the future based in the actual data and actions, hence validating that vergence is a behavior capable of increasing the causal relations among the pixels and the actions.

3.1 Setup

For the three controllers we tested four different sensor configurations: (1) the average of the left and right image. (2) The inverse log-polar of the average of the left and right image. (3) The log-polar of the average of the left and right image, and (4) a single image, the left camera (Fig. 3B). We used four different log-polar transformations (M = 8, 12, 20 and 40 which reduce the size of the image to 17%, 27%, 43%, and 83% respectively.) For each transformation we ran 8 different experiments for all different kinds of images.

3.2 Results

First, we compared different morphologies using a controller which performs an appropriate vergence. We evaluated in the experiment how the proposed sensor morphology can keep the predictability while it reduces the number of pixels. We compared the measures of transfer entropy of the left image against the average and the inverse average log-polar. As we can see in Figs. 4A-C the causality in all these sensor morphologies changes less than 5%, which means that the pixels in the center are dominant in the causal relation.

The tracker kept the zero disparity region in the center of the image. Therefore, in the log-polar transformation the receptors sample more the object than the periphery. We tested different number of receptors in the average log-polar morphology to see how the causality could be affected. In Figs. 4D-G we presented the results for four different examples. We found out that the reduction of receptors does not decrease the causality. Therefore this sensor morphology keeps the information structure with fewer pixels. This result could be used

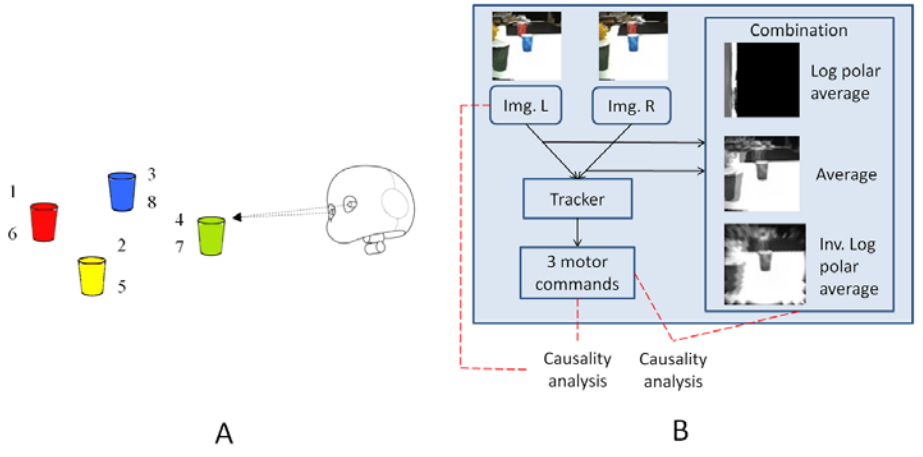


Fig. 3. Experimental setup. (A) The robot is looking at the different cups in the sequence given by the numbers, after 7 the robots starts again with 1. (B) Causal analysis among different sensor and control configurations.

in order to reduce the number of inputs in a developmental algorithm, taking advantage of the sensor morphology.

The different controllers represent different "qualities" of the vergence behavior. As shown in Figs. 4G-I the more accurate the control for vergence, the more causality appears in the sensorimotor loop. From this result we imply that that if the robot looks for predictability in terms of its sensorimotor coupling it has to do vergence.

4 Discussion

The log-polar transform and the average of the two images force the robot to develop vergence, because on the one hand, the log-polar transform allows to better sample the center of the image, and on the other hand, the average of both cameras blurred regions in the image that are not in the zero disparity region. Therefore vergence is aligning the zero disparity region in the center of the image, where the robot has more receptors. The more precise this behavior, the bigger the causal relation among pixels and actions.

The log-polar transform reduces the computational load, and additionally improves the learning, because these are the pixels with the higher causal relations even when the inputs are reduced to 17%. With a normal Cartesian pixel array the rest of the pixels in the learning process are just noise, due to the lack of structure, and in this sense the perception of the agent is decreased.

The causality can be interpreted as the predictability, which allows the agent to develop a model of the world [18]. If the agent is not able to perform vergence then the predictability decreases as it is presented in the experimental results. This means that the learning capability is limited by the predictive capacity of

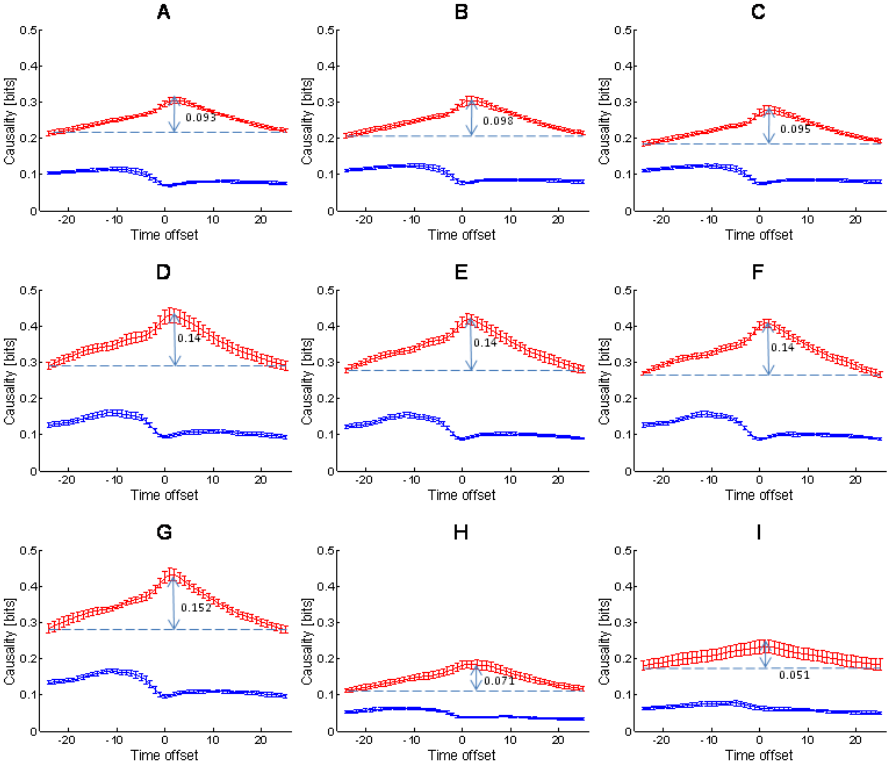


Fig. 4. Transfer entropy among pixels and motor signals. Plots A to I display the average causality as in Eq.(5), $T_{S\rightarrow M}$ (blue), $T_{M\rightarrow S}$ (red). In plots A to G the 3 DOF of the active vision system were controlled independently. (A) Left image. (B) Average image. (C) Average inverse log-polar image with M=8. (D) Average log-polar image with M=40. (E) Average log-polar image with M=20. (F) Average log-polar image with M=12. (G) Average log-polar image with M=8. (H) One camera tracked the object while the other mirrors its movement. The causality was calculated using the average log-polar transform with M=8. (I) The controller is equal to the one used in A, but with the addition of noise in the control signal sent to the left camera. The causality presented in (I) is using the average log-polar transform with M=8.

the sensorimotor loop. In other words, the robot is limited by the "quality" of its control. In this sense the sensor morphology and the combination of different sensor modalities shape the possible developmental behavior.

5 Conclusions

In this study, we implemented a set of biologically plausible information processing mechanisms based on the human vision system. We analyzed the transfer entropy as a function of the sensor morphology and the controller. Our experimental results demonstrate how an appropriate morphology reduces the amount of inputs and increases the predictability in the sensorimotor loop. The reduction of inputs to a system, and the increment of causal relations among motor actions and inputs are key aspects that increase the applicability of developmental algorithms in robots.

The vision system allows us to generate a belief of the environment beyond the simple 3D perception or spatial distribution. Thanks to the interaction with the world and the coupling with other sensor inputs, visual information allows prediction. Our capacity to use our attention towards what it is needed, like a reflex, and the capacity of prediction of our visual system, are two features that makes our vision system a fascinating tool to handle the world, and it is an incredibly complex system that is not easy to isolate or emulate in an artificial platform. In this experiment we show how from the coupling between the visual system and the proprioceptive system the vergence could emerge under the developmental mechanism of predictability. The possible extension of this result might be the development of an attention systems based not just on visual data but in the relations among different sensor systems. The development of the attention system then enables the agent to extract the information relevant for its own tasks providing the substrate for the emergence of behaviors such as eye hand coordination.

In the perspective of human infants our results show that the build up of behavior might be a result of better information structure. Actions like vergence allow us to predict better to understand better the environment, and the integration of several sensor modalities can therefore generate more complex final behaviors in order to achieve structure in several sensor systems.

Acknowledgments. This work was supported in part by the EU Project IST-2004-004370 ROBOTCUB and by the EU Project FP7-ICT-231864 EC-CEROBOT. We would also like to thank Alejandro Hernandez for his valuable comments.

References

1. Lungarella, M., Sporns, O.: Mapping information flow in sensorimotor networks. PLoS Comp. Bio. 2(10), e14 (2006)
2. Pfeifer, R., Lungarella, M., Iida, F.: Self-organization, embodiment, and biologically inspired robotics. Science 318, 1088–1093 (2007)

3. Koerding, K.P., Wolpert, D.M.: Bayesian decision theory in sensorimotor control. Trends Cogn. Sci. 10, 319–326 (2006)
4. Thelen, E., Smith, L.: A dynamic systems approach to the development of cognition and action. MIT Press/Bradford (1994)
5. Schmidhuber, J.: Driven by Compression Progress: A Simple Principle Explains Essential Aspects of Subjective Beauty, Novelty, Surprise, Interestingness, Attention, Curiosity, Creativity, Art, Science, Music, Jokes. In: Anticipatory Behavior in Adaptive Learning Systems, from Sensorimotor to Higher-level Cognitive Capabilities. LNCS (LNAI), pp. 48–76. Springer, Heidelberg (2009)
6. Oudeyer, P.-Y., Kaplan, F., Hafner, V.: Intrinsic Motivation Systems for Autonomous Mental Development. IEEE Trans. on Evol. Comp. 11(2), 265–286 (2007)
7. Barto, A., Singh, S., Chentanez, N.: Intrinsically motivated learning of hierarchical collections of skills. In: 3rd Int. Conf. Devel. Learn., pp. 112–119. IEEE Press, San Diego (2004)
8. Smith, P., Cowie, H., Blades, M.: Understanding children's development. Blackwell, Malden (1998)
9. Tondel, G., Candy, T.: Human infants' accommodation responses to dynamic stimuli. Investigative Ophthalmology & Visual Science 48(2), 949–956 (2007)
10. Aslin, R.N.: Development of binocular fixation in human infants. Journal of Exp. Child Psy. 23(1), 133–150 (1977)
11. Schreiber, T.: Measuring information transfer. Phys. Rev. Lett. 85, 461–464 (2000)
12. Beira, R., Lopes, M., Praça, M., Santos-Victor, J., Bernardino, A., Metta, G., Becchi, F., Saltarén, R.: Design of the Robot-Cub (iCub) Head. In: Conf. on Rob. and Auto., pp. 94–100. IEEE Press, Orlando (2006)
13. Birch, E., Petrig, B.: FPL and VEP Measures of Fusion, Stereopsis and Stereoacuity in Normal Infants. Vision Res. 36(9), 1321–1327 (1996)
14. Birch, E., Morale, S., Jeffrey, B., Oconnor, A., Fawcett, S.: Measurement of stereoacuity outcomes at ages 1 to 24 months: Randot stereocards. Journal of Ame. Asso. for Ped. Opht. and Stra. 9(1), 31–36 (2005)
15. Nothdurf, H.: Texture discrimination by cells in the cat lateral geniculate nucleus. Exp. Brain Res. 82, 48–66 (1990)
16. Itti, L., Koch, C.: Computational Modeling of Visual Attention. Nat. Rev. Neuro. 2(3), 194–203 (2001)
17. Wheatstone, C.: Contributions to the physiology of vision.-Part the First. On some remarkable, and hitherto unobserved, phenomena of binocular vision. Phil. Trans. of the Royal Soci. of London 128, 371–394 (1838)
18. Pearl, J.: Causality: Models, reasoning, and inference. Cambridge University Press, Cambridge (2000)

Adapting Preshaped Grasping Movements Using Vision Descriptors

Oliver Krömer, Renaud Detry, Justus Piater, and Jan Peters

Max Planck Inistitute for Biological Cybernetics
Spemannstr. 38, 72076 Tübignen, Germany
{oliverkro,Jan.Peters}@tuebingen.mpg.de,
{Renaud.Detry,Justus.Piater}@ulg.ac.be

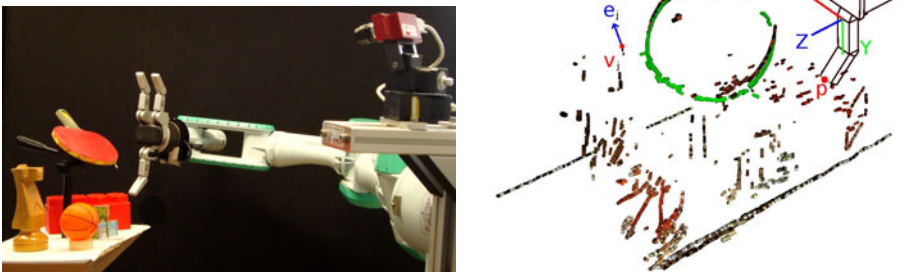
Abstract. Grasping is one of the most important abilities needed for future service robots. In the task of picking up an object from between clutter, traditional robotics approaches would determine a suitable grasping point and then use a movement planner to reach the goal. The planner would require precise and accurate information about the environment and long computation times, both of which are often not available. Therefore, methods are needed that execute grasps robustly even with imprecise information gathered only from standard stereo vision. We propose techniques that reactively modify the robot's learned motor primitives based on non-parametric potential fields centered on the Early Cognitive Vision descriptors. These allow both obstacle avoidance, and the adapting of finger motions to the object's local geometry. The methods were tested on a real robot, where they led to improved adaptability and quality of grasping actions.

1 Introduction

Consider grasping an object at a specific point in a cluttered space, a common task for future service robots. Avoiding collisions is easy for humans, as is preshaping the hand to match the shape of the object to be grasped. Most adults perform these actions quickly and without excessive planning. All of these actions occur before the hand comes into contact with the object, and can therefore be accomplished using stereo vision [1,2]. In contrast, robots often struggle with executing this task, and rely on specially designed sensors (e.g., laser scanner, ERFID) to get accurate and complete representations of the object and environment [3,4], followed by lengthy planning phases in simulation [5].

To avoid excessive planning, a robot can employ a sensor-based controller, which adjusts its motions online when in the proximity of obstacles or other external stimuli [6]. Sensors such as time-of-flight cameras, ultrasonic sonar arrays, and laser range finders are favored for these purposes due to their relatively dense sampling abilities [7,8]. Stereo vision systems, while usually giving sparser readings, have also been used for obstacle detection, especially in the field of mobile robots. However, these methods often rely on task-specific prior knowledge (e.g, assume the ground is flat) and are designed to avoid obstacles completely [8,9],

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 156–166, 2010.



A. Scene B. ECV Representation

Fig. 1. A) The robot used in our experiments and an example of a grasping task in a cluttered environment. **B)** The green ECVDs represent the object to be grasped, while the surrounding ECVDs in the scene are clutter. The coordinate frame of one of the robot's fingers and variables used in section 2 are shown. The x-y-z coordinate system is located at the base of the finger, with z orthogonal to the palm, and y in the direction of the extended finger. The marked ECVD on the left signifies the j^{th} descriptor, with its position at $\mathbf{v}_j = (v_{jx}, v_{jy}, v_{jz})^T$, and edge direction $\mathbf{e}_j = (e_{jx}, e_{jy}, e_{jz})^T$ of unit length. The position of the finger tip is given by $\mathbf{p} = (p_x, p_y, p_z)^T$.

while the robot must get close to the object for grasping tasks. In terms of robot manipulators, the research has focused on coarse object representations of novel objects [10,11,12,13] and using additional sensor arrays when in close proximity to the object [14,15].

In this paper, we propose a sensor-based robot controller that can perform human inspired grasping motions, including preshaping of the hand, smooth and adaptive motion trajectories, and obstacle avoidance, using only stereo vision to detect the environment. The controller uses potential field methods [6], which treat the robot's state as a particle in a force-field; i.e. the robot is attracted to a goal state, and repelled from obstacles.

The system uses the dynamical system motor primitive (DMP) framework [16, 17] for the attractor field, which are capable of encoding complex trajectories and adapting to different grasp locations. These DMPs are implemented as a passive dynamical system superimposed with an external force; i.e.,

$$\ddot{y} = \alpha_z(\beta_z \tau^{-2}(g - y) - \tau^{-1}\dot{y}) + a\tau^{-2} f(x), \tag{1}$$

where α_z and β_z are constants, τ controls the duration of the primitive, a is an amplitude, $f(x)$ is a nonlinear function, and g is the goal for the state variable y. The variable $x \in [0, 1]$ is the state of a canonical system $\dot{x} = -\tau x$, which ensures that the different hand and arm motions are synchronized. The function $f(x)$ is used to encode the trajectory for reaching the goal state, and takes the form $f(x) = (\sum_{i=1}^{M} \psi_i)^{-1} \sum_{j=1}^{M} \psi_j(x) w_j x$, where $\psi(x)$ are M Gaussian basis functions, and w are weights. The weights w can be programmed through imitation

learning [18]. The DMPs treat the goal state g as an adjustable variable and ensure that this final state is always reached.

The scene's visual representation is used to augment the DMP motions and form the basis of the repelling field. The scene description needs to be in 3D, work at a fine scale to maintain geometric details, and represent the scene sparsely to reduce the number of calculations required per time step. The Early Cognitive Vision system of Pugeault et al. [19,20] (see Fig. 1) fulfills these requirements by extracting edge features from the observed scene. The system subsequently localizes and orientates these edges in 3D space [21], with the resulting features known as early cognitive vision descriptors (ECVD) [19]. By using a large number of small ECVDs, any arbitrary object/scene can be represented.

The methods for generating the DMP and ECVD based potential fields are detailed in Section 2. In Section 3, the system is tested on a real robot and shown to be capable of avoiding obstacles and adapting the fingers to the local geometry of the object for improved grasps using only stereo vision.

2 Methods for Reactive Grasping

The methods proposed in this section were inspired by human movements. Human grasping movements can be modeled as two linked components, transportation and preshaping, synchronized by a shared timer or canonical system [22,23]. Transportation refers to the actions of the arm in moving the hand, while the preshaping controls the opening and subsequent closing of the fingers [24].

Humans perform the reaching/transportation component in a task-specific combination of retina and hand coordinates [25], which allows for easier specification of object trajectories in a manipulation task than joint coordinates would and also results in a reduction in dimensionality.

Similar to the transportation component, the main purpose of the finger posture component is to preshape the hand by extending the fingers sufficiently for them to pass around the object upon approach, and then close on the object simultaneously for a good grasp [22,24]. Over-extending the fingers is undesirable as it makes collisions with the environment more likely and is usually restricted to situations where the shape of the object is uncertain [22,26].

The DMP and ECVD based potential field implementations are described in Sections 2.1 and 2.2. Section 2.3 proposes methods that improves the interpolation of grasping movements to new grasp locations.

2.1 Regular Dynamical Motor Primitives for Grasping

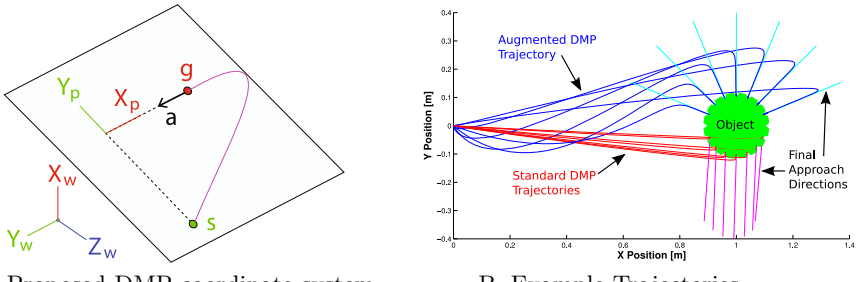
The first step towards specifying the grasping movements is to define an attractor field as a DMP that encodes the desired movements given no obstacles. The principal features that need to be defined for these DMPs are the goal positions, and the generic shape of the trajectories to reach the goal.

Determining the goal posture of the hand using the ECVDs has been investigated in a previous paper [27]. Possible grasp locations were hypothesized from

the geometry and color features of the ECVDs, and subsequently used to create a kernel density estimate of suitable grasps. It was then refined by evaluating grasps on the real system. However, this grasp synthesizer only gives the desired location and orientation of the hand and not the exact finger locations.

Using the ECVDs, the goal position of each finger is determined by first estimating a local contact plane for the object in the finger coordinate system shown in Fig. 1. If the region to be grasped is not planar, it can still be linearly approximated as such for each finger to give good results. To ensure the approximation is accurate in the proximity of the finger, the influence of the i^{th} ECVD is weighted by $w_i = \exp(-\sigma_x^{-2} v_{ix}^2 - \sigma_y^{-2} v_{iy}^2 - \sigma_z^{-2} v_{iz}^2)$, where σ_x, σ_y, and σ_z are length scale constants that reflect the finger's length and width, and $\mathbf{v}_i$ is the position of the ECVD in the finger reference frame. The hand orientation was chosen such that the Z direction of the finger should be approximately parallel to the contact plane, which reduces the problem to describing the plane as a line in the 2D X-Y space. The X-Y gradient of the plane is approximated by $\phi = (\sum_{i=1}^{N} w_i)^{-1} \sum_{i=1}^{N} w_i \arctan(e_{iy}/e_{ix})$, where N is the number of vision descriptors, and $\mathbf{e}_i$ is the direction of the i^{th} edge. The desired Y position of the fingertip is then given by $\tilde{p}_y = (\sum_{i=1}^{N} w_i)^{-1} \sum_{i=1}^{N} (w_i v_{iy} - \tan(\phi) w_i v_{ix})$, which can be converted to joint angles using the inverse kinematics of the hand.

Many of the beneficial traits of human movements, including smooth motions and small overshoots for obstacle avoidance [28, 24, 23], can be transferred to DMPs through imitation learning. To demonstrate grasping motions, we used a VICON motion tracking system to record the movements of a human test subject during a grasping task. It is not necessary for the object used for the demonstration to match that grasped by the robot later. VICON markers were



A. Proposed DMP coordinate system B. Example Trajectories

Fig. 2. A) The above diagram shows the the coordinate systems for the transportation DMPs. The axes X_w-Y_w-Z_w are the world coordinate system, while X_p-Y_p-Z_p is the coordinate system in which the DMP is specified. The trajectory of the DMP is shown by the **pink** line, starting at the **green point**, and ending at the **red point**. Axis X_p is parallel to the approach direction of the hand (the **black arrow a**). Axis Y_p is perpendicular to X_p, and pointing from the start **s** towards the goal **g**.
B) The plot shows reaching trajectories, wherein the x and y values are governed by two DMPs sharing a canonical system. The standard DMPs and the augmented DMPs described in Section 2.3 are presented along with their respective final approach directions.

only required on the back of the hand and finger tips. As the reaching trajectories are encoded in task space rather than joint space, the correspondence problem of the arm was not an issue for the imitation learning step. Details for imitation learning of DMPs using locally weighted regression can be found in [18].

As DMPs are provably stable [17], they are safe to execute on a robot and also ensure that the final arm and finger postures will also always be achieved when physically possible. The repelling field must maintain this stability.

2.2 Adapting the Motor Primitives with Vision Descriptors

Having specified the basic grasping movements, a repelling field refines the motions in order to include obstacle avoidance for the transportation and ensure that the finger tips do not collide with the object during the hand's approach.

The repelling field is based on ECVDs, which can be understood as small line segments of an object's edges localized in 3D (see Fig. 1). The repelling potential fields for ECVDs are characterized by two main features; i.e., the repelling forces of multiple ECVDs describing a single line do not superimpose, and the field should not stop DMPs from reaching their ultimate goals. The system therefore uses a Nadaraya-Watson model [29] of the form

$$u_a = -s(x)\frac{\sum_{i=1}^{N} r_i c_{ai}}{\sum_{j=1}^{N} r_j},$$

to generate a suitable repelling field, where r_i is a weight assigned to the i^{th} ECVD, s is the strength of the overall field, x is the state of the DMPs' canonical system, and c_{ai} is the repelling force for a single descriptor. Subscript a specifies if the detractor field is for the finger motions "f" or the reaching movements "h".

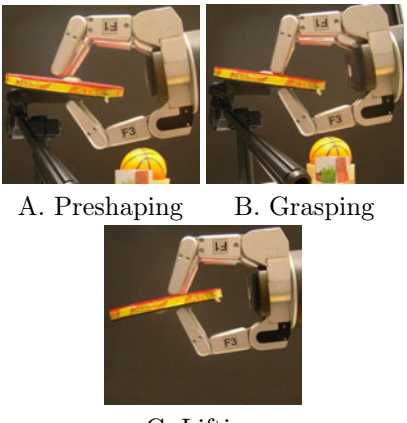


Fig. 3. The three main phases of a basic grasp are demonstrated. The preshaping of the hand (**A**) tries to pose the fingers to match the object's geometry. The grasping (**B**) then closes the three fingers at the same rate until they secure the object. Finally (**C**) the object is lifted. The objects on the bottom **A** and **B** are clutter that had to be avoided.

The weight of an ECVD for collision avoidance is given by $r_i = \exp(-(\mathbf{v}_i - \mathbf{p})^{\mathrm{T}}\mathbf{h}(\mathbf{v}_i - \mathbf{p}))$, where $\mathbf{v}_i$ is the position of the i^{th} ECVD in the local coordinate system, $\mathbf{h}$ is a vector of width parameters, and $\mathbf{p}$ is the finger tip position, as shown in Fig. 1. A suitable set of width parameters are $\mathbf{h} = 2[w,\ l,\ l]^{\mathrm{T}}$, where w and l are the width and length of the finger respectively.

The reaching and finger movements react differently to edges and employ different types of basis functions c_{fi} and c_{hi} for their potential fields. For the fingers,

the individual potential fields are logistic sigmoid functions about the edge of each ECVD of the form $\rho(1+\exp(d_i\sigma_c^{-2}))^{-1}$, where $d_i = \left\|(\mathbf{p}-\mathbf{v}_i) - \mathbf{e}_i(\mathbf{p}-\mathbf{v}_i)^{\mathrm{T}}\mathbf{e}_i\right\|$ is the distance from the finger to the edge, $\rho \geq 0$ is a scaling parameter, and $\sigma_c \geq 0$ is a length parameter. Differentiating the potential field results in a force term of $c_{fi} = \rho \exp\left(d_i\sigma_c^{-2}\right)\left(1+\exp\left(d_i\sigma_c^{-2}\right)\right)^{-2}$. As the logistic sigmoid is monotonically increasing, the repelling always forces the fingers open further to move their tips around the ECVDs and thus ensure that they always approach the object from the outside. Similarly, a symmetrical potential function can be employed to force the hand closed when near ECVDs pertaining to obstacles.

The reaching motion uses basis functions of the form $\varrho\exp(-0.5\mathbf{d}_i^{\mathrm{T}}\mathbf{d}_i\sigma_d^{-2})$, where $\mathbf{d}_i = (\mathbf{q}-\mathbf{v}_i) - \mathbf{e}_i(\mathbf{q}-\mathbf{v}_i)^{\mathrm{T}}\mathbf{e}_i$ is the distance from the end effector position, $\mathbf{q}$, to the edge, and $\varrho \geq 0$ and $\sigma_d \geq 0$ are scale and length parameters respectively. Differentiating the potential with respect to $\mathbf{d}_i$ gives a force term in the Y direction of $c_{hi} = \varrho(\mathbf{d}_i.\mathbf{Y})\sigma_d^{-2}\exp(-0.5\mathbf{d}_i^{\mathrm{T}}\mathbf{d}_i\sigma_d^{-2})$, which can be interpreted as a radial force from the edge with an exponentially decaying magnitude.

To synchronize the repelling field with the DMPs and ensure the repelling strength is zero at the end of a motion, the strength s is coupled to the canonical system of the DMPs. Hence, $s(x) = (\sum_{j=1}^{M}\psi_j(x))^{-1}\sum_{i=1}^{M}\psi_i(x)w_i x$, where x is the value of the canonical system, ψ are the DMP basis functions, and w specify the varying strength of the field during the trajectory. To reflect the human tendency towards more precise movements during the last 30% of a motion [28], the strength function was set to give the highest strengths during the first 70% of the motion for the reaching trajectories, and the last 30% for the finger movements.

The repelling fields of both the grasping and reaching components have now been defined, and can be superimposed into the DMP framework as

$$\ddot{y} = \left(\alpha_z(\beta_z\tau^{-2}(g-y) - \tau^{-1}\dot{y}) + a\tau^{-2}f(x)\right) - \tau^{-2}u_a,$$

which then represents the complete ECVD and DMP based potential field.

2.3 Generalizing Dynamical Motor Primitives for Grasping

Having defined the potential field for a single grasping motion, we must generalize the movements to new target grasps. By interpolating the trajectories in a task-specific manner, the number of example trajectories required from the demonstrator for imitation learning can be greatly decreased. While the goal states of DMPs can be set arbitrarily, the approach direction to the grasp cannot be easily defined and the amplitude of the trajectory can be unnecessarily sensitive to changes in the start position y_0 and the goal position g.

The correct approach direction can be maintained by using a task-specific coordinate system. We propose the X_p-Y_p-Z_p coordinate system shown in Fig. 2, which dedicates one axis $\mathbf{x}_p$ specifically to the approach direction. The majority of the unobstructed reaching motion will lie in a plane defined by the starting point, the goal location, and the final approach direction, which we use to define our second axis $\mathbf{y}_p$. The final axis $\mathbf{z}_p$ is given by $\mathbf{z}_p = \mathbf{x}_p \times \mathbf{y}_p$.

The second problem relates to the sensitivity of scaling motions with ranges greater than $\|y_0 - g\|$, which grasping motions require to move around the outside of objects. The system can be desensitized to variations in $y_0 - g$ by employing the amplitude term $a = \|\eta(g - y_0) + (1 - \eta)(g_T - y_{0T})\|$ instead of the standard $a = (g - y_0)$ [16], where g_T and y_{0T} are the goal and start positions of the training data respectively, and $\eta \in [0, 1]$ is a weighting hyperparameter that controls how conservative the generalization is. By taking the absolute value of the amplitude, the approach direction is specified solely by the choice of X_p-Y_p-Z_p coordinate system and not the amplitude term. This amplitude term is a generalization of the amplitude proposed by Park et al. [12], which corresponds to the special case of $\eta = 0$. Example interpolations of a transportation trajectory can be seen in Fig. 2.

Fig. 4. Examples of different approach directions are presented, all based off of a single human demonstration

3 Grasping Experiments

The methods described in Section 2 were implemented and evaluated on a real robot platform consisting of a Videre stereo camera, a Barrett hand, and a 7-degrees-of-freedom Mitsubishi PA10 arm, as shown in Fig. 1.

3.1 Grasping Experiment Procedure

To test the system's obstacle avoidance ability, the robot was given the task of grasping an object without hitting surrounding clutter (see Fig. 1). Each trial begins with an estimate of the pose of the object relative to the robot [30] and setting the desired grasp location. The model's ECVD are then projected into the scene, and the robot attempts to perform the grasp and lift the object off the table.

If the hand collides with an obstacle or knocks the object down during its approach, the trial is marked as a failure. Grasp locations on the object were predefined, and all successful trials had to lift the object from its stand (see Fig. 3). After each grasp attempt, the hand reverses along the same approach direction, but with a static preshaping of the hand in order to determine if collisions would have occurred if the proposed controller had not been used. The experiment consisted of 50 trials and were varied to include different approach directions and locations around the object.

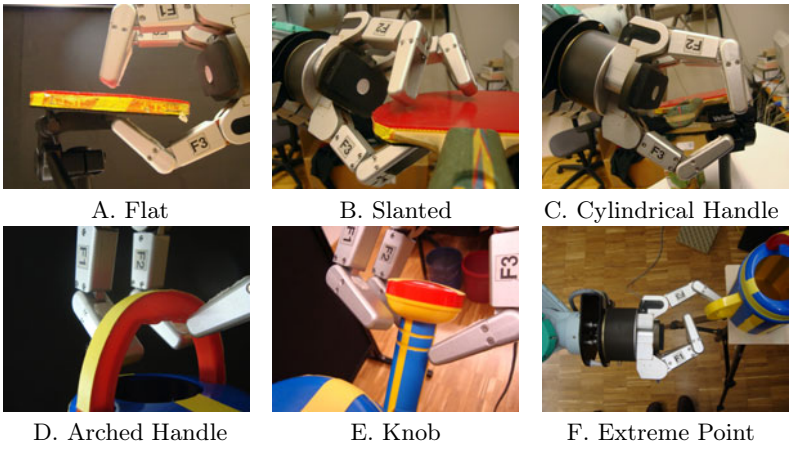


Fig. 5. Pictures **A** and **B** show the system adjusting to different plane angles. Images **C** and **D** demonstrate the preshaping for different types of handles. Picture **E** shows the preshaping for a circular disc structure, such as a door knob, and manages to get its fingers closely behind the object. Picture **F** shows a preshape where the object was too far away to be reached by two of the fingers, but still hooks the object with 1 finger.

Additional trials were performed on another object to further explore how the system's preshaping ability adapts to different object geometries. The clutter was removed in these trials to demonstrate the range of grasps that a single human demonstration can easily be generalized to.

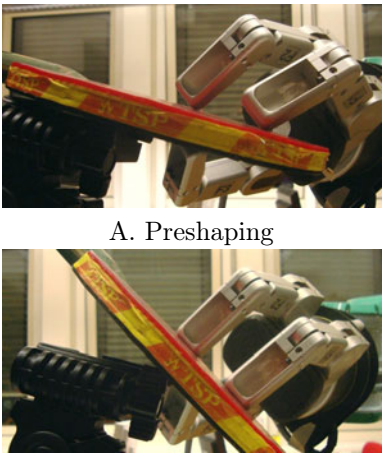
3.2 Experimental Results

The repelling field and preshaping of the hand allowed the system to handle the cluttered environment that the object had been placed in, which was not a trivial task. The hand came into contact with the clutter for an estimated 8% of the grasp attempts, but never more than a glancing contact. When the proposed controller was deactivated and a static preshape was used, the hand collided with one or more pieces of clutter in 86% of the trials. Thus, the proposed sensor-based controller led to a factor of ten decrease in the number of contacts with the clutter. The few instances when the hand did collide with the obstacles were the result of obstacles being partially occluded, and thus not fully represented by the ECVDs. This problem represents the main restriction of the current method, which can be overcome by simply using multiple views to accumulate the ECVD representation of the scene, as described in [19,20]. The repelling fields of the fingers ensured that the hand always opened sufficiently to accept the object without colliding with it.

Using only a single human demonstration, the robot could perform a wide range of reaching movements with varied approach directions, as demonstrated in Fig. 4. Requiring fewer demonstrations hastens the imitation learning process, while still allowing the robot to perform smooth and natural reaching motions.

The incorporation of ECVDs allowed the fingers to adapt to a wide variety of different object geometries, as shown in Fig. 5, and place the finger tips very close to the object before applying the grasp. This close proximity to the object restricts how much the object can move during the final grasping phase, as the fingers make contact with the object at approximately the same time, and leads to grasps being applied in a more controlled manner. An example of a controlled grasp is shown in Fig. 6, which would not be possible without the proposed preshaping, as the finger on the right would have made first contact with the paddle and simply knocked it down.

The results ultimately show that our hypothesis was correct and the proposed methods represent a suitable basis for avoiding obstacles without relying on a complicated path planner, and using only stereo vision information.



A. Preshaping

B. Grasping

Fig. 6. The preshaping allows for more controlled grasping. (A) The preshape has matched the local geometry of the object. When grasping, the two fingers on the left immediately pinch the paddle, while the finger on the right turns the paddle about the pinched point. (B) The grasping ends when the paddle has become aligned with all three finger tips.

4 Conclusions

The proposed methods augment dynamical system motor primitives to incorporate Early Cognitive Vision descriptors by using potential field methods, and represent important tools that a robot needs to execute preshaped grasps of an object in a cluttered environment using stereo vision. The techniques allow for preshaping the fingers to match the geometry of the object and shaping the trajectory of the hand around objects. The controller was tested on a real robot, and was not only successful at performing the task, but also requires very few demonstrations for imitation learning, improves obstacle avoidance, and allows for more controlled grasps to be performed.

References

1. Bard, C., Troccaz, J., Vercelli, G.: Shape analysis and hand preshaping for grasping. In: IROS Proceedings (1991)
2. Iberall, T.: Grasp planning for human prehension. In: ICAI Proceedings (1987)

3. Morales, A., Asfour, T., Azad, P., Knoop, S., Dillmann, R.: Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands. In: IROS, pp. 5663–5668 (2006)
4. Xue, Z., Kasper, A., Zoellner, J.M., Dillmann, R.: An automatic grasp planning system for service robots. In: Proceedings of International Conference on Advanced Robotics, ICAR (2009)
5. Bertram, D., Kuffner, J., Dillmann, R., Asfour, T.: An integrated approach to inverse kinematics and path planning for redundant manipulators. In: ICRA, pp. 1874–1879 (2006)
6. Spong, M.W., Hutchinson, S., Vidyasagar, M.: Robot Modeling and Control. In: WSE (2005)
7. Khatib, M.: Sensor-based motion control for mobile robots (1996)
8. Sabe, K., Fukuchi, M., Gutmann, J.-S., Ohashi, T., Kawamoto, K., Yoshigahara, T.: Obstacle avoidance and path planning for humanoid robots using stereo vision. In: ICRA, pp. 592–597 (2004)
9. And, S.L.: Visual sonar: Fast obstacle avoidance using monocular vision (2003)
10. Tegin, J., Ekvall, S., Kragic, D., Wikander, J., Iliev, B.: Demonstration based learning and control for automatic grasping. In: Demonstration based Learning and Control for Automatic Grasping (2008)
11. Miller, A.T., Knoop, S., Christensen, H.I., Allen, P.K.: Automatic grasp planning using shape primitives. In: Proceedings of the International Conference on Robotics and Automation, ICRA (2003)
12. Park, D.-H., Hoffmann, H., Pastor, P., Schaal, S.: Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In: IEEE International Conference on Humanoid Robots(HUMANOIDS) (2008)
13. Bley, F., Schmirgel, V., Kraiss, K.-F.: Mobile manipulation based on generic object knowledge. In: Proceedings of Robot and Human Interactive Communication, ROMAN (2006)
14. Hsiao, K., Nangeroni, P., Huber, M., Saxena, A., Ng, A.: Reactive grasping using optical proximity sensors. In: ICRA Proceedings (2009)
15. Steffen, J., Haschke, R., Ritter, H.: Experience-based and tactile-driven dynamic grasp control. In: IRS Proceedings (2007)
16. Ijspeert, A.J., Nakanishi, J., Schaal, S.: Learning attractor landscapes for learning motor primitives. In: NIPS (2003)
17. Schaal, S., Peters, J., Nakanishi, J., Ijspeert, A.: Learning movement primitives. In: ISRR Proceedings (2003)
18. Ijspeert, A.J., Nakanishi, J., Schaal, S.: Movement imitation with nonlinear dynamical systems in humanoid robots. In: ICRA (2002)
19. Pugeault, N.: Early Cognitive Vision: Feedback Mechanisms for the Disambiguation of Early Visual Representation. Vdm Verlag Dr. Mueller (2008)
20. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
21. Krueger, N., Lappe, M., Woergoetter, F.: Biologically motivated multimodal processing of visual primitives. The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour (2004)
22. Chieffi, S., Gentilucci, M.: Coordination between the transport and the grasp components during prehension movements (1993)
23. Oztop, E., Kawato, M.: Models for the control of grasping. In: Sensorimotor Control of Grasping: Physiology and Pathophysiology. Cambridge University Press, Cambridge (2009)

24. Jeannerod, M.: Grasping Objects: The Hand as a Pattern Recognition Device. In: Perspectives of Motor Behaviour and Its Neural Basis (1997)
25. Graziano, M.S.: Progress in understanding spatial coordinate systems in the primate brain. In: Neuron (2006)
26. Oztop, E., Bradley, N.S., Arbib, M.A.: Infant grasp learning: a computational model (2004)
27. Detry, R., Kroemer, O., Popovic, M., Touati, Y., Baseski, E., Krueger, N., Peters, J., Piater, J.: Object-specific grasp affordance densities. In: ICDL (2009)
28. Jeannerod, M.: The study of hand movements during grasping. A historical perspective. In: Sensorimotor Control of Grasping: Physiology and Pathophysiology. Cambridge University Press, Cambridge (2009)
29. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
30. Detry, R., Pugeault, N., Piater, J.: Probabilistic pose recovery using learned hierarchical object models. In: International Cognitive Vision Workshop (2008)

Multimodal Predictive Control in Crickets

Mark Payne[1], Berthold Hedwig[2], and Barbara Webb[1]

[1] IPAB, School of Informatics, University of Edinburgh, U.K.
[2] Department of Zoology, University of Cambridge, U.K.
mark.payne@dunelm.org.uk, bh202@cam.ac.uk, bwebb@inf.ed.ac.uk

Abstract. A combination of behavioural testing and robotic modelling was used to investigate the interaction between sound localisation (phonotaxis) and optomotor following in crickets. Three hypotheses describing simple interactions — summation, gain modulation and chaining — were eliminated, leaving efference copy as the most likely mechanism. A speculative but plausible model for predicting re-afference was implemented and evaluated on a robot.

1 Introduction

Nervous systems process afferent sensory signals to produce efferent motor output. Afferent signals can be further subdivided into those that originate from external causes (exafferent signals) and those arising due to self-generated causes (reafferent signals). For example, in an insect's natural habitat, exafferent optical flow might be caused by the wind blowing it around as it flies. In an experimental situation, a visual pattern rotated around an insect provides an exafferent signal that usually evokes the optomotor response, i.e., the insect rotates in the same direction, to stabilise itself with respect to its surroundings. But optical flow is also produced whenever the insect moves itself, e.g., when a female cricket steers towards the sound of a chirping male (Fig. 1, top-left). The motion sensitive visual neurons involved in the optomotor response will be activated by this reafferent signal, but responding in the same way as to exafferent input could be maladaptive.

von Holst was first to note explicitly that if an animal were to respond to reafferent optical flow with an optomotor response, this would tend to counteract all self-initiated turns [1]. He proposed that the solution is for the animal to predict (and thus eliminate) the reafferent signal based on copies of its efferent (motor) signals (Fig. 1, top-right). However, using efference copy is not trivial because the sensory consequences of an action are generally quite different in magnitude, duration or shape from the motor command, hence a predictive (forward) model is required to calculate the transformation, as shown in the diagram.

Before assuming that efference copy is necessary, and hence examining how it might be implemented in insect brains, it is important to exclude alternative simpler possibilities. Efference copy is an example of a more general class of signals termed corollary discharges. These are branching pathways from motor outputs that may influence other sensory systems by inhibition, facilitation, or

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 167–177, 2010.

any other kind of modulation (Fig. 1, bottom-right). A special instance of this arrangement would be where the corollary discharge modulates the optomotor gain to zero (outright inhibition), i.e., the animal simply ignores all (rotational) optical flow when generating its own (turning) movements. A further possibility is the "chained subsystem" hypothesis, in which auditory signals control turning via the optomotor pathway, by altering the optic flow signal in such a way that the resulting optomotor response will include a turn in the required direction (Fig. 1, bottom-left). Finally, it is also possible that simple summation (Fig. 1, top-left) might suffice, despite von Holst's argument. For example, intentional turns could be too fast or too slow to activate the optomotor response; this has been proposed to hold for flies making rapid saccades [2].

The four possibilities outlined in Fig. 1 have been previously investigated for crickets using hardware models of auditory and visual sensors on a robot [3] [4], inspired by mathematical and analogue electronic models in Collett's investigation of behavioural co-ordination in hoverflies [5]. However a major limitation in these earlier studies was a lack of behavioural data characterising the cricket optomotor response. Here we combine behavioural experiments on crickets with robotic models to enable stronger conclusions to be drawn regarding the possible mechanisms of multimodal control.

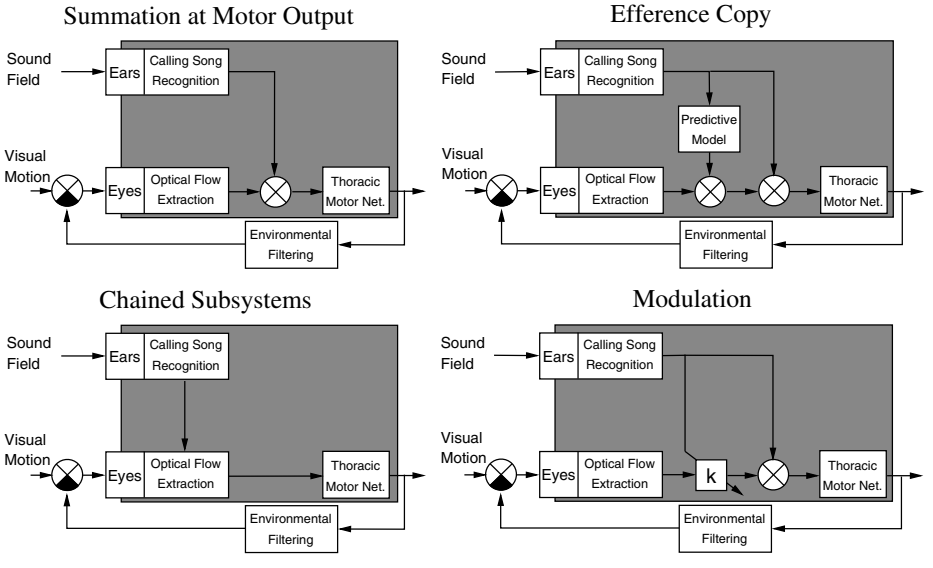


Fig. 1. Combining auditory and visual signals. In summation, the two systems act in parallel and do not distinguish exafferent and reafferent inputs. In efference copy, a copy of the motor command from the phonotaxis system is modified by a predictive model and interacts with the afferent visual signal, cancelling out the portion due to reafference. In modulation, the auditory pathway branches as for efference copy, but interacts with the optomotor pathway simply by modifying its gain. In chained subsystems the auditory pathway is arranged in series with the optomotor pathway.

2 Methods

Cricket Experiments: Behavioural tests were performed using an open-loop trackball system, described in [6]. Female crickets (*Gryllus bimaculatus*, 7–20 days post-moult) were tethered using a pin attached to the third thoracic tergites, and were placed so that their legs turned a light air-suspended sphere (diameter 56.5 mm, weight 3 g). Movements of the ball were registered with an optical mouse chip and recorded using PC software written in Labview. The trackball was located inside a darkened anechoic box. Visual stimuli were provided by a translucent vertically-striped cylinder which could be rotated around the insects using a programmable servomotor. The pattern was illuminated by a circular fluorescent tube around the outside of the cylinder. The auditory stimuli consisted of simulated cricket songs played from a speaker that could be positioned at different azimuth angles, 57 cm from the animal.

Robot Experiments: A Khepera II (K-Team, Switzerland) was used as the mobile base. A custom-made auditory sensor based on [7] and a camera (K-Team K2D), fitted with a hemispherical mirror that gave it a 360-degree view of the horizon, were added to model the crickets' ears and eyes. The control programme was executed on a PC communicating with the robot over a serial cable. The robot was tested in a sound-proofed box of dimensions $122 \times 175 \times 122$ cm using the same simulated songs as used for the cricket. Visual stimuli consisted of black stripes fixed to the white internal walls. The robot's path was recorded using an overhead camera.

The robot's auditory system uses a neural network as described in [4], which replicates both the directionality and pattern selectivity of female cricket responses to calling songs. The output of the network is two spike trains, one for each side of the body, and the relative activity can be used to determine the robot's direction of rotation. The optomotor system is based on two previous models of optomotor behaviour in flies [8] [9]. 360 intensity values are extracted around the horizon, and the transient nature of the response of lamina monopolar cells is simulated by passing these through a high-pass filter with time constant $\tau = 58$ ms. Local motion detection is accomplished, following the Hassenstein-Reichardt model [10], by correlating the output of each unit with the signal from the adjacent unit delayed via a low-pass filter with $\tau = 93$ ms. The local motion signals are summed to provide the optomotor signal which can drive rotation of the robot in the corresponding direction.

The interaction of the two sensory systems is described in the appropriate sections below. For full implementation details of the robot's controller see [11].

3 Results

Characterising the optomotor response rules out chained subsystems: Data to characterise the optomotor response of the cricket were collected using the open loop trackball and visual stimulus apparatus described in section 2.

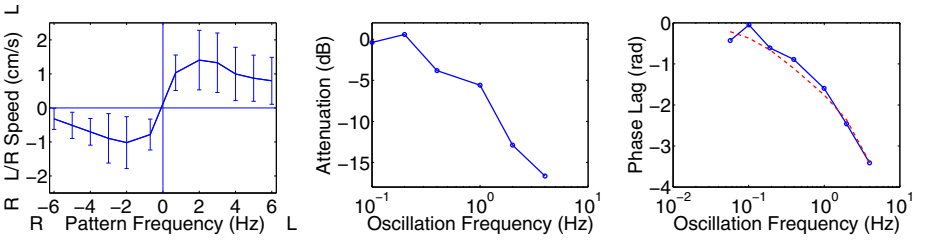


Fig. 2. Left: Characteristic curve for unidirectional pattern motion. Centre: Amplitude / frequency plot showing ratio between observed steering amplitude and reference steering amplitude (assuming no filtering) for frequencies from 0.1 to 0.4 Hz. Right: Phase / frequency plot. Dotted line shows phase for a first order low-pass filter with cut-off frequency 0.3Hz and lag 76 ms.

Figure 2 (left) shows the asymptotic turning speed of the cricket for unidirectional visual rotation at pattern frequencies up to 6 Hz. This curve has the log-normal shape characteristic of motion detection based on the delay-and-correlate principle [10]. The maximum response was produced for a pattern frequency of 2 Hz.

Stimulation of the crickets with an oscillating grating resulted in an oscillating steering pattern. Fourier transforms of the steering traces were computed, and the amplitudes and phase lags were read off for input frequencies from 0.05 to 4 Hz. These were used to check for temporal filtering effects. In the absence of temporal filtering the response to an oscillating grating would be predicted by applying the characteristic function of Fig. 2 (left) to the sine wave describing the stimulus motion. In fact, the amplitudes read off from the transformed cricket data became progressively smaller relative to this reference value as the input frequency increased. Figure 2 (centre) shows this relationship in the standard form of the Bode amplitude plot. Although it is noisy, the amplitude plot is comparable to that which would be expected for a first-order low pass filter with a cut-off frequency between 0.2 and 0.4 Hz. This interpretation is supported by an examination of the phase-lag of the cricket's movements (Fig. 2, right) if a short fixed lag is introduced. This lag is assumed to represent the combination of synaptic and axonal delays.

The chained subsystems hypothesis (Fig. 1, bottom-left), by requiring that phonotaxis responses pass through the optomotor pathway, would predict that phonotaxis behaviour should show comparable amplitude and phase characteristics. We can test this by looking the phonotactic steering responses of the cricket to certain song patterns, and comparing this to the response predicted by applying low-pass filters with different frequency cut-offs to the same song patterns. The two patterns used are a typical calling song of pulses of around 4.7 kHz with a pulse repetition interval 42 ms grouped into six-pulse "chirps", and a "split" song in which alternate sound pulses are presented from opposite sides of the body. Looking at the ratio of the steering amplitudes to these two stimuli gives an indication of the likely frequency cut-off for phonotactic steering. Fig. 3 shows that this ratio for the cricket ($1.7/1.3 = 1.3$) is much higher than that seen for a

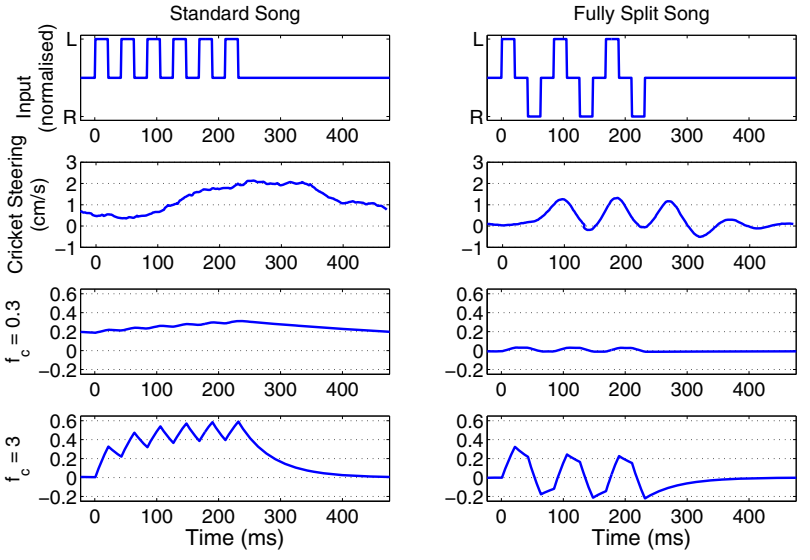


Fig. 3. Standard calling song and split song patterns (*top row*), averaged steering pattern of female crickets to each song (*second row*, redrawn from [6]) and effect of low-pass filtering of song patterns with cutoff frequencies of 0.3 Hz and 3 Hz (*lower two rows*). The crickets' phonotaxis response is matched by the 3 Hz cut-off, not the 0.3 Hz that would be predicted if the chained subsystems hypothesis were to hold.

filter with 0.3 Hz cut-off ($0.126/0.041 = 3.1$) and is better matched with cut-off frequency of 3 Hz. This is much faster than the optomotor system, so it seems unlikely that the chained subsystems hypothesis is correct.

Cricket behaviour with both stimuli rules out modulation: The modulation model (Figure 1, bottom-right) predicts that the influence of a moving grating (as characterised in Fig. 2, left) will be altered, and possibly inhibited entirely, in the presence of an active response of the cricket to calling song. As this response depends on the direction of the song, it might be expected the modulation should be more pronounced for a more laterally located song. We tested for this, following [12], by measuring the turning response of the cricket to sound broadcast at 0°, ±45° and ±90° under three different visual conditions: with a stationary black and white striped grating of period sixty degrees, or rotating the grating to create leftward optical flow or rightward optical flow at a pattern frequency of 1 Hz.

Figure 4 (left) shows that the characteristic curve for open-loop turning to sound within the stationary visual environment is shifted up or down by optical flow. In order to test the possibility that the size of the shift alters depending on the direction of the sound a one-way repeated measures ANOVA was conducted on the difference between the upper and lower curves. No significant effect of speaker direction was found ($F=2.1$, $p=0.1038$). The average size of shift is

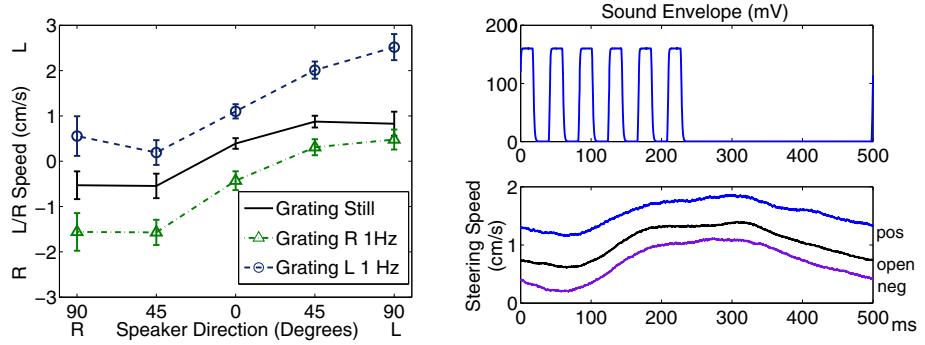


Fig. 4. Walking speeds for nine crickets stimulated with calling song and grating combined (*left*), means ± standard error. Mean L–R speed over one chirp period during positive, negative and open-loop visual stimulation (*right*), means from 13 crickets.

0.91 cm/s (SD across crickets = 0.33). This is not significantly smaller than the shift (0.98 cm/s) that would be predicted by interpolation from the characteristic curve of Figure 2 (unpaired t-test (2-tailed), t = -0.5981, df = 8, p = 0.57). Phonotaxis does not appear to modulate the optomotor response.

However, the auditory stimuli used, standard calling song (Fig. 3, upper left), mixes pulses of sound with periods of silence. Perhaps the modulation only occurs during pulses, so looking at the averaged turning response is misleading. We thus carried out a further experiment in which the visual stimulus was moved only when a phonotactic turning response was detected. The movement either simulates the visual flow that a cricket would experience in normal closed loop behaviour so that a left turn results in rightward visual motion and vice-versa (negative feedback), or is inverted (positive feedback). Figure 4 shows the average rotation of crickets during a 500 ms chirp period. It is clear that different optomotor stimuli experienced during rotation to sound produce different responses, with the response for negative feedback significantly weaker than for no feedback (Wilcoxon signed ranks test, T=0, $p<0.0003$) and the response for positive feedback significantly stronger(T=1, $p<0.0005$). Turning to sound does not suppress the optomotor response.

Robot behaviour with both stimuli rules out summation: In the above experiments, the shift in the turning responses to combined sound and optical flow could be characterised as addition, i.e., it is quite well predicted by simply summing the responses seen to each stimulus on its own. However, it is also consistent with efference copy: under this hypothesis, the open-loop behaviour would already represent the animal responding to the absence of the expected re-afference; turning should be reduced if the correct negative feedback is experienced, and increased if the incorrect postive feedback is experienced. To distinguish between these possibilities we tested the summation hypothesis on the robot in closed loop control, to determine whether von Holst's argument — an optomotor response running in parallel with phonotaxis would stop the

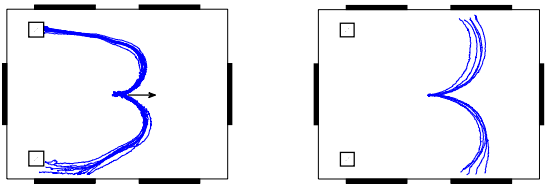


Fig. 5. Sound localisation behaviour of the robot for 10 trials for each speaker (squares). Arena (97 × 128 cm) shown in plan view, arrow shows initial robot position and orientation. Auditory system alone (*left*) and with optomotor system added (*right*).

animal from making its intended turns — applies in practice if we tune the dynamics of the two subsystems to match the observed response of the cricket to the individual stimuli. The tuning (frequency response and gain) of the robot's auditory and visual systems were tuned using the data above and also using observed turning and walking speeds of crickets moving freely in an arena.

Fig. 5 (left) shows that when the phonotaxis system alone was active the robot initially made a tight turn and then proceeded towards the active speaker. When the optomotor system was also active (Fig. 5 (right)) the turns were much broader in radius, and the robot encountered the edge of the arena before being able to turn and face the speaker. It appears the tendency of the optomotor system to stabilise the robot's course does interfere substantially with the attempt to turn towards sound.

4 A Robot Model with Efference Copy

So far we have argued that a close characterisation of the cricket's optomotor response is inconsistent with three hypotheses about how it may be combined with phonotaxis. First, the phonotaxis response seems to be filtered at a higher cut-off frequency than that found for the optomotor response, suggesting it is unlikely that the former is controlled via the latter as a chained subsystem. Second, adding optomotor stimuli during phonotaxis leads to a consistent shift in angular velocity for all sound directions, both during and between actual phonotactic turns, which makes it unlikely that the phonotaxis system is modulating (certainly it is not simply suppressing) the optomotor response. Third, if the two systems are simply run in parallel with the outputs added, the optomotor response should dramatically interfere with turning towards sound, yet this is clearly not a problem for the cricket under normal conditions.

Consequently we suggest that the cricket uses efference copy to cancel out expected optic flow, but still reacts to any difference in the optic flow from that expected. We propose here a speculative model for efference copy circuitry in the insect brain, shown schematically in Fig. 6. Added to the existing phonotaxis and optomotor systems is a predictive system. This receives as input a copy of the auditory system's output, via two delay lines set to 170 ms to compensate

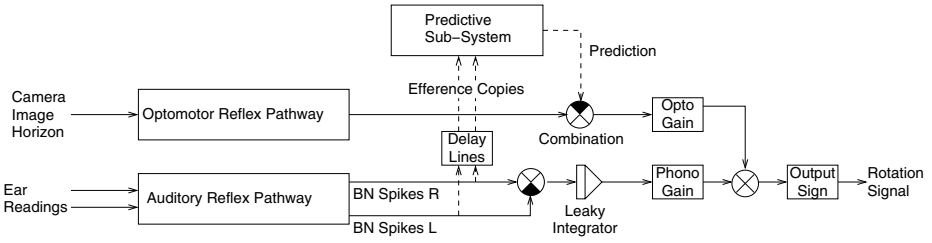


Fig. 6. Architecture of the neural controller showing signals of the reflex pathways (*solid lines*) and signals of the predictive sub-system (*dotted lines*)

for the delay inherent in the robot's visual feedback. The output is a continuous prediction of the output of the optomotor system, and this quantity is subtracted from the optomotor reflex pathway, in order to allow the robot to respond only to the exafferent component of the optical flow signal.

Unlike the auditory and optomotor pathways this predictive component cannot be based upon known neural circuitry from the cricket or other insects. Instead a liquid state machine (LSM) neural network is used [13]. It is not claimed that any particular part of the cricket brain is equivalent to an LSM; this type of model is preferred to alternatives only because it provides the necessary functionality using principles that exist in insect and other brains. The most important of these is the idea that memory and sustained activity can be produced through reverberating activity in recurrent networks. These effects are essential to the cross-model prediction under consideration, since filtering in both the motor and optical systems means that sequences of short, discrete sound pulses may result in continuous movement with sensory consequences that last beyond the duration of the input.

The particular implementation of the LSM uses a pool of 135 integrate-and-fire neurons, randomly interconnected with a mixture of excitatory and inhibitory synapses. Each input spike train was connected by excitatory synapses to randomly chosen 30 % of the neurons of this recurrent pool. Pulses of sound picked up by the auditory system result in reverberating activity which decays away over time. The statistics of the connectivity within the recurrent pool came from [13], which was based on rat cortex, although these are not the only possible values that could produce a viable LSM.

The output of the LSM is calculated by a linear "readout unit", which receives one input from all the recurrent neurons and computes a weighted sum. To find a set of weights for the readout unit the output of the predictive model was disconnected from the optomotor pathway, and the auditory system was used to perform phonotaxis while the outputs of both the recurrent neurons and the optomotor system were recorded. Having collected this training set, linear regression could be used to calculate a least-squares fit. In principle, this learning could occur online, i.e., the weights could be continuously adapted to improve the prediction, using the error between the prediction and the observed optomotor signal.

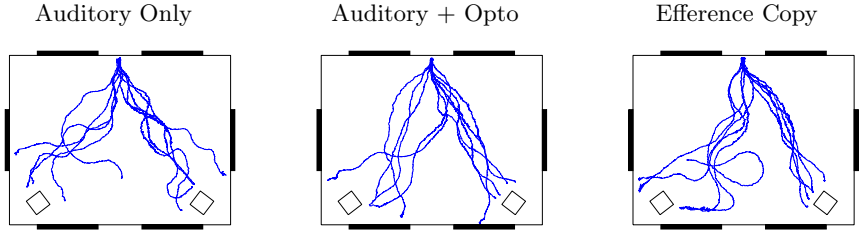


Fig. 7. Tracks of the robot turning towards a standard calling song in the arena under conditions of random motor disturbance. Phonotaxis system only (*left*), phono and opto systems summed at the output (*middle*) and with directional prediction (*right*). Black bars indicate the visual pattern, squares the speaker locations.

The robot was tested using the auditory system alone, using simple summation of auditory and optomotor output, and using the prediction mechanism just described, under conditions of random motor disturbances. This was a random rotatory bias, superimposed on the robot's motor output, and changed at random intervals with a mean interval of one second. Without the optomotor system, the robot can only correct for these disturbances by re-orienting to sound (Fig. 7, left), whereas it should be able to detect and compensate for them more smoothly by using the resulting optic flow. Under simple summation (Fig. 7, middle), it should also 'correct' for optic flow detected when turning towards sound, so the efference copy mechanism (Fig. 7, right) should produce the most successful behaviour. The behaviour of the robot in Fig. 7 shows less difference between the controllers than expected. For the phonotaxis system alone (left), the robot generally manages to reach the active speaker, although the paths are far more convoluted than seen without the added random turns (Figure 5, left). Adding the optomotor system (middle) does seem to make the paths straighter, suggesting that it can compensate for the random turns. However, unlike the previous test (Figure 5, right) the robot does not need to make a tight turn to reach the speaker, so the optomotor system is not actively interfering with phonotaxis. The results with prediction (right) are rather mixed, with particularly smooth paths to the right-hand speaker but substantial deviations in its approach to the left hand speaker. We suppose this may be due to some asymmetries in sound propagation in the two corners of the arena. As a consequence, a comparison of the directness of the paths (using an ANOVA on the D statistic defined in [4]) under the three conditions showed a non-significant difference ($F=0.82$, $p=0.4029$). It may be that a more challenging task is needed to reveal whether the efference copy mechanism is indeed the most efficacious solution.

5 Discussion

The idea of efference copy — a copy of a motor command that is transformed to act as a sensory prediction — is an elegant and general model of how animals

might deal with the sensory consequences of their own actions. It is also a potentially complex model, since the prediction task is essentially that of forward modelling. While the existence of such internal models seems increasingly well established for vertebrates [14] it seems plausible that simpler animals such as insects might employ more basic mechanisms for dealing with the problem. We have examined three possible alternatives for cricket auditory and visual control: chained subsystems, modulation/inhibition of one system by the other, and simple summation under the assumption that the subsystems do not actively interfere with each other.

Our behavioural experiments on the cricket, characterising the sensitivity and frequency response of the optomotor system, suggest that the auditory steering signal is not subjected to the same kind of filtering as the optomotor response, which makes the chained subsystems hypothesis unlikely. We also found no evidence of inhibition or modulation of the optomotor response during turns made to sound. Using this behavioural data to constrain a robot model, we found that simple summation would predict strong interference between the subsystems. Consequently we believe the cricket may indeed be predicting its own re-afferent visual input.

Where might such a predictive component be found in the cricket brain? The mushroom bodies are a structure which might merit investigation from this point of view. Although their primary inputs are apparently olfactory, the lobe-extrinsic neurons that connect with other parts of the protocerebrum are highly multimodal [15], form recurrent connections, display sustained activity, and thus might support the computation of cross-modal predictions between multiple sensory systems. Certainly the numbers of neurons and connections used in the current model are small enough not to preclude the possibility that insects might compute with similarly complex forward models.

References

1. von Holst, E.: The Behavioural Physiology of Animals and Man. Methuen, London (1973)
2. Egelhaaf, M.: Dynamic Properties of two Control Systems underlying Visually Guided Turning in House-Flies. J. Comp. Physiol. A 161, 777–783 (1987)
3. Webb, B., Harrison, R.: Integrating Sensorimotor Systems in a Robot Model of Cricket Behaviour. SPIE 4196, 113–124 (2000)
4. Webb, B., Reeve, R.: Reafferent or Redundant: Integration of Phonotaxis and Optomotor Behaviour in Crickets and Robots. Adaptive Behavior 11(3), 137–158 (2003)
5. Collett, T.: Angular Tracking and the Optomotor Response: An Analysis of Visual Reflex Interaction in a Hoverfly. J. Comp. Physiol. A 140, 145–158 (1980)
6. Hedwig, B., Poulet, J.: Mechanisms Underlying Phonotactic Steering in the Cricket Gryllus Bimaculatus Revealed with a Fast Trackball System. J. Exp. Biol. 208, 915–927 (2005)
7. Lund, H., Webb, B., Hallam, J.: A Robot Attracted to the Cricket Species Gryllus bimaculatus. In: 4th European Conference on Artificial Life, pp. 246–255. MIT Press/Bradford Books (1997)

8. Huber, S., Bülthoff, H.: Simulation and Robot Implementation of Visual Orientation Behaviour of Flies. In: From Animals to Animats, vol. 5, pp. 77–85. MIT Press, Cambridge (1998)
9. Harrison, R., Koch, C.: A Silicon Implementation of the Fly's Optomotor Control System. Neural Computation 12, 2291–2304 (2000)
10. Hassenstein, B., Reichardt, W.: Systemtheoretische Analyse der Zeit-, Reihenfolgen-, und Vorzeichenauswertung bei der Bewegungsperzeption des Rüsselkäfers Chlorophanus. Z. Naturforschung 11(b), 513–524 (1956)
11. Payne, M.: Co-ordinating Behaviours in an Insect Biorobot. Unpublished Ph.D. Thesis, University of Edinburgh, UK (2010)
12. Böhm, H., Schildberger, K., Huber, F.: Visual and Acoustic Control in the Cricket Gryllus bimaculatus. J. Exp. Biol. 159, 235–248 (1991)
13. Maass, W., Natschläger, T.: Real-Time Computation Without Stable States: A New Framework for Neural Computation Based on Perturbations. Neural Computation 14, 2531–2560 (2002)
14. Kawato, M.: Internal Models for Motor Control and Trajectory Planning. Curr. Opin. Neurobiology 9(6), 718–727 (1999)
15. Schildberger, K.: Multimodal Interneurons In The Cricket Brain - Properties of Identified Extrinsic Mushroom Body Cells. J. Comp. Physiol. A 154(1), 74–79 (1984)

Tactile Discrimination Using Template Classifiers: Towards a Model of Feature Extraction in Mammalian Vibrissal Systems

Mathew H. Evans[1], Charles W. Fox[1], Martin J. Pearson[2], and Tony J. Prescott[1]

[1] Active Touch Laboratory, Psychology Department,
Western Bank, Sheffield, S10 2TP, UK
mat.evans@shef.ac.uk
[2] Bristol Robotics Laboratory, Du Pont Building, Bristol Business Park,
Coldharbour Lane, Frenchay, Bristol, BS16 1QD, UK

Abstract. Rats and other whiskered mammals are capable of making sophisticated sensory discriminations using tactile signals from their facial whiskers (vibrissae). As part of a programme of work to develop biomimetic technologies for vibrissal sensing, including whiskered robots, we are devising algorithms for the fast extraction of object parameters from whisker deflection data. Previous work has demonstrated that radial distance to contact can be estimated from forces measured at the base of the whisker shaft. We show that in the case of a moving object contacting a whisker, the measured force can be ambiguous in distinguishing a nearby object moving slowly from a more distant object moving rapidly. This ambiguity can be resolved by simultaneously extracting object position and speed from the whisker deflection time series – that is by attending to the dynamics of the whisker's interaction with the object. We compare a simple classifier with an adaptive EM (Expectation Maximisation) classifier. Both systems are effective at simultaneously extracting the two parameters, the EM-classifier showing similar performance to a handpicked template classifier. We propose that adaptive classification algorithms can provide insights into the types of computations performed in the rat vibrissal system when the animal is faced with a discrimination task.

Introduction

Rats, mice and other whiskered mammals can discriminate a variety of tactile object properties using only their facial whiskers (vibrissae). For instance, rats are able to discriminate surface textures, with different degrees of roughness, with similar acuity to the human fingertip [4]; the Etruscan shrew - the smallest living mammal - can recognise and localise prey animals (insects) from a small number of fleeting whisker contacts, sufficient to allow fast and precisely targeted attacks [1]; and sea mammals such as seals and walruses are able to make judgements about object size, shape, and direction of movement, using only tactile signals from their vibrissae [5].

Several properties of the vibrissal system make it stand out as an interesting model system in which to investigate theories about the sensory guidance of behaviour. First, in tactile sensing systems generally, the sensory apparatus is usually brought into contact

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 178–187, 2010.

with objects in a deliberate and controlled manner. Whether it is a person exploring an object contour with their fingertips or a rat palpating (whisking) its vibrissae against a surface, purposive control or active sensing is key to information acquisition [14]. Second, the processing of tactile signals may require a relatively small number of stages. For instance, in the rat, there are multiple closed loops connecting vibrissal signals to actuation mechanisms such that new information can begin to influence behaviour after passing through just a small number of synapses [12]. Processing can also be very fast. For instance, whisker contact signals can reach the barrel cortex – from where they can begin to effect processing in behaviour-related areas such as the motor cortex, the superior colliculus, and cerebellum – in just 7 milliseconds [7]. The rat brain thus appears to be tuned to pick out the behaviourally-relevant aspects of vibrissal signals rapidly and in just a small number of steps.

Inspired by the vibrissal systems of mammals, we are working to develop artificial whisker systems for fast and accurate tactile discrimination – which could be useful for mobile robots – and at the same time can be used to test theories of mammalian sensorimotor control. Previous work has shown that information about texture, distance to contact, and shape can be extracted from signals obtained when an artificial whisker is moved against a surface [9] [8] [11] [6]. Our current work extends these findings in several directions. First, by exploring how classifier systems can be trained to extract a range of different tactile properties from whiskers signals with relatively little pre-processing of the sensor input. Second, by showing that such systems can extract multiple features simultaneously from the same signal. Third, by investigating the effects of active control of the sensor apparatus on the problem of tactile feature discrimination. At the same time, though described elsewhere, we are developing decision-making algorithms for these systems that can optimise the speed-accuracy tradeoff [13].

Developing models of whisker based perception has been problematic. In passive sensory modalities such as vision and audition it is generally quite easy to present stimuli to a passive sensor on a robot, or images and tones can be simulated and used to train a computational model. There is no obvious analog for tactile stimuli, and the true nature of tactile stimuli is too poorly understood to be simulated accurately. Whiskers are especially difficult to simulate accurately, as they have very low mass but high spring constants when modelled as a series of masses on rotational springs, leading to numerical instabilities. Additionally when the parameters of a whisker-object contact become more numerous (e.g. speed and radial distance to contact, surface texture, orientation and softness etc) it becomes very difficult to constrain the contact and generate reliable signals in either simulated or physical robots. For these reasons acquiring sufficient examples of carefully controlled whisker contacts with tactile stimuli to train models and classifiers has proved difficult. To facilitate the study of artificial vibrissal sensing we therefore present here a novel system for generating large sets of tactile stimuli and deflection signals. An XY positioning robot is programmed to move objects into an artificial whisker sensor in an accurate and highly repeatable manner (Fig.1). Deflections for the whisker are streamed to a PC, and can be processed in real time to control subsequent movement of the robot arm. Under passive deflections the object moved by the robot arm makes contact with the artificial whisker and deflects the whisker through a large angle. When deflection reaches a critical point the whisker loses friction with the

object, deforms and deflects past the object and goes through oscillatory ringing until the energy dissipates and the whisker comes to rest. However, in addition to passive touch experiments we are also able to use our experimental setup to investigate active sensing. In this case we mimic a control policy that we have observed in rats in our own laboratory whereby the protraction of a whisker ceases rapidly on contact with a surface and whisker then begins to retract [14]. In contrast to the passive case, this policy, which we call Minimal Impingement (MI), keeps the amplitude and duration of whisker deflection within a limited range, and also keeps whisker ringing after contact to a minimum. An additional benefit is that the forces acting on the whisker are much smaller, meaning whisker breakage is less likely – an important consideration for autonomous robotics.

The remainder of this paper motivates and describes a discrimination algorithm developed using this test-bed and shows its utility under differing parameters for accurately extracting object features from vibrissal deflection signals. In future work these algorithms will be transferred to operate in less constrained circumstances onboard mobile robot platforms such the SCRATCHbot robot described in our partner paper [15].

(a) (b)

Fig. 1. The XY positioning robot (a) from above, to show the range of movement available. (b) From the side.

Simultaneous Radial Distance and Speed Task

The object properties we chose to manipulate and attempt to recognize were radial distance to contact from the base, and contact speed. The task is to recognise these two parameters *simultaneously*. To investigate this problem we presented a vertical pole to the whisker at a range of radial distances from the base, and at different movement speeds. We show that, depending on the features used for classification, radial distance detection is confounded with contact speed. Previous work [3] [10] has shown that a rat could encode the radial distance to contact along a whisker by monitoring the magnitude

of forces (or moments) at the base. Others have suggested that the increased firing rate of cells in the whisker sensory nerve, for contacts close to the base, could be due to the increased moments at the whisker base. Static beam equations, and analyses relying on instantaneous measures of moments do not account for the dynamic properties of objects. If an object collides with a whisker at the same location but at different speeds it will induce different forces at the base. For example, under the right conditions the moment at the base will be the same for a slowly moving object contacting near the base, and a fast object near the tip (see Fig.2 for a demonstration of this). This ambiguity in the signal cannot be accounted for with a single observation, an additional observation or feature must be found in order to discriminate these two properties of the collision. Successful classification relies either on finding the contact speed before conducting a radial distance estimation, or discriminating both properties simultaneously. In the analysis we assess a simple template-based classifier and compare its performance to a classification using an adaptive template classifier, or EM (Expectation Maximisation) algorithm [2] based classifier. Previously we have shown that templates can be used for discriminating tactile features in simulation [8], and that spectral templates can be used to discriminate whisker deflection signals from floor surface textures in a real world environment [6]. In these cases the templates were hand-picked from the data set for better classification. In the present study we show that template based classification can be used to successfully discriminate ambiguous whisker signals in hardware, and that the templates can be found adaptively using a simple Hebb-like learning algorithm.

Methods

The XY positioning robot. An XY positioning robot (Yamaha-PXYX, Yamaha Robotics) (see Fig.1) was used to move objects into the whisker. The robot has a movement range of 350x650mm, and can move up to 720mm/s. Repeatability of the robot is $\pm$0.01mm, and the maximum load it can carry is 1.5kg. Objects are carried by the robot into an artificial whisker fixed to the table, as this allows us to control the contact as carefully as possible. Moving the whisker into an object would cause the whisker to oscillate unpredictably during movement between contacts, and as a result each contact would be slightly different. A controller (Yamaha RCX 222, 2-axis robot controller) takes instructions from a PC through an RS232 cable, and the controller interprets the instructions, completes path integration, and drives the motors. Instructions for the robot are generated inside a Matlab (www.mathworks.com) loop, and can be easily updated during robot operation, depending on the whisker input.

The Whisker. A whisker sensor was taken from the SCRATCHbot robot platform (described in detail in [15] [16]). Technical details of the whisker can be found in [6]. A tapered, flexible plastic whisker, $\approx$5 times scale models of a rat whisker, was mounted into an inflexible rubber-filled 'follicle' case (Fig.3(a)). A tri-axis hall effect sensor mounted in the follicle case measures the deflection of a magnet fixed to the base of the whisker shaft (Fig.3(b). The hall effect sensor IC was programmed to generate two voltages, corresponding to the magnitude of the whisker base deflection in two directions, x and y.

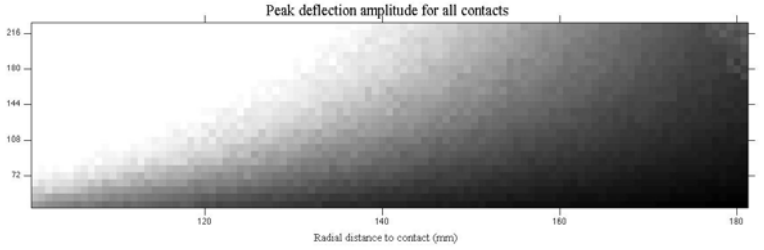


Fig. 2. Peak deflection magnitude for each speed-radial distance pair. Deflection magnitude (brightness of pixel), or force, has been used in the past as a discriminator of radial distance to contact. Here a given radial distance results in differing deflection magnitudes depending on movement speed. If deflection magnitude were a sufficient criteria for radial distance estimation, the brightness would decrease evenly across all speeds (Y axis).

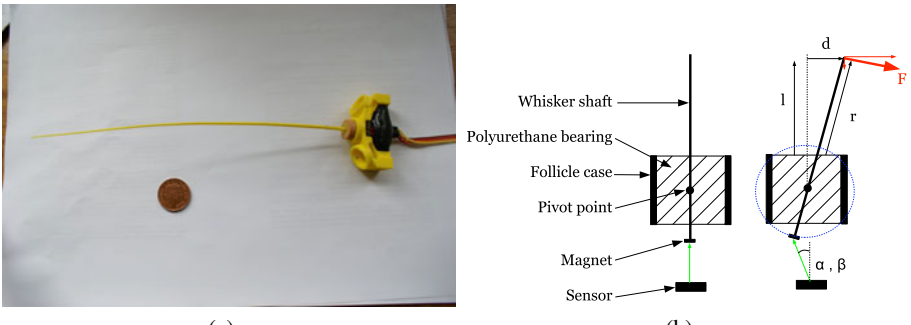


Fig. 3. Artificial whisker shaft and follicle, with a UK 1 pence coin for scale (b)Diagram of the artificial whisker Hall effect sensor

Data. Deflections of the whisker were transmitted through the hall effect sensors to a LabJack UE9 USB data acquisition card (www.labjack.com) at a rate of 1 kHz for each of the x and y directions. Each trial lasted 4s. This data was sent to a computer through the BRAHMS middle- ware (brahms.sourceforge.net) for analysis in Matlab.

Robot control. Minimal impingement was implemented by instructing the robot to move an object into the whisker at a given speed until a deflection threshold is crossed, at which point the robot retracts the object as fast as possible (720mm/s). Temporal latency for the loop is $\approx$ 300ms from initial contact due to the controller duty cycle. Though this latency is short enough for the present study, we are working to reduce this latency by gaining more direct control over the robot motors.

The task. Preliminary investigations highlighted that the closest contact that could be made by the whisker at any reasonable speed without saturating the Hall effect sensor was $\approx$80mm from the base. Contacts at less than 5mm from the tip did not deflect the base of the whisker for long enough before slipping past to allow an MI type contact. Therefore, the 185mm length whiskers provide a 100mm range of radial distances. Contact speeds above 216mm/s either cause the whisker to slip past the object before a

retraction, or saturates the sensors. 36mm/s was the lower bound on the speed here. Contacts were sampled at radial distance intervals of 1mm, and speed intervals of ≈7mm/s over the previously described ranges. In total 101 radial distances and 26 speeds were sampled, giving 2626 different radial distance and speed combinations. Contact combinations were randomly interleaved to limit any affects of changing whisker properties. For each contact combination the whisker was deflected by the robot in both a clockwise and anticlockwise directions (-ve and +ve in x), ensuring that the whisker did not bend over time through repeated unilateral deflections. The experiment was performed twice to generate sufficient data for classification.

Data from each trial was stored separately. Deflections from the clockwise robot movement trials (-ve in x) were converted so all data samples were equivalent. Trials were ordered into arrays by speed and radial distance to contact. Each trial was aligned to peak deflection, and shortened to only the 325ms either side of the peak deflection (751 data samples in total).

Analysis

The data were separated into training and test sets that were each complete data sets of 26 speeds and 101 radial distances. Signals were placed in the training or test sets at random from the original data. In each case classifiers were developed on the training sets, and performance was determined on the test set.

Template based classification. Template based classification involves recording example sensory data as templates during a training phase, and comparing the stored templates to novel data during the test phase. By systematically comparing the novel data to signals encountered previously, a classification can be made by declaring which of the stored templates the novel signal is most similar to. In the present study each template corresponds to a speed-radial distance pair. Classification based on these templates is therefore simultaneous classification of both speed and radial distance. From the training data set a subset of trials – representative of the larger set – were stored in an array as templates. The number of templates chosen were dependent on the experimental condition. During the test phase, trials were taken at random from the test set as inputs to the classifier. An element-wise sum of squared errors calculation was made between the input I and each template T_i,

$$e(T_i) = \sum_{t=1}^{n} (I(t) - T_i(t))^2. \tag{1}$$

where n is the length of the template, in samples. The template with the lowest sum of squared error was determined the winner, and a recording was made in an output array of the estimated speed and radial distance to contact of the input trial. We conducted the experiment with a full complement of templates (26x101) as a benchmark for classification, as well as with two reduced sets of templates, (13x50 and 7x26 templates respectively).

Adaptive EM template based classifier. Picking templates by hand, though successful, is an inefficient method of developing a classifier. A better method is to adaptively learn

a set of templates in an unsupervised way, to find a set that covers the bulk of the data space for a given number of templates. Such adaptation can be performed with an EM algorithm [2]. Each time a classification is made it is verified against the true value of the two contact parameters. If the classification was correct, the winning template is modified to be more like the input pattern. If the classification was incorrect, the appropriate losing template is modified to be more like the input pattern. This template modification is achieved by taking a weighted mean of the template and the input,

$$T = \frac{(I + \alpha T)}{1 + \alpha} \tag{2}$$

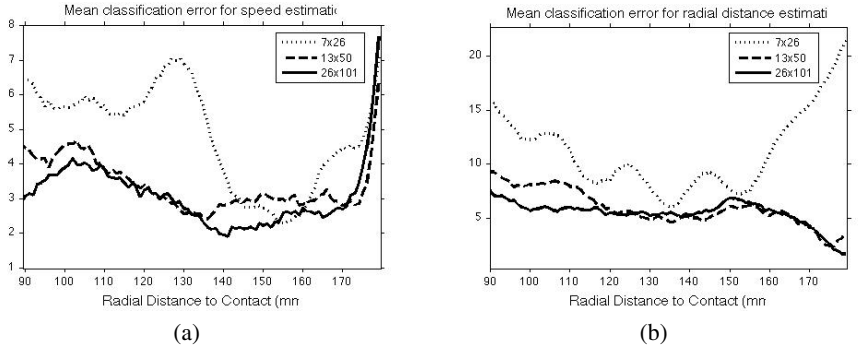
where α is the learning rate of the classifier. By increasing $alpha$ it is possible to learn quicker, but it is less likely that an optimal value will be reached. (The positive and negative modifications to templates may be viewed as Hebbian and anti-Hebbian learning respectively, for templates as linear neurons.) We decided to run the adaptive classifier with a reduced set of templates (13x50), to see if it was possible to achieve performance close to or equivalent to that of hand picked templates.

Results

Template based classifier. The template based classifier with 13x50 templates was capable of successfully classifying 65% of inputs to within 50mm/s of speed and 10mm of radial distance. This performance decreases as the number of templates used is reduced.

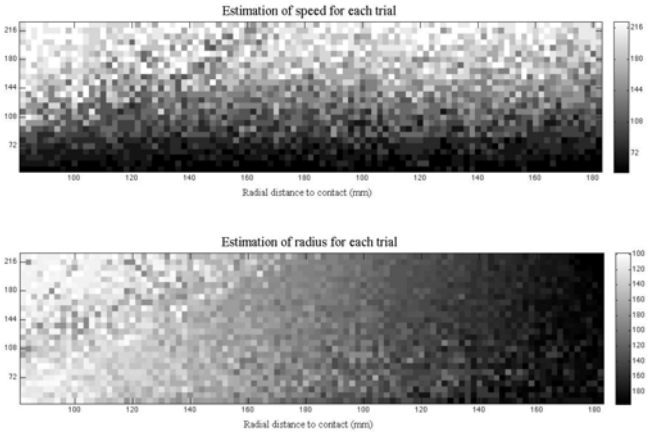
Adaptive EM template classifier. We found that after training the Adaptive EM template classifier was capable of classifying 66% of inputs to within 50mm/s of speed and 10mm of radial distance.

Fig.5 shows the results in graphical format. Input signals are shown, arranged for speed and radial distance. Pixel brightness indicates the value assigned to that input

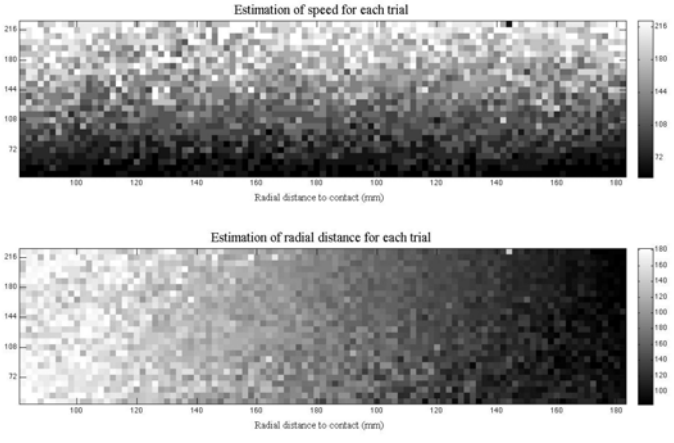


(a) (b)

Fig. 4. Mean classification error for each radial distance along the whisker, for different numbers of templates. Y axis corresponds to mean distance between the real and predicted values of speed (a) and radial distance (b). Error is at its lowest in the portion of the whisker near, though not at, the tip.



(a)



(b)

Fig. 5. Classification performance of the EM template classifier (a), against that of a classifier with hand picked templates

signal. A completely correct classification of speed (Fig.5(a) would appear as a gradual transition from dark to light vertically along the Y axis, and a corresponding transition from light to dark horizontally along the X axis for correct radial distance detection (Fig.5b). We can see that both classifiers achieve some degree of correct classification, as well as a number of mistakes. Classification tends to be best for larger radial distances (contacts nearer the tip, lower right region of each plot). Contacts at high speed and near the base tend to be systematically misclassified (upper left region of each plot). Fig.4 shows the mean error of classification for each point along the whisker. Error is lowest in the region near, though not at, the whisker tip. Performance of the classifiers with

reduced sets of templates is good, though the performance of the classifier with the fewest (7x25) templates is less reliable over certain regions of the space.

Discussion

By collecting a large data set and exhaustively tiling a feature space we have shown that certain features in the whisker signal that are ambiguous in isolation can be simultaneously discriminated using a robust, and computationally cheap adaptive classified system. Previously we have shown [9][6] that successful classification of surface properties, such as texture, is highly dependent on knowing the location of the surface and the nature of the contact. We believe that the first steps towards the goal of simultaneously extracting a range of relevant object properties have now been taken. The success of the classifiers at contacts nearer the tip may also go some way to explain the whisking behaviour seen in rats. Rats appear to control their whiskers so as to make contacts at or near the whisker tip [14], possibly because this creates signals that vary more predictably across contact parameters allowing the animal to make better judgements. More generally, keeping whisker deflection amplitude and duration within a limited range using active sensing strategies will allow the development of classifiers that are more sensitive to smaller changes in the input. Indeed, the data presented here suggests that contacts of this type are easier to discriminate over a particular section of the whisker, suggesting a 'sweet spot' of whisk speed and whisker contact location. The development of fast, adaptive classifiers for tactile feature discrimination could also provide insights into signal processing in areas of the vibrissal system such as the barrel cortex where the rat is known to be able to rapidly extract behaviourally-relevant properties of the stimulus in a small number of processing stages. Ullman et al [17] have proposed that the visual system operates through hierarchies of progressively more complex adaptive feature-matching templates. The ideas investigated in the current paper might therefore be considered as the first step towards identifying a similar, general scheme for understanding cortical processing in the domain of touch.

References

1. Anjum, F., Turni, H., Mulder, P.G.H., van der Burg, J., Brecht, M.: Tactile guidance of prey capture in etruscan shrews. Proc. Natl. Acad. Sci. USA 103(44), 16544–16549 (2006)
2. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
3. Birdwell, J.A., Solomon, J.H., Thajchayapong, M., Taylor, M.A., Cheely, M., Towal, R.B., Conradt, J., Hartmann, M.J.Z.: Biomechanical models for radial distance determination by the rat vibrissal system. J. Neurophysiol. 98(4), 2439–2455 (2007)
4. Carvell, G.E., Simons, D.J.: Biometric analyses of vibrissal tactile discrimination in the rat. J. Neurosci. 10(8), 2638–2648 (1990)
5. Dehnhardt, G., Ducker, G.: Tactual discrimination of size and shape by a california sea lion (zalophus californianus). Animal Learning and Behavior 24(4), 366–374 (1996)

6. Evans, M., Fox, C.W., Pearson, M.J., Prescott, T.J.: Spectral template based classification of robotic whisker sensor signals in a floor texture discrimination task. In: Kyriacou, T., Nehmzow, U., Melhuish, C., Witkowski, M. (eds.) Proceedings of Towards Autonomous Robotic Systems (TAROS 2009), pp. 19–24 (2009)
7. Ferezou, I., Bolea, S., Petersen, C.C.H.: Visualizing the cortical representation of whisker touch: voltage-sensitive dye imaging in freely moving mice. Neuron. 50(4), 617–629 (2006)
8. Fox, C., Evans, M., Pearson, M., Prescott, T.J.: Towards temporal inference for shape recognition from whiskers. In: Ramamoorthy, S., Hayes, G.M. (eds.) Towards Autonomous Robotic Systems, pp. 226–233 (2008)
9. Fox, C.W., Mitchinson, B., Pearson, M.J., Pipe, A.G., Prescott, T.J.: Contact type dependency of texture classification in a whiskered mobile robot. Autonomous Robots (2009)
10. Kaneko, M., Kanayama, N., Tsuji, T.: Active antenna for contact sensing. IEEE Transactions on Robotics and Automation 14(2), 278–291 (1998)
11. Kim, D., Moller, R.: Biomimetic whiskers for shape recognition. Robotics and Autonomous Systems 55, 229–243 (2007)
12. Kleinfeld, D., Ahissar, E., Diamond, M.E.: Active sensation: insights from the rodent vibrissa sensorimotor system. Curr. Opin. Neurobiol. 16(4), 435–444 (2006)
13. Lepora, N.F., Evans, M., Fox, C.W., Diamond, M.E., Gurney, K., Prescott, T.J.: Naive bayes texture classification applied to whisker data from a moving robot. In: World Congress on Computational Intelligence (submitted 2010)
14. Mitchinson, B., Martin, C.J., Grant, R.A., Prescott, T.J.: Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. Proc. Biol. Sci. 274(1613), 1035–1041 (2007)
15. Pearson, M.J., Mitchinson, B., Welsby, J., Pipe, T.G., Prescott, T.J.: Scratchbot: Active tactile sensing in a whiskered mobile robot. Submitted to SAB (2010)
16. Prescott, T.J., Pearson, M.J., Mitchinson, B., Sullivan, J.C.W., Pipe, A.G.: Whisking with robots from rat vibrissae to biomimetic technology for active touch. IEEE Robotics and Automation Magazine 16(3), 42–50 (2009)
17. Ullman, S., Vidal-Naquet, M., Sali, E.: Visual features of intermediate complexity and their use in classification. Nat. Neurosci. 5(7), 682–687 (2002)

A Supramodal Vibrissa Tactile and Auditory Model for Texture Recognition

Mathieu Bernard[1,2], Steve N'Guyen[1,2], Patrick Pirim[2], Agnès Guillot[1], Jean-Arcady Meyer[1], and Bruno Gas[1]

[1] Institut des Systèmes Intelligents et de Robotique,
UPMC Paris 6, CNRS UMR 7222, 4 place Jussieu, 75252 Paris cedex 05, France
[2] Brain Vision Systems, 32 rue du Javelot, 75013 Paris, France

Abstract. Audition and touch endow spectral processing abilities allowing texture recognition and discrimination. Rat whiskers sensory system exhibits, as the cochlea, resonance property decomposing the signal over frequencies. Moreover, there exists strong psychophysical and biological interactions between auditory and somatosensory corteces concerning texture analysis. Inspired by these similarities, this paper introduce a "supramodal" model allowing both vibrissa tactile and auditory texture recognition. Two gammatone based resonant filterbanks are used for cochlea and whiskers array modeling. Each filterbank is then linked to a feature extraction algorithm, inspired by data recorded in the rats barrel cortex, and finally to a multilayer perceptron. Results clearly show the ability of the model for texture recognition in both auditory and tactile tuning. Moreover, recent studies suggest that this resonance property plays a role in texture discrimination. Experiments presented here provide elements in the direction of this resonance hypothesis.

1 Introduction

Spectral information carried by the tactile and auditory systems is a primary cue used in cognitive tasks like speech or music perception in audition as well as surface or object recognition in touch. These abilities depend strongly on our perception of complex stimuli like surface textures through the skin or acoustic timbres through audition. Humans are able to discriminate textured surfaces by touch only, by auditory signals only and both touch and audition [1], with similar discrimination performance in each case.

Among the somatosensory systems, rat whiskers are capable of fine texture discrimination. By actively whisking their vibrissae, rats extract information about the spatial properties of a surface or object, including size, shape and texture [2] [3]. Each vibrissa has a strong frequency tuning around a resonant frequency [4] [5], allowing to increase the sensitivity of the vibrissa sensory system to an ecologically relevant range of low-amplitude, high-frequency stimuli [6]. The rat facial whiskers array allows the decomposition of the signal into an ensemble of components at different frequencies. There exist strong parallels between vibrissa tactile and auditory encoding. Inner hair cells - sensitive cells of

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 188–198, 2010.

the auditory system disposed along the basilar membrane of the cochlea - have also a strong frequency selectivity and decompose the auditory signal over resonant frequencies, from high frequencies in the cochlear base to low frequencies in the apex [7]. Thus, both cochlea and whiskers decompose a signal into frequency components, each sensory cell tuned to a particular resonant frequency. A tonotopic organization is found in the auditory pathway [7], especially in subcortical centers, as well as in the somatosensory pathway [8].

More recently, it has been shown that auditory and somatosensory systems interact in texture perception, both on psychophysical and biological levels (see [9] for a recent review). For example, Yau *et al.* [10] found a cross-sensory interference between auditory and touch temporal frequency channels in human: a touch-based texture discrimination task could be disturbed by a frequency-dependent auditory signal in a systematic manner. More precisely, given that the auditory system is specialized for spectral analysis, Yau *et al.* [11] recently suggested that spectral analysis of tactile signals is processed in the caudo-medial belt area, a region of the auditory cortex. They also proposed clear and testable predictions about underlying physiology.

This present contribution propose a same model for both vibrissa tactile and auditory texture recognition. In this way it can be characterized as supramodal. This work is inspired by the strong parallels between vibrissa and cochlea transduction in the one hand, and by crossmodal auditory and tactile interaction within texture discrimination in the other hand. We propose in Section 2 a whiskers array model based on a classical cochlear model using gammatone filterbank [12]. Each filterbank output is then plugged to a bioinspired multichannel feature extraction algorithm presented in Section 3. By estimating the instantaneous amplitude-frequency product of the signal, this algorithm is closely inspired by data recorded in the rat somatosensory cortex, which are supposed to be related to temporal frequency estimation [8]. Then, a multilayer perceptron is used to discriminate textures with this feature. Section 4 is devoted to the experimental part of this paper. A set of experiences are proposed in order to show texture classification skills for both tactile and auditory modalities with several sets of textures. A specific attention will be made to the influence of the resonance effect on the performance. Finally, a discussion about the results and there incidence is provided in Section 5.

2 Cochlear and Whiskers Array Models

There exists several cochlear models in the literature. Some trying to reproduce internal (both active and passive) phenomena in the cochlea, *e.g.* basilar membrane elasticity or fluid mechanics [13]. In a more abstract level, the cochlea can be seen as a filterbank [14], decomposing the signal over frequency channels. One classical filterbank cochlear model is the gammatone auditory filterbank introduced by Patterson *et al.* in [12].

Rat whiskers have also been modeled, both with mathematical models and robotic implementations. Neimark *et al.* [4] proposed a model of vibrissa as a thin

elastic beam, extended in [5], and have experimentally tested resonance-related predictions. These studies suggest to their author that whiskers resonance should be an important property for performing fine texture discrimination. Whiskered robotic sensors have also been developed for texture discrimination, both with real rat whiskers [15] or with artificial systems [16], [17]. Nevertheless, in spite of the similarities between cochlear and whiskers resonance properties, filterbank based whiskers modeling has not been investigated yet.

In one hand we briefly introduce gammatone filters and their use in a cochlea tuning. Because there exist fine parameters for human cochlea modeling which are not provided for rat cochlea, human parameters are used in this paper. On the second hand, we propose an adaptation of this cochlear model for modeling a rat whiskers array. The free C implementation of gammatone filters provided by Ma[1] is used in this paper.

2.1 Gammatone Filterbank for Cochlear Modeling

Gammatone filters are well know to fit basilar membrane response to an impulse stimuli [12]. It appears that this is still similar for the movements of a vibrissa base [5]. In temporal domain, the impulse response of a gammatone filter is:

$$g(t) = at^{n-1}cos(2\pi f_c t + \phi)e^{-2\pi bt}. \quad (1)$$

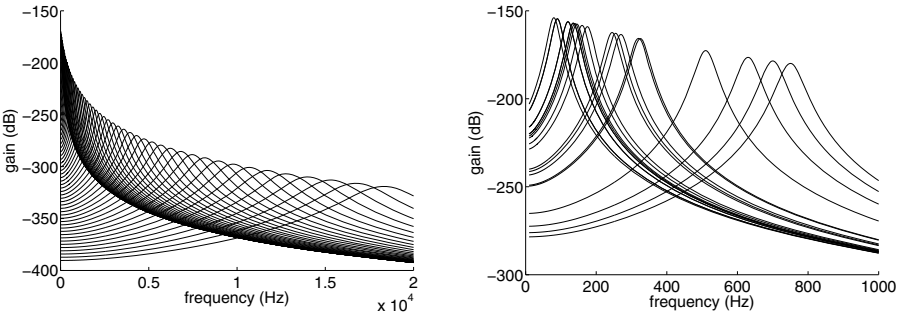
Parameters of the filter are f_c, b and n: f_c is the center frequency of the filter, b determines the duration of the impulse response and thus the bandwidth of the filter, n is the order of the filter and largely determines the slope of the skirts. A gammatone of order 4 best fits human auditory filter shapes and is used in this paper. Slaney proposed in [18] a general formulation of the bandwidth of a 4^{th} order filter in function of the center frequency f_c, the asymptotic filter quality at large frequencies q_{ear} and the minimum bandwidth for low frequencies channels b_{min}. Glasberg and Moore [19] estimated these parameters from psychoacoustical human data and proposed $q_{ear} = 9.26$ and $b_{min} = 24.7$.

Moreover Slaney [18] addresses a solution for channel spacing along the frequency axis, assuming the number of channels n_c, the lowest and the highest center frequencies f_l and f_h are known. Transfer functions of a 50 channels gammatone filterbank tuned as cochlear model are plotted on Fig. 1(a). The human auditive frequency range were used ($f_l = 20$ Hz and $f_h = 20$ kHz) both with the Glasberg and Moore parameters.

2.2 Adaptations for Whiskers Array Modeling

As we said above, the filterbank model of whiskers array is derived from the human cochlear one. It means that each vibrissa is represented by a well-tuned gammatone filter. There are two strong arguments in favor of this approach for

[1] Ma, N.: On efficient implementation of gammatone filters. http://www.dcs.shef.ac.uk/∼ning/resources/gammatone.



(a) Cochlear model (b) Whiskers array model

Fig. 1. Transfer functions of the gammatone filterbanks. (a): cochlear model, 50 channels, $f_l = 20$ Hz, $f_h = 20$ kHz, $q_{ear} = 9.26$ and $b_{min} = 24.7$. (b): whisker array model, 20 vibrissae, center frequencies from [4], $q_{ear} = 35$ and $b_{min} = 15$.

modeling whiskers. Firstly, it emphasis on the resonance phenomena which is understood as a key mechanism for temporal frequency perception and thus for texture discrimination. Secondly, gammatone filter and rat vibrissa have quite similar impulse responses [5].

The problem is now to adapt the whiskers model from the cochlea to fit as close as possible biomechanical properties of rat's whiskers. In [4], Neimark *et al.* provided the center frequencies of 20 vibrissa measured on a living rat. 5 vibrissa arcs are described, each composed of 4 whiskers. These values are used in order to fix the center frequency of each whisker gammatone filter. Andermann *et al.* [20] identified populations of neurons in the trigeminal ganglion and the primary somatosensory cortex of rats that respond to whisker vibrations only within a narrow band of frequencies centered at resonance. Nevertheless, to the best of our knowledge, there doesn't exist any experimental data directly concerning the bandwidth of whisker filters. Assuming the lack of data, we fixed $q_{ear} = 35$ and $b_{min} = 15$. These values provide a bandwidth from 111 Hz from low resonant whiskers to 233 Hz for high resonant ones (mean frequency is 145 Hz). In comparison with other existent models, this approach deals with simplicity and offers a functional view of vibrissa transcription mechanisms: as in the cochlear model, biomechanical details of whisker transduction are omitted. Obviously, the counterpart is a lack a biological fidelity. Transfer functions of the generated filters used as whiskers array model are showed in Fig. 1(b).

3 Feature Extraction for Texture Discrimination

Model presented in the precedent section aim to reproduce basilar membrane and vibrissa motion in response to a stimuli. To serve as basis for feature extraction, a peak extractor algorithm is plugged to each filterbank channel output,

mimicking the sensory nerve spike train activity. If the current sample is not a local extrema of the signal, its value is set to 0. This approach relies on the strong hypothesis that the peaks thus characterized provide enough information to describe a texture. Such hypothesis is reinforced by the fact that, when Licklider and Pollack [21] assessed the effects of various signal distortions in human speech recognition, they found that "infinite clipping" - a treatment that only kept a signal's periodicity - did not prevent speech recognition in humans.

In a previous work [16], we experienced texture discrimination on a whiskered robotic platform with a feature extraction algorithm inspired by data recorded in the rat's barrel cortex, which estimates the instantaneous mean power of a multichannel spike train. Using an elastomer-based whiskers array [22], the robot was able to discriminate 8 different sandpapers with a mean performance above 90%.

Each rat's whisker projects to a precise part of its somatosensory cortex, in a structure named "barrel cortex". A barrel is a discrete neural structure that receives an input principally from a given whisker, with a little influence from neighboring ones [23]. Neuronal base of texture representation have been investigated in anesthetized rats's barrel cortex [8]. By stimulating a whisker with a pure sinusoid fully described by its amplitude A and its frequency f and by recording the induced neural activity in the barrel cortex, it appears that the neural activity most probably encodes a quantity homogeneous to the product Af and called the equivalent noise level.

The feature extraction algorithm used in this paper estimates the instantaneous frequency f through the inverses of the time intervals between successive peaks. Thus the peak amplitude is multiplied by the estimate frequency and accumulates within a time window. In addition of this instantaneous mean power calculation, we also use in Section 4 both A and f alone as features for texture discrimination. These three feature extraction algorithms are summarized in Fig. 2.

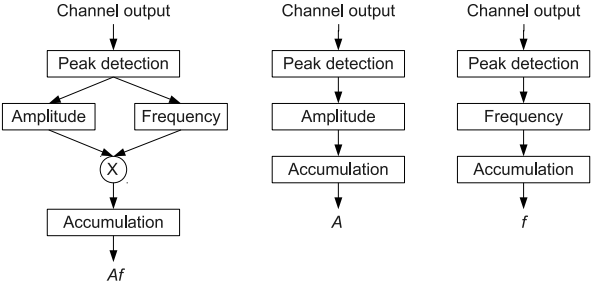


Fig. 2. Feature extraction algorithms for a single channel. From left to right: equivalent noise level estimation by Af product, amplitude A only, frequency f only. Accumulation is done over a constant time window.

4 Experiments

In order to demonstrate that the two models presented above are able to discriminate textures, several sets of experiences are proposed. The first one presents the texture discrimination skills, both in cochlear and whiskers tuning. The other ones are focusing on precise aspects of the model in order to illustrate the resonance effect and the whiskers bandwidth influence on the results. These experiments share the same data acquisition which is described before the results.

4.1 Experimental Device

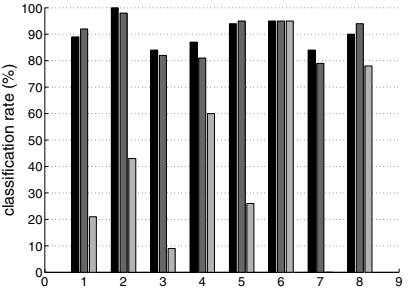
Sets of sandpapers are classically used in texture discrimination experiments on whiskers system. Neimark *et al.* [4] modeled a sandpaper in the frequency domain as a three peaks spectrum, with a dominant activity of low frequencies. According to this, a set of 8 textures is generated for each experiment (see Fig. 3). The lowest frequency peak is amplified to 6 dB whereas the two others are amplified to 3 dB. Other frequencies are attenuated to -3 dB. 40 seconds input files are generated with a sample frequency of 44100 Hz for the cochlea textures set and 5000 Hz for the whiskers set. Thus, each file is passed through its associate gammatone filterbank and feature extraction algorithms. The time window is set to 100 ms, which match the period of a typical rat whisking movement. For each texture 400 data vectors is computed, 300 for learning and 100 for testing. Vectors dimension is equal to the number of channels of their related filterbank and each vector contains values accumulated by the feature extraction algorithm over one time window.

Having thus obtained an input vector for each trial, we fed it into a simple three layer perceptron to perform supervised learning. The hidden layer had the same dimension than the input layer and the input vector. Obviously the output layer contains 8 neurons, one per texture. We used the FANN library [24] with the iRPROP training algorithm [25]. The final classification was done by a winner take all on the 8 output neurons.

4.2 Experimental Results

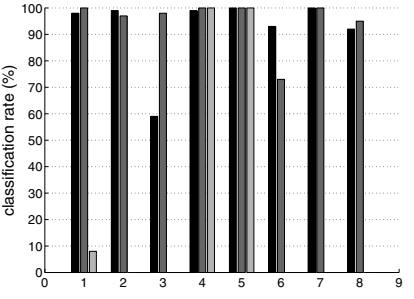
The first experiment presents the texture discrimination skills of the cochlear and whiskers array model described above. Filterbanks used are those showed in Fig. 1 and cover a frequency range between 20 Hz and 20 kHz for the cochlea and between 80 Hz and 800 Hz for whiskers. For each experience, the three feature extraction algorithms are compared over a set of 8 different textures. Classification results are presented in Fig. 3, both with the detailed spectral composition of the different textures. This experiment clearly shows the ability of the model for texture discrimination, with a mean classification rate near 90% for Af and A features. Moreover, it is shown that the f feature alone is irrelevant for texture discrimination: results are greater than the chance level (12.5% for 8 textures) but clearly under the amplitude based features (see Fig. 3 for details).

Texture	Peak 1	Peak 2	Peak 3
1	250	420	710
2	100	200	500
3	90	350	800
4	200	350	800
5	250	500	600
6	65	250	600
7	90	600	800
8	150	550	850



(a) Whiskers model texture discrimination. Mean rate: $Af = 90.4\%$, $A = 89.5\%$ and $f = 41.5\%$

Texture	Peak 1	Peak 2	Peak 3
1	1000	3000	5000
2	100	1000	10000
3	500	2000	7000
4	150	3000	7000
5	200	10000	15000
6	300	5000	10000
7	500	1000	3000
8	200	1000	2000



(b) Cochlear model texture discrimination. Mean rate: $Af = 92.5\%$, $A = 95.4\%$ and $f = 26.0\%$

Fig. 3. Classification results for the model in cochlea tuning and whiskers array tuning, tested with 8 different textures. Left column: three peaks textures set used for discrimination, frequencies are given in Hz. Right column: classification rate for the 8 textures and the 3 features (Af in dark grey, A in pure grey and f in clear grey).

As showed in Fig. 4, where the comparison of the A and f features response to a pure sinus near the resonance frequency of the A4 vibrissa ($c_f = 630$ Hz) is provided, the f feature is sensitive to the resonance effect lesser than the A feature. The frequency based pattern becomes consequently less discriminative than the amplitude based pattern, that explains the irrelevance of the f feature for texture discrimination.

The major limitation of the gammatone based whiskers array model, as we said above, is the lack of biological data concerning the bandwidth of the filters. In the previous experiments, we used arbitrary fixed bandwidth values. Figure 5 shows the evolution of the mean classification rate in function of the whiskers filters mean bandwidth, from 5 Hz to 500 Hz. The same textures as in Fig. 3(a) are used on the three features Af, A and f. Experimental results confirm those of Fig. 3(a) by showing the weak performance of the f feature alone. Moreover, one can observe that the model follows the same behavior for the three features: it doesn't work with minimal bandwidth value (the classification rate is equal

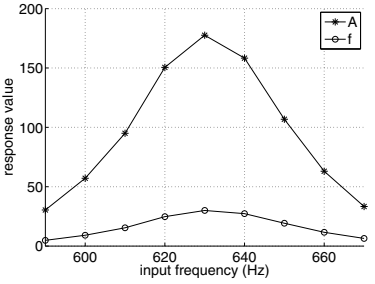


Fig. 4. Comparison of the A and f features response to a pure sinus near the resonance frequency of the A4 vibrissa ($c_f = 630$ Hz)

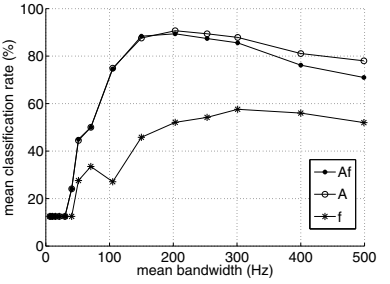
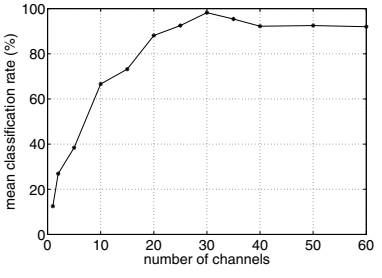
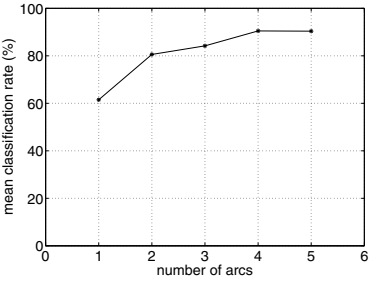


Fig. 5. Influence of the whiskers filters mean bandwidth on the classification rate. The set of 8 textures of Fig. 3(a) is used for the 3 features Af, A and f.



(a) Cochlear model



(b) Whiskers array model

Fig. 6. Influence of the number of filters on the mean classification rate of a set of textures. Input textures are the same as in Fig. 3. Af feature is used.

to the chance level upon 30 Hz) but quickly rises the maximal classification rate for a mean bandwidth near 200 Hz for Af and A or near 300 Hz for f. Above this value, the mean performance is slowly decreasing.

The last experiment, presented in Fig. 6, tests the influence of the number of channels on the discrimination performance. Input textures were the same as previously (see Fig. 3) for both modalities and the equivalent noise level estimation Af was the only feature took into account. Nevertheless, due to morphological differences between the cochlea and a whiskers array, channel reduction is done by a different way. Thus, cochlear channels are disposed along a frequency range thanks to the Slaney channel spacing solution [18]: decreasing the number of channels will reduce the channel density but the same frequency range will be covered. Fig. 6(a) plots the evolution of the mean classification rate from 1 to 60 cochlear channels. In the other way, whiskers are organized in arcs over the rat's face [4], each arc containing whiskers with similar length and resonance properties. The channel reduction is done arc by arc following the length gradient from rostral arcs to caudal ones. Figure 6(b) plots the evolution of the mean classification rate from 1 to 5 arcs composing the whiskers array

model. Results summarized on Fig. 6 show that the percentage of successful discrimination quickly rises with the number of channels and reaches values over 80% when 20 channels or 2 arcs at least are concerned. This result confirms previously obtained ones in [15], [16] about whiskers.

5 Discussion

Audition and touch endow spectral processing abilities allowing texture discrimination. Rat whiskers sensory system exhibits, as the cochlea, resonance property decomposing the signal over frequencies. Moreover, there exists strong psychophysical and biological interactions between auditory and somatosensory corteces concerning texture analysis. Inspired by these similarities, this paper proposes a vibrissa tactile model and auditory model sharing exactly the same mechanisms for texture recognition.

Based on gammatone filterbanks, cochlea and whiskers array models aim at reproducing the sensitive cell motion and offer a simple functional view of transduction. Nevertheless important aspects of transduction are omitted. Thus, active perception skills (such as damping adaptation or whisking movements in touch or cochlear adaptation in audition) are not modeled here, as well as mechanoreceptors, viewed as simple peak extractors. More specifically on the whiskers model, the lack of biological records about vibrissae bandwidth is a key problem for having precise parameters values. Moreover, by viewing the whiskers pad as a filterbank, we omit the spatial organization of vibrissae over the face and are only interested in its spectral organization, through the reproduction of the first order resonance frequency of whiskers.

Feature extraction used for texture discrimination is inspired by biological data recorded in the rat barrel cortex. Confirming previously obtained results on elastomer-based artificial whiskers [16], our results (Fig. 3) show that the Af feature allows fine texture discrimination in both artificial and simulated context. This finding is an argument in favor of the so-called kinetic signature hypothesis [8] which stands that each vibrissa encodes a specific signature of the touched surface in term of magnitude and temporal pattern. Nevertheless one can observe on Fig. 3 that the A feature alone have a very similar performance than Af, whereas the mean inter-peak frequency f is not able to fine texture recognition. The amplitude A reproduces the motion of the basilar membrane and the base of a vibrissa. Thus, when a filter resonates, its "motion" becomes more important and this activity is reflected by the A pattern. In the other way, the instantaneous frequency estimation f is not sharply sensitive to this resonance phenomena, as plotted in Fig. 4. From these findings we suggest that, by changing the A pattern over frequencies, the resonance property improves the discrimination skills of the model and thus its global performance. This suggestion is enhanced by the results presented in Fig. 4 and Fig. 6. Likewise, the fact that our results suggest that the texture discrimination capacities depends both on the number of channels and amplitude activity patterns seems to back up the resonance hypothesis [4]. This hypothesis stands that the self resonance

property of the vibrissa plays a crucial role in vibration transduction and helps to enhance texture perception.

The whiskers array and cochlea model used in this paper provide a supramodal representation of the input signal. Although two distinct feature extractors are used in spite of a real crossmodal one as hypothesized in [11], this work is going in the direction of a multimodal integration of low-level fundamental sensory dimensions. Future work will be devoted to the implementation of this supramodal model on a robotic platform [26] in order to investigate crossmodal recognition of vibrissa tactile sensory signal and auditory signal produced by the contact of whiskers on a surface.

This paper shows that gammatone based filters, in spite of the lack of biomechanical precision, are suitable filters for texture discrimination. This kind of functional model may help in tactile transduction and neural spectral processing understanding by proposing a more abstract view of the transduction process and enhancing the resonant property.

Acknowledgment

This work as been funded by the EC Integrated Project ICEA (Integrating Cognition, Emotion and Autonomy), IST-027819-IP.

References

1. Lederman, S.J.: Auditory texture perception. Perception 8, 93–103 (1979)
2. Carvell, G.E., Simons, D.J.: Biometric analyses of vibrissal tactile discrimination in the rat. J. Neurosci. 10(8), 2638–2648 (1990)
3. Brecht, M., Preilowski, B., Merzenich, M.M.: Functional architecture of the mystacial vibrissae. Behav. Brain Res. 84, 81–97 (1997)
4. Neimark, M.A., Andermann, M., Hopfield, J., Moore, C.: Vibrissa resonance as a transduction mechanism for tactile encoding. J. Neurosci. 23(16), 6499–6509 (2003)
5. Hartmann, M.J., Johnson, N.J., Towal, R.B., Assad, C.: Mechanical characteristics of rat vibrissae: Resonant frequencies and damping in isolated whiskers and in the awake behaving animal. J. Neurosci. 23(16), 6510–6519 (2003)
6. Andermann, M., Moore, C.: Mechanical resonance enhances the sensitivity of the vibrissa sensory system to near-threshold stimuli. Brain Res. 1235, 74–81 (2008)
7. Warren, R.M.: Auditory Perception: An Analysis and Synthesis. Cambridge University Press, Cambridge (2008)
8. Arabzadeh, E., Panzeri, S., Diamond, M.E.: Whisker vibration information carried by rat barrel cortex neurons. J. Neurosci. 24(26), 6011–6020 (2004)
9. Foxe, J.J.: Multisensory integration: frequency tuning of audio-tactile integration. Curr. Biol. 19(9), R373–R375 (2009)
10. Yau, J.M., Olenczak, J.B., Dammann, J.F., Bensmaia, S.J.: Temporal frequency channels are linked across audition and touch. Curr. Biol. 19(7), 561–566 (2009)
11. Yau, J.M., Hollins, M., Bensmaia, S.J.: Textural timbre: the perception of surface microtexture depends in part on multimodal spectral cues. Commun. Integr. Biol. 2(4), 1–3 (2009)

12. Patterson, R., Robinson, K., Holdsworth, J., McKeown, D., Zhang, C., Allerhand, M.: Complex sounds and auditoty images. In: Auditory Physiology and Perception, Proc. 9th Int. Symp. on Hearing (1992)
13. Givelberg, E., Bunn, J.: A comprehensive three-dimensional model of the cochlea. J. Comput. Phys. 191(2), 377–391 (2003)
14. Lyon, R.: A computational model of filtering, detection, and compression in the cochlea. In: IEEE Int Conf. of Acoustics, Speech, and Signal Processing, pp. 1282–1285 (1982)
15. Fend, M., Bovet, S., Yokoi, H., Pfeifer, R.: An active artificial whisker array for texture discrimination. In: IROS IEEE/RSJ Int. Conf., pp. 1044–1049 (2003)
16. N'Guyen, S., Pirim, P., Meyer, J.-A.: Tactile texture discrimination in the robot-rat psikharpax. In: BIOSIGNALS 2010, 3rd Int. Conf. on Bio-Inspired Systems and Signal Processing, Valencia, Spain (2010)
17. Pearson, M.J., Pipe, A.G., Melhuish, C., Mitchinson, B., Prescott, T.J.: Whiskerbot: A robotic active touch system modeled on the rat whisker sensory system. Adapt. Behav. 15, 223–240 (2007)
18. Slaney, M.: An efficient implementation of the patterson-holdsworth auditory filter bank. Technical report, Apple Computer Technical Report 35 (1993)
19. Glasberg, B., Moore, B.: Derivation of auditory filter shapes from notched-noise data. Hear. Res. 47, 103–138 (1990)
20. Andermann, M., Ritt, J., Neimark, M.A., Moore, C.: Neural correlates of vibrissa resonance: Band-pass and somatotopic representation of high-frequency stimuli. Neuron 42, 451–463 (2004)
21. Licklider, J.C., Pollack, I.: Effects of differentiation, integration and infinite peak clipping upon the intelligibility of speech. Am. Acoust. Soc. J. 20(1), 42–51 (1948)
22. N'Guyen, S., Pirim, P., Meyer, J.-A.: Elastomer-based tactile sensor array for the artificial rat psikharpax. In: ISEF 2009 - XIV Int. Symp. on Electromagnetic Fields in Mechatronics, Electrical and Electronic Engineering (2009)
23. Petersen, R.S., Diamond, M.E.: Spatial-temporal distribution of whisker-evoked activity in rat somatosensory cortex and the coding of stimulus location. J. Neurosci. 20(16), 6135–6143 (2000)
24. Nissen, S.: Implementation of a fast artificial neural network library. Technical report, Department of Computer Science University of Copenhagen (2003)
25. Igel, C., Hüskel, M.: Improving the rprop learning algorithm. In: Second International Symposium on Neural Computation, pp. 115–121 (2000)
26. Meyer, J.-A., Guillot, A., Girard, B., Khamassi, M., Pirim, P., Berthoz, A.: The Psikharpax project: Towards building an artificial rat. Robot Auton. Syst. 50(4), 211–223 (2005)

Learning to Look in Different Environments: An Active-Vision Model Which Learns and Readapts Visual Routines

Dimitri Ognibene[1], Giovanni Pezzulo[2], and Gianluca Baldassare[1,⋆]

[1] Istituto di Scienze e Tecnologie della Cognizione,
Consiglio Nazionale delle Ricerche,
Via San Martino della Battaglia 44, 00185 Rome, Italy
[2] Istituto di Linguistica Computazionale "Antonio Zampolli",
Consiglio Nazionale delle Ricerche,
Via Giuseppe Moruzzi 1, 56124 Pisa, Italy

Abstract. One of the main claims of the active vision framework is that finding data on the basis of task requirements is more efficient than reconstructing the whole scene by performing a complete visual scan. To be successful, this approach requires that agents learn *visual routines* to direct overt attention to locations with the information needed to accomplish the task. In ecological conditions, learning such visual routines is difficult due to the partial observability of the world, the changes in the environment, and the fact that learning signals might be indirect. This paper uses a reinforcement-learning actor-critic model to study how visual routines can be formed, and then adapted when the environment changes, in a system endowed with a controllable gaze and reaching capabilities. The tests of the model show that: (a) the autonomously-developed visual routines are strongly dependent on the task and the statistical properties of the environment; (b) when the statistics of the environment change, the performance of the system remains rather stable thanks to the re-use of previously discovered visual routines while the visual exploration policy remains for long time sub-optimal. We conclude that the model has a robust behaviour but the acquisition of an optimal visual exploration policy is particularly hard given its complex dependence on statistical properties of the environment, showing another of the difficulties that adaptive active vision agents must face.

1 Introduction

The *information-processing framework* of vision, initiated with Marr's theory [1], describes attention and vision as processes aimed at building 'objective' general-purpose representations of the environment that can be used to guide

⋆ This research was funded by the European Projects *HUMANOBS – Humanoids that Learn Socio-Communicative Skills through Observation*, contract no FP7-ICT-STREP-231453, and *IM-CLeVeR – Intrinsically Motivated Cumulative Learning Versatile Robots*, contract no FP7-ICT-IP-231722.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 199–210, 2010.

any relevant perceptual process and action [2]. This approach has a strong limit in that building representations totally detached from the specific needs of the system produces scene representations with an overwhelming amount of non-needed information. The *active vision approach* [3] proposes instead to use a perceptual system with a highly-sensitive *fovea* to explore the environment and extract only the information needed to pursue current goals. This dramatically reduces visual computations as these can be performed only on limited portions of the scene, similarly to what happens in humans [4].

Given this strategy, the difficulty of performing complex reconstruction of the environment is in great part transferred to the difficulty of controlling the gaze in an efficient manner. Experiments on overt visual attention in ecological conditions show that eye-movement patterns are organised on the basis of task-dependent *visual routines* [5,6]. These are stereotyped sequences of elemental operations related to the accomplishment of specific perceptual or sensorimotor tasks, which involve eye movements and specific processing of foveated scene regions (e.g. contour tracing and location storing). Visual routines are a fundamental concept for active vision as they constitute the means through which visual behaviour actually selects only specific portions of the scenes to perform high cost visual-processing operations. Several behavioural studies, on tasks like face recognition [7] and visuo-motor control [8,9], have shown that human are able to readapt their visual routines for task specific demands.

In ecological conditions, learning visual routines is difficult due to: (a) the many dimensions of the visual space; (b) the partial observability caused by the environment properties and by the presence of a fovea and a periphery with a limited perception; (c) the need to pursue goals in changing conditions; (d) the fact that the mechanisms learning to control the eye can make leverage only on *indirect* rewarding signals, for example related to the effects of manipulation actions and not directly to eye movements.

Notwithstanding the importance of visual routines and these difficulties, most studies on active perception and autonomous robotics have not studied the *processes of learning and re-adaptation* related to them. In this respect, the goal of this work is starting to study in a systematic fashion how visual routines are first *learned* and then *re-adapted* when the environment changes. To this purpose, we use a reinforcement-learning bio-inspired embodied model controlling a camera-arm robot (some components of the architecture were presented in [10] and [11]). The system has some features which make it well suited for this study within a bio-inspired active vision context: (a) it assumes a strong coupling between visual and arm control; (b) has a simplified bottom-up attention component and a rather sophisticated top-down attention component; (c) is based on neural maps which allow the formation of distributed quantitative internal representations easily studied in a graphical way.

Related studies. Previous studies on adaptive active vision have so far focused the previously mentioned topics in isolation. In [12] an artificial fovea is controlled by an adaptive neural controller. Without a teacher, this learns trajectories causing the fovea to find targets in simple visual scenes and to track moving

targets. In [13] a model is proposed which integrates bottom-up and top-down attention processes. The system used Q-learning to find objects located in fixed positions in crowded rooms. Although interesting, the model needs hand-made knowledge of the target appearance and learning is specific for objects in one specific context, so it has to be retrained when objects move or scenes change. In [14] a model is proposed to solve active sensing problems under uncertainty. A reinforcement learning algorithm allows it to develop active sensing strategies to decide which uncertainties to reduce. However, in this study the model of the task is known a-priori and motor control is hardwired. In [15,16,17] evolutionary learning techniques are used for developing adaptive active vision systems. These approaches are robust to the *perceptual aliasing problem*, however they do not allow on-line adaptation to changing environments.

2 The Model

The architecture of the model (Figure 1.a) integrates two components: (a) an attention control component formed by a bottom-up and top-down attention sub-component; (b) an arm control component. These components are based on common bio-inspired computational principles: (a) population codes (here 2D neural maps) used to represent sensorimotor information [18,19]; (b) dynamic neural-field networks used to integrate information and select actions based on neural competitions [20,21]; (c) a progressive development of skills (cf. [11]). We now present an overview of the components and then describe them in detail.

The setup used to test the model is a simulated version of a real system presented in [10] (see Figure 1.a), formed by a down-looking camera and a 2-DOFs robotic arm. The arm horizontal working plane is made up by a computer screen where the task stimuli appear. The input image of the model activates a *periphery map* that implements bottom-up attention. The central part of the input image (*fovea*) feeds a *reinforcement-learning actor-critic* component that learns to predict the position of relevant visual elements based on foveated cues (top-down attention). A leaky-neuron *potential action map (PAM)* integrates in time these predictions. A *saliency map* sums up the information from the periphery map and the PAM and selects the next eye movement corresponding to the most active neurons (neural competition). Each eye fixation point, encoded in a *eye posture map*, suggests a potential arm target to an *arm posture map* which (a) performs the eye posture → arm posture inverse kinematic and (b) implements a second neural competition which triggers reaching movements when the eye fixates the same location for about three consecutive time steps. If the reached target is the correct one (red object), the actor-critic component gets rewarded otherwise it gets slightly punished (as a metaphor of energy consumption).

2.1 Attention Control Components

Periphery Map (Bottom-Up Attention). The input image is a 240×320 pixel RGB image. A 30×40 gray periphery map **pm** is extracted from it: first the

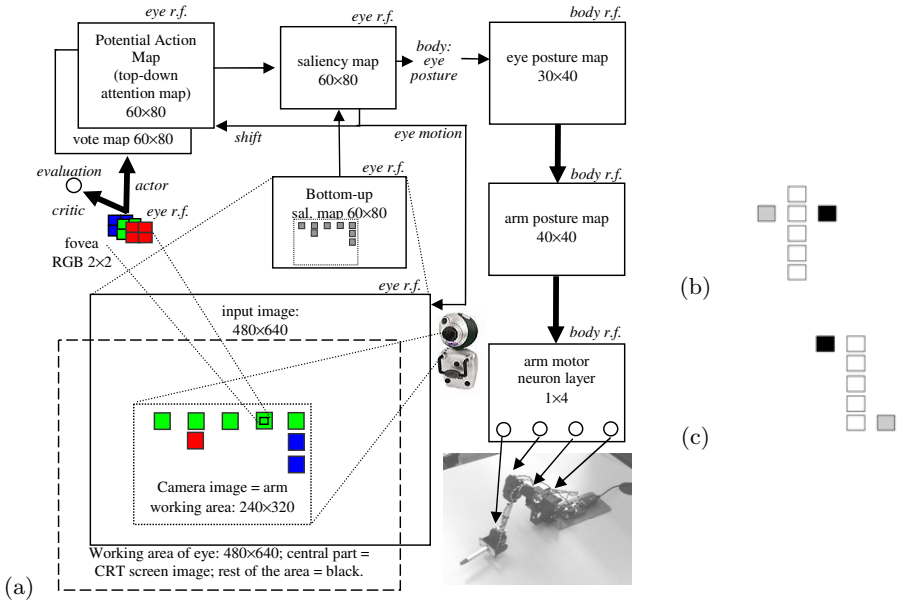


Fig. 1. (a) Eye-arm control architecture. (b) Example of environment from family **L**, with target (gray = red) on the left, a distractor (dark gray = blue) on the right of a line of 2 to 5 cues (light grey = green). (c) Example of environment from family **R**, with switched placement of the target and distractor.

input image is divided into 30×40 blocks of 8×8 pixels each, then the RGB color values of the pixels of each block are averaged to obtain a gray value. As objects (uniformly coloured squares) are shown on a black background, the simple gray image is enough to reveal their presence: a more sophisticated bottom-up saliency (e.g. as that of [22]) is not needed for the purposes of this work.

Actor-Critic Component (Top-Down Attention) The fovea is composed of a 2×2 RGB pixel image (encoded in vector $\mathbf{f}$) extracted from the centre of the input image. The fovea image is fed into two feedforward neural networks forming a reinforcement-learning actor-critic architecture [23], a biologically plausible model of trial-and-error learning in organisms [24,25]. The *critic* is a network with a linear output unit v_t which learns to evaluate the current state on the basis of the expected future discounted rewards. The system gets a reward r_t after the execution of a reaching action, and this, together with v_t, is used to compute the *surprise signal* s_t (or 'TD error', [23]) used to update the critic's weights (vector $\mathbf{w}^c$) and the actor's weights (matrix $\mathbf{W}^a$). The *actor* is a network whose output layer is a *vote map* of 60×80 sigmoid neurons (encoded in vector $\mathbf{vm}$) which signal to the PAM the possible positions of rewarded targets with respect to the currently foveated visual cue ($\gamma = 0.9$; T is the transpose operator):

$$v_t = \mathbf{w}^{cT}\mathbf{f} \quad s_t = (r_t + \gamma\, v_t) - v_{t-1} \tag{1}$$

$$\mathbf{vm} = g\,[\mathbf{W}^a\mathbf{f}] \quad g[x] = 1/(1+e^{-x}) \tag{2}$$

The critic is trained on the basis of s_t, used as error signal, and the input $\mathbf{f}$ [23]. The actor is trained with a Hebb rule involving the activation of the saliency map (vector $\mathbf{sm}_t$), which encodes the last eye displacement (see below), and the input $\mathbf{f}$, so as to increase or decrease the probability of doing the same saccadic movement again on the basis of the surprise signal s_t [10] ($\eta^c = 10^{-7}$, $\eta^a = 10^{-5}$; $\bullet$ is the entrywise product operator):

$$\mathbf{w}^c_{t+1} = \mathbf{w}^c_t + \eta^c \, s_t \, \mathbf{f}_t \tag{3}$$

$$\mathbf{W}^a_{t+1} = \mathbf{W}^a_t + \eta^a \, s_t \, \mathbf{sm}_t \bullet (\mathbf{vm}_t \bullet (1 - \mathbf{vm}_t)) \, \mathbf{f}^T_t \tag{4}$$

Potential Action Map (Top-Down Attention Memory). The PAM is formed by 60×80 leaky neurons (vector **pam**) and accumulates evidence, furnished by the vote map **vm** via topological connections, on the possible positions of rewarded targets. Importantly, during each saccade the map activation is shifted in the direction opposite to the eye motion to maintain eye-centred representations (as it might happen in real organisms, see [26]). The PAM is reset each time the input image changes (also this might happen in real organisms [27]).

Saliency Map. The 60×80 saliency map (encoded in vector **sm**) selects saccade movements on the basis of the sum of the topological input signals **pm** and **pam**. The saccade movement is selected by first identifying the unit with the maximum activation and then by activating the map with a Gaussian population code centred on it (the Gaussian function has a width $\sigma = 1$). The eye movement is the average of the winning neurons' preferred eye displacement $(\Delta x, \Delta y)$. This selection mechanism, based on the maximum function, is a computationally fast approximation of a neural dynamic competition process (e.g., cf. [21]).

Biology. Empirical evidence indicates that the cortical area of the frontal eye field (FEF) exhibits properties similar to those of the saliency map integrating bottom-up and top-down information to drive overt and covert attention [28]. Another possible location for this integration is the posterior parietal cortex [29]. Bottom-up (pre-attentive) saliency processes take place in a parallel in relation to various aspects of the retina image such as color, orientation, and motion, and the resulting information is then integrated at higher levels such as the FEF and the parietal cortex. These processes are performed with increasing abstraction in the retina, the lateral geniculate nucleus, the visual cortex, and the extrastriate visual cortex. The top down influence on attention control mainly originates from prefrontal cortex based on the subject's goals and motivations and the environment context. One type of top-down influence reaching FEF neurons is related to a *template* of the target to which attention must be allocated. Other type is related to the spatial relationship between objects that human can acquire even unconsciously as shown in experiment on contextual cueing [30]. This kind of knowledge might be encoded in the hippocampal system. Neurobiologial data on how and where saliency maps can be implemented in the brain can be found in [31,32,29]. The trial-and-error learning processes performed by the model might correspond to the processes taking place in the portions of the basal ganglia dedicated to the control of the eye (striatum and substantia nigra pars reticulata [33]).

2.2 Arm Control Components

As the paper is focussed on attention processes, we now illustrate the main aspects of the arm control components and refer the reader to [11] for details.

Eye Posture Map. This 30×40 neuron map encodes the current eye posture as a Gaussian population code (encoded in vector **emp**; $\sigma = 0.3$).

Arm Posture Map. This is a 40×40 map (vector **apm**) which represents the output layer of a neural network whose weights ($\mathbf{W}^{apmk}$) are pre-trained with a Kohonen algorithm to encode the arm postures in a 2D space. During tests, a neural neural competition [21] takes place in the map (similarly to what happens in real organisms [20]), selects a target for reaching actions, and triggers them when any neuron achieves a certain threshold.

Arm Posture Readout Layer. This is a layer of four sigmoid neurons (vector **aprl**) that encode the desired arm joint angles issued to the arm simulated servos. The map is activated by the arm posture map through connection weights encoded in the matrix $\mathbf{W}^{aprl}$.

Training. The arm components were trained before the experiments illustrated in Section 3. This pre-training is divided in three succeeding learning phases based on random movements of the arm (*motor babbling*). In these phases the system: (a) performs a vector quantization of postures within the arm posture map on the basis of the Kohonen algorithm; (b) learns with a delta rule the inverse kinematic mapping ($\mathbf{W}^{apm}$) between the gaze direction corresponding to the seen hand (**epm**) and the corresponding arm posture encoded by the Kohonen map (**apm**) on the basis of $\mathbf{W}^{apmk}$; (c) trains the arm posture readout map ($\mathbf{W}^{aprl}$) with a delta rule.

3 Experimental Setup

In a previous work we showed that the system was able to learn in few trials to interact with several environments sharing an underling structure, like an simplified tree where the target was an apple always under the foliages. To test how the architecture behaves when exposed to environments based on contrasting rules a task was designed where the environment was randomly selected from one of 2 'families' of environments in every trial. We wanted also to study what could happen when the system has to update its knowledge to tackle a new context for which it is good but not optimal. To test this, the two families were presented with unbalanced frequencies and then these frequencies were switched.

The objects of both families of environments were uniformly-coloured red/green/blue rectangles with width $2.96cm$ and height $2.94cm$ and were placed on the vertexes of a 5×6 grid spaced $5.2cm$ and $3.8cm$ apart. In the family **L** (which stands for 'Left'), used in 75% of trials, the red *target* was on the left of a 'line' formed by 2 to 5 green *cues* (randomly positioned on one of the 2nd, 3rd, 4th or 5th column of vertexes). A blue *distractor* was set on the right side of the cue line (see Figure 1.b). In the family **R** ('Right'), presented in 25% of trials, the placement of the target and distractor was switched with respect to the cue line (see Figure 1.c). The attentional system was first trained for 60,000 steps in this

condition, and then for further 60,000 steps with the frequency of presentation of environments of family **L** and **R** switched to respectively 75% and 25%.

4 Results

Figure 2.a shows the average reward during learning, measured for the two different families of environments separately. The performance with **L** environments increases quickly and then reaches a stable steady state whereas for **R** environments increases slower and shows oscillations.

Figure 2.b shows the evolution of the average number of saccades per trial during learning divided for the two different families of environments. Learning time looks similar for the two families with some advantages for **L**, but **R** shows oscillations. The final average number of saccades is 5.86 for **L** and 7.08 for **R**. Both plots reach a steady level, so the final different number of saccades with **L** in comparison to **R** (approximately one saccade less on average) does not reflect a different level of skill but rather a different strategy. In this respect, the analysis of the behaviour of the system shows that it learns an exploration policy that initially assumes to tackle an environment from family **L**. In the presence of an environment from family **R** this assumption fails, and the system looks directly on the other side of the array rather than exploring the cues again. This strategy allows the system to solve the task with only one additional step.

Figure 3 presents an analysis of visual routines. In particular, Figure 3.a shows the most frequent sequences (during learning) of the first three 'eye actions' (saccade towards left, saccade towards right, saccade on the current object column) per trial. Figure 3.c shows the evolution of the frequencies of the same sequences during learning. The most frequent action sequence is 'left, stay, stay', and is the sequence the model uses when exposed to environments of family **L**. The second most frequent sequence is 'stay, stay, stay'. This sequence is the most selected in the first trials, when the system oscillates between cues, whereas its frequency

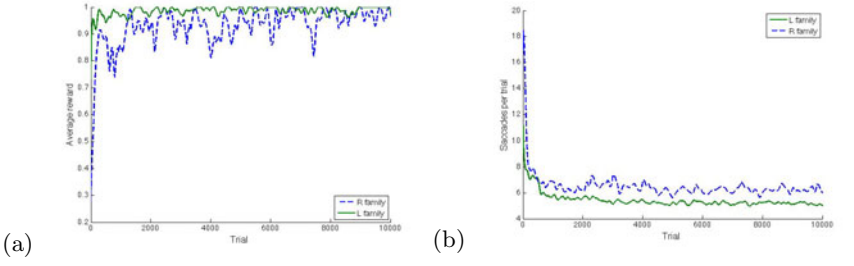


Fig. 2. (a) Average reward of the system (y-axis) during learning (x-axis) measured separately for the two families of environments. Using a moving window of 100 trials, the trials inside the window were separated in two groups corresponding to the two families and the total reward taken for each group was divided by the number of trials in each group. (b) Average number of saccades per trial (y-axis) during learning (x-axis) measured separately for the two families of environments.

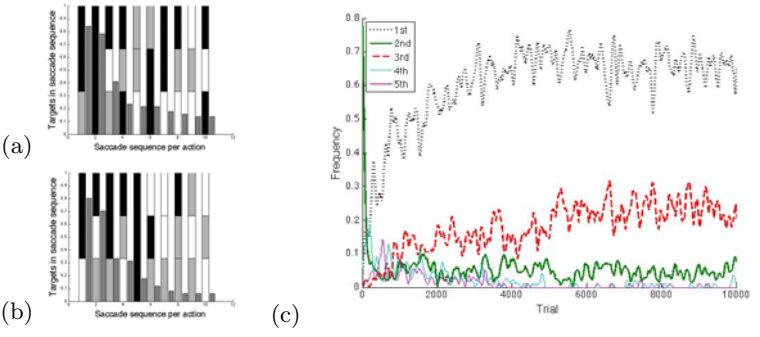


Fig. 3. (a,b) Different eye-action sequences (segmented histogram bars: only three actions per sequence are considered) with the ten highest frequencies (gray bars) exhibited by the system during learning. For each sequence, the colours of the bars represent three different possible eye-actions: saccades towards left (gray), towards right (white), and on the current column (black). Note that the first action of each new trial/scene was removed when the agent gazed the background, and kept when it gazed an object, as the first action depended on the random starting position of the eye. (c) Evolution of eye movement sequences during learning. The plots are sorted by the max frequency they had during learning. Only the first five sequences are shown.

lowers as soon as the system learns to inhibit the entire column of cues, and is then used only when the system reaches the target to trigger the arm movement. The third most frequent sequence, 'left, right, stay', has the same first action of the most frequent sequence but then takes the gaze back to the right and then stops: this is the sequence used by the system to solve an environment of family **R**. This sequence is the last to be learnt, and together with the first and second sequences covers about 90% of all sequences. Overall, this strategy shows that the system initially acts as if it were in an **L** environment, and when it collects information contrary to this assumption it goes straight from the distractor to the target. The other sequences reach a relatively high frequency after the system learns to inhibit the column of cues, and the system has not yet discovered the relationship between the cues and the target, but then they get a rather low frequency after the system has learned the whole strategy.

This policy, acquired by the model to tackle the two environment families is not a composition of the two policies which the system acquires to tackle each family separately. In fact, if a simulation is run with only one of the two families, the family **L** and the family **R** lead to the use of respectively the first sequence of Figure 3.a and the first sequence of Figure 3.b with a frequency, in each case, of about 90%. The internal representations of stimuli acquired by the system are not a simple combination of the representations of the strategies for the two families of environments. Figures 4.a,b show the activation patterns of the vote map when the system foveates a cue or a distractor after it has been trained with environments chosen from families **L** and **R** with a frequency of respectively 75% and 25%. Figures 4.c,d show the activation patterns of the vote map after the system has been trained with environments drawn from only family **L** (the patterns after

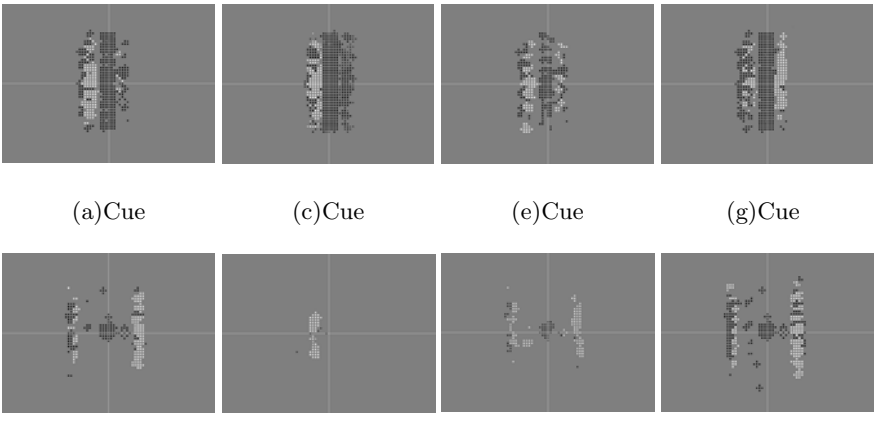


Fig. 4. (a,b) Vote map activation patterns in correspondence to a cue or a distractor after 60,000 training steps with environments of both families. (c,d) Vote map activation patterns in correspondence to a cue or a distractor after 60,000 training steps with environments of family **L**. (e,f) Vote map activation patterns in correspondence to a cue or a distractor after 10,000 training steps with environments of both families. (g,h) Vote maps on the stimuli of the model after 60,000 training step when exposed to both families of environments after changing frequency of family **L** to 25% and frequency of family **R** to 75%.

training with only family **R** have a mirror structure with respect to the vertical axis). Figures 4.a,b show that the patterns developed with both families is not a combination of the maps obtained with the separate training with each family as such combination would have been something like an average of topologically corresponding locations of the patterns related to the two families learned singularly. Rather, it contains features which allow the system to express the visual routines described above. This is shown especially by the activation pattern related to the distractor when the system is trained with only a family. In such pattern, contrary to what happens with the training with both families, there is no coding for the position of the target but rather for a movement back to the cue.

The structure of the vote-map activation patterns can also help clarifying the oscillations of performance observed with family **R** and shown in Figure 2.a. Considering the vote map activation patterns built in the first learning phase (Figure 4.e,f), the final vote map activation patterns (Figure 4.a,b), and the vote map activation patterns developed when the system is exposed only to one family (Figure 4.c,d), it is apparent that the most frequent task interferes with the exploration of the less frequent one during learning (see also Figure 2.b).

4.1 Readapting to New Environment Statistics

After 60,000 steps of training with the families **L** and **R** at respectively 75% and 25%, the two frequencies were switched. As we have seen, at this point the

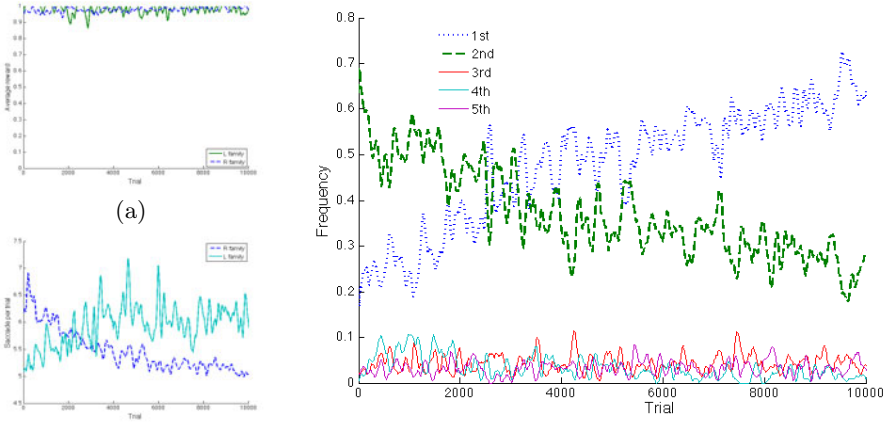


Fig. 5. (a) Average reward during learning after changing the frequency of family **L** and **R** to respectively 25% and 75%. (b) Average number of saccades during learning after changing the frequency of family **L** and **R** to respectively 25% and 75%. (c) Frequencies of the five most frequent saccade sequences (first 3 saccades per trial) during learning, after changing the frequency of family **L** and **R** to respectively 25% and 75%.

system has already stabilised on its maximum level of performance. Switching the frequencies of the two tasks does not substantially affect the performance, as shown in Figure 5.a. The performance with family **L** has some decrease whereas the performance with family **R** has a little increment, but after a transient phase performance with both families is again at about the same maximum level and the performance oscillations with family **R** are even reduced.

Interestingly, Figure 5.b shows that after about 2,000 trials the system becomes faster in solving the tasks of family **R** than those of family **L**, so following the switch in frequency of the two families. The nature of this policy shift is revealed by Figure 5.c. This indicates that after switching the former most frequent sequence, 'left, stay, stay', progressively decreases to a very low level. The optimal policy for family **R**, 'right, stay, stay', steadily increases until becomes the new most frequent sequence. The sequence 'right, left, stay' appears slowly: this is the 'back-up' sequence for the new infrequent environments of family **L** which allows going directly to the target after finding the distractor.

The re-adaptation of the policy is quite slow because the difference in performance is rather stable after the shift, and this implies a small error signal for the actor. In particular, right after the shift the distractor (now encountered often) is a predictor of the target, so the value that the critic assigns to it is close to that of the target and the corresponding error is low. Moreover, when the switch takes place the connection weights are already rather high but often with a sign opposite to the required one (compare Figure 4.a,b and Figure 4.g,h).

5 Conclusions

This work used a bio-inspired reinforcement-learning model for controlling an eye-arm system to study learning and readaptation of *visual routines*. The model was tested with two contrasting tasks which posed aliasing problems and were presented with different frequencies. The model solved the two tasks by developing visual routines dependent on both on the structure and frequency of the tasks. The analysis of the internal representations revealed how the system organised to support such visual routines. Interestingly, the system starts exploring the environment on the basis of the expectation of tackling the most frequent task. If evidence is collected against this expectation, the system changes it and solves the less frequent task with only one additional saccade. The analysis of representations also indicates that the system has learned the spatial (stochastic) relationships between the elements of the objects useful for finding the task-dependent information within the scene. When after learning the frequency of presentation of the environments is switched the performance of the system does not substantially decrease thanks to the re-use of the previously acquired visual routines, then slowly improved on the basis of the reward feedback.

Overall, the study shows that when *active vision* systems have *learning capabilities* the discovery and re-adaptation of visual routines generates a number of very interesting phenomena such as the ability to tackle several environments at the same time, and the dependency of the optimal visual exploration policy on the statistical properties of the environment.

References

1. Marr, D.: Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. Freeman, New York (1982)
2. Fermuller, C., Aloimonos, Y.: Vision and action. Image Vision Comput. 13(10), 725–744 (1995)
3. Ballard, D.: Animate vision. AI 48, 57–86 (1991)
4. Treisman, A.M., Gelade, G.: A feature-integration theory of attention. Cognit. Psychol. 12(1), 97–136 (1980)
5. Ullman, S.: Visual routines. Cognition 18(1-3), 97–159 (1984)
6. Hayhoe, M.: Vision using routines: A functional account of vision. Visual Cognition 7(1-2-3), 43–64 (2000)
7. Heisz, J.J., Shore, D.I.: More efficient scanning for familiar faces. J. Vis. 8(1), 1–10 (2008)
8. Sailer, U., Flanagan, J.R., Johansson, R.S.: Eye-hand coordination during learning of a novel visuomotor task. J. Neurosci. 25(39), 8833–8842 (2005)
9. Land, M.F.: Eye movements and the control of actions in everyday life. Prog. Retin. Eye Res. 25(3), 296–324 (2006)
10. Ognibene, D., Balkenius, C., Baldassarre, G.: Integrating epistemic action (active vision) and pragmatic action (reaching): A neural architecture for camera-arm robots. In: Asada, M., Hallam, J.C.T., Meyer, J.-A., Tani, J. (eds.) SAB 2008. LNCS (LNAI), vol. 5040, pp. 220–229. Springer, Heidelberg (2008)

11. Ognibene, D., Rega, A., Baldassarre, G.: A model of reaching that integrates reinforcement learning and population encoding of postures. In: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J., Marocco, D., Meyer, J.-A., Miglino, O., Parisi, D. (eds.) From Animals to Animats 9: Proceedings of the Ninth International Conference on the Simulation of Adaptive Behavior (SAB 2006), September 2006, pp. 381–393. Springer, Heidelberg (2006)
12. Schmidhuber, J., Huber, R.: Learning to generate artificial fovea trajectories for target detection. Int. J. Neural Syst. 2(1-2), 135–141 (1991)
13. Minut, S., Mahadevan, S.: A reinforcement learning model of selective visual attention. In: AGENTS 2001: Proceedings of the Fifth International Conference on Autonomous Agents, pp. 457–464. ACM, New York (2001)
14. Kwok, C., Fox, D.: Reinforcement learning for sensing strategies. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2004 (2004)
15. de Croon, G.: Adaptive Active Vision. PhD thesis, Universiteit Maastricht (2008)
16. Suzuki, M., Floreano, D.: Enactive robot vision. Adapt. Behav. 16(2-3), 122–128 (2008)
17. Mirolli, M., Ferrauto, T., Nolfi, S.: Categorisation through evidence accumulation in an active vision system. Connection Science (subm)
18. Pouget, A., Ducom, J.C., Torri, J., Bavelier, D.: Multisensory spatial representations in eye-centered coordinates for reaching. Cognition 83(1), B1–B11 (2002)
19. Pouget, A., Zhang, K., Deneve, S., Latham, P.E.: Statistically efficient estimation using population coding. Neural Comput. 10(2), 373–401 (1998)
20. Cisek, P.: Integrated neural processes for defining potential actions and deciding between them: a computational model. J. Neurosci. 26, 9761–9770 (2006)
21. Erlhagen, W., Schöner, G.: Dynamic field theory of movement preparation. Psychol. Rev. 109(3), 545–572 (2002)
22. Itti, L., Koch, C.: Feature combination strategies for saliency-based visual attention systems. Journal of Electronic Imaging 10(1), 161–169 (2001)
23. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
24. Houk, J., Adams, J., Barto, A.: A model of how the basal ganglia generate and use neural signals that predict reinforcement, pp. 249–270 (1995)
25. Mannella, F., Baldassarre, G.: A neural-network reinforcement-learning model of domestic chicks that learn to localise the centre of closed arenas. Phil. Trans. R. Soc. B 362(383-401), 333–356 (2007)
26. Dominey, P.F., Arbib, M.A.: A cortico-subcortical model for generation of spatially accurate sequential saccades. Cereb. Cortex. 2(2), 153–175 (1992)
27. Klein: Inhibition of return. Trends Cogn. Sci. 4(4), 138–147 (2000)
28. Thompson, K.G., Bichot, N.P.: A visual salience map in the primate frontal eye field. Prog. Brain Res. 147, 251–262 (2005)
29. Treue, S.: Visual attention: the where, what, how and why of saliency. Curr. Opin. Neurobiol. 13(4), 428–432 (2003)
30. Chun: Contextual cueing of visual attention. Trends Cogn. Sci. 4(5), 170–178 (2000)
31. Silver, M.A., Kastner, S.: Topographic maps in human frontal and parietal cortex. Trends Cogn. Sci. 13(11), 488–495 (2009)
32. Cutsuridis, V.: A cognitive model of saliency, attention, and picture scanning. Cognitive Computation 1, 292–299 (2009)
33. Hikosaka, O., Takikawa, Y., Kawagoe, R.: Role of the basal ganglia in the control of purposive saccadic eye movements. Physiol. Rev. 80(3), 953–978 (2000)

Estimating Relative Positions of Multiple Objects in the Weakly Electric Fish

Miyoung Sim and DaeEun Kim

Biological Cybernetics Lab
School of Electrical and Electronic Engineering,
Younsei University,
Schinchon, Seoul, 120-749,
Corea (South Korea)
{simmi,daeeun}@yonsei.ac.kr
http://cog.yonsei.ac.kr

Abstract. The weakly electric fish have a specialized sensor system of electrolocation. They use the electric field to identify the location of a target object. When the object is near the electric fish, the distortion of electric field is observed and this distortion draws a bell-shaped curve along electroreceptors on the whole surface of weakly electric fish. The ratio of maximal slope to maximal amplitude in this electric image is called relative slope. It is already known that the relative slope can be a distance measure of a target object. However, if there are more than one object, a bell-shaped electric image for each object is superposed. So it is difficult to extract the relative slope information for each object. It means that the location of each target object can hardly be estimated among background objects. In this paper, we provide a new mechanism to figure out the position of a target object, based on the spatiotemporal information of electrosenses.

Keywords: weakly electric fish, electrosensory system, electrolocation, relative slope, distance measure, spatiotemporal sensing.

1 Introduction

Weakly electric fishes have a specialized electrolocation system to identify a target object, explore in their surroundings, and communicate with conspecifics. The weakly electric fish generates the electric field and detect the distortion of the electric field. Readings of many electroreceptors generate the electric image. Electric images give us the information of how weakly electric fish find preys or recognize the environment. They have tuberous receptors on the whole skin surface, and the sensor readings are recorded as the intensity of electric potential dependent on the distance of a target object from the receptor.

Sensory systems can be divided into two types, passive and active sensory systems. The electrosensory system of the weakly electric fish belongs to active sensory system, because they use their electric organ (EO) as a source of energy.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 211–220, 2010.

Weakly electric fish emit self-generated electric field and detect the reflected signal from any object. The bell-shaped curve of electric potential influenced by an object can be observed with a lot of electroreceptors on the skin surface along the rostrocaudal line. This electric signal possesses the information to localize the target object. The localization of the target is a fundamental and necessary function of animals to capture preys and avoid their predators. Weakly electric fish use electrolocation to localize the target with electric image. In the three-dimensional space, weakly electric fish have to identify rostrocaudal (from head to tail), lateral (side direction), and dorsoventral (from dorsal to ventral) position of a target object.

A weakly electric fish has about 14,000 'tuberous electroreceptors' and readings of these electroreceptors produce an electric image [1]. The stimulus intensity is inversely proportional to the distance of an object from the fish. When we see the sensor readings of electroreceptors along the rostrocaudal axis, the position of maximum amplitude is the same as the rostrocaudal position of an object. However, identifying the lateral distance of a target object is not simple. The distance from the midline of a fish, the conductivity and size of a target object, all of them influence the electric image. There have been researches to observe changes of electric field on the skin surface along the rostrocaudal line and it is demonstrated that the ratio of the maximal slope to maximal amplitude is dependent on the distance of a target object but has no change for varying sizes of the object [2,3,4]. It suggests a possible distance measure with electrosenses of weakly electric fish [5].

Chen et al. [6] showed an experiment with a moving object near the body surface of weakly electric fish. Stimulus due to a moving object has been recorded at a specific sensory position on the surface of the weakly electric fish with a given time span. In this case, the time spot with the maximal intensity indicates the time when the distance between the target object and the sensor is shortest. Also the full-width at half-maximum (FWHM) can be used as a distance measure, irrespective of the conductivity and size of a target object.

The relative slope and FWHM have different measure styles; the one is related to a spatial distribution of sensor readings, and the other has temporal property. However, both measures take sensor readings for a target object at varying positions of electroreceptors. Weakly electric fish swim back and forth frequently when they capture their prey [7]. We suggest that weakly electric fish can take an advantage of spatiotemporal information acquired from a distribution of electrosenses with their active movements. In this paper, we will see the effect of the back and forth swimming with spatiotemporal information when there are more than one object near the fish. It is expected the combination of the spatial and temporal information processing can give us a hint about how weakly electric fish discriminate a target object from background objects. We introduce particular features acquired from the spatiotemporal pattern of electric image and further explain the mechanism of how weakly electric fish identify target objects in complex environment. This study can provide us with a possible localization mechanism of an aquatic vehicle in the underwater.

2 Method

2.1 Modeling Electric Field

A weakly electric fish has an electric organ (EO) transformed from muscle and nerve cells [8,9]. We study the electroreception mechanism of the electric fish *Apteronotus albifrons*. The electric field is produced by the EO and the EO is modeled as a set of positive and negative poles [10,6]. The simulation of electrosenses can gives a good illustration about the mechanism of how weakly electric fish use their electrolocation to identify a target object although the measured values of a sensor is extremely small to obtain clean electric image [11].

The EO is composed of n positive poles and one negative pole and the sum of charges of electric poles are zero. These are distributed along the mid-line axis of the weakly electric fish with a fixed interval. The electric potential, $V(\boldsymbol{x})$, and electric field, $E(\boldsymbol{x})$, measured at $\boldsymbol{x}$ are derived, respectively as,

$$V(\boldsymbol{x}) = \sum_{i=1}^{n} \frac{q/n}{|\boldsymbol{x} - \boldsymbol{x}_P^i|} - \frac{q}{|\boldsymbol{x} - \boldsymbol{x}_N|} \tag{1}$$

$$E(\boldsymbol{x}) = -\nabla V(\boldsymbol{x}) = \sum_{i=1}^{n} \frac{q/n}{|\boldsymbol{x} - \boldsymbol{x}_P^i|^3}(\boldsymbol{x} - \boldsymbol{x}_P^i) - \frac{q}{|\boldsymbol{x} - \boldsymbol{x}_N|^3}(\boldsymbol{x} - \boldsymbol{x}_N) \tag{2}$$

where $\boldsymbol{x}_P^i$ is the position of the i-th positive pole, $\boldsymbol{x}_N$ the position of one negative pole, and q is the normalized magnitude of electric charge that has generally the value from $8mV$ to $20mV$ [6]. To consider the normal component for an electroreceptor, the transdermal potential difference $V_{td}(\boldsymbol{x}_s)$ is calculated as,

$$V_{td}(\boldsymbol{x}) = E(\boldsymbol{x}) \cdot \hat{n}(\boldsymbol{x})\rho \tag{3}$$

where $\hat{n}(\boldsymbol{x})$ is the normal vector at the measured point, and ρ is the ratio of resistivity of the surface of the fish and water, ρ_{skin}/ρ_{water}. The distortion of the electric field due to a spherical object at $\boldsymbol{x}$ can be derived [10,6] as,

$$\Delta V(\boldsymbol{x}) = \chi \frac{r^3 E(\boldsymbol{x}_{obj}) \cdot (\boldsymbol{x} - \boldsymbol{x}_{obj})}{|\boldsymbol{x} - \boldsymbol{x}_{obj}|^3} \tag{4}$$

where $E(\boldsymbol{x}_{obj})$ is the electric field at the center position, r the radius of a sphere, and χ is the electrical contrast that ranges from 1 at a perfect conductor to -0.5 at a perfect insulator [6].

Weakly electric fish have a lot of electroreceptors over the surface of the body and in this paper we consider a group of electroreceptors distributed along the rostrocaudal axis. The collection of stimulus at each electroreceptor forms a spatial electric image (one-dimensional curve). The electroreceptor reads the distortion of transdermal potential caused by an object.

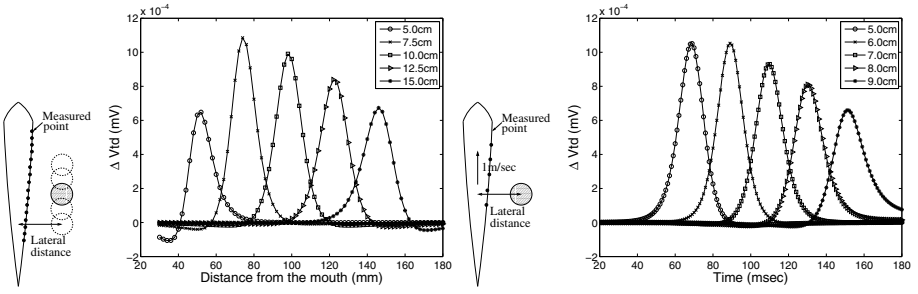


Fig. 1. Electric images with fixed lateral distance and size of a target object (a) spatial sensor readings with varying rostrocaudal positions of a target object from the mouth (b) temporal sensor readings with varying sensor positions from the mouth (the velocity of the weakly electric fish is $1m/sec$)

2.2 Electrolocation and Relative Slope

Weakly electric fish use electrolocation to localize a target object. Fig. 1 (a) shows sensor readings of electroreceptors on the skin surface. It displays spatial electric images when the lateral distance and size of a target object are fixed. The maximal amplitude depends on the relative rostrocaudal and lateral position as well as the size and conductivity of the target object. It is noted that the rostrocaudal position of the target object can be found in the electric image with the position of the maximum intensity. For spatial electric image (Fig. 1 (a)), the position with maximum intensity matches the rostrocaudal position of the target object from the mouth of a fish. We also simulate the sensor value of a specific electroreceptor when the fish moves forward. For temporal electric image - see Fig. 1 (b), which shows sensor readings of an electroreceptor in the time course, the maximal amplitude can be observed at the time when the distance between the target object and the measured point is shortest. However, the lateral distance is not directly extracted in the electric images for both experiments, unlike the rostrocaudal position of a target object.

When there is only one target object near the weakly electric fish, the localization is simple. The rostrocaudal position of a target object is identified by the position of a maximal amplitude. The relative slope and FWHM are the measure of the lateral distance of a target object [2,3,4,5,6]. The relative slope is the ratio of the maximum slope to the maximum amplitude in the curve of sensor readings. Eventually the relative slope can be applied to both spatial sensor readings and temporal sensor readings in Fig. 1. However, it may not be suitable to estimate the distances with relative slope when there are multiple neighboring objects close each other. Still unknown is the mechanism to extract the distance infomation of a target object when there exist background signals or more than one target object.

Assume there are n electroreceptors at $x_1, x_2, ..., x_n$ on the skin surface along the mid-line rostrocaudal axis. Each electroreceptor measures the transdermal

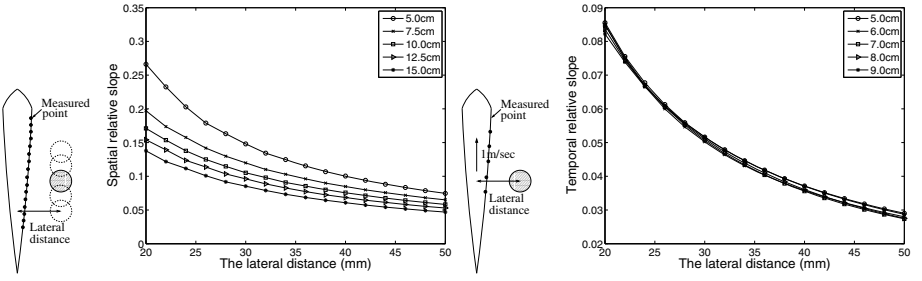


Fig. 2. Relative slopes with the lateral distance of the target object from $2cm$ to $5cm$ (a) spatial relative slope with varying rostrocaudal positions from the mouth (b) temporal relative slope with varying sensor positions from the mouth (the velocity of the weakly electric fish is $1m/sec$)

potential difference at a given position x_i in a given time t, that is, $I(x_i, t)$. We define spatial relative slope as relative slope with respect to the spatial axis, that is, $max_i \{I(x_{i+1}, t) - I(x_i, t)\}/max_i \{I(x_i, t)\}$.

Similarly we also define temporal relative slope as relative slope with respect to the temporal axis, that is, $\max_k \{I(x, t_{k+1}) - I(x, t_k)\}/\max_k \{I(x, t_k)\}$.

3 Experiments

3.1 Relative Slope

Fig. 2 (a) shows spatial relative slopes for varying rostrocaudal positions of a target object. When the rostrocaudal position of the object is changed, the relative slope curve changes. The variation of relative slope due to the rostrocaudal position is displayed in Fig. 3. If the position is closer to the tail, the slope curve moves to the lower level. Relative slopes are significantly affected by the rostrocaudal position of the object. Thus, to estimate the distance of an object, we need to record all the set of relative slope curves for varying rostrocaudal positions.

Now consider the temporal relative slope at one specific electroreceptor. How are temporal relative slopes changed when the stimulus-recording position is changed? Fig. 2 (b) shows relative slope curves at each measured point of sensors when the fish swims forward. We observe almost no change of relative slope pattern (see Fig. 3), since the sensors follow nearly the same track of body movement with a static target object.

In fact, electroreceptors are not uniformly distributed and the spatial relative slope can be affected by this distribution. In contrast, the temporal relative slope with a fish's movements seems to be independent of the relative rostrocaudal position of the target object. When the weakly electric fish swims forward or the target object moves along the rostrocaudal line, the recording of a sensor depends only on the relative lateral distance. Measuring the sensor readings even

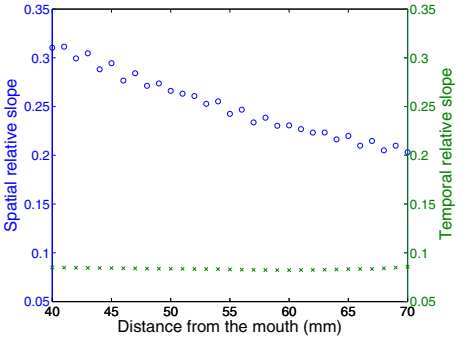


Fig. 3. Relative slopes with fixed lateral distance and size of the target object (o: spatial relative slope with varying rostrocaudal positions of the target object from the mouth, x: temporal relative slope with a static target object and varying sensor positions from the mouth)

at different sensor positions can only slightly influence the temporal relative slope. This property is useful to detect the lateral distance of a target object. We will see whether or not this property is still valid for multiple objects.

3.2 Electric Image with More Than One Object

From the back and forth swimming of a weakly electric fish, we can obtain the spatiotemporal information of an electric image. From this spatiotemporal information, we suggest the mechanism of identifying positions of multiple objects with the relative slopes. We consider differentiating electric image at spatial and temporal axis.

We can see the intensity difference of neighboring electroreceptors, $I(x_{i+1}, t) - I(x_i, t)$, along the spatial axis (see Fig. 4 (b)) or along the temporal axis (see Fig. 4 (c)). For both cases, we can easily find two peak amplitudes each of which corresponds to an object, and say two different objects are detectable with the electrosenses.

Similar to the above procedure, the relative difference of sensor intensity between neighboring time spots can be calculated, that is, $I(x, t_{k+1}) - I(x, t_k)$, - see Fig. 4 (d). It corresponds to the temporal slope, if $|t_{k+1} - t_k|$ is small enough. The temporal slope can be measured along the spatial axis and the temporal axis, respectively. Those examples are displayed in Fig. 4 (e)-(f).

Here, we consider four cases for spatiotemporal information processing, spatial slope along the spatial axis and the temporal axis, and also temporal slope along the spatial axis and the temporal axis. This idea can be extended into the relative slope concept with respect to the spatial axis and time axis, which requires calculating the ratio of maximum slope to maximal amplitude. For each maximal slope, the local maximal value of an electric image can be found and thus we can obtain spatial relative slopes and temporal relative slopes in the local

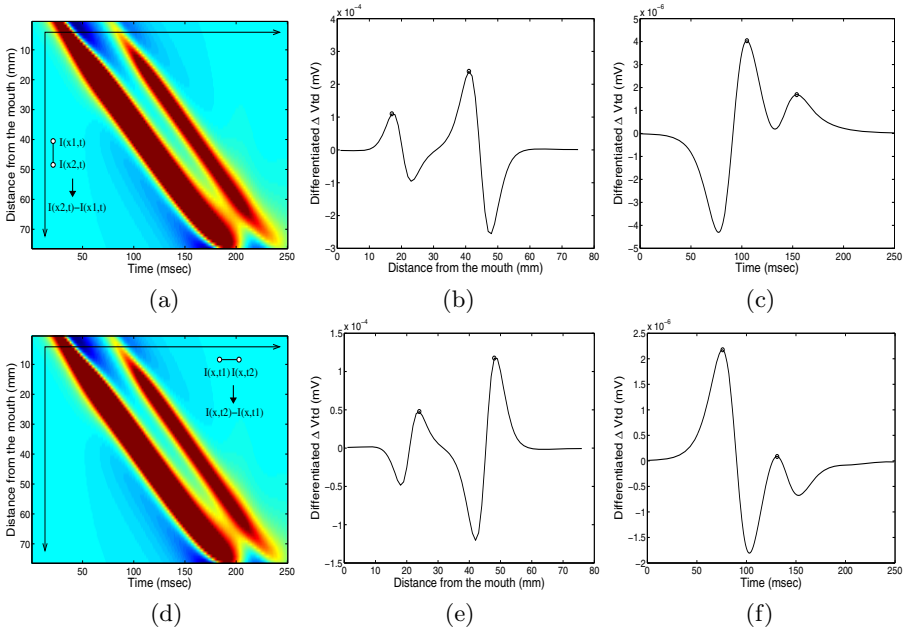


Fig. 4. Differentiation of electric images over two objects (a) spatial slope diagram (b) intensity difference between neighboring sensors along the spatial axis (c) along the time axis (d) temporal slope diagram (e) temporal intensity difference along the spatial axis (f) along the time axis (two objects whose radius is $0.8cm$, $1.2cm$ are located at $7cm$, $12cm$ from the mouth, respectively)

zone of time and space. Each local zone is assigned for one object. Localization of each object can be obtained through the relative slope in the local zone.

Fig. 5 (a)-(b) show the spatial slope, that is, differentiation with respect to rostrocaudal position, along the spatial axis and the temporal axis, respectively. For each case, the relative slope is calculated with the maximum amplitude along its reference axis. Fig. 5 (a) is equivalent to the spatial relative slope. We can redefine the spatial relative slope as a space-to-spatial slope.

$$Space\text{–}to\text{–}spatial\ slope = \frac{\max_i \{I(x_{i+1}, t_k) - I(x_i, t_k)\}}{\max_i \{I(x_i, t_k)\}} \tag{5}$$

Fig. 5 (b) shows the relative slope curves when a pair of sensor positions change along the rostrocaudal line of a fish with two static target objects. The spatial slope along the temporal axis can be defined as a time-to-spatial slope.

$$Time\text{–}to\text{–}spatial\ slope = \frac{\max_k \{I(x_{i+1}, t_k) - I(x_i, t_k)\}}{\max_k \{I(x_i, t_k)\}} \tag{6}$$

The relative slopes for small and large objects are closely overlapped. It implies that the relative slopes can be used for distance measure, irrespective of the size

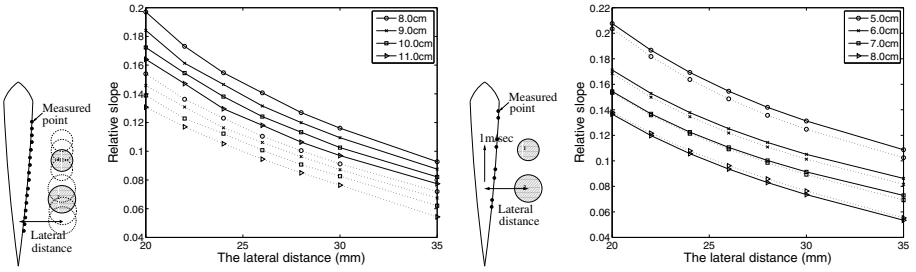


Fig. 5. Spatial slopes of electric image with two objects along the reference axis (a) spatial slope along the spatial axis (b) spatial slope along the time axis; 'solid line' first small object 'dotted line' large object (more detailed explanation is given in the text)

of multiple target objects. The measure seems less affected by the superposition of electric potentials from multiple objects.

Fig. 6 (a)-(b) show the temporal slope, that is, differentiation with respect to time, along the spatial axis and the temporal axis, respectively. Similar to the above procedure, the relative slope is calculated with the maximum amplitude along its reference axis. A space-to-temporal slope which is the temporal slope with respect to the spatial axis can be defined as

$$Space\text{–}to\text{–}temporal\ slope = \frac{\max_i \{I(x_i, t_{k+1}) - I(x_i, t_k)\}}{\max_i \{I(x_i, t_k)\}} \tag{7}$$

and space-to-temporal slope curves of two objects are shown in Fig. 6 (a).

Fig. 6 (b) diagram is equivalent to the temporal relative slope mentioned above. It is redefined as a time-to-temporal slope as,

$$Time\text{–}to\text{–}temporal\ slope = \frac{\max_k \{I(x_i, t_{k+1}) - I(x_i, t_k)\}}{\max_k \{I(x_i, t_k)\}} \tag{8}$$

The temporal slope pattern over the spatial axis depends on the size of object and the rostrocaudal position. So it would not be a good distance measure of multiple objects. Temporal relative slope is nearly independent of the rostrocaudal position and sensor position, if only one target object is available - see Fig. 2 (b). Here, when multiple objects are near the fish, the transdermal potential of an electroreceptor is influenced by those objects, and the relative slope pattern changes depending on the object size - see Fig. 6 (b). This discrepancy is related to the effect of superposition of each electric potential curve, which modifies the maximal amplitude and maximal temporal slope. Especially, the effect of size and rostrocaudal position is larger for a small-sized object, since the electric potential from the larger object makes more distortion over the electric field of the small-sized object.

From the experimental result, the spatial slope along the temporal axis as shown in Fig. 5 (b) provides a reasonable measure for the lateral distance independent of the size of objects. This measure can be applied to multiple target

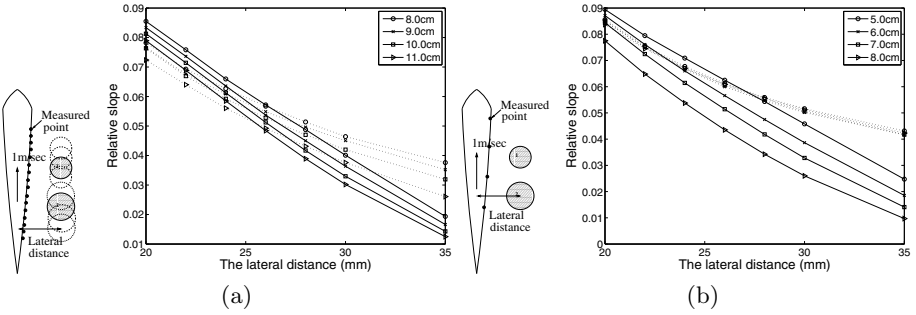


Fig. 6. Temporal slopes of electric image with two objects along the reference axis (a) temporal slope along the spatial axis (b) temporal slope along the time axis (solid: small object, dotted: large object, more detailed explanation is given in the text)

objects in the environment, if the inter-distance between a pair of objects is not too small, assuming only small distortion of electric potentials is given. Thus, the spatiotemporal information of electroreceptors is more useful for multiple target objects than a pure spatial information or a pure temporal information.

When there are three objects near a weakly electric fish, the effect of superposition is larger, and the relative slope variation becomes remarkable depending on varying sizes and varying rostrocaudal positions. However, if the distance between objects is large enough to keep a little distortion of electric potentials, the spatial slope along the temporal axis provides the distance information of multiple objects.

In fact, electric images are affected by surroundings such as rocks, water plants, conspecifics, or preys [12]. Therefore, the electric image is the result of 'superposition' of electric potentials influenced by each single object. In this study, we suggest a possible localization method when there are multiple objects near a weakly electric fish. Intensity difference of two neighboring sensors along the temporal axis shows the relative distance of objects approximately. However, when a pair of objects are very close, each relative slope is severely affected by the interference of electric potentials by the objects and in this case it is difficult to extract relative slopes or the distance information appropriately.

4 Conclusion

The relative slope, the ratio of maximum slope to maximal amplitude over electroreceptor readings, is known as a distance measure independent of the size and conductivity of a target object [2,3,4]. However, if there are multiple objects near the weakly electric fish, the localization problem of objects appears.

The relative slope or FWHM has difficulty in estimating the distance of multiple target objects. The intensity difference of electrosenses between neighboring sensors can be monitored along the time axis when the fish swims forward or backward. It provides the spatiotemporal relative slope and furthermore, the

lateral distance of multiple objects from the fish body. The measure is independent of the size of objects.

Electrolocation can be used to localize target objects in the underwater. The mechanism of localizing multiple objects is helpful to develop the electrosensory system of aquatic robots. The electrosensory system can find not only a position of a target but also another properties, size and electrical characteristics [10,2,3,4], In this paper, we assume the electrosensor signal for each object is preserved and the signals are superposed for multiple objects. Interference among the signals has been ignored in this paper but we need to consider it for accurate modeling of electrosenses in the future work.

Acknowledgement. This work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MEST) (No.2009-0080661).

References

1. MacIver, M.: The computational neuroethology of weakly electric fish: body modeling, motion analysis, and sensory signal estimation. PhD thesis, University of Illinois (2001)
2. von der Emde, G., Schwarz, S., Gomez, L., Budelli, R., Grant, K.: Electric fish measure distance in the dark. Nature 395, 890–894 (1998)
3. von der Emde, G.: Active electrolocation of objects in weakly electric fish. Journal of Experimental Biology 202(10), 1205–1215 (1999)
4. Schwarz, S., von der Emde, G.: Distance discrimination during active electrolocation in the weakly electric fish *Gnathonemus petersii*. Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology 186(12), 1185–1197 (2001)
5. Sicardi, E., Caputi, A., Budelli, R.: Physical basis of distance discrimination in weakly electric fish. Physica A: Statistical Mechanics and its Applications 283(1-2), 86–93 (2000)
6. Chen, L., House, J., Krahe, R., Nelson, M.: Modeling signal and background components of electrosensory scenes. Journal of Comparative Physiology A: Sensory, Neural, and Behavioral Physiology 191(4), 331–345 (2005)
7. Lannoo, M., Lannoo, S.: Why do electric fishes swim backwards? An hypothesis based on gymnotiform foraging behavior interpreted through sensory constraints. Environmental Biology of Fishes 36(2), 157–165 (1993)
8. Lissmann, H., Machin, K.: The mechanism of object location in Gymnarchus niloticus and similar fish. Journal of Experimental Biology 35(2), 451–486 (1958)
9. Kramer, B.: Electroreception and communication in fishes. Gustav Fischer (1996)
10. Rasnow, B.: The effects of simple objects on the electric field of Apteronotus. Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology 178(3), 397–411 (1996)
11. Heiligenberg, W.: Theoretical and experimental approaches to spatial aspects of electrolocation. Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology 103(3), 247–272 (1975)
12. Budelli, R., Caputi, A., Gomez, L., Rother, D., Grant, K.: The electric image in *Gnathonemus petersii*. Journal of Physiology-Paris 96(5-6), 421–429 (2002)

From Force Control and Sensory-Motor Informations to Mass Discrimination*

Sébastien Razakarivony[1], Philippe Gaussier[1], and Fathi Ben Ouezdou[2]

[1] ETIS-UMR 8051 CNRS, Université de Cergy Pontoise - ENSEA
6, Avenue du Ponceau, 95000 Cergy, France
[2] LISV-EA 4048, Université de Versailles Saint Quentin
10-12, Avenue de l'Europe, 78140 Vélizy, France
sebastien.razakarivony@ensea.fr, gaussier@ensea.fr, ouezdou@lisv.uvsq.fr

Abstract. Human adults know that usually, big objects are heavier than small ones if these objects are quite similar, in the same material for example. They have a general idea of the weight affordances about the every-day life objects. This paper presents a neural network architecture coupled with a simple linear actuator using force control, designed to use sensory-motor and visual informations during manipulation to learn how to recognize objects of different masses. After learning the association of sensory-motor informations through time with a particular object, our architecture can discriminate different masses and give relevant information for unknown objects, consequently, the objects are associated to some of their inferred physical properties.

1 Introduction

Muscles determine a part of the abilities of the intelligence of a system, as showed in [1]. Muscles properties are indeed important in motor control[2]. Nowadays, almost all used actuators in robotics are electrical motors associated with gearbox reduction allowing usually only speed or position control. Such actuators are subjected to friction and the command given does not correspond exactly to the effect. These two problems represent a loss of information. This is usually solved thanks to addition of sensors on the above mentioned actuators. Furthermore, the power to size ratio is relatively small for electric motors inducing an increase for the mass whenever high power is needed. To minimize information loss and have better size-power compromise, linear hydraulic motors can be an interesting alternative and could be a better model of muscle since it will be able to reproduce either isotonic (i.e constant passive force) or isometric (i.e. fixed configuration with controlled force) functioning modes. Indeed, in this paper, we will present a neural network architecture inspired from the cerebellum [3], [4], [5] which uses a force control (as human muscles) to recognize objects thanks to the alteration of sensory-motor informations due to their different masses.

* This work is within the ANR project Interact ANR-09-CORD-014. The authors would like to thank P. Andry and Y. Delevoye for their help.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 221–231, 2010.

We show that weight affordances [6] can emerge from the prediction learning of hand trajectory when lifting objects. This architecture will be implemented in a further work on a 2 degrees of freedom (d.o.f) hydraulic robot equipped with differential pressure sensors (fig.1 right) but yet only simulations have been performed. In the first part, the simulation model is presented. In the second part, a first learning architecture is used to learn sensory-motor consequences of motor orders i.e. predicts which errors are made when the mass is changed. Finally, a more complete neural network learns to recognize object, thanks to the previous architecture. The results presented here used the simulated physical system of fig.1(left). Some of the tests have been also performed with a simulated actuator of fig.1(middle) and the results are discussed in the last part.

2 Simulated Physical System

The linear hydraulic motor allows force control. However, it is difficult to model such an actuator. Our main interest is not to provide an accurate simulation of the linear hydraulic motor itself (because it will be replaced by the real physical device) but rather to study how a force control can be exploited to learn object discrimination through sensory-motor associations. This is a first step to validate the architecture in a simple case. Our simulation simply models a force on an object, and presented as follow (fig. 1). An object of a mass m is pulled with a linear actuator with a force F. For the sake of simplicity, only one translational d.o.f is used along the vertical axis, with a minimal and a maximal position. The forces applied to the system are the weight $m.g$ with g the gravity acceleration ($9.80\ m.s^{-2}$), friction $k.v(t)$ where $v(t)$ is the speed (velocity) and k a coefficient of friction($7.0\ kg.s^{-1}$) and finally $a(t)$ the current acceleration. Applying Newton equation to the pulled mass leads to : $m.a(t) = F(t) - m.g - k.v(t)$. Integrating the above equation with $(0, 0)$ as initial conditions on speed and position and constant force $F(t) = F$ yields to the following solutions for speed $v(t)$ and actuator top position $z(t)$:

$$v(t) = -\frac{mg - F}{k} e^{(\frac{-k}{m}t)} + \frac{mg - F}{k}; z(t) = \frac{(m.g - F)m}{k^2} e^{(\frac{-k}{m}t)} + \frac{m.g - F}{k} t \quad (1)$$

The simulation system returns $v(t)$ and $z(t)$. In order to use the same tools for simple and complex simulations (where no analytical solution is available) v(t) and z(t) are computed according to a fourth-order Runge-Kutta method. Two sensors scale them between 0 and 1. The simulated physical system is calibrated to obtain a credible trajectory, i.e. not always flat or always crashed at maximum position.

3 Learning Speed and Position through Time

The mass cannot be determined easily from equation 1 alone since $z(t)$ is non linear from m and the inversion is not easy (at least with a simple on line neural

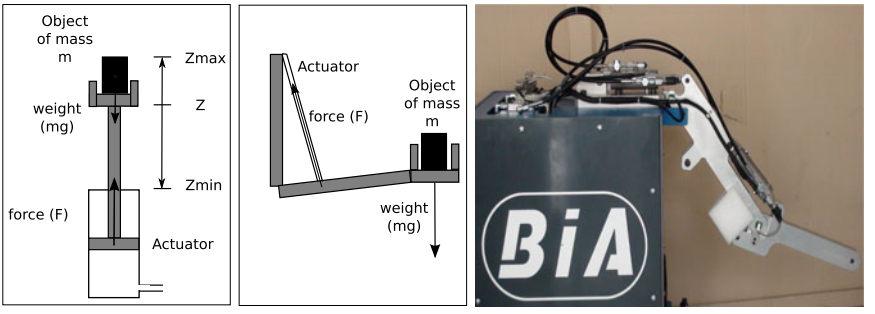


Fig. 1. left: The object of mass m can be moved within the range (zmin, zmax). A modifiable force F, its weight P and friction force are applied to this object. middle: Same but with a rotational d.o.f. right: The hydraulic actuator built by the firm BIA allowing force control for the future experiments.

network). In a static experiment, a differential pressure sensor could give the mass of an object, but we want the actuator to interact with the object and let the property emerge without such a sensor. Our aim is to allow our architecture to detect changes in sensory-motor perception to recognize objects and associate a mass. In this part, sensory-motor perception is learned with different interactions with the same object to know if a simple system can learn several interactions and an interaction is better than another to detect the changes.

In a first approach, preliminary experiments showed that (1) Kohonen maps [7] are not sufficient to achieve proper categorization regrouping v(t), z(t) and F(t). Because of the different dynamics of the variables, after learning the map can only discriminate the values of one or two variables but nenver simultaneously the force, speed and position [1]. (2) in the case of position control the force profile is too complex to be learned (too much variations and oscillations and changed only during the transitory state).

Therefore our architecture is based on the learning of v(t) and z(t). The subsystem that enables learning (fig. 2) is composed of a spectrum generator and a group performing the least mean square algorithm [8]. The spectrum generator is activated each time the actuator is interacting with the object and returns a battery of activities (fig. 2). It matches to a model of the parallel fibers in the cerebellum, which send a spectral response to a given event [3]. These activities allow the learning subsystem to associate the interaction to its consequences through time. The least mean square algorithm used to model conditioning tries to predict $(v(t), z(t))$ from the time spectrum generator. The learning equation is the Widrow-Hoff rule [8], $\Delta w_{ij} = \epsilon * (S^d - S) * I_{ij}$, where w_{ij} is the weight between one granular cell j of the line i of the spectrum generator, I_{ij} the activity of the same cell, S^d the variable to learn (v and z) returned by the sensors,

[1] The only way to allow map convergence would be to perfectly now the time scales of each variable to adapt the learning rates and neighborhoods, hence this learning would be pointless.

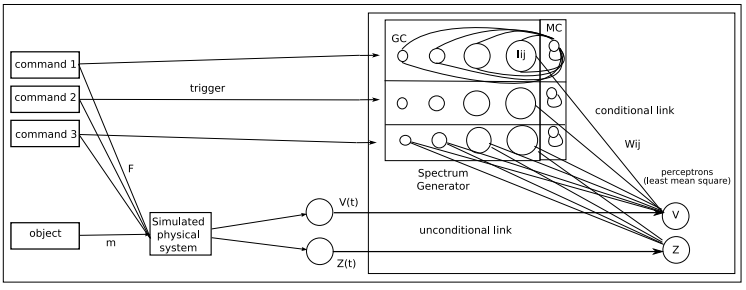


Fig. 2. For one interaction, one command is activated. A force is applied to the physical system. The object is of mass m. Each battery of the spectrum generator is activated by one command. For each battery, 30 cells were used (not all represented in the figure), which represent the granular cells (GC). The GC interact with a mossy cell (MC) (not all links represented), Their resulting activities are different gaussians. The gaussian shape is related to the size of the GC. Each GC is linked to the two learning neurons (not all links represented). These neurons match their activities to the sensors activities.

S what the learning subsystem returns as estimate position or speed, and ϵ the learning rate. The activities of the cells of the spectrum generator are corresponding to a gaussian $exp(\frac{-(t-me_i)^2}{\sigma_i^2})$, where me_i and σ_i are the mean and standard deviation of each cells, plus an offset in time. This mean and standard deviation increased in a linear way according to the size of the granular cells. It is now well know that least mean square algorithm is equivalent to the Rescorla-Wagner [9] brain conditioning rule [10]. A sensory-motor feedback of position exists with the tendon proprioception. The speed is in fact only a subtraction of two positions, so it is not irrelevant to use information of speed also. The use of linear hydraulic motors which have a force control corresponds to the force control of muscles through the activation process. All these aspects make our architecture biologically plausible. It is interesting to know if there is a better interaction to use in order to detect the sensory-motor alterations when the mass is changed. Our architecture thus learns to match different force to their consequences. The architecture needs several examples for each sensory-motor experiment to learn accurately. Each example is divided in three phases, (that is to say three interactions with an object). During each phase, the linear actuator pulls the mass with a given force. For each different force, a different row of the spectrum generator is activated and the force set is different. The three phases are repeated until the convergence of the learning subsystem. After running various simulations with an increasing number of examples, it appears that a really good approximation was obtained when the number of examples is near 3000 for each phase. For a position and a speed which vary between 0 and 1, the normalized mean square error was under $2.5 * 10^{-4}$ for both and the standard deviation of these errors is around $5 * 10^{-4}$. This high number of interactions will imply a long learning phase on a real robot which may be a strong

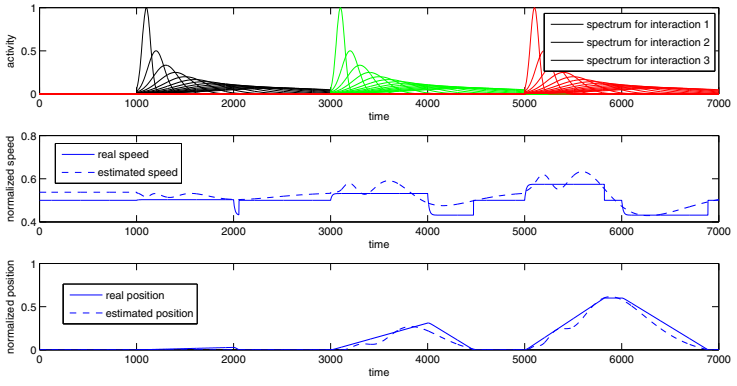


Fig. 3. From up to bottom : Activities of the spectrum generator for each manipulation. Speed estimate and real speed, position estimate and real position (real in solid line and estimate in dotted line). The results presented are obtained after 300 interactions for each manipulation. The first phase begins at t = 1000.

limitation for the proposed architecture. The results on fig. 3 show that good estimations can be obtained after 300 interactions (mean square error around $6 * 10^{-3}$ and standard deviation around $9 * 10^{-3}$), which is still a large number, but allows several experiments in a reasonable time. An important point is that the least mean square convergence assumes a stochastic gradient descent, while here, the examples are presented continuously through time. Consequently, the time sequence is not well learned. To solve this problem, the learning subsystem learns at a given iteration of the Runge Kutta algorithm only with a probability of $1/N$, (N is fixed during the experiment), otherwise, weights are not modified. This randomization could correspond to a neuromodulation effect or more simply to the dynamics of an inhibitory network inducing a rhythmic activity. If the frequency of this rhythm is coprime to the frequency of the experiment, some kind of random learning is obtained. The synapses are assumed to be randomly selected. In order to keep the architecture simple, global random triggering of the learning is chosen. The fig.4 (left) shows the progress of the errors according to N. The increase of the randomization leads to better results, for both errors and their standard deviations. However, the gain is in fact even greater, because the learning subsystem has an expected value of learning of only one time out of N, so N times less informations from the sensors are actually used to learn. With a normalization, the results presented in fig. 4 (middle) are obtained. Since we want to minimize the number of samples to implement our architecture on a real actuator and have fast on line adaptation, only the real number of samples is important. The fig. 4(right) shows the improvement of the mean square position error function of the increase of number of sensory-motor experiments. The random rule always reduces drastically the error level. The speed error, the standard deviations of speed error and position error have the same kind of evolution profile and the same order of magnitude. Thanks to the sensory-motor

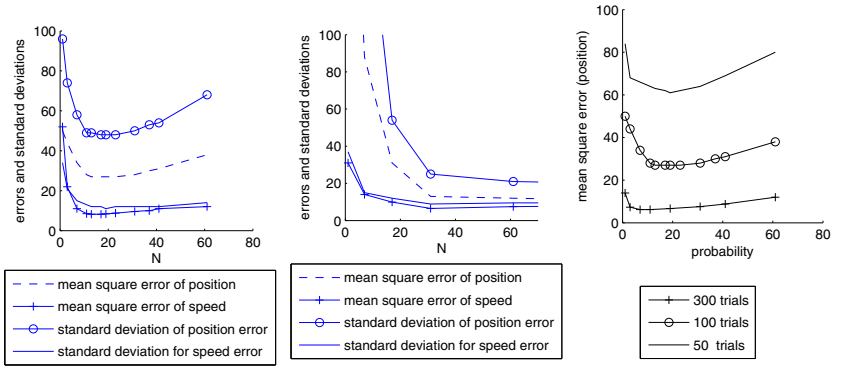


Fig. 4. Results of the mean square errors on position and speed estimations and their standard deviations functions of N, with $1/N$ the probability that the weights are changed due to learning for one iteration of the simulation. On the left: number of non normalized examples. In the middle : number of normalized examples. On the right: results of the mean square position error estimation functions of N, showing the influence of the number of examples presented to the learning subsystem.

interactions, our architecture can learn the different consequences on our actuator for a given experiment. Using a random learning increases precision. Learning the speed and position for a given mass allows us to detect a change in the mass.

4 Sensory-Motor Prediction and Error Detection

Once the learning is completed, it is interesting to see if the architecture is able to discriminate learned objects. The actuator interacts with objects and tries to predict their speeds and positions. The manipulated objects can be one which has been used for learning, called a learned object with its learned mass, or another one with a different mass. The mass used during one interaction is called the test mass of the test object. The errors for each couple (Learned Mass, Test Mass) were calculated and reported in fig.5. The same kind of figures can be observed with the standard deviation of position error, the speed and the standard deviation of the speed error. With heavy objects or with very light objects, the actuator is blocked to a maximal error when the mass is too far from the learned mass because of the maximal and minimal positions of the actuator. A comparison of the different phases leads to the following observation: the larger the amplitude of movement is, the higher the errors are, including the error on the learned object. This could be a problem, however, it is in fact good news : even if the error on the learned object is greater, the difference between this error and the others is greater too. Consequently, large movements give more discernible errors, and our architecture can be improved to exploit these differences of errors. For a test mass far from the learned mass, the manipulation error is always greater that the error on the learned object; so a minimum valley

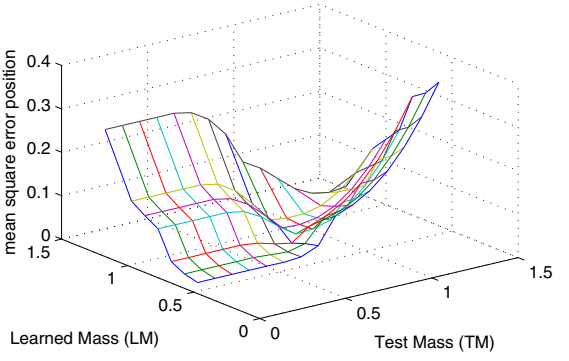


Fig. 5. Mean square position error (vertical axis) in function of test mass (TM) and learned mass (LM). For $|TM-LM| < TM/10$, a minimum valley is observed. However, the minimum is not always reached for TM = LM.

can be observed. However, for some test mass too close to the learned mass, the manipulation error may be smaller than the error on the learned object, i.e. the argument of the minimum of error during manipulation is not always the learned mass. This property can be explained by a non-perfect learning. The very heavy objects are easy to learn, because the actuator cannot move them, so the position and speed are always equal to zero. However, when the learned mass is heavy but can be slightly moved, these small movements are not learned. The error during manipulation is greater than if the actuator tries to interact with a too heavy object, because for this one the error will always be null.

Hence, we know that error in sensory-motor prediction increases when the system does not use the right hypothesis (i.e. the right predictor). However, the result can be exploited only at the end of the interaction : during one particular interaction, the current manipulation error can be greater than the error on the learned object, even if the used mass is far from the learned mass. This is also due to an imperfect or incomplete learning. Thanks to this experiment, we know that our model can be used to detect a change in the mass of an object. If several neurons learn to predict the speed and position according to a given object and command profile then the cumulative error in prediction is lower on the neurons trained with the nearest mass. Next section will focus on using this property to build an architecture able to recognize an object according to its perceived mass.

5 Object Discrimination from Sensory-Motor Perception

Once the meaning of errors is understood, our learning subsystem is changed to use these errors to recognize objects (fig. 6). Only the information of position error is chosen, because it had better results than the informations given by speed or standard deviations of the different errors. M objects, each one with a different mass were considered. The actuator executes only one phase instead of

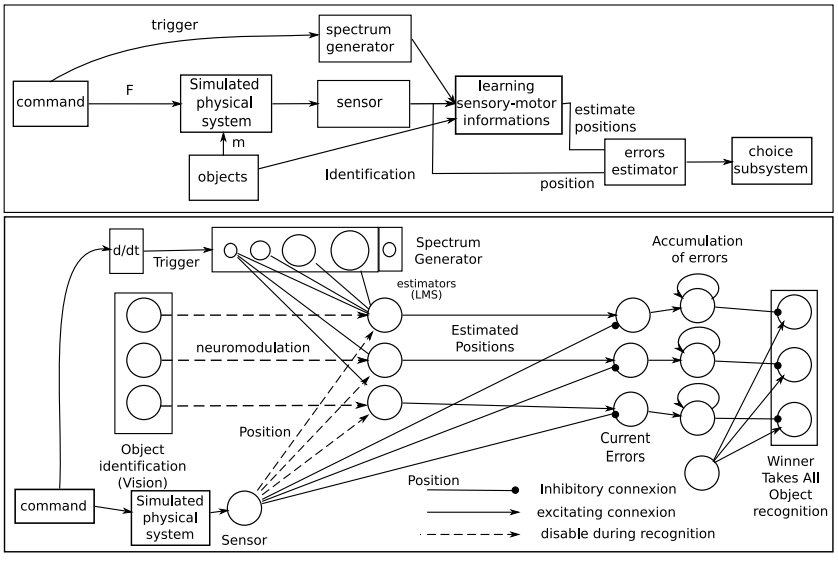


Fig. 6. The object triggers one learning estimator during the learning part. During the test part, each learning estimator return an estimate position. For each estimate position, a current square error is calculated. The errors are cumulated to get an equivalent to the mean square errors. The learning estimator with the lower mean square error is the winner and corresponds to the label recognized by our architecture.

three, with large movements while avoiding the maximal position. Consequently, only one row of cells is used in the spectrum generator. The learning subsystem is divided in M estimators. Each of them is associated with one object. To do so, when one part of the actuator is interacting with a specific object, the learning rates of the $M-1$ other estimators are set to zero. It could represent the effect of visual perception for instance, which triggers only one learning estimator. Once learning achieved, the actuator interacts with different objects. Some of them correspond to one of learned masses while some others not. The current square error is calculated during the manipulation for the M estimators. Accumulators sum the errors through time. Then, a weight labelling group selects the estimator that obtained the lower accumulated error. The object which has been learned by this estimator is considered as the recognized object. At the beginning of a new manipulation, the error accumulators are reset to zero. The results through time are presented in fig. 7. One can see on these figures that through time, the architecture associates the learned object of decreasing weights successively until it reaches the true one and stabilizes. This can be explained by the fact that the learning estimators learn the movement mean. As the actuator moves the object up, the mean position is always increasing. The heavier is the object, the lower the trajectory will be, so the mean of this trajectory increases when weight of the object decreases. Consequently, when the errors are accumulated, the heaviest object is recognized because the position is still low, but when

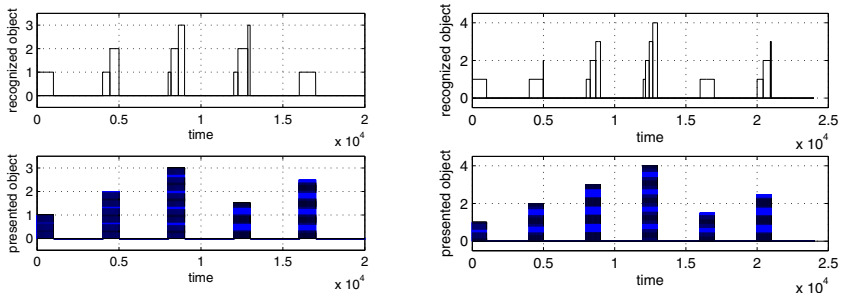


Fig. 7. Left picture: Results obtained after the interaction with the different objects (mass of 1.3, 1.0, 0.7, 8.15 and 1.15 kg). The three first objects were learned. Upper part: number returned by the labelling group. Lower part: label of the test object (from 1 to 3). The learning was performed for 60 sensory-motor experiments for each object. We can see that the learning is sufficient to recognize correctly the 3 objects. Right picture: Results obtained after the interaction with the different objects, mass of 1.3, 1.1, 9.0, 0.7, 1.15 and 1.0 kg. The four first objects were learned. Upper part, number returned by the labelling. Lower part, label of the test object (from 1 to 4). The learning was performed for 60 sensory-motor experiments by object.

the mean position continues to increase, the error corresponding to the heaviest object increases more. Even with a low number of samples, a good discrimination can be obtained, as shown in the fig. 7. If the number of examples drops too low (under 60), the discrimination can fail. The unknown objects are recognized as one of the learned object with a close mass, and we can see that even if the neural network recognizes the unknown objects as one of the nearest object in term of weight, the final response is not reached at the same time than for the learned object. So it is possible to estimate for how long the architecture recognized the object. This estimation is related to the reliability of the discrimination.

6 Discussion, Conclusion and Further Work

These experiments show that thanks to a rough model of the cerebellum activity, a force control and a simple mechanism to extract errors on predicted measures, it is possible to associate an object to specific sensory-motor feedback. The larger the movement is, the easier the alterations in the sensory-motor are detected, even if it is better to avoid the maximum and minimum positions. A random learning on the sequence improves also the learning. Each final response is associated to a specific time, which can provide an information of quality (by the delay), so we could then change our architecture to learn to estimate the properties of such information.

Some problems are coming from the simulated actuator. First, the applied force is the asked one, but with the real actuator, the asked force will not perfectly correspond to the effective force, so we will have to look again at the force

proprioception. The coefficient of friction is set by hand and is not realistic, but is set to observe some non linear behavior. This simulation just represents a falling mass in a changing gravity well, which does not correspond to a real actuator. During the experiment it also appeared that the range of possible masses was small due to our simulation, that is why we cannot learn a lot of different mass and the trajectories are relatively simple. In a further work, this architecture will be implemented on a real robot arm and will have to be tested on it, we hope to access a larger panel of masses, from a few grams to several kilos. In addition, some similar tests have just been performed with a simulation of an arm as presented in fig. 1(middle). The results are close to those presented in this paper, and our architecture is still able to recognize objects. Although, the speed variable was more pertinent than the position variable.

The fact that weight is learned by the association of sensory-motor information from the muscles is relevant from a biological point of view: in [11], it is shown that weight perception is linked to the inertial tensor of an object, (i.e. difficulties to make it move around an axis), which corresponds to the effort of muscles acting around the elbow when moving an object. Other experiments are planed after the implementation on a robot arm, to test the size-weight illusion [12] (when someone is confronted with two objects of same weights but with one larger than the other, the person says the smaller is heavier). Then we could present a new architecture that could associate size to sensory-motor feedback. Another point that makes our architecture biologically plausible is the independence in perception and sensory-motor predictions for the size-weight illusion[13]. Such illusion could not be observed with only a static measure, it has to be related to visual perception. A vision learning subsystem should be added to our architecture and used as the learning trigger, in order to observe such illusion. The size illusion has an influence on how we lift objects [14], so we would like to observe such behavior on our future architecture. We could also generalize and use the global context instead of only a vision subsystem, to increase the system efficiency to discriminate different sensory-motor situations, in order to have a context-dependent sequence learning [15]. In further work, we would like to have a control model of the manipulation of different objects. In order to have a framework for control and social interaction [16], the MOSAIC model [17] could be a base for this future work. Because this architecture allows to predict the sensory-motor perception, we will be able to have a control on it [18]. We will also base our future work on [19], to use the prediction of the cerebellum to have a good control strategy. We planned to set up same experiments taken from [20]. One of them is described as follows : a human person stands, his hand on the trajectory of a pendulum. He will try to stop its run without moving. The force of the hand is measured to know how the person prepares and predicts the effect of the pendulum function of what he knows about this object. Our architecture could learn to recognize an object and then prepare to interact with it. Finally, the implementation on a real actuator (fig.1 right) is currently carried out.

References

1. Pfeifer, R., Bongard, J.: How the Body Shapes the Way We Think: A New View of Intelligence. MIT Press, Cambridge (2007)
2. Paolo, E.D., Buehrmann, T.: Biological actuators are not just springs. From Animals to Animats 9, 89–100 (2006)
3. Bullock, D., Fiala, C., Grossberg, S.: A neural model of timed response learning in the cerebellum. Neural Networks 7(6-7), 1101–1114 (1994)
4. Grossberg, S.: On learning of spatiotemporal patterns by networks with ordered sensory motor components. Studies in Applied Mathematics 48, 105–132 (1969)
5. Banquet, J., Gaussier, P., et al.: Cognitive Science perspectives on personality and emotion. In: Matthews, G. (ed.) Space-Time Order and Hierarchy in Fronto-Hippocampal System: A nNeural Basis of Personality, ch. 4 (1997)
6. Gibson, J.: The ecological approach to visual perception. Lawrence Erlbaum Associates, Mahwah (1986)
7. Kohonen, T.: Self-organization formation of topologically correct feature maps. Biological Cybernetics 43, 59–69 (1982)
8. Widrow, B., Hoff, M.: Adaptive switching circuits. In: IRE WESCON Convention Records, pp. 96–104 (1960)
9. Rescorla, R., Wagner, A.: A theory of pavlovian conditioning: Variations in the effectiveness or reinforcement and non-reinforcement. In: Black, A.H., Prokasy, W.E. (eds.) Classical Conditioning II: Current Research and Theory (1972)
10. Sutton, R., Barto, A.: Toward a modern theory of adaptative networks: Expectation and prediction. Psychological Review, 135–170 (1981)
11. Amazeen, E.L., Turvey, M.: Weight perception and haptic size-weight illusion are function of the inertia tensor. Journal of Experimental Psychology 22, 213–232 (1996)
12. Charpentier, A.: Analyse experimentale de quelques elements de la sensation de poids —experimental study of some aspects of weight perception—. In: Archives de Physiologie Normales et Pathologies, pp. 122–135 (1891)
13. Flanagan, J., Beltzner, M.A.: Independence of perceptual and sensorymotor predictions in the size-weight illusion. Nature Neuroscience 3, 737–741 (2000)
14. Brenner, E., Smeets, J.: Size illusion influences how we lift but not how we grasp and object. Experimental Brain Research, 473–476 (1996)
15. Berthouze, L., Tijsseling, A.: A neural model for context-dependent sequence learning. Neural Processing Letters 23, 27–45 (2006)
16. Wolpert, D., Doya, K., Kawato, M.: A unifying computational framework for motor control and social interaction. The Royal Society, pp. 593–602 (2003)
17. Haruno, M., Wolpert, D., Kawato, M.: Mosaic model for sensorimotor learning and control. Neural Computation 13, 2201–2220 (2001)
18. Flanagan, J., Roland, P., Johansson, S., Wolpert, D.: Prediction precedes control in motor learning. Current Biology, 146–150 (2003)
19. Shadmehr, R.: Computational approaches to motor control. Encyclopedia of Neuroscience 3, 9–17 (1986)
20. Delevoye, Y., et al.: Efficiency of grip force adjustments for impulsive loading during imposed and actively produced collisions. Quarterly Journal of Experimental Psychology, 1113–1128 (2006)

The Approach Behaviour of the Hawkmoth *Manduca sexta* toward Multi-modal Stimuli: A Simulation Model

Anna Balkenius[1], Marie Dacke[2], and Christian Balkenius[3]

[1] Chemical Ecology Group, Swedish University of Agricultural Sciences, Alnarp, Sweden
[2] Vision Group, Lund University, Lund, Sweden
[3] Lund University Cognitive Science, Lund, Sweden

Abstract. We present two models of the behaviour of the hawkmoth *Manduca sexta* when it approaches an artificial flower with an olfactory, visual or multimodal cue. The first model treats each condition separately while the second model combines both types of sensory cues in a single model. Both models reproduce several characteristic properties of the hawkmoth behaviour including its goal direction and velocity profile for different stimulus types. In addition, the second model accounts for the interaction between visual and olfactory cues.

1 Introduction

Two important sensory modalities for most insects during foraging are vision and olfaction. Depending on whether the insect is day or night active, one of these modalities is often more important than the other [1] [2]. When a nectar feeding insect approaches a flower it could use either visual or olfactory cues. It is also possible that the insect combines both modalities since multisensory cues enhance reaction times and lower error rates [3].

We have used the hawkmoth *Manduca sexta* as a basis for building computational models that investigate multi-modal interactions between colour and odour. This is an ideal animal for this type of studies since we know a lot about their receptors and the processing of both olfactory and visual information. *M. sexta* is a crepuscular-nocturnal hawkmoth native to the Americas and the larva is a pest on the tobacco plant. It is a pollinator that is a fast flyer that hovers in front of flowers while extending its proboscis to probe the flower and drink the nectar. The proboscis is very long (8-10 cm) and is probably guided by both vision and mechanoreceptors.

The eyes of *M. sexta* are superposition compound eyes with well developed tapeta adapted to activity during night. The moth has visual receptors in the green, blue and ultraviolet range. They have an innate preference for blue flowers even though most blooming flowers during the night are white or bright yellow. This shows that they can quickly learn to visit the white and yellow flowers instead of endlessly searching for blue flowers during the night.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 232–241, 2010.

In behavioural experiments with hawkmoths we have shown that colour can overshadow odour learning [4]. We also discovered that colour could both enhance and suppress processing of different odour responses in the mushroom body of hawkmoths [5]. There is thus clear evidence for an interaction between vision and olfaction in the selection of flowers.

Approach behaviours have been much studied within the modeling community staring with Braitenberg's vehicles that illustrated how simple systems can show goal-directed behaviours [6]. However, such models have usually only addressed on a modality at a time, whether it be vision [7], olfaction [8] [9] or phonotaxis [10]. Here we address the question of how the different modalities interact in the control of flight as the moth approaches a flower. There are easily observed differences in the behaviour of the moths dependning on the available stimulus modalities [11]. For the multimodal and visual stimulus, the moths slow down as they approach the flower and hover a few centimetres in front of it while extending their proboscis to feed. In some cases, the moths circle around the flower as they feed. For the odour stimulus the behaviour is different as the moths do not slow down completely and will instead pass over the target only to turn quickly back as they pass apparently in an attempt to find the invisible flower.

To develop a model of these behaviours, the flight paths of the moths were recorded using unimodal visual or odour cues or using a cue consisting of both modalities. In a second step, a tracking system was used to determine the position and orientation of the moth at 100 Hz during the approach of the flowers. Finally, a model was developed that could reproduce the behaviour of the moth in the three conditions. The parameters of the model were fitted using system identification methods [12].

The goal of the simulation model was to characterise these behaviour and to investigate how complex a model would have to be to reproduce them. Are different models needed for the different sensory conditions? Is it sufficient to use the angle and distance to the target as input? Are linear models sufficient or are more complex relations between input and behaviour control necessary?

2 Data Collection

The hawkmoth *Manduca sexta* (Lepidoptera: Sphingidae) was used for the experiments where they were allowed to fly in a circular arena with a diameter of 1.5 m. The arena was surrounded by a wall with a height of 0.5 m and covered with a transparent net. One of three different feeding-targets was placed at the centre of the arena: a flower-like blue target; an invisible, scented target; or a scented flowerlike blue target. The odour used to scent the targets was Bergamot oil that contains odours that many night-blooming flowers release [13]. The approach of a hawkmoth to the feeding target could thus be analysed under three experimental conditions: only a visual stimulus (V), only an odour stimulus (O) or a multimodal stimulus consisting of the artificial flower with an odour (M).

The image of the arena was recoded at 100 frames/s. The moths were released into the circular arena one at the time, and always from different directions. The

recording of the state variables started the first time the moth came closer to the centrally placed target than 400 mm. The locations of the target stimulus was manually indicated in one of the images, $\langle x_T, y_T \rangle$ under each experimental condition.

The analysis of the recorded image sequences was done in four stages: First the animals were localised using an adaptive foreground detection method [14] followed by the removal of shadows. Next, the detected foreground pixels were clustered to find the centroid of the foreground pixels which indicates the preliminary position of the moth [15]. Finally, a number of positions before and after the current time where averaged using a Gaussian window to produce a smooth trajectory. The result of the preliminary tracking was used to initiate the position of an active shape model [16]. The shape model consisted of a number of contour points around the moth together with contour normals estimated from a standard moth image. The initial orientation of the model was set to the direction of the second moment of the foreground pixels. In addition, the last estimated position and orientation of the moth was used as an alternative hypothesis. The best match of the shape model was iteratively calculated and the relative match of the different hypotheses were compared to obtain the estimated location and orientation of the moth $\langle x, y, \theta \rangle$. The resulting position estimation was accurate to approximately a pixel resolution which corresponds to 1.7 mm in the recorded scene.

In the last, stage a number of values were calculated (Fig. 1). For the moth the velocity was mapped onto forward, lateral and rotational components $\langle v_x, v_y, v_\theta \rangle$. The distance and direction to the target stimulus, $\langle d_T, \theta_T \rangle$ relative to the moth were also calculated.

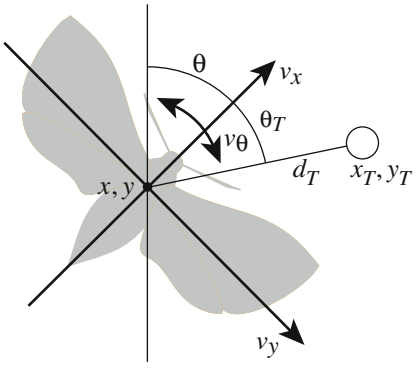


Fig. 1. The parameters used to describe the flight of the moth

3 Separate Models

To obtain a model of the behaviour of the moths in the experiments, we first set up a model equation for the moth. The velocity vector was considered as a

function of the two variables θ_T and d_T. There are however several symmetries in the behaviour that we need to take into account to get an accurate model. We must assume that if the forward acceleration depends on the direction of the target, it should be the same whether the target is on the left or the right side. We thus used the absolute value of θ_T in the equation for the forward acceleration. Since the opposite must be true for the turning and lateral movement, these equations used the original value of θ_T. An interaction term between the two variables was also included in the model. We consequently get the following initial model equations for the velocity vector:

$$\begin{aligned} v_x^* &= \alpha_0 + \alpha_1|\theta_T| + \alpha_2 d_T + \alpha_3|\theta_T|d_T \\ v_y^* &= \beta_0 + \beta_1\theta_T + \beta_2 d_T + \beta_3\theta_T d_T \\ v_\theta^* &= \gamma_0 + \gamma_1\theta_T + \gamma_2 d_T + \gamma_3\theta_T d_T \end{aligned} \tag{1}$$

Since we assumed that the moth would not turn or move laterally without any detected target, the values of β_0 and γ_0 were set to 0. The other parameters were individually estimated for each of the three velocity components using the following procedure. Starting with the first order model we first calculated the Akaike's information measure for each possible first order model. In the second step we selected the model with the lowest entropy and tested whether the addition of the interaction term would decrease the entropy measure. Finally, the optimal parameters for the selected model were calculated for each of the three conditions.

Although this procedure resulted in model parameters that very closely fit the recorded data, the simulations (described below) showed that there was a clear case of overfitting since the model did not generalize to new initial conditions. The problem was identified as the relatively large coefficients that the method had assigned to the correlation terms. To increase the generality of the generated models we removed the interaction term for all equations. This resulted in the values presented in Table 1. The estimated parameters for d_T were 0 for lateral velocity and rotation and are not shown in the table.

Table 1. Parameters

condition		α_0	α_1	α_2		β_1		γ_1
V	v_x	-0.0008942	-0.0009464	0.0168	v_y	-0.0295	v_θ	0.0033
M		-0.0004606	0.0014671	0.0055		-0.0145		0.0004
O		0.0016516	0.0001849	0.0105		-0.0064		-0.0001

Based on Eq. 1 and the estimated parameters we simulated the behaviour of the moth in using two sets of equations. First, the desired velocities calculated by Eq. 1 were used as the set-ponts for a proportional controller:

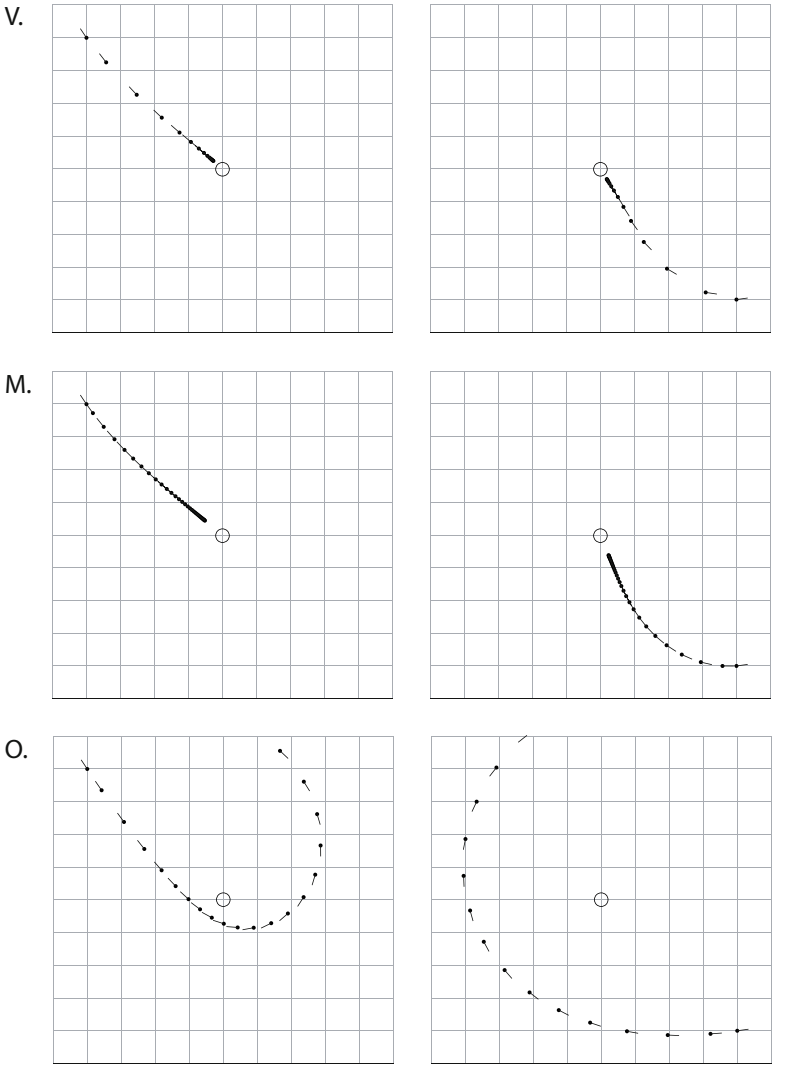


Fig. 2. The simulated approach behaviour of the moth toward a visual (V), multimodal (M) and olfactory (O) stimulus. The dot indicates the head of the simulated moth and the line shows the direction of the body. The circle in the middle marks the location of the target stimulus. The left and the right figures shows the behaviour for two different initial conditions.

$$\begin{aligned} v_{x,t+1} &= v_{x,t} + p(v^*_{x,t} - v_{x,t}) \\ v_{y,t+1} &= v_{y,t} + p(v^*_{y,t} - v_{y,t}) \\ v_{\theta,t+1} &= \theta_t + p(v^*_{\theta,t} - v_{\theta,t}) \end{aligned} \tag{2}$$

The constant p is the gain of the controller and was set to 0.1 in the simulations. Given a starting position $\langle x_0, y_0, \theta_0 \rangle$, subsequent positions were calculated using Euler's method by first calculating the velocities from the accelerations given by Eq. 1 and then updating the position in the following way:

$$\begin{aligned} \theta_{t+1} &= \theta_t + v_{\theta,t+1} \\ x_{t+1} &= x_t + v_{x,t+1} cos\theta_{t+1} + v_{y,t+1} sin\theta_{t+1} \\ y_{t+1} &= y_t + v_{y,t+1} sin\theta_{t+1} - v_{y,t+1} cos\theta_{t+1} \end{aligned} \tag{3}$$

Fig. 2 shows six simulations of the approach of the flower for each of the three stimulus conditions. The behaviour for the visual (V) and multimodal (M) conditions are very similar although the velocity profiles are somewhat different. In both cases the simulated moth turns toward the target and slows down as it approaches it and stops a small distance from the flower. This is the phase where the real moths hover while foraging from the flower.

The simulated behaviour when there is no visual cue also parallel that of the real moth. The speed is higher overall and although the speed decreases as the target is approached, it does not reach zero and the simulated moth will pass the target. In parallel with the real moth, the model moth will turn back as it has passed the target location. The model also illustrates that the turning of the moth is not sufficient to reach the target if the initial angle to the target is too large. This results in a much more spread out flight when there is only a unimodal odour stimulus.

To reproduce the circling around the flower as the moths fed, we hypothesised that the shape of the artificial flowers used, where the proboscis was inserted in a

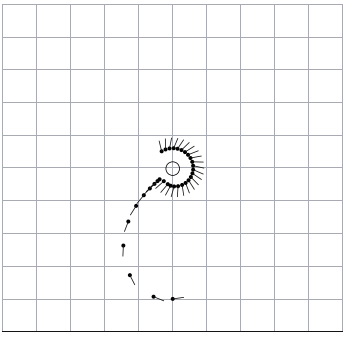


Fig. 3. The model accurately predicts that the moth will rotate around the flower

thin slit that went around the flower, would produce a lateral force on the animal through the proboscis. When such a force was added to the model, the behaviour in Fig. 3 was generated. Here, the simulated moth stays at approximately the same distance from the flower during foraging but circles around it in the way seen in some of the experiments with real moths.

4 A Combined Model

The model above uses different sets of parameters for each of the stimulus conditions. To develop a single model that could be used in all conditions, we repeated the procedure above with two additional inputs V and O that indicated whether a visual or an odour input was available. A value of 1 indicates that the corresponding modality is present and a value of 0 indicates that it is not.We also included a new variable $c_T = (1 - d_T)$ that better captures that the speed increases at increased distance from the target. Since we only investigate data close to the target, we need not consider the case when $d_T > 1$ which means that c_T is always positive.

To simplify the model, we gradually removed parameters that made no significant contribution to the final equations. This procedure reduced the three separate models with 18 parameters to a single model with only 6 parameters. The resulting equations are show below and the parameters are listed in Table 2.

$$\begin{aligned} v_x^* &= \alpha_0 + V\alpha_1 + \alpha_2 c_T \\ v_y^* &= V\beta\theta_T \\ v_\theta^* &= \gamma_0\theta_T + V\gamma_1\theta_T \end{aligned} \tag{4}$$

As can be seen, the variable O is not used in the final model as it did not contribute to its performance. This implies that the model gives identical predictions for visual and multimodal stimuli. Note also that the model can not be used to describe the behaviour of the moth when no target is present at all. If this had been the goal, it is unlikely that the variable O could have been omitted.

Table 2. Parameters for the combined model

α_0	α_1	α_2	β	γ_0	γ_1
0.010	0.002	-0.008	0.0003	-0.005	-0.008

The combined model was simulated in the same way as the previous models and the results for the same initial conditions as the separate models are shown in Fig. 2. As can be seen in Fig. 4, the combined model produces similar behaviour as the three separate models although it uses much fewer parameters.

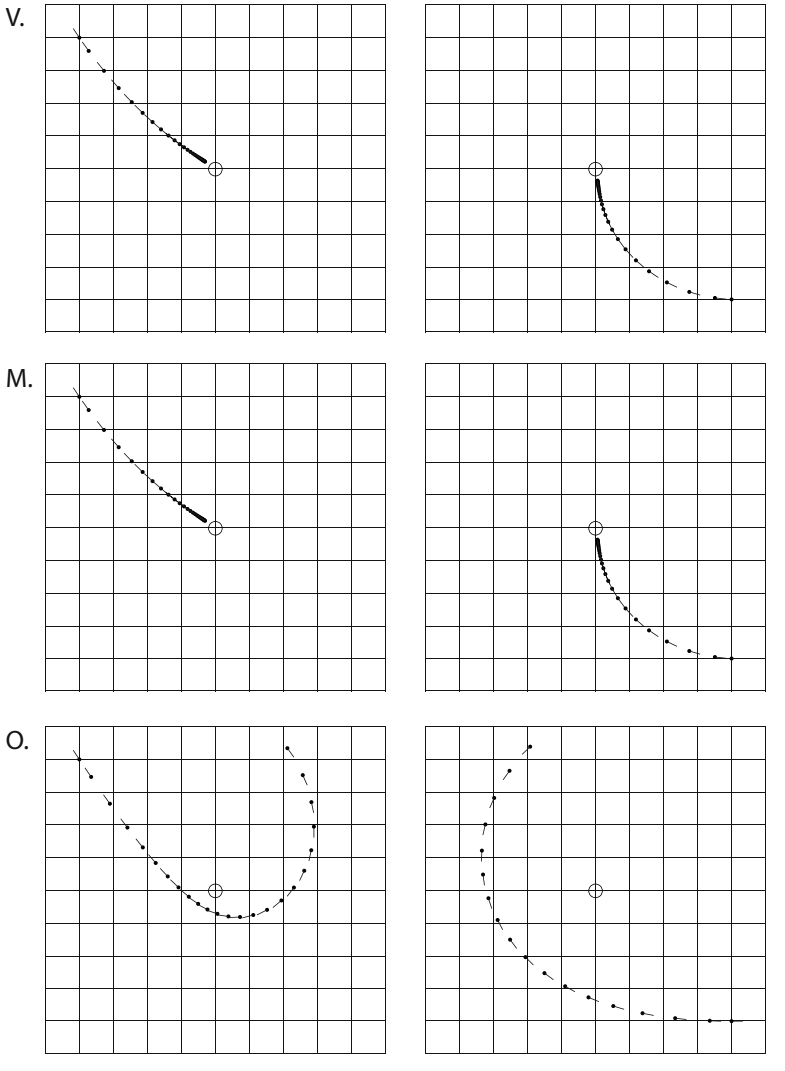


Fig. 4. The simulated behaviour of the combined model toward a visual (V), multimodal (M) and olfactory (O) stimulus. Although much fewer parameters are used, the behaviours are very similar to those in Fig. 2.

5 Discussion

We have presented a model of the approach behaviour of the hawkmoth *M. sexta* close to a flower stimulus. The model reproduces the behaviour of the moth for visual, olfactory and multimodal cues and is able to account for the interaction between the two modalities (Eq. 4). It was shown that a single linear model with six parameters could describe the behaviour of the moths in all stimulus conditions. This model only needed the location of target relative to the animal as input together with one signal that indicates the presence of a visual stimulus.

The model captures both that the velocity of the speed is higher when a visual stimulus is not present and that it decelerates as it approaches the target. In addition, the model describes that the moth only flies sideways when it can see the target and that a purely olfactory stimulus does not induce lateral movement. Finally, the model illustrates the differences in turning velocity toward the target between an olfactory and a visual stimulus.

Although our initial goal was to generate the model entirely from the data, it was clear that this was not possible. There were two factors that needed to be considered to obtain a satisfactory model. The first was that we needed to include a number of assumptions to limit the structure of the model. One of these was that the behaviour of the moth was symmetrical with respect to the direction to the flower. Without this assumption, the generated model would include a bias toward turning in one direction as a result of a corresponding bias in the collected data. Since this only reflects that a different number of animals approached the flower from the left or the right, it should clearly not be included in the model.

Another problem with the method used was that it was very prone to overfitting. When the higher order interaction terms were included in the model equations, these would generally be included by the automatic method to obtain a better fit to the collected data. However, this would dramatically decrease the generality of the model as it would no longer be able to reproduce the behaviour at arbitrary starting positions. These terms were consequently removed from the models equations which resulted in much simpler and more robust models. It is possible that the higher order terms could have been included if we had used a larger data set and this is something that will be investigated in the future.

Another extension would be to investigate non-linear model equations. Such equations will give a much larger set of possible models, but this too would require a much larger data set to determine if the more complex models would add anything to the explanation of the behaviour of the moths. In the future we would like to extend this work with a more realistic model of the sensory processing of the moth. This would allow us to study the influence of different visual configurations on the reliability of the estimation of θ_T and d_T. Some first steps toward this goal was taken in our previous model of learning in the moth [17]. We would also like to include the behaviour further away from the flower since the present model only addresses the final phase of foraging.

Acknowledgements

The support from The Swedish Research Council Formas and VR is gratefully acknowledged.

References

1. Andersson, S., Dobson, H.E.: Behavioral foraging responses by the butterfly *eliconius melpomene to Lantana camara* floral scent. J. Chem. Ecol. 29, 2302–2318 (2003)
2. Balkenius, A., Rosén, W., Kelber, A.: The relative importance of olfaction and vision in a diurnal and a nocturnal hawkmoth. J. Comp. Physiol. A 192(4), 431–437 (2006)
3. Stein, B., Stanford, T.: Multisensory integration: current issues from the perspective of the single neuron. Nature Reviews Neuroscience 9(4), 255–266 (2008)
4. Balkenius, A., Kelber, A.: Colour preferences influences odour learning in the hawkmoth, *Macroglossum stellatarum*. Naturwissenschaften 93(5), 255–258 (2006)
5. Balkenius, A., Bisch-Knaden, S., Hansson, B.: Interaction of visual and odour cues in the mushroom body of the hawkmoth manduca sexta. J. Exp. Biol. 212, 535–541 (2009)
6. Braitenberg, V.: Vehicles. MIT Press, Cambridge (1984)
7. Biró, Z., Ziemke, T.: Evolution of visually-guided approach behavior in recurrent artificial neural network robot controllers. In: Pfeifer, R., Blumberg, B., Meyer, J. (eds.) From animals to animats 5: proceedings of the Fifth International Conference on Simulation of Adaptive Behavior, pp. 73–77. MIT Press, Cambridge (1998)
8. Grasso, F., Consi, T., Mountain, D., Atema, J.: Locating odor sources in turbulence with a lobster inspired robot. In: Maes, P., Matarik, M., Meyer, J.A., Pollack, J., Wilson, S.W. (eds.) From Animals to Animats 4, vol. 4, pp. 104–112. MIT Press, Cambridge (1996)
9. Webb, B.: Robots, crickets and ants: models of neural control of chemotaxis and phonotaxis. Neural Networks 11(7-8), 1479–1496 (1998)
10. Webb, B.: Spiking neuron controllers for a sound localising robot. Dynamical Systems, Wave-Based Computation and Neuro-Inspired Robots, 119–130 (2008)
11. Balkenius, A., Dacke, M.: Flight behaviour of the hawkmoth manduca sexta toward unimodal and multimodal targets. (submitted, 2010)
12. Ljung, L.: System Identification: Theory for the User, 2nd edn. Prentice Hall PTR, Englewood Cliffs (1998)
13. Raguso, R., Pichersky, E.: A day in the life of a linalool molecule: chemical communicaion in a plant-pollinator system. Plant Species Biol. 14, 95–120 (1999)
14. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Proceedings of 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Colins, CO, vol. 2 (1999)
15. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience Publication, Hoboken (2000)
16. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models—their training and application. Comput. Vis. Image Underst. 61(1), 38–59 (1995)
17. Balkenius, A., Kelber, A., Balkenius, C.: Simulations of learning and behaviour in the hawkmoth deilephila elpenor. In: ICSAB: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior on From Animals to Animats, pp. 85–92. MIT Press, Cambridge (2002)

The Indiana Experiment: Investigating the Role of Anticipation and Attention in a Dynamic Environment

Birger Johansson[1] and Christian Balkenius[2]

[1] University of Technology, Sydney, Australia
m@birgerjohansson.com
www.birgerjohansson.com
[2] Lund University, Cognitive Science, Lund, Sweden
christian.balkenius@lucs.lu.se
www.lucs.lu.se/christian.balkenius

Abstract. We investigating the role of anticipation and attention in a dynamic environment in a number of large scale simulations of an agent that tries to negotiate a number of gates that continuously open and close. In particular we have looked at learning mechanisms that can predict the future positions of the gates and control strategies that will allow the agent to pass through the gates unharmed. The simulations reported below use the AARC architecture [1]. This architecture combines a large number of different cognitive mechanisms. In Experiment 1, the task for the agent is to pass through a single gate and in Experiment 2, to pass through three successive gates. The results shows that the AARC architecture is flexible enough to handle very diverse situations. It is also somewhat surprising that linear predictors are sufficient in most cases.

1 Introduction

Imagine yourself in the depths of the Peruvian jungle trying to avoid a number off traps inside an ancient temple. Any careless step will set off the traps or even trigger the occasional giant boulder. As doors open and close, you have to predict the right time to negotiate the deadly obstacles and your movements need to be timed with utmost precision. You need to decide how long to study the different obstacles before trying to move forward. Wait too long and the giant boulder will hit you from behind. Act too soon and you will not have a correct picture of the movements of the doors and risk getting crushed.

Even in less adventurous situations it is often necessary to adapt actions to dynamic objects in the environment, for example when crossing the street or while catching a ball. There are many cognitive mechanisms that need to work together in such a situation. We need to attend to the relevant aspects of the environment to learn about the dynamics of different objects. If many things are happening at once, it becomes necessary to choose where to look at each time. We can also perform epistemic actions of different kinds. An example would be

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 242–251, 2010.

to move around to get a better view of some object. A more advanced action would influence the environment in some way to increase our knowledge of it. Furthermore, we need to adapt our actions so that they interact appropriately with objects. This involves moving in phase with dynamic objects and producing the correct timing of movements.

Very little research has previously addressed the problem of prediction in dynamical environment although some work have been done within the RoboCup community [2] [3] [4] [5]. However, this work has focused on the behavior of other agents rather than on inherently dynamic environments. Similarly, within control engineering there is an extensive literature on tracking, but it is mainly directed towards pursuit of freely moving objects [6] [7] [8] [9] [10]. We have previously shown that anticipation can be of great benefit for a robot in situations that are too complicated for a purely reactive system to handle [11]. Anticipatory agents can have an advantage over systems that typically do not handle dynamic changes in the environment. Although anticipation may help agents to avoid future problems, more anticipation is not necessarily better [12]. If an agent anticipates too far into the future, there is a risk that it will see too many potential problems and avoid doing anything at all.

Here we investigate the cognitive mechanisms that are required to control movements in these types of situations and have performed a number of large scale simulations of an agent that tries to negotiate a number of gates that continuously open and close. In particular we have looked at learning mechanisms that can predict the future positions of the gates and control strategies that will allow the agent to pass through the gates unharmed. This also involves the control of attention in such a way that the agent can learn about the obstacles in an optimal way. The simulations reported below use the AARC architecture [1]. This architecture emphasizes the role of (A)nticipation and (A)ttention in (R)obot (C)ontrol and combines a large number of different cognitive mechanisms to produce behaviors that includes low-level motor control, path-planning, control of attention and anticipatory mechanisms that compensate for different types of system delays as well as anticipation of future states of the world. Currently, the largest implementation of the AARC architecture consists of more than 300 modules implementing different algorithms and 1000 connections between these modules. AARC is built on top of the Ikaros framework that provides an infrastructure for cognitive simulation and robot control [13]. AARC and Ikaros together offer a simple way to switch between experiments in real robots and pure simulations. The experiments reported below were run as large-scale simulations on the parallel computer Milleotto at the Center for Scientific and Technical Computing for Research at Lund University (LUNARC). Milleotto is an IBM blade-centre solution with a total of 1008 processor cores.

2 The Architecture

For the task in this paper, four parts of the AARC architecture are especially important: The visual part of the agent, the learning system that learns the

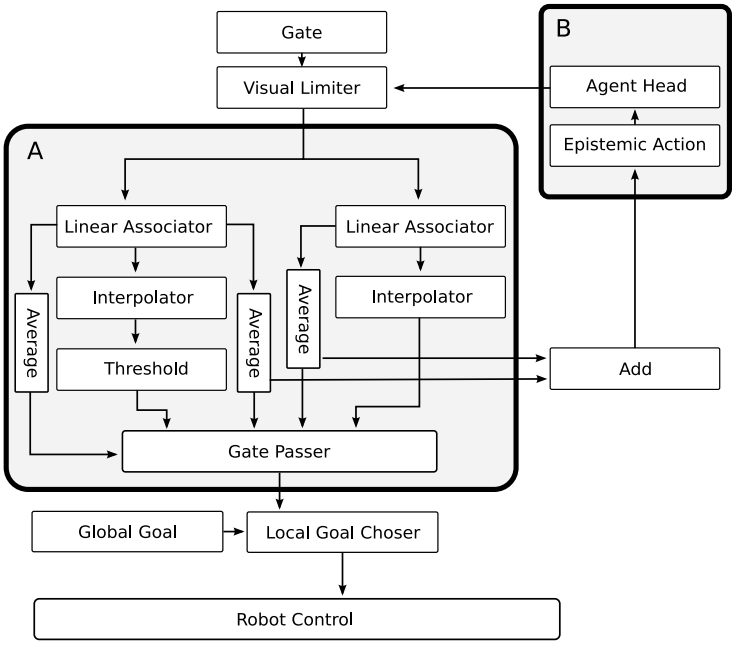


Fig. 1. Overview of the AARC architecture with the gate predictor (A) and epistemic action control (B)

behavior of each gate, the epistemic action control and the agent control system. These parts consist of a number of Ikaros modules and are connected as shown in Fig. 1. Only the most important connections are shown in this figure.

The behavior of each simulated gate is controlled by a gate module. These modules regulate the opening duration, the position of the gate and possible movements of the gate over time. The gate module can open the gates, either at a constant velocity or by following a sinusoidal function. The gate information is subsequently filtered through a visual filter module. The filter module conceals the parameters of the gate, if the agent is not currently able to see the gate due to obstacles or because it is attending a different area in the environment.

Part A of Fig. 1 shows the gate learning and the prediction part of the system. Two linear associators learn the position of the gate and how the gates open and close. This learning is based on a sequence of coordinates that describe the position and opening of the gates. The predicted positions of the gates are formed as linear combinations of these sequences. The learning of the linear associators can be adjusted by changing the learning rate α and the learning momentum β [14].

The predicted position and the predicted gate opening are then passed forward to the Gate Passer module. The linear associators calculate a prediction error between the predicted values and the actual values. The average prediction errors are also forwarded to the Gate Passer module. The gate's average error are also sent to the Epistemic Action Control system which is indicated in part B of Fig. 1.

The Epistemic Action Control directs the attention of the agent according to one of several attention strategies. Either it can direct the attention toward the next coming gate, the gate with the highest prediction error or the gate farthest away etc. The Epistemic Action module forwards the selected interest point in the agent's environment to the Agent Head module, which then tries to direct the attention towards this area. The agent's new head direction is then forwarded to the Visual Limiter. If the Epistemic Action module cannot find any interesting focus point in its surroundings, it will start to explore the world using random head movements.

All the inputs from the linear associators and the average modules are connected to the Gate Passing module. This module predicts how long it will take for the agent to get to the gate from its current position, and compares this to the predicted input, to decide if it possible to pass though the gate. The output from the Gate Passing module indicates if the gate can be passed from the current position together with an estimation of how certain this prediction is. All the modules in part A, together with the gate and visual limiter module, are duplicated for each gate that are present in the environment.

For each of the gates in the environment, part A is connected to the Goal Chooser module which determines a local goal for the agent. To decide on a local goal, it uses the global goal provided and divides it into subgoals for each gate. If the certainties of the gates are lower than the confidence threshold, the agent is not allowed to try to pass though the gates. Instead it will stop the agent and allow it to explore its surrounding further using the epistemic action control.

The control system uses the local goal position to steer the agent to this point. The control part is not shown in this overview but consist of a large number of modules like agent navigation, environment modeling and latency compensation and has been thoroughly described elsewhere [15]. The simulation of the agent is performed on a very low level where all the details of the motor control and the physical properties of the agent are included.

3 Experiment 1: A Single Gate

In experiment 1, the task for the agent was to move from left to right in a simple environment with a single gate that opens and closes at regular intervals Fig. 2A. The agent must learn to predict when the gate will be open and use this information to select when to pass. The agent can move freely in the environment, but has a specific goal on the other side of the gate that it attempts to reach. To observe the gate and learn about its behavior, the agent must learn about the movements of the gate through directing its attention toward it. Finally, the agent needs to decide when its model of the gate is sufficiently accurate to allow it to pass through without risk.

Simulation 1A investigated the role of the two learning parameters α and β on the success of the learning. During this simulation, the agent did not move around in the environment. Its only function was to look at the gate to try

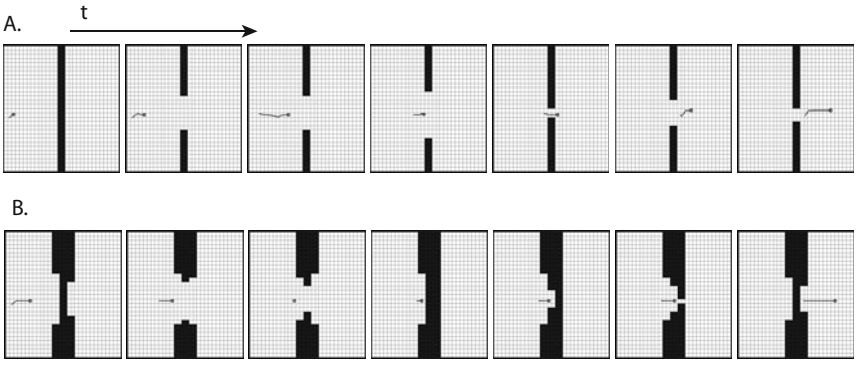


Fig. 2. The behavior of the simulated agent (circle) over time in (A) Experiment 1 and (B) Experiment 2. The black regions indicate the walls and the moving gates. See text for further explanation.

to predict how it would behave. Since the gates only opened and closed and did not move in this simulation, there is only a single value that needs to be predicted. The value of α was varied from 0.001 to 0.4096 with a doubling for each new session. The value of β was varied from 0.0 to 1.0 in steps of 0.1. Each data point represents the average of 20 trials with identical α and β, but with different initial position of the gate. There were 2860 trials in total.

Simulation 1B looked at the role of the gate confidence level γ. This parameter was varied from 0.001 to 1.024 with a doubling between each measurement. We measured the time of a single trial where the agent first had to learn the behavior of the gate and then had to pass to the other side. The maximal time allowed for a trial was 500 seconds. After that time, the trial was considered a failure if the agent had not passed the gate. The trial also failed if the agent did not manage to pass through the gate. We also measured the number of trials that succeeded for each confidence level. Each value is an average of 30 trials. There were 330 trials in total.

Simulation 1C tested how important the time margin δ was for the success on the task. We changed both the margin before and after passing the gate. As in experiment 1B, we measured the success rate and the average time of 20 trials for each time margin. The total number of trials was 1980.

Fig. 2A shows a typical simulation run for a single agent. The black regions show a single gate that opens and closes at regular intervals. As can be seen, the agent which is represented by the small circle approaches the gate in such a way that it will pass when it is open. The line connected to the agent shows the 100 previous positions of the agent.

The result of the first simulation are shown in Fig. 3. As can be seen, the values of α and β are not critical since the system learned the behavior of the gate very quickly in all but the borderline cases with very low α or β.

The second simulation showed that the confidence level did not influence the result very much over 0.2 (Fig. 3). With a lower value, the success rate decreases

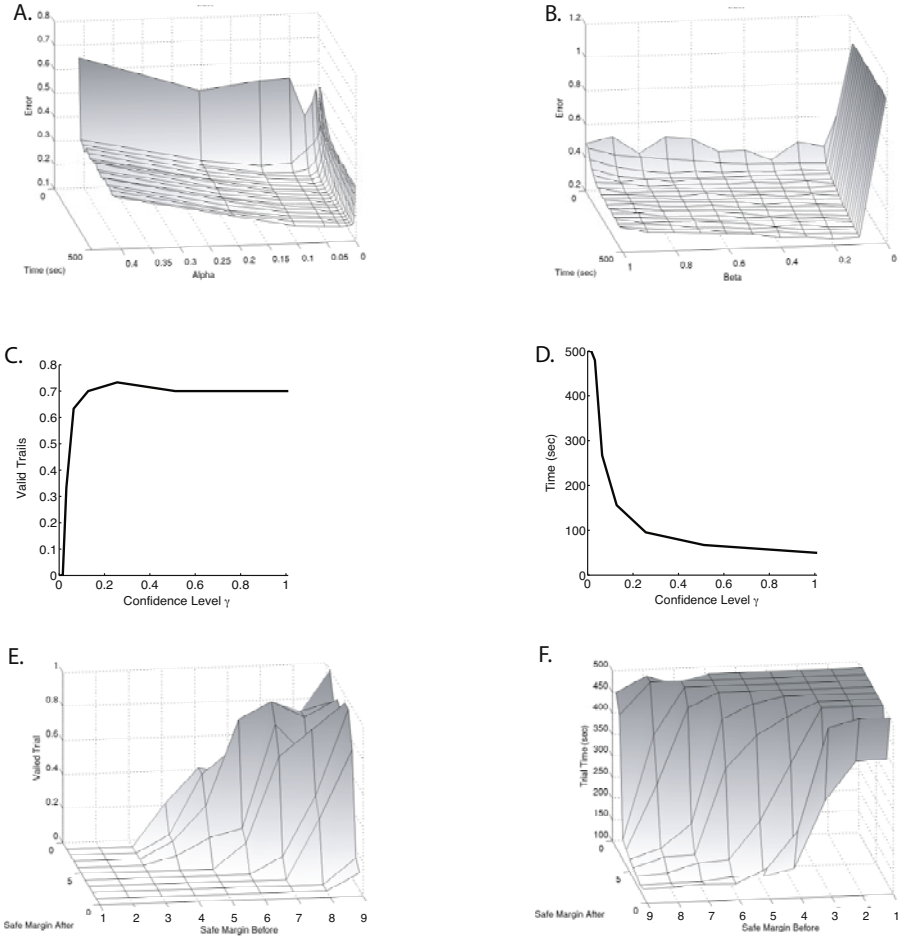


Fig. 3. A., B. The effect of different values of α and β on the learning of the single gate in Experiment 1. C. The trial time as a function of the confidence level γ. D. The success rate as a function of the confidence level γ. E. Number of valid trails for different safety margins before and after the gate. F. Trial time as a function of the safety margins before and after the gate.

quickly and the time for a trial gradually increases to its maximum level. Finally, the third simulation showed that the safety margins before and after the gate had similar influence on the success and timing of the task (Fig. 3).

4 Experiment 2: Three Successive Gates

The simulations in Experiment 2 followed a similar pattern as Experiment 1 with the important distinction that the agent had to pass three successive gates. This makes the problem much harder and shows the power of the suggested mechanisms.

Simulation 2A investigate the role of the two learning parameters α and β on the success of the learning. As in experiment 1, the value of α was varied from 0.001 to 0.4096 with a doubling for each new session. The value of β was varied from 0.0 to 1.0 in steps of 0.1. Each data point represents the average of 20 trials with identical α and β, but with different initial position of the gates. There were 2860 trials in total.

Simulation 2B looked at the role of the gate confidence level γ. This parameter was varied from 0.001 to 1.024 with a doubling between each measurement. Each value was tested in 30 trials.

Simulation 2C tested how important the time margin δ was for the success on this extended task. We changed both the margin before and after passing the gate. As in experiment 1CB, we measured the success rate and the average time of 20 trials for each time margin. The total number of trials was 1980.

Fig. 2B shows the typical behavior of the agent in the environment with three successive gates. The agent first approaches the gates, but then stops and waits until the positions of the three gates will allow the agent to pass. Note that the last two gates are not open when the agent enters, but as can be seen, the last three images, the gates open just before the agent passes.

As in experiment 1, the results are relatively insensitive to the values of $\alpha$and β. This is true for the first as well as the last gate (Fig. 4). In experiment 2B, the number of valid trials decreased with increased confidence threshold while the trial time decreased (Fig. 4). Finally in experiment 2C, it was shown that the safety margin made an abrupt change in the trial time above 5 for the 'before margin'. For a lower 'before margin', the trial time is much reduced regardless of the value for the 'after margin' (Fig. 4).

5 Discussion

We have shown in two simulation experiments how it is possible to use linear predictive systems to anticipate how one or several successive moving obstacles will behave which will allow a simulated robot to pass through the obstacles unharmed. This ability depends on an initial learning phase were the agent allocates it attentional resources in such a way that it is able to observe the behavior of the obstacles in an efficient way. We also shows how the agent is able

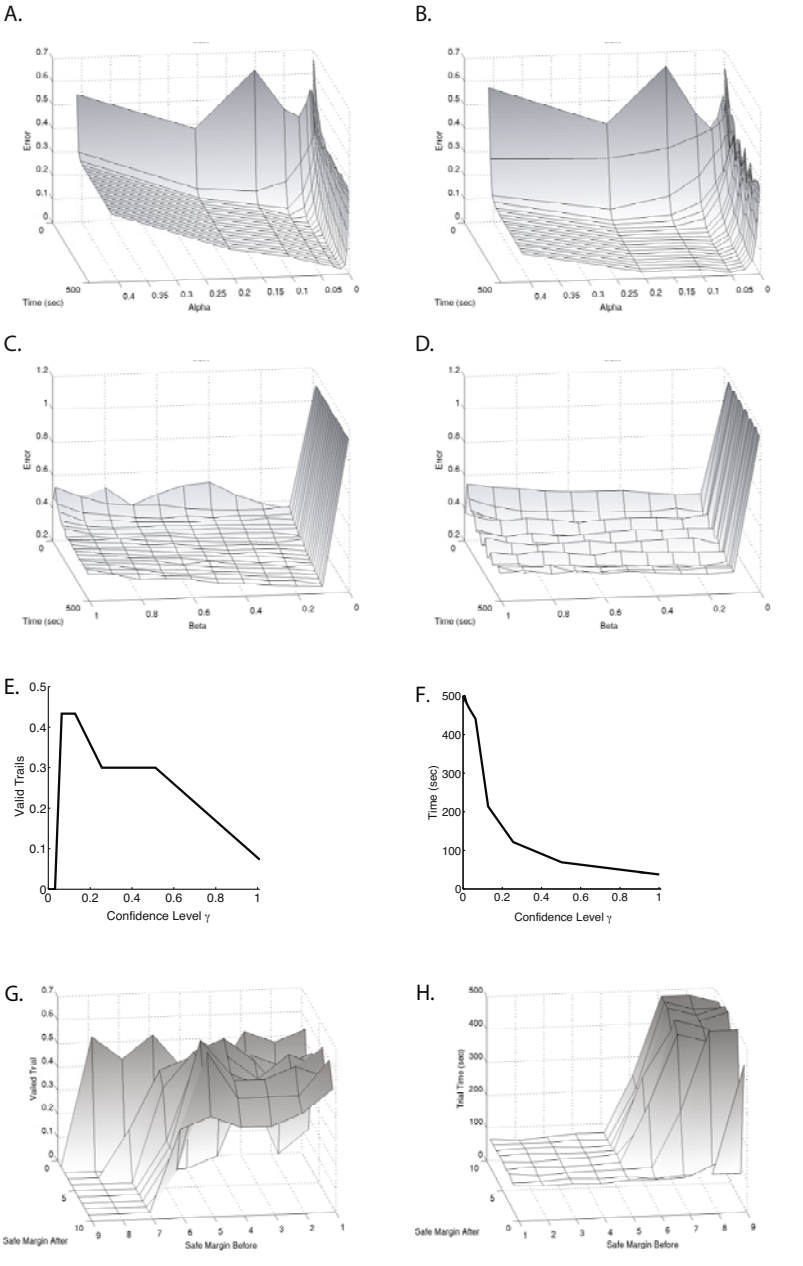


Fig. 4. The effect of different values of α and β on the learning of gate 1 (A and C) and gate 3 (B and D) in Experiment 2. E. The number of valid trials in experiment 2 for different confidence levels. F. The trial time as a function of the confidence level. The effect of different safety margins on the success in experiment 2. G. The number of valid trials with different safety margins. H. The trial time for different safety margins.

to measure how confident it is about the behavior of the obstacles and how this could guide attention.

Experiment 2B showed a similar results as experiment 1B for the trail time for different confidence levels. However, the number of valid trials is markedly lower. The number of valid trials also decreases with increased confidence level which can be contrasted with the result from experiment 1 where the success rate did not change with the confidence value.

It is also somewhat surprising that linear predictors are sufficient in most cases. However, in earlier experiments with gates that moved according to a triangle wave rather than a sinusoidal, the linear system did not work as well. Although it would be possible to use a non-liner system instead, we believe that a better solution is to add systems that detect changes in simple behaviors rather than building very large models (cf. [16]). For example, a triangular movement can be seen as two simple linear movements after each other. Each of these are simple to learn and if this is combined with a system for switching between models, it will become trivial to anticipate this type of movement.

In the future, we would like to investigate how the AARC architecture can be used in more complex situations involving many different types of moving obstacles as well as other goal directed agents. More challenging environments could be investigated by interfacing the system to different types of computer games [17].

In summary, we have shown how a robot control system can be designed that is able to autonomously learn about the properties of dynamic objects in the environment and accurately time its movement to avoid such objects.

References

1. Johansson, B.: Anticipation and Attention in Robot Control. PhD thesis, Lund University (2009)
2. Ball, D., Wyeth, G.: Modeling and exploiting behavior patterns in dynamic environments. In: IEEE/RSJ International Conference Intelligent on Robots and Systems, vol. 2, pp. 1371–1376 (2004)
3. Barakova, E.: Prediction of rapidly changing environmental dynamics for real time behavior adaptation using visual information. In: Würtz, L.M. (ed.) Proceedings of the 4th Workshop on Dynamic Perception, Bochum, Germany, pp. 147–152. IOS Press, Amsterdam (2002)
4. Sharifi, M., Mousavian, H., Aavani, A.: Predicting the future state of the robocup simulation environment: heuristic and neural networks approaches. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 32–37. IEEE, Los Alamitos (2003)
5. Veloso, M., Stone, P., Bowling, M.: Anticipation as a key for collaboration in a team of agents: A case study in robotic soccer. In: Schenker, P.S., McKee, G.T. (eds.) Proceedings of SPIE Sensor Fusion and Decentralized Control in Robotic Systems II, Bellingham, vol. 3839, pp. 134–143 (1999)
6. Ding, Z., Hong, L.: An interacting multiple model algorithm with a switching markov chain. Mathematical and Computer Modelling 25(1), 1–9 (1997)

7. Efe, M., Atherton, D.P.: Maneuvering target tracking with an adaptive kalman filter. In: Proceedings of the 37th IEEE Conference on Decision and Control, vol. 1 (1998)
8. Bando, T., Shibata, T., Doya, K., Ishii, S.: Switching particle filters for efficient visual tracking. Robotics and Autonomous Systems 54(10), 873–884 (2006)
9. Bar-Shalom, Y.: Recursive tracking algorithms: from the kalman filter to intelligent trackers for cluttered environment. In: Proceedings of IEEE International Conference on Control and Applications, ICCON 1989, pp. 675–680 (1989)
10. Morelande, M.R., Challa, S.: Manoeuvring target tracking in clutter using particle filters. IEEE Transactions on Aerospace and Electronic Systems 41(1), 252–270 (2005)
11. Johansson, B., Balkenius, C.: An experimental study of anticipation in simple robot navigation. In: Butz, M.V., Sigaud, O., Pezzulo, G., Baldassarre, G. (eds.) ABiALS 2006. LNCS (LNAI), vol. 4520, pp. 365–378. Springer, Heidelberg (2007)
12. Johansson, B., Balkenius, C.: Prediction time in anticipatory systems. In: Pezzulo, G., Butz, M.V., Sigaud, O., Baldassarre, G. (eds.) ABiALS 2008, vol. 5499, pp. 283–300. Springer, Heidelberg (2009)
13. Balkenius, C., Morén, J., Johansson, B., Johnsson, M.: Ikaros: Building cognitive models for robots. Advanced Engineering Informatics 24(1), 40–48 (2010)
14. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons, New York (2001)
15. Johansson, B., Balkenius, C.: Learning to anticipate the movements of intermittently occluded objects. In: Schlesinger, M., Berthouze, L., Balkenius, C. (eds.) Eighth International Conference on Epigenetic Robotics, Lund University Cognitive Studies, vol. 139 (2008)
16. Prem, E., Hörtnagl, E., Dorffner, G.: Growing event memories for autonomous robots. In: Proceedings of the Workshop on Growing Artifacts that Live: Basic Principles and Future Trends (2002)
17. Laird, J.E.: It knows what you're going to do: Adding anticipation to a quakebot. In: Proceedings of the Fifth International Conference on Autonomous Agents, Canada, pp. 385–392. ACM Press, New York (May 2001)

Attentional Mechanisms for Lateral Line Sensing through Spectral Analysis

Otar Akanyeti, Camilla Fiazza, and Paolo Fiorini

Department of Computer Science, University of Verona, Italy

Abstract. For autonomy in underwater robotics it is essential to develop context-driven controllers, capable of leading from perception to action without human intervention. One of the key challenges in this area is to extract reliable information from noisy sensor signals in a fast and efficient manner. In this context, we present a novelty-detection mechanism for lateral line sensing; this mechanism is meant to highlight interesting stimuli and separate them from the background, by bringing into focus new frequencies appearing in the environment. The method is fast and computationally cheap; additionally, it paves the way for characterization and classification of detected novelties. We present a testing framework to explore how to integrate frequency-related, temporal and spatial information and we demonstrate the viability of this approach in a multiple dipole-source environment.

1 Introduction

Sonar and vision systems for underwater robots can cover many operational niches but are of limited applicability in close-distance, low-visibility scenarios or when awareness of currents is key. Studies in hydrodynamic imaging and information extraction from artificial lateral lines [1, 2, 3] stand out as an alternative, inspired by the main sensing organ of fish.

The lateral line is a distributed sensor system composed of specialized cell sensors known as neuromasts. Fish are equipped with two types of sensors (canal and surface neuromasts), which respectively vehicle pressure and flow information. The sensing properties of a neuromast are critically dependent on its gelatinous cap, the cupula, which serves as the mechanical link between the environment and the neuromast itself.

There has been significant effort in understanding how fish exploit the lateral line to recognize different flow regimes, differentiate between prey and predators and interact with vortices to harvest energy in organized turbulence. Such understanding may lead to developing fast and efficient bio-inspired signal processing algorithms – and thus to providing autonomous underwater vehicles with real-time contextual guidance in complex environments.

So far, lateral line research has been focused on detecting and characterizing the flow field generated by a constant-volume sinusoidally vibrating sphere (localization [4, 5, 6, 7], object recognition [8]). The vibrating sphere emulates the beating tail of aquatic animals that generate dipole-dominant flow fields ([9]).

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 252–262, 2010.

The scenarios generally investigated consist of a single dipole source excited in still water. The lateral line organ is simulated as an array of sensors fixed in place – most often as a linear array, although 3D spatial arrangements have also been explored ([10]). Considerations on how the environment reflects on signal propagation from source to sensors are generally omitted in theoretical studies; experimental studies so far place artificial sensors only in still water.

On the other hand, in real-world scenarios such as the deployment of an underwater robot, the signal detected by the lateral line is heavily influenced by flow alterations, related to: i) other objects moving nearby, ii) effects of the fish's own movement and iii) the environmental currents in the surrounding water. Such "background" signals collectively behave as highly non-linear interference, which makes direct analysis of lateral line signals rather complex, if not impossible in real time. Therefore, intermediate signal processing mechanisms are needed to separate interesting information from the rest, for further elaboration. Filtering hydrodynamic background signals is not a straightforward task and there are no established techniques available as of now.

1.1 Problem Definition: Filtering Background Signals

In this paper, we address the background filtering problem from a control-related point of view. We interpret as the background the perception of fish when fish-flow interaction in a given environment is in a steady state (stable) — eg. fish Kármán gaiting in Kármán streets ([11]) or free swimming in steady flows. Our goal is highlighting where sensor signals start differing from the expected ones. Such differences are relevant from a control perspective, as they suggest some variations in fish-flow interaction — eg. fish losing stability in Kármán streets or a new object appearing in the proximity of a steadily-swimming fish. In these circumstances, further processing is necessary to select the appropriate control action.

Within this context, we instantiate a novelty-detection approach as a method for filtering background signals. Our method consists of two phases: i) initializing the filter to the background by modeling "common" perception, and ii) highlighting any perception that differs significantly from the model. The proposed method is inspired by novelty detection in animals; however, it should be considered as an engineering solution, rather than an attempt to model a biological system.

In general, attentional mechanisms are used to select particular inputs of interest from the sensor signals, thus reducing the computational cost and the complexity of dealing with the world. Recent studies in fish neural circuitry suggest that fish may employ similar mechanisms for filtering out the effects of their own motion. For instance, the octavolateralis efferent system in the central nervous system functions as a part of a feedback or feed-forward control system, which modifies the sensitivity of the lateral line to flows created by the animal's own movement ([12, 13]).

2 Novelty Filter

Sensor readings during stable fish-flow interactions usually exhibit a high degree of periodicity — eg. stable vortex shedding frequency in schools and in Kármán streets. Thus, we find it advantageous to model common perception in the frequency domain. This choice is encouraged by recent studies on fish, suggesting that the lateral line organ is mechanically engineered to detect frequency-related information and that fish neural processing is attuned to frequency information. The filtering properties of neuromasts can vary significantly depending on the size, shape and elastic properties of their cupulae. Resonating cupulae with high stiffness are particularly interesting, because they can be tuned to detect narrow frequency bands. Lateral line canals also behave as high-pass filters ([14, 15]).

Initializing the novelty filter: In Phase I, for each sensor i on the lateral line, we observe the input signal for T_1 seconds and consider it on $\bar{N}$ subintervals of equal duration; we compute the frequency spectrum on each subinterval and calculate the mean $m_i(f)$ and standard deviation $\sigma_i(f)$ of each frequency in the $\bar{N}$ spectra. In this study we consider spectral amplitude only. The mean $m_i(f)$ and standard deviation $\sigma_i(f)$ become the coefficients of the novelty filter; $m_i(f)$ describes the expected frequency spectrum for common perception and $\sigma_i(f)$ indicates how much deviation from $m_i(f)$ occurs due to uncertainties in lateral line sensing (environmental variations, measurement noise, etc.).

Novelty detection: After identifying the filter coefficients, in Phase II we highlight where current perception differs from the model. The novelty detection mechanism is represented in Figure 1.

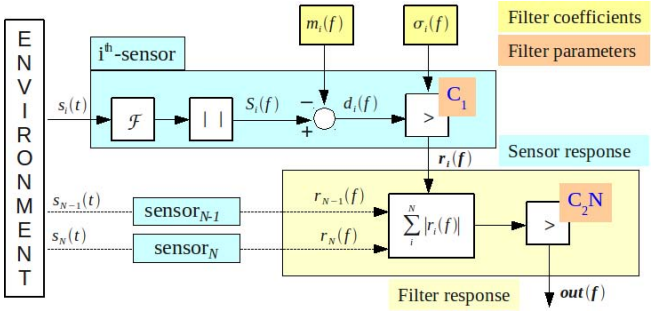


Fig. 1. The novelty detection schema. The i-th sensor computes $|r_i(f)|$ as $(||F[s_i(t)]| - m_i(f)| \geq C_1\sigma_i(f))$, whereas the overall filter calculates $out(f)$ as $\left(\sum_{k=1}^{N} |r_k(f)| \geq C_2 N\right)$. The sign of $r_i(f)$ follows that of $(|F[s_i(t)](f)| - m_i(f))$.

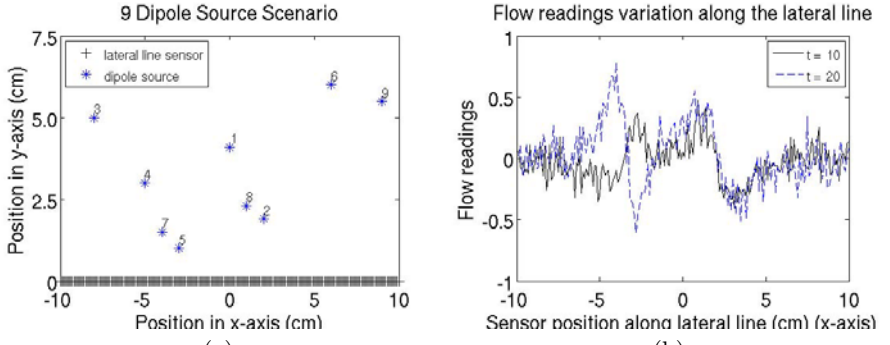
For each sensor i, we first compute the difference $d_i(f)$ between the amplitude of the frequency spectrum $S_i(f)$ of the current input signal and the expected spectrum $m_i(f)$. If $d_i(f)$ is larger than $C_1\sigma_i(f)$, the sensor fires $(|r_i(f)| = 1)$,

indicating that something unusual has been detected at frequency f. Finally, if the number of sensors firing at f exceeds C_2 percent of the total number of sensors, the filter agrees on novelty ($out(f) = 1$).

The threshold parameters C_1 and C_2 determine the sensitivity of the novelty filter. In environments with a high degree of uncertainty, choosing large C_1 and C_2 helps to reduce false novelty detections but may also lead to missing some novel stimuli.

3 Materials and Methods

We test our novelty-detection method through experiments in MATLAB. All experiments are carried out within the xy-plane (2D). We choose the randomly generated 9 dipole-source scenario in Figure 2a as the experimental environment. Each dipole source oscillates sinusoidally around a fixed position; the parameters of each dipole source are shown in Table 1. We consider the flow field generated by the 9 dipole sources as the background. Changes in the characteristic parameters of a dipole source in the scenario (position or angle of vibration) and addition or removal of a dipole source into or from the environment are considered events leading to a novel situation.



(a) (b)

Fig. 2. The 9 dipole source scenario (left). Positions are random within an L-by-7.5cm rectangular region. Flow readings ν_r along the lateral line at two different time stamps (right).

3.1 A Testing Framework for Information Extraction

In this paper we also propose a framework to combine novelty detection with temporal and spatial information. We test novelty detection in four different experimental situations (adding or removing a dipole source, relocating it, changing its direction of oscillation) and systematically explore how different events reflect in the sensor picture. We also wish to relate spatial position of the source involved in the novel event and number of sensors detecting novelty; furthermore, we test the spatial thresholds of sensitivity for novelty detection of the individual events.

Table 1. Diameter (a), position (x_0, y_0), oscillation frequency (f) and angle (α) for each of the dipole sources in the experimental scenario of figure 2a

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
x_0 (cm)	0	2	-8	-5	-3	6	-4	1	9
y_0 (cm)	4.1	1.9	5	3	1	6	1.5	2.3	5.5
α (deg.)	0	90	45	0	30	10	5	70	25
f (hz)	10	15	20	50	55	60	90	95	100
a (cm)	1	0.94	0.89	0.52	0.48	0.50	0.2	0.25	0.19

The combined information leads to successful 2D localization of a source of novelty that is moving in space. The overall goal is organizing parallel processing of sensor data to relate spatio-temporal information to frequency information and build a coherent picture from the sensor stream.

3.2 Simulating the Sensor Readings in the Dipole Source Scenario

We focus on modeling the flow parallel to the lateral line (x-direction), to simulate the sensor inputs of surface neuromast. We consider a sphere, initially placed in (x_0, y_0), which oscillates in still water in the plane of the lateral line and moves with $\mu(t) = \frac{\rho w^2 a^3}{y_0^3} sin(wt)$; here ρ is the density of water, a is the diameter of the sphere, w is the pulse and t is the time stamp. The flow field ν generated by the sphere, for an arbitrary axis of vibration making an angle α with the x-axis, is described by [16]:

$$\nu(x, y, t) = \nu_{\parallel}(x, y = 0, t)cos(\alpha) + \nu_{\perp}(x, y = 0, t)sin(\alpha) \tag{1}$$

Here $\nu_{\parallel}$ and $\nu_{\perp}$ are, respectively, the flow field functions — as read along the x-axis— in case of parallel ($\alpha = 0$) and perpendicular ($\alpha = 90$) dipole vibration with respect to the lateral line:

$$\nu_{\parallel}(x, y, t) = -\frac{\mu(t)}{4\pi}\left[\frac{(y - y_0)^2 - 2(x - x_0)^2}{[(x - x_0)^2 + (y - y_0)^2]^{\frac{5}{2}}} + \frac{(y + y_0)^2 - 2(x - x_0)^2}{[(x - x_0)^2 + (y + y_0)^2]^{\frac{5}{2}}}\right],$$

$$\nu_{\perp}(x, y, t) = \frac{3\mu(t)}{4\pi}\left[\frac{(y - y_0)(x - x_0)}{[(x - x_0)^2 + (y - y_0)^2]^{\frac{5}{2}}} - \frac{(y + y_0)(x - x_0)}{[(x - x_0)^2 + (y + y_0)^2]^{\frac{5}{2}}}\right] \tag{2}$$

In order to scale up to the 9 dipole source scenario, the overall flow field ν_r is obtained as the superposition of the individual flow fields. To make data more realistic, we also introduce white noise $\hat{n}$ with SNR = 10 into the resultant flow readings. If ν_j is the flow generated by the j-th dipole source, then $\nu_r(t) = \sum_{j=1}^{9} \nu_j(t) + \hat{n}(t)$ along a lateral line of length $L = 20cm$. We consider $N = 201$

equally spaced sensors; intersensor distance is $l = 0.1cm$. The lateral line is fixed in space and its center located at $(0, 0)$. The sampling rate of the sensors is chosen as 200Hz, so as to capture the frequency band up to 100Hz; a sampling rate of 200Hz is biologically plausible ([7]). Figure 2b illustrates flow readings along the lateral line at two different time stamps ($t = 10, t = 20$).

Modeling common perception through spectral analysis. To model "usual" perception in the 9 dipole-source scenario, for each sensor i we observe the input readings for $10s$ and consider the signal on 10 subintervals of equal duration ($1s$ each). For each subinterval, we compute the amplitude of the frequency spectrum. We sample the resulting frequency band ($0 - 100$Hz) with 100 points (uniform frequency spacing with 1Hz intervals). Finally, we calculate the mean $m_i(f)$ and standard deviation ($\sigma_i(f)$) of the 10 spectra. Figure 3 illustrates the $m(f)$ values of three lateral line sensors (s_{51}, s_{101} and s_{151}). Note that the expected spectrum varies significantly across sensors, depending on which dipole sources lie in the proximity. For instance, the vibration frequency of dipole source d_5 ($f = 55$Hz) is detected easily by s_{51} but not by s_{151}, as s_{51} is much closer to d_5 than s_{151}.

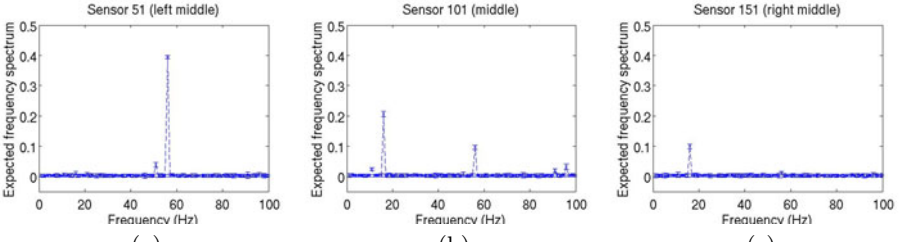


Fig. 3. The expected frequency spectrum $m_i(f)$ for sensor 51 (a), sensor 101 (b) and sensor 151 (c). Note that sensor 1 is located at the left extremity of the lateral line.

4 Experiments and Results

After modeling the background, we test our novelty filter in four different test conditions: i) when a new dipole source ($x_0 = 4cm$, $y_0 = 2cm$, $\alpha = 0°$, $f = 2$Hz and $a = 0.78cm$) is introduced, ii) when dipole source d_7 is removed, iii) when dipole source d_1 is moved to a different position and iv) when the angle of vibration of d_1 is changed. We chose the threshold parameters of the filter as $C_1 = 3$ and $C_2 = 10\%$; thus, a minimum of 20 sensors simultaneously reporting novelty are needed to reach a consensus on novelty for a particular frequency. Figure 4 presents the response of the novelty filter in each scenario, according to the processing steps described in Figure 1.

Exp 1. Introducing a new dipole source: Figure 4 (top left) presents a histogram showing the number of sensors that output 1 at each frequency,

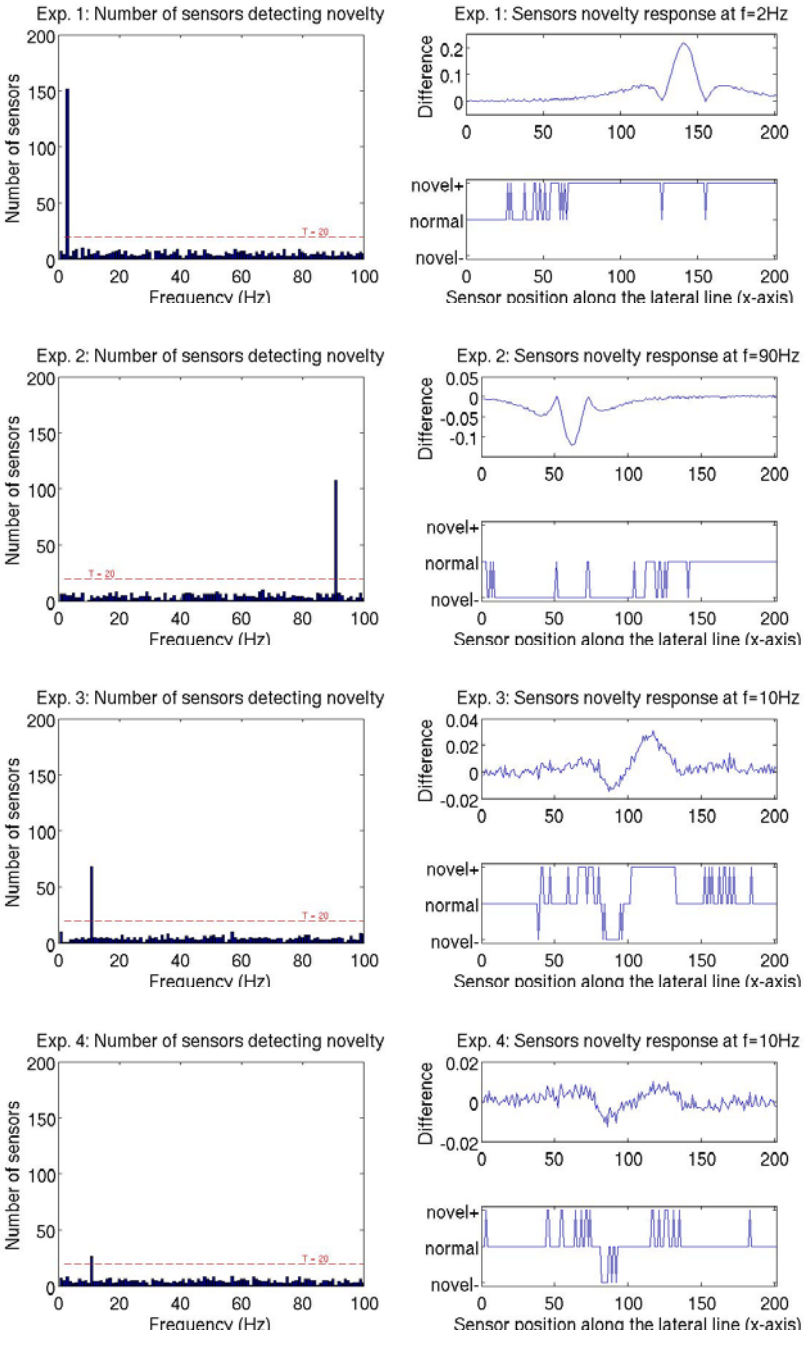


Fig. 4. Exp. 1: source addition (first row); Exp. 2: source removal (second row), Exp. 3: source displacement (third row), Exp. 4: source rotation (last row). The red dashed line shows the minimum number of sensors required to agree on novelty.

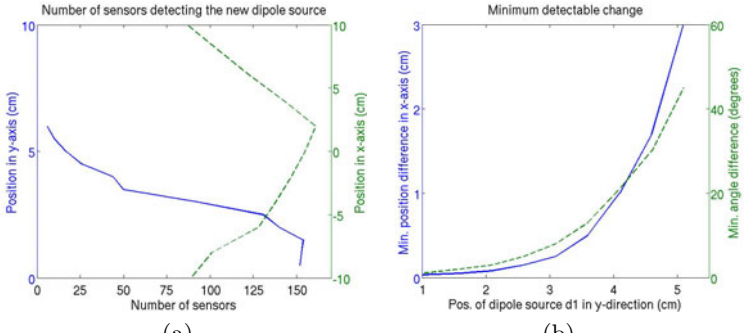
when the new dipole source has been introduced. At $f = 2$Hz, the number of sensors exceeds 20, resulting in novelty detection. Figure 4 (top right) shows the behavior of each sensor at $f = 2$Hz: the difference between the current and expected frequency spectrum (above) and the response $r_i(2)$ of each sensor (below).

We also examine the relationship between the position of the new dipole source and the number of sensors detecting it. We change the position of the dipole source and count the number of sensors with positive response. When moving the dipole source away from the lateral line (along the y-axis), fewer sensors fire —the influence of the dipole source on the lateral line decreases with distance. After $5cm$, the number drops below 20 and the filter stops detecting novelty. When moving the dipole source along the x-axis, the highest number of active sensors (153 sensors) is obtained when the dipole source is close to the center of the lateral line; as expected, the effective area of the dipole source is maximized at the center. The number of active sensors decreases symmetrically as we move away from the center (Figure 5a).

Exp 2. Removing dipole source d_7: After removing dipole source d_7 from the environment, the filter detects novelty at $f = 90$Hz (Figure 4 (second row)). The negative difference indicates that the frequency component at 90Hz is not as strong as it used to be, suggesting that a source is no longer present.

Exp 3. Changing the position of dipole source d_1: When dipole source d_1 is moved from $(0, 4.1)$ to $(1, 3.5)$, the filter responds with a peak at $f = 10$Hz (Figure 4 (third row)). To measure the sensitivity of the filter to minimum positional difference, we first change the position of d_1 gradually in the x-direction. The filter first detects a novelty at a distance of $1cm$. With d_1 closer to the lateral line, the filter detects smaller differences, as the intensity of the dipole stimulus increases with proximity. (Figure 5b).

Exp 4. Changing the angle of vibration of d_1: We also evaluate the performance of the filter by changing the oscillation angle of d_1. The minimum



(a) (b)

Fig. 5. Position of the new dipole source versus the number of sensors with positive response (left). The minimum positional and angular change that can be detected by the filter depending on the position of the new dipole source (right).

angular difference detected by the filter is 22°, when d_1 is positioned at $x = 0$, $y = 4.1$ (Figure 4, bottom left). Again, the closer the dipole source is, the higher the sensitivity (Figure 5b).

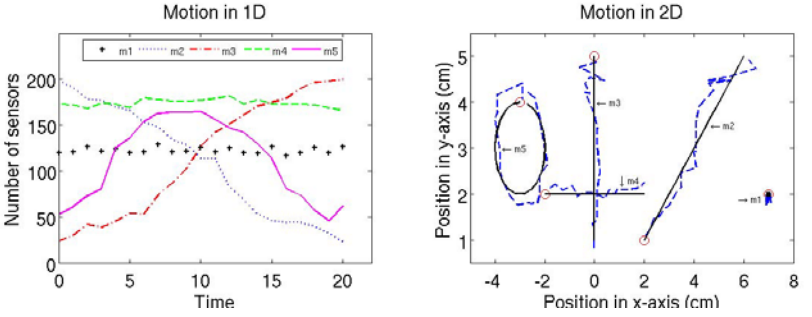
5 Characterization of the Detected Novelty

Highlighting novel frequencies as soon as they appear in the environment is quite useful for fish. For instance, fish entering into a Kármán street can harvest energy by synchronizing the tail beat frequency with the vortex shedding frequency ([11]). Moreover, a preliminary classification between prey and predators can be made on the basis of frequency by interpreting a novel frequency as the tail beat frequency of an incoming moving agent (fish). The tail beat frequency reveals information about the size and family of a fish —for example, for subcarangiform swimmers body length and tail beat frequency are inversely proportional ([17]).

In the case of Experiment 1 and 2, the difference between the current and expected frequency spectrum in the novelty filter (Figure 4, right column) is equivalent to the flow field generated by the added/removed dipole source. This flow field is a well-defined one-to-one function, which depends on (x_0, y_0, α) (see Section 3). Therefore, pattern recognition techniques (eg. RBF networks, SOM, support vector machines, or other) can be used to identify a dipole source and estimate its position concurrently. Additionally, assuming that the detected novelty is to be identified as a dipole source, several methods to estimate position have been proposed in the literature; examples are wavelet decomposition ([5]) or analysis of the main features (local minima, maxima and zero crossing points) of the detected signal ([4]). For instance, in Exp. 1 the sensor with highest novelty response (s_{141}) indicates the position of the dipole source in the x-direction, whereas the distance between the two sensors detecting local minima (s_{128} and s_{155}) is proportional to the position in y-direction (Figure 4 (top, right)).

Motion detection: To detect the motion of a dipole source, one possibility is estimating its position continuously using one of the techniques just mentioned. An alternative and simpler technique exploits the novelty filter: to detect motion along the y-axis, we analyze the number of sensors with positive response, as a function of time. If the number is approximately constant, we deduce that the object is at same distance, whereas gradual increments suggests that the object is moving towards the lateral line. Figure 6a illustrates the trends in sensor number for five different types of dipole source motion: i) stationary (m1), ii) moving away from the lateral line (m2), iii) approaching the lateral line (m3), iv) moving parallel to the lateral line (m4) and iv) moving in a closed loop trajectory (circle, m5).

To detect motion along the x-axis, we also track the position of the sensor experiencing the strongest difference between actual and expected frequency content. To estimate motion along the y-axis, we rely on the relation between y-position of a novel dipole source and number of sensors detecting novelty, derived from Experiment 1 (Figure 5a); the relation is injective and its inverse can be employed to yield an estimate of distance. The combined estimates for position in x and y yield a rough estimate for the 2D trajectory of a moving object in the field of view of the lateral line (Figure 6b).



(a) (b)

Fig. 6. Total number of sensors with positive response in case of a moving dipole source (left). Extracting the 2D trajectory of the moving dipole source (right). Red circle, dashed and continuous lines are, respectively, the starting position, the estimated trajectory and the actual trajectory of the dipole source. Position in the y-axis is estimated though the number of active sensors, by exploting the implicit relation given in Figure 5a.

6 Conclusion and Future Work

In this paper we have presented a method for processing information from a lateral line sensor array, combining novelty detection with event characterization. Signals are processed in parallel to highlight significant events on the basis of frequency information; interesting stimuli are thus separated from their hydrodynamic background and can be further explored in terms of space and time information. The main advantage of the proposed method lies in reducing the complexity of the signals to process. We show that our method leads naturally towards characterization of the detected novelties and propose a testing framework to systematically explore the link between events and sensor readings. We successfully tested our method in a multi-dipole source environment and demonstrated the usefulness of our approach by achieving 2D localization of a moving source of novelty.

Future work: We are currently investigating scenarios in which the lateral line is not fixed in place and can also be subject to changes in shape, in order to reproduce the bending curve of fish. Preliminary results favor our method. The fact that fish employ different locomotion patterns in different flow regimes also leads to further interest in characterizing highlighted novelties in terms of flow changes. We are also working on improving our method by introducing phase analysis of the frequency spectrum. Phase information can be useful in detecting flow direction and speed in streams, in localizing nearby objects and in synchronizing with the vortices in Kármán streets. Finally, we are also developing an artificial lateral line to test our method in real-world experiments.

This work and its proposed developments form part of the ongoing research in project FILOSE (Fish LOcomotion and SEnsing), supported by the European Union under the Seventh Framework Programme (FP7-ICT-2007-3).

References

[1] Fernandez, V.I., Hou, S.M., Hover, F.S., Lang, J.H., Triantafyllou, M.S.: Lateral-Line Inspired MEMS-Array Pressure Sensing for Passive Underwater Navigation. Technical Report, MIT, Cambridge (2007)
[2] Yang, Y., Chen, J., Engel, J., Pandya, S., Chen, N., Tucker, C., Coombs, S., Jones, D.L., Liu, C.: Distant touch hydrodynamic imaging with an artificial lateral line. In: Proceedings of the National Academy of Sciences of the United States of America (2006)
[3] Martiny, N., Sosnowski, S., Khnlenz, K., Hirche, S., Nie, Y., Franosch, J.P., van Hemmen, J.L.: Design of a Lateral-Line Sensor for an Autonomous Underwater Vehicle. In: Preprints of the 8th IFAC International Conference on Manoeuvring and Control of Marine Craft, Brazil, pp. 292–297 (2009)
[4] Franosch, J.P., Sichert, A.B., Suttner, M.D., van Hemmen, J.L.: Estimating position and velocity of a submerged moving object by the clawed frog Xenopus and by fish—A cybernetic approach. Journal of Biological Cybernetics 83, 231–238 (2005)
[5] Blake, B.C., van Netten, S.M.: Source location encoding in the fish lateral line canal. The Journal of Experimental Biology 209, 1548–1559 (2006)
[6] Goulet, J., Engelmann, J., Chagnaud, B.P., Franosch, J.P., Suttner, M.D., van Hemmen, J.L.: Object localization through the lateral line system of fish: theory and experiment. Journal of Computational Physiology 194, 1–17 (2008)
[7] Coombs, S., Patton, P.: Lateral line stimulation patterns and prey orienting behaviour in the Lake Michigan mottled sculpin (Cottus bairdi). Journal of Computational Physiology 195, 279–297 (2009)
[8] Sichert, A.B., Bamler, R., van Hemmen, J.L.: Hydrodynamic object recognition. Physical Review Letters (2009)
[9] Kalmijin, A.J.: Hydrodynamic and acoustic field detection. In: Atema, J., Fay, R.R., Popper, A.N., Tavolga, W.N. (eds.) Sensory Biology of Aquatic Animals, pp. 83–130. Springer, New York (1988)
[10] Yang, Y., Nguyen, N., Chen, N., Lockwood, M., Tucker, C., Hu, H., Bleckmann, H., Liu, C., Jones, D.L.: Artificial lateral line with biomimetic neuromasts to emulate fish sensing. Bioinspiration and Biomimetics 5(1), 016001 (2010)
[11] Liao, J.C., Beal, D.N., Lauder, G.V., Triantafyllou, M.S.: The Kármán gait: novel body kinematics of rainbow trout swimming in a vortex street. The Journal of Experimental Biology 206, 1059–1073 (2003)
[12] Roberts, B.L., Meredith, G.E.: The efferent system. In: Coombs, S., Görner, P., Münz, H. (eds.) In the Mechanosensory Lateral Line: Neurobiology and Evolution, pp. 445–460. Springer, Heidelberg (1989)
[13] Coombs, S.: Smart skins: Information processing by lateral line flow sensors. Autonomous Robots 11, 255–261 (2001)
[14] McHenry, M.J., Strother, J.A., van Netten, S.M.: Mechanical filtering by the boundary layer and fluid-structure interaction in the superficial neuromast of the fish lateral line system. Journal of Computational Physiology 194, 795–810
[15] van Netten, S.M.: Hydrodynamic detection by cupulae in a lateral line canal: functional relations between physics and physiology. Biol. Cybern. 94, 67–85
[16] Lamb, H.: Hydrodynamics, 6th edn. Cambridge University Press, Cambridge (1932)
[17] Wu, T.Y.: Introduction to the scaling of aquatic animal locomotion. In: Pedley, T.J. (ed.) Scale effects in animal locomotion, pp. 203–232. Academic Press, London (1977)

BeeIP: Bee-Inspired Protocol for Routing in Mobile Ad-Hoc Networks

Alexandros Giagkos and Myra S. Wilson

Dept. of Computer Science, Aberystwyth University, Penglais, SY23 3BD, Wales, UK
{aag07,mxw}@aber.ac.uk

Abstract. We introduce a new bee-inspired routing protocol for mobile ad hoc networks. Emphasis is given to the ability of bees to evaluate paths by considering several quality factors. In order to achieve similar behaviour in the networking environment, BeeIP is using cross-layering. Fetching parameters from the lower PHY and MAC layers to the core of the protocol, offers the artificial bees the ability to make predictions about the link's future performance. Our approach is compared with two well-known routing protocols in the area, the destination sequenced distance-vector protocol (DSDV), and the adaptive on-demand distance vector protocol (AODV). The outcome shows that BeeIP achieves higher data delivery rates and less control overhead than DSDV, and slightly better results compared to AODV, initializing less route discovery processes.

Keywords: Bee-Inspired, Network, Routing, Cross-Layer, MANETs.

1 Introduction

Nodes in a mobile ad hoc environment face two major challenges, the mobility of the network participants and their resource constraints. Firstly, the movement of the nodes lead to network topology changes and frequent path breaks. Secondly, as nodes act as both transmission endpoints and routers, they generate their own traffic as well as route the traffic generated by others. This requires more energy being spent, and also, increases the complexity of routing. [1]

The routing algorithms for mobile ad hoc networks (MANETs) can be broadly categorized as proactive, reactive or hybrid. Protocols that use proactive algorithms periodically send control packets to collect information about the network state and update their routing tables accordingly. Such examples are the destination sequenced distance-vector protocol (DSDV) [2] and the optimized link state routing protocol (OLSR) [3].

Contrarily, reactive algorithms find routes on-demand. They do not maintain routes between all the nodes in the topology. Rather, routes are established only when needed through a route discovery process, in which a route request is broadcast. Examples of reactive protocols are the dynamic source routing (DSR) [4] and the adaptive on-demand distance vector (AODV) [5].

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 263–272, 2010.

The third category, hybrid, contains ideas borrowed from both proactive and reactive paradigms. Generally, hybrid protocols separate the network topology into zones. Routing is determined proactively within each zone, and reactively outside it. The advantage of such a combination is the increased overall scalability and optimization within the zones. One well-known hybrid example is the zone routing protocol (ZRP) [6].

All these approaches point out the need for adaptation in routing. Protocols have to be able to adapt to topological changes and provide optimal results. Examples of such adaptive behaviour come from the study of Nature and in particular natural networks (e.g. insects). The first algorithm which presented a detailed scheme for network routing based on ant colony principles is ARA [7]. This routing algorithm is inspired by the pheromone laying behaviour of ant colonies.

In this paper we present a new routing protocol for MANETs, called BeeIP, which is designed to provide routing solutions inspired by the foraging principles of bees. Cross-layering is used in order to utilize parameters of lower layers and be able to calculate the performance of the links between the sources and the destinations [8].

2 Related Work

In 2004, H.F. Wedde, M. Farooq, and Y. Zhang were the first to present BeeHive [9], a novel routing algorithm for wireless networks inspired by the communicative and evaluative methods and procedures of bees.

More specifically, BeeHive is built around two types of agents, the short distance and the long distance agents which are proactively generated at the nodes and are designed after the way bee foragers respond to bee dances. The responsibility of both types of agents is to explore the network and to evaluate the quality of paths that they traverse, in order to update node routing tables. Short distance agents are allowed to move only up to a restricted number of hops in the network, whereas long distance agents have to collect and disseminate routing information in the complete topology.

Moreover, BeeHive has been extensively tested and evaluated. Its results conclude that while it achieves similar or better performance compared to state-of-the-art routing algorithms, bee agents occupy smaller bandwidth and require significantly less processing time compared to the agents of existing algorithms.

BeeHive has been an inspiration to further research and enhancements. In 2005, H.F. Wedde et al have proposed BeeAdHoc [10], a routing algorithm for energy efficient routing in MANETs. By utilizing two types of agents, scouts and foragers, BeeAdHoc is able to reactively search for routing solutions, consuming less energy compared to existing state-of-the-art approaches. The major difference of our approach and BeeAdHoc is on how the quality of the links is calculated, and the way of evaluating their performance. Due to the early stage of this work, we were not able to present a comparison between our approach and BeeAdHoc, however, this is part of our future plan.

The rest of this paper is organized as follows. In Section 3 we give an overview of the key points of biological bee behaviour in respect to both scouting and foraging. In Section 4 we present our design model. Section 5 includes the first simulation experiments and results. Section 6 contains our conclusion and plan for future work.

3 Biological Background

In Nature, a bee explores the surroundings of the hive in order to detect possible sources of food. Once a source is found, the scout returns back to the hive to report her findings and to recruit other hive members to start foraging. Both reporting and recruiting are done by performing a special dance.

In his book [11] von Frisch presented the understanding of the dependence of the bee dances on the profitability of foraging activity. He has shown that although the pattern of bees' dance is determined fundamentally by the distance of and direction to a source of food, whether dancing will take place depends on many factors that may significantly change the bees' behaviour. Examples of such factors are the sweetness of the sugar solution in the food, the ease of obtaining and carrying it back to the hive, the distance of the food source to the hive, and the amount of energy required during the particular foraging process.

It is also crucial to mention that the special dance is not performed only by scout bees. Each time a successful forager returns back to the hive she can also perform the foraging dance (serving as a scout at the same time), and report any improvement or deterioration of the currently working path. Furthermore, if the path's reliability is becoming very poor, the forager can also refuse to dance and, hence, stop recruiting new members.

4 Design Model

BeeIP is a routing protocol which models the collaborative behaviour of simple artificial bee agents to build enough knowledge in order to establish communication links between two nodes, and allow data to be transferred across them.

The base of any assumptions made in our design is that every time there is a need for a link to be established, the sender node will behave as being the hive, the destination node will behave as being the source of food, and all the intermediate nodes will consistite the path that a bee forager needs to traverse while flying from one endpoint to the other.

4.1 Agents

The model uses three types of agents in the form of data packets. The scout, the ack_scout, and the forager.

Scout: They are sent when a scouting process is initialized in order to discover new paths towards a given destination. This happens each time there is a new request from the upper layer and previous routing knowledge is unsatisfactory. A scout is transmitted using broadcast to all neighbouring nodes. This technique benefits not only the propagation of the initial request, but also the introduction of the transmitting node to its neighbourhood.

Apart from the details of the scouting process, scouts also carry important information about their sender's state. A node's state is a group of attributes that describe the situation in which the node is at the time of broadcasting the scout packet. Cross-layering between PHY, MAC and network layers allows the routing protocol to know the current energy and speed levels of the node, as well as the size of the interface queue.

Furthermore, upon receiving a scout, neighbouring nodes are able to discover evidence about the link's quality between them and the scout's sender. This evidence is the one-way transmission delay of the link, and the scout packet's signal power. The latter is an indication of the distance and the clearance of the intermediate area. The information above is stored internally and is used to calculate the local reliability level of the pair, i.e. the sender and the receiver of the scout packet. The local reliability level plays a very important role to the overall path quality and the decision making of the foragers.

Following that, the receiving node can either propagate the scout packet further if it is not the scout's destination, or create an ack_scout to send back. Loops are avoided by tagging each scout packet with a unique scouting ID.

Ack_scout: Once the scout reaches its destination the scouting is considered successful and an ack_scout packet is created. Ack_scouts use a source routing fashion to travel back to the source, using unicast transmission. Therefore, the route that was followed towards the destination is used in reverse. On their way back, ack_scouts acknowledge the success of the scouting to both the intermediate nodes and the source node.

Forager: When BeeIP is unable to transmit a data packet, it stores it into a local queue and starts a new scouting process for its destination. This decreases the packet loss due to incomplete routing information. Once an ack_scout returns back and acknowledges the existence of a path, all packets for the corresponding destination in the queue are being transmitted.

The way they do this, is by using the most important agent type of BeeIP, the forager. Foragers are specially crafted packets that have three important roles. Firstly, they carry (in form of payload) the data packets from the source to the destination. Secondly, they are used to update neighbouring nodes' states and links' information, just like scouts did in the first place. Thirdly, foragers are constantly monitoring the path they traverse for any improvements. Technically speaking, foragers collect the differences between the local reliability levels, calculated by using the current forager, and the local reliability levels calculated by the previous forager's visit, and report the summation back to the hive. In a TCP connection, this is done when carrying TCP ACK packets. The

summation represents the total reliability level of the path, hence, the global reliability level.

4.2 Local Reliability Level

The local reliability level describes the one-way performance between a pair of nodes in the topology. It is the BeeIP's way of measuring how good or bad a transmission can be, by using this particular pair. The direction of the measurement is towards the source of the transmission and it is a combination of the neighbour's state and the network link between the endpoints.

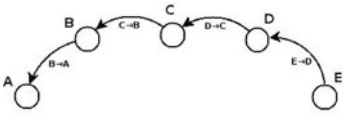


Fig. 1. A simple scouting

Figure 1 illustrates a simple example. Node A is the source (hive), node E is the destination (flower), and nodes B, C and D consistute the intermediate path from A to E. A forager that returns back to A has to follow the path through the nodes D, C, B, and A, pick up each pair's reliability improvement, and submit the result to A. Furthermore, the reliability level of pair (E,D) is a combination of E's state, and the path $E \rightarrow D$.

In total, there are five parameters that can be extracted from a node's state and a path such as the above. (i) E's speed level, (ii) E's energy level, (iii) the path's transmission delay based on the forager packet, (iv) the forager packet's transmission power, and (v) the queuing delay of the path, based on E's reported queue size.

Each one of the above parameter plays a significant role in the local quality of the pair. The speed of a node affects their transmissions and can lead to a weak or even broken link. Similarly, the energy level of the node dominates its ability to transmit clear signals to full transmission range. The signal power of the orager packet is used to give an idea for both the distance and the area between the nodes. Finally, both queueing and transmission delays alter the quality of the link. On one hand, the transmission delay describes the difficulties experienced because of the bandwidth of the link. On the other hand, the queueing delay describes the difficulties caused by traffic loads. Note that the propagation and processing delays are factored out, since they are insignificantly small.

Although the parameters' similarity is that they all affect the reliability level of the pair, their values are of different scales. Table 1 shows the minimum and maximum accepted values for each parameter during the simulation experiments. In order to use them properly, all values need to be normalized to the same scale: min 0, max 20. BeeIP achieves scale normalization of values by performing linear transformation. If α_1, β_1 and α_2, β_2 the minimum and maximum numbers of the first and second scale respectively, and χ is the number to be normalized to ψ then,

$$\psi = \frac{\alpha_2 + (\chi - \alpha_1) * (\beta_2 - \alpha_2)}{(\beta_1 - \alpha_1)} \quad (1)$$

Notice that speed, and both queueing and transmission delays are adversely affecting the performance. For instance, a node's speed equal to 0 does not affect the transmission, as it does not alter the distance between the transmission endpoints. In order to tackle this issue, these three parameters are normalized in reverse.

Table 1. Local reliability parameters and scales

	Signal Pow[1]	Speed	Energy	Q-Delay[2]	Tx-Delay[3]
min	1.258925e-10 W (-69 dBm)	0 m/s	0 W*h	0 s	0.0006 s
max	7.943282e-10 W (-61 dBm)	10 m/s	10 W*h	0.075 s	0.0120 s

Once all values are put on the same scale, the local reliability level is calculated using a simple weighting system. This is required because not all of the parameters have the same influence on the performance. Obviously, a very weak signal strength can be an indication of either a long distance between two nodes or the appearance of an obstacle. In both cases, it requires immediate action. This does not happen with the queueing delay. The latter may affect the performance, however, it does not necessarily involve a link break. The weighting system is shown at Table 2.

Table 2. Weighting system and factors

Parameter:	Signal Pow	Speed	Energy	Q-Delay	Tx-Delay
Weight (w):	0.40	0.20	0.20	0.15	0.05

Then, the local reliability level of the pair is finally defined by the formula:

$$rel_{local} = pow' * w_{pow} + speed' * w_{speed} + energy' * w_{energy} + qd' * w_{qd} + txd' * w_{txd} \quad (2)$$

where pow' is the normalized value of the signal's power, etc.

Every time a forager visits a new node during its flight back to the hive, the knowledge it brings with it as well as its own transmission are used to calculate the new local reliability level of the corresponding pair. Once calculated, the number is compared with the previous available local reliability level. The difference of the two is then reported back to the forager which continues its journey to the next hop in the path.

The difference of the two local reliability levels, previous and new, describes the improvement of the pair since the last use. In addition, the new local reliability level is stored internally to be used for future calculations.

[1] Proxim. ORiNOCO 11b Client PC Card Specification for open range environment.
[2] Maximum queue size is set to 50 packets.
[3] 11Mbit bandwidth. Minimum 76 bytes and maximum 1500 bytes packet size.

4.3 Global Reliability Level

A bee forager that finally arrives at its hive, carries the summation of all the local reliability differences collected on its way back. This number is called the global reliability level and is an indication of the link's quality as experienced during the last forager's flight. In BeeIP it is defined as follows:

$$rel_{global} = \sum_{n=1}^{m} (rel_{local-new_{N_{n+1} \rightarrow N_n}} - rel_{local-prev_{N_{n+1} \rightarrow N_n}}) \tag{3}$$

where m is the total number of nodes in an numerically ordered path, and $N_{n+1} \rightarrow N_n$ the pair of nodes with direction towards the source node (N_1).

Likewise in local, the global reliability level is compared to the one obtained from the previous flight. The difference of the two represents the improvement or the deterioration of the quality of the path. However, the number by itself can only give a dim idea since it is a result of one transmission only, which, depending on the environmental and network conditions may lead to negative assumptions. In order to utilize these numbers correctly and be able to make predictions about the quality of the link and its status in future, we use a 10x2 matrix of the last 10 instances of incoming foragers and apply regression analysis to the values. Time is used for the first column, and the difference of the new and previous global reliability levels for the second column. The output matrix has the form of:

$$\begin{pmatrix} 2.823042 & 0.32 \\ 2.825661 & 1.46 \\ .. & \\ 2.854530 & -0.25 \end{pmatrix}$$

Using Pearson's correlation coefficient [12], we are allowed to make predictions based on the strength of the linear dependence between the two. The correlation coefficient r is defined by the formula:

$$r = \frac{\sum_{i=1}^{k}(t_i - \mu t)(rel_{global_i} - \mu rel_{global})}{\sqrt{\sum_{i=1}^{k}(t_i - \mu t)^2}\sqrt{\sum_{i=1}^{k}(rel_{global_i} - \mu rel_{global})^2}} \tag{4}$$

where t_i the time of receiving rel_{global_i}, μt the mean of the time column values, and k the matrix row number (10 by default).

The correlation coefficient result ranges from -1 to 1. A value of 1 implies that a linear equation describes the relationship between t's and global reliability differences perfectly, i.e. when t increases, the improvement increases too. On the contrary, a value of -1 implies that the improvement decreases as t increases, i.e. the path becomes weak. Values near 0 imply that there is no linear correlation between the two, and we are not able to make any serious predictions.

Similarly to Nature, where bee foragers may dance vigorously if the quality of the path is becoming better or even stop dancing when the path is very poor, artificial foragers are able to judge whether to recruit other members or initialize a new scouting process. At this early stage of this work, BeeIP is able to detect weak links by comparing r to a threshold (-0.8) and re-send new scouts if it finds it necessary.

5 Simulation Experiments and Results

In order to evaluate the performance of BeeIP, we have used ns-2 network simulator. We have performed experiments with static scenarios of 20, 40, 60, 80 and 100 nodes in 300x300 m^2, 500x00 m^2, .., and 1100x1100 m^2 areas. Each node carries a single wireless card, the configuration of which is set to match ORiNOCO11b Wireless Card, 11Mbps, 802.11b for 160m in open range environment.

The nodes are moving in random directions with randomly selected speeds between 1m/s (walking speed) and maximum 10m/s. Two nodes, fairly far from each other, are picked up to serve as the source (bee hive) and the destination (flower) of an TCP/FTP connection in each scenario. The initial energy level is set to 36000 Joules (or 10 watt-hours).[4] The simulation time is set to 600 seconds. Our results are compared to those of AODV and DSDV protocols, under the same topological conditions. In order to factor out any implementation related errors to our comparisons, we use the implementations which are distributed with ns-2 simulator.

In figure 2, we study the successful packet delivery ratio of the three protocols. Unsurprisingly, the reactive nature of BeeIP has a clear advantage over DSDV which becomes weaker as the number of nodes is increased. This is due to the large number of required control packets, in order to collect enough information and build DSDV routing tables. Compared to AODV, BeeIP has a slightly better performance. The reason behind that, is that although they both apply reactive schemes, BeeIP is able to detect when a link is about to break faster, and then switches to another one.

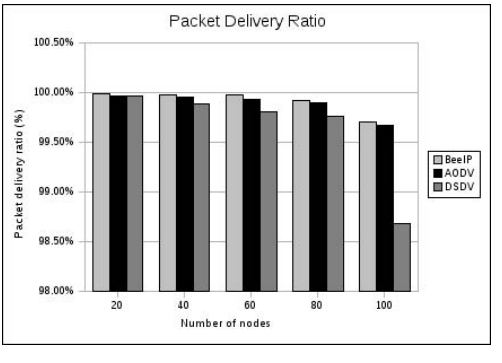


Fig. 2. Packet delivery ratio vs. number of nodes

Furthermore, we have measured the packet loss of the three approaches (figure 3). For a number of nodes lower than 40, BeeIP scores less packet loss than AODV. Between 40 and 60 nodes, BeeIP performs quite steadily and better than AODV. However, we notice a big increase after 60 nodes, which although is still better than AODV's, it triggers our interest for future improvements. Finally, both BeeIP and AODV packet losses are significantly lower than DSDV's.

[4] Almost $^1/_{25}$ of the battery capacity of a fully charged Pioneer P3-AT all-terrain robot (http://www.activrobots.com/ROBOTS/specs.html).

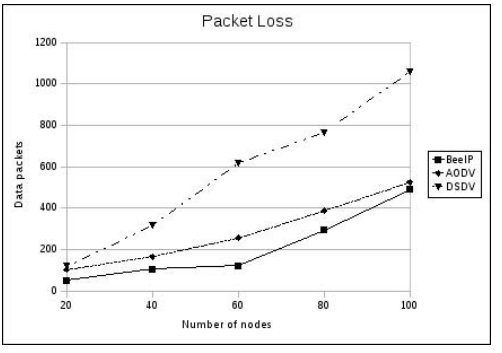


Fig. 3. Packet loss vs. number of nodes

The control overhead of the three protocols is shown in figure 4. Although things get worse for higher number of nodes, under the same circumstances, BeeIP sends less control packets during the static simulation scenario, than AODV and DSDV protocols.

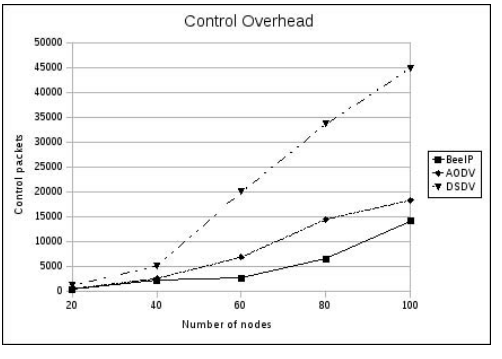


Fig. 4. Control overhead vs. number of nodes

In a reactive point of view, BeeIP and AODV are compared based on the successful route discoveries and the successful data packet deliveries. All experiments have shown that BeeIP managed to deliver more data packets successfully using less route discoveries. Table 3 summarizes these results. For example, BeeIP was able to send 369399 data packets by using 14 links during the simulation, whereas AODV sent 310912 using 15 links.

Table 3. Successfully established links during simulation (packets sent)

	20	40	60	80	100
BeeIP:	14 (369399)	27 (333449)	146 (381166)	184 (355629)	271 (163024)
AODV:	15 (310912)	39 (330284)	196 (374010)	220 (354899)	343 (157042)

6 Conclusion

In this paper we have introduced BeeIP, a new bee-inspired routing protocol for mobile ad hoc networks. We have also compared the first simulation results of our approach with two state-of-the-art protocols, AODV and DSDV. The simulation experiments have shown that BeeIP performs better than DSDV and slightly better than AODV in terms of packet delivery ratio and packet loss. Furthermore, BeeIP was able to deliver more data packets successfully, initializing less route discovery processes than AODV under the same network conditions.

Our future work includes the improvement of our design, in order to support multiple paths for each transmission which will be selected based on their quality via artificial bee dancing. This will increase the life of the network and the delivery ratio of the protocol. Finally, we need to add support for stateless transport protocols such as UDP. The results of these improvements as well as new features, will be compared to AODV, DSDV and the biologically inspired AntHocNet and BeeAdHoc, something that we did not include in this paper due to the early stage of the work.

References

1. Murthy, C., Siva, R., Manoj, B.S.: Ad Hoc Wireless Networks Architectures and Protocols. Prentice Hall, Englewood Cliffs (2004)
2. Perins, C.E., Bhagwat, P.: Highly dynamic destination-sequenced distance vector (DSDV) for mobile computers. In: ACM SIGCOMM 1994 (1994)
3. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR). IETF RFC3626, http://www.ietf.org/rfc/rfc3626.txt
4. Johnson, B.D., Maltz, D.A.: Dynamic Source Routing in Ad-Hoc Wireless Networks. In: Imielinski, T., Korth, H. (eds.) Mobile Computing, pp. 153–181. Kluwer, Dordrecht (1996)
5. Perkins, C.E., Royer, E.M., Das, S.R.: Ad hoc on demand distance vector routing. IETF RFC3561, http://www.ietf.org/rfc/rfc3561.txt
6. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR). IETF RFC3626, http://www.ietf.org/rfc/rfc3626.txt
7. Gunes, M., Sorges, U., Bouazizi, I.: ARA - The Ant-Colony Based Routing Algorithms for MANETs. IWAHN (2002)
8. Giagkos, A., Wilson, M.S.: A Cross-Layer Design for Bee-Inspired Routing Protocols in MANETs. In: Kyriacou, T. (ed.) Proceedings of TAROS 2009, pp. 25–32. Ulster University (2009)
9. Wedde, H.F., Farooq, M., Zhang, Y.: BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior. In: Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T. (eds.) ANTS 2004. LNCS, vol. 3172, pp. 83–94. Springer, Heidelberg (2004)
10. Wedde, H.F., Farooq, M., Pannenbaecker, T., Vogel, B., Mueller, C., Meth, J., Jeruschkat, R.: BeeAdHoc: An Energy Efficient Routing Algorithm for Mobile Ad Hoc Networks Inspired by Bee Behavior. In: GECCO. ACM Press, New York (2005)
11. von Frisch, K.: The Dance Language and Orientation of Bees. Oxford University Press, Oxford (1967)
12. Read, T.R.C., Noel, A.C., Cressie, N.A.C.: Goodness-of-fit statistics for discrete multivariate data. Springer, Heidelberg (1988)

Simulating Human Table Tennis with a Biomimetic Robot Setup

Katharina Mülling, Jens Kober, and Jan Peters

Max Planck Institute for Biological Cybernetics, Department of Empirical Inference
Spemannstr. 38, 72076 Tübingen, Germany
{katharina.muelling,jens.kober,jan.peters}@tuebingen.mpg.de

Abstract. Playing table tennis is a difficult motor task which requires fast movements, accurate control and adaptation to task parameters. Although human beings see and move slower than most robot systems they outperform all table tennis robots significantly. In this paper we study human table tennis and present a robot system that mimics human striking behavior. Therefore we model the human movements involved in hitting a table tennis ball using discrete movement stages and the virtual hitting point hypothesis. The resulting model is implemented on an anthropomorphic robot arm with 7 degrees of freedom using robotics methods. We verify the functionality of the model both in a physical realistic simulation of an anthropomorphic robot arm and on a real Barrett WAM$^{\text{TM}}$.

Keywords: biomimetic table tennis, anthropomorphic robot arms.

1 Introduction

Table tennis has long fascinated roboticists as a particularly difficult task. The main work on robot table tennis started in 1983 [3] with a robot ping pong competition and ended in 1993 [2,12,9,10,8] when the competition came to an end, but single groups continued work until today [15,14,1]. These early approaches used smart engineering to overcome inherent problems like movement generation, orientation of the racket and vision in an human inhabited environment. Furthermore, they used a much smaller table and modified table tennis rules [3]. In contrast to these approaches, we use an anthropomorphic robot arm with seven degrees of freedoms (DoFs) and concentrate on generating smooth movements that properly distribute the forces over the different DoFs. Therefore, we employ a biomimetic approach for trajectory generation and movement adaptation.

Table tennis requires fast and accurate movements to achieve high playing performance. However, for such quick and forceful movements, the human central nervous system has little time to process feedback about the environment and has to rely largely on feedforward components [21] such as accurate task models as well as predictions about the opponent and the ball. Understanding how humans perform so well in such a complex task as table tennis may yield essential knowledge for skill execution in robotics. In this project, it is our goal to construct

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 273–282, 2010.

a model of table tennis striking movements based on known hypotheses of human motor control in table tennis. We want to get a step closer to understanding which basic building blocks are required for generic robot skill execution systems. We describe the construction of a robot ping pong player, with seven DoFs, that is capable of returning a ball on an International Table Tennis Federation (ITTF) standard sized table served by a ball cannon. We focus particularly on modeling the arm trajectories in striking movements based on human table tennis data using a multi-stage model [16]. We end up with a method that successfully adapts the stroke according to the movement of the ball. The setup works sufficiently well in simulation and on a real Barrett WAM[1].

In this paper, we will proceed as follows. In Section 2, we present all relevant background on modeling a table tennis stroke based on biological hypotheses such that we are able to obtain a model of a table tennis stroke in Section 3. In Section 4, we present the results of our implementation and show that the proposed model works well in simulation and on the real robot.

2 Modeling Striking Movement in Human Table Tennis

In this section, we present background information on modeling table tennis from a racket sports perspective. In particular, we focus on movement stages, motion selection and parameterization, and movement generation. At the end of each of these sections, we will outline which computational concepts arise from the biological hypotheses.

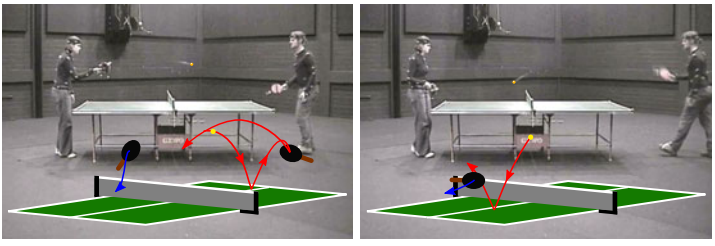
2.1 Movement Stages of a Stroke

Table tennis exhibits a regular, modular structure that has been studied by Ramanantsoa and Durey [16]. They analyzed a top player and proposed a spatial adjustment of four movement stages with respect to certain ball events, i.e., bouncing, net crossing and stroke. According to their hypothesis, the following four stages can be distinguished during playing of experts and, to make them more understandable, we have labeled them according to their functionality:

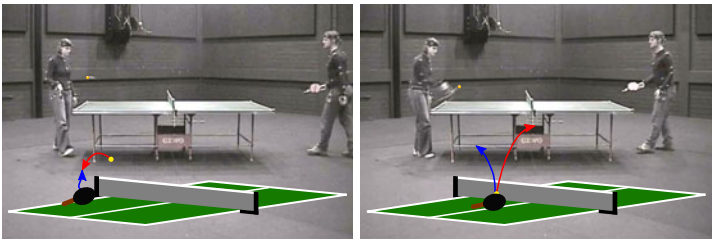
Awaiting Stage. The ball moves towards the opponent who hits it back towards the net. The racket is moving downwards. At the end of this stage the racket will be in a plane parallel to the table surface.

Preparation Stage. The ball comes towards the player, has already passed the net and will bounce off the table during this stage. The racket is moving backwards in order to prepare the stroke. For forehand strokes the racket is in the same plane as it is in the awaiting phase. For backhand strokes the racket moves on a frontal plane nearly perpendicular to the plane in the awaiting stage. The player chooses a hitting point where he plans to hit the ball to which we refer as the virtual hitting point.

[1] Note, that a preliminary version with no real robot results and a simplified dynamical model has been presented at a German local conference.



(a) Awaiting Stage (b) Preparation St.



(c) Hitting Stage (d) Finishing Stage

Fig. 1. This figure illustrates the four movement stages of Ramanantsoa et al. [16] recorded in a Vicon motion capture system where (a) shows the Awaiting Stage in which the opponent is observed, (b) the Preparation Stage in which the stroke is prepared, (c) the Hitting Stage in which the ball is intercepted, and (d) the Finishing Stage. The red arrow shows the movement of the ball in the phase and the blue arrow the movement of the racket.

Hitting Stage. The ball moves towards the virtual hitting point where the player intercepts it. In a first substage final adjustments are done. In the second substage the racket moves towards the virtual hitting point until it hits the ball in a circular movement. For expert players the duration of this phase appears to be constant and lasts approximately $80ms$. At the point of impact the lateral velocity (in the direction of the small table side) is zero and the velocity in direction of the long table side reaches its apex.

Finishing Stage. After having been hit, the ball is on the return path to the opponent while the racket is moving upwards to a stopping position. This stage ends with the ball crossing the net and the velocity of the racket tending to zero.

We have verified the stages suggested by Ramanantsoa and Durey [16] in a VICON motion capture setup for two intermediate players where each of the stages can be observed distinctively (see Figure 1). From a computational point of view, this model corresponds to a finite state automaton.

2.2 Movement Selection and Goal Determination

As humans appear to rely on elementary motor programs [18], it is likely that pre-structured movement commands are employed for each of the four stages.

These motor programs are adapted to the environmental stimuli at the beginning of each stage. Motor programs determine the order and timing of the muscle contractions and, by doing so, define the shape of the action produced. Sensory information can further modify motor programs to generate rapid corrections in the case of changing environmental demands as found in table tennis [5]. The system is only altering the parameters of the movement such as duration, amplitude, and the final goal position of the movement [18]. This is supported by the experiments in [20], which demonstrated that expert players exhibit a consistent spatial and temporal movement pattern in table tennis. The authors of [20] concluded that a professional player chooses a movement program for which the execution time is known from their repertoire and decides when to initiate the drive. This observation is known as operational timing hypothesis.

The problem of what information is used for initiating the movement is not yet solved. Most likely humans use the so-called *time to contact*, i.e., is the time until an object reaches the observer, to control the timing of their stroke stages. Lee [13] suggested that humans determine the time to contact by an optical variable τ that is specified as the inverse of the relative rate of dilation of a retinal image of an object. Using the operational timing hypothesis, biomimetic system has to initiate the chosen movement program when τ reaches a critical value.

We represent one set of movement programs for a specific forehand as splines. The start and end position, velocity and acceleration of the stages as well as the durations of the movements are given by pre-defined values which are fixed while the end and start conditions of the hitting and finishing stage, respectively, can be selected freely. Here we use the hitting point which is adapted according to the incoming ball and the desired return.

2.3 Movement Generation

Assuming that movement stages, selection and initiation are known, we need to discuss how the different strokes are generated. There are infinitely many ways to generate racket trajectories and, due to redundancies in the arm, there are also numerous different arm posture to execute the same task-space trajectory in joint-space. In order to find generative principles underlying the movement generation, neuroscientists often turn to optimal control [19]. One approach is the use of cost functions which allow the computation of trajectory formation for arm movements. Most cost functions focus primarily on reaching and pointing movements where one can observe a bell-shape velocity curve as well as a clear relationship between movement duration and amplitude. However, this does not hold for striking sports. Cruse et al. [6] suggested a cost function for the control of the human arm movement based on the comfort of the posture. For each joint, the cost is induced by proximity to a comfort posture in joint-space, i.e., the cost is minimal if the joint angles are the same as for the comfort posture and increases with the distance between comfort posture and joint position. For

movement generation, this cost is minimized. We employ this cost function to select a comfortable joint configuration at the hitting point (see Section 3.3).

3 A Biologically-Inspired Trajectory Generator for Table Tennis Strokes

In this section, we will discuss how the parts of the behavioral model presented in Section 2 can be implemented as a mathematical model suitable for real-time execution on a robot. For doing so, we proceed as follows: first, we present all required components in an overview. Subsequently, we discuss the details of the dynamics model for table tennis in Section 3.2, the computation of the goal parameters in Section 3.3 and the trajectory generation in Section 3.4.

3.1 General Assumptions

As outlined in Section 2.1, we assume the movement stages of the model by Ramanantsoa et al. [16] and use a finite state automaton to represent this model. In order to realize each of these four stages, the system has to detect the ball and determine its position $\mathbf{p}_b$. Due to noise in the vision processing, the system needs to filter this information.

To generate the arm trajectories, we have to determine the constraints for the movements of each joint of the arm in each stage. While desired final joint configurations suffice for the awaiting, preparation and finishing stages, the hitting stage requires a well-chosen movement goal which is the hardest to realize. The system has to first choose a point on the court of the opponent where the ball needs to be returned[2]. Secondly, we have to determine the intersection point of the ball and the racket, which specify the virtual hitting point $\mathbf{p}_e$. The hitting point is determined by the location where the ball trajectory intersects a virtual hitting plane in the forehand area of the robot. Based on the choice of these two points, the necessary batting position, orientation and velocity of the racket are chosen as goal parameters for the hitting movement. More details on the computations involved are given in Section 3.3.

Movement initiation is triggered in accordance with the movement stages and using the movement goals, i.e., when the time of the predicted ball intersecting the virtual hitting point $\boldsymbol{p}_{\mathrm{e}}$ is less than a threshold, the hitting movement is initiated. This step requires the system to predict when the ball is going to reach the virtual hitting plane. The current hitting time can be determined by predicting the trajectory of the ball using the physical model of the aerodynamic and bouncing behavior of the ball described in Section 3.2. Following the suggestion in [4] that some online adaptation of the movement can take place, we update the virtual hitting point if the estimates changes drastically. For the determination of the movement program, we rely upon a spline-based representation for encoding the trajectory. More details are given in Section 3.4.

[2] Humans choose this point as part of a higher level strategy. To date, we choose them in an ad-hoc fashion not conditioned on the opponent.

3.2 Dynamics Model

To predict the position and velocity of the ball at time t_1 based on the ones at time t_0, we have to model the aerodynamics of the ball and the physics of a ball's bounce off of a table. For modeling the ballistic flight of the ball we have to consider air drag, gravity and spin. As the latter is hard to determine, our model currently neglects the spin. For a table tennis ball we can assume that the air drag is proportional to the square of the velocity of the ball. Using symplectic Euler integration, we can implement the following model in discrete time form:

$$\boldsymbol{a_k} = \boldsymbol{g} - C\|\boldsymbol{v_k}\|\boldsymbol{v_k} \qquad \boldsymbol{v_{k+1}} = \boldsymbol{v_k} + \boldsymbol{a_k}\Delta t \qquad \boldsymbol{p_{k+1}} = \boldsymbol{p_k} + \boldsymbol{v_{k+1}}\Delta t, \tag{1}$$

where $\boldsymbol{p}$ denotes the position of the ball, $\boldsymbol{v}$ is the velocity, $\boldsymbol{a}$ denotes the acceleration, $\boldsymbol{g} = -9.81m/s^2[0,0,1]^T$ is the gravity, $C = c_w\rho A/(2m)$, c_w is the drag coefficient, ρ is the density of the air, A is the size of the ball surface and m is the mass of the table tennis ball.

For the bouncing behavior of the ball we assume a velocity change in z-direction only. This change in velocity $v_z = -\varepsilon_T v_z$ is determined by the coefficient of restitution ε_T.

3.3 Determining the Goal Parameters

After determining the virtual hitting point, the system can freely choose the height z_{net} at which the returning ball passes the net as well as the positions x_{b}, y_{b} where the ball will bounce on the opponents courts. The y-axis is along the net and the x-axis is aligned with the long side of the table. The choice of these three variables belongs to the higher level functionality and is not covered in this model, we instead draw them from a distribution of plausible values. To determine the goal parameters, we have to first calculate the desired outgoing velocity vector $\boldsymbol{O}$ of the ball which corresponds to the desired velocity of the ball after the impact with the racket. Directly from it, we can also determine the required velocity and orientation of the racket.

Desired Outgoing Vector. Based on the dynamics model derived in Section 3.2, we obtain 5 non-linear equations with 5 unknowns, i.e., the time until the ball reaches the opponents court, the time until the ball reaches the net and the desired outgoing vector (3 components). Since these equations are non linear in the variables of interests, we have to solve the problem numerically. Therefore, we need to use a globally convergent solver for nonlinear equation systems, which combines the Newton-Raphson update with a modification for global convergence [7].

Goal Orientation. The orientation of the end-effector is specified as a rotation that transforms the normal vector $\boldsymbol{n}_e$ to the desired normal vector $\boldsymbol{n}_{\text{ed}}$ given by

$$\boldsymbol{n}_{\text{rd}} = \frac{\boldsymbol{O} - \boldsymbol{I}}{\|\boldsymbol{O} - \boldsymbol{I}\|}, \tag{2}$$

(a) Awaiting Stage (b) Preparation St. (c) Hitting Stage (d) Finishing Stage

Fig. 2. The figure shows the different phases on the real robot. Note that the black circles are part of our safety system and have nothing to do with the task.

where $\boldsymbol{O}$ is the velocity of the outgoing ball after the ball-racket impact and $\boldsymbol{I}$ is the velocity vector of the incoming ball at the virtual hitting point before impact. Note that we assume only a speed change $\boldsymbol{O} - \boldsymbol{I}$ in the normal direction $\boldsymbol{n}$. The rotation is defined in terms of quaternions by

$$q_{\mathrm{ed}'} = q_{\mathrm{rd}} q_{\mathrm{yrot}}, \tag{3}$$

where q_{yrot} is the quaternion that describes the rotation from the racket to the end-effector and $q_{\mathrm{rd}} = (\cos(\theta/2), \boldsymbol{u}\sin(\theta/2))$, with $\theta = \boldsymbol{n}_{\mathrm{e}}^T \boldsymbol{n}_{\mathrm{rd}}/(\|\boldsymbol{n}_{\mathrm{e}}\|\|\boldsymbol{n}_{\mathrm{rd}}\|)$ and $\boldsymbol{u} = \boldsymbol{n}_{\mathrm{e}} \times \boldsymbol{n}_{\mathrm{rd}}/\|\boldsymbol{n}_{\mathrm{e}} \times \boldsymbol{n}_{\mathrm{rd}}\|$, is the quaternion that defines the transformation of the normal of the end-effector $\boldsymbol{n}_e$ to the desired racket normal $\boldsymbol{n}_{\mathrm{rd}}$. As there exist infinitely many racket orientations that have the same racket normal, we need to determine the final orientation depending on a preferred end-effector position. The resulting quaternion of the end-effector q_{ed} is determined by the rotation about the normal of the racket. The orientation with the corresponding joint values is chosen to yield the minimum distance to the comfort position in joint space is used as a desired racket orientation.

Required Racket Velocity. Next we have to calculate the velocity vector for the end-effector at the time of the ball's interception. We can describe the relation between the components of the incoming and ingoing velocity vector parallel to the racket norm using

$$O_{||} - v = \varepsilon_R(-I_{||} + v), \tag{4}$$

where ε_{R} denotes the coefficient of restitution of the racket and v the speed of the racket along its normal. This equation can be solved for v yielding the desired racket velocity.

3.4 Trajectory Generation

For the execution of the movements, we need a representation to obtain position $\boldsymbol{q}(t)$, velocity $\dot{\boldsymbol{q}}(t)$ and accelerations $\ddot{\boldsymbol{q}}(t)$ of the joints of the manipulator at each point in time t so that it can be executed with an inverse dynamics based controller. We used fifth order polynomials $\boldsymbol{q} = \sum_{j=0}^{5} \boldsymbol{a}_j t^j$ to represent the trajectory of all stages. Such polynomials are the minimal sufficient representation, generate

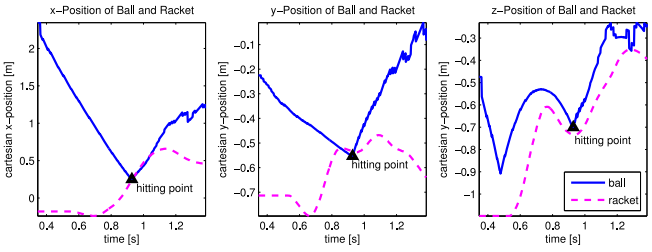


Fig. 3. This figure shows the movement of the racket and the ball on the real robot for one stroke movement. The hitting point is indicated by the black triangle.

smooth trajectories and can be evaluated quickly as well as easily. Applying the four stage model of Ramanantsoa et al. [16], we can determine four different spline phases consisting of splines interpolating between fixed initial and final positions. As the trajectory of the hitting and finishing state depends on the hitting point, trajectories have to be calculated jointly at the beginning of the hitting stage and have to be recalculated every time the virtual hitting point is updated.

4 Evaluations

In this section, we demonstrate that the presented biomimetic robot table tennis model can be used effectively in a setup where the ball is served by a ball cannon. Firstly, we present the simulated setup for the table tennis task. Secondly, we implement the model on a real robot.

We employ a Barrett WAM arm with seven DoFs that is capable of high speed motion. A standard table tennis racket is attached to the end-effector. The robot arm interacts with a standard sized table and a table tennis ball according to the ITTF rules. The ball is served randomly by a ball cannon to the right half of the table. This range corresponds roughly to an area of $1m^2$. The virtual hitting point is determined as the intersection point of the ball and the virtual hitting plane discussed in Section 3 (it covers the whole $1m^2$). The ball is tracked using a stereo vision system with a sampling rate of 60 frames per second and the vision information is filtered using an extended Kalman Filter based on the dynamics model described in Section 3.2.

4.1 Simulated Setup

We employed the SL framework [17] to create a simulation of an anthropomorphic robot arm. Subsequently, we used a model of the flight and the bouncing behavior of the ball as described in Section 3.2. We model the noise and delay of the vision system. The coefficients of restitution of both racket-ball and ball-table interactions were determined in a VICON setup.

The table tennis system is capable of returning an incoming volley to the opponents court which was served by a ball cannon at random times and to randomly selected positions. In an evaluation setup where the ball cannon served

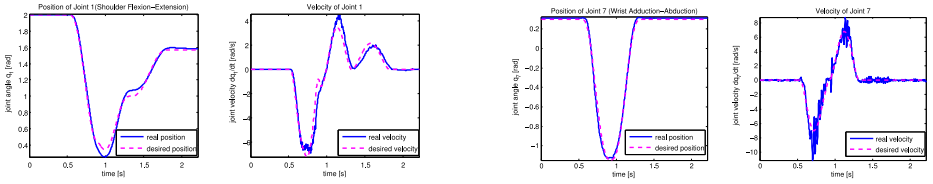


Fig. 4. This figure shows the trajectories for representative joint positions and velocities for one stroke movement. Note that the tracking errors are often due to low-gain control.

the ball 10,000 times to a random position in the work-space of the robot, the system was able to return 98% of the balls. In 75% of the trials the ball was returned to the opponent's court. The mean distance of the position of the racket mid point from the ball at the moment of contact is 1.8 cm. This result could be further improved by optimizing the trajectory generation in joint space.

4.2 Application on a Barrett WAM™

We have subsequently set up the same framework on a real robot using two partially overlapping stereo-setups for visual input.We are going to detail the arising differences here. An extended Kalman filter, based on a ballistic flight model with estimated restitution factors, tracks the ball well. However, the prediction of the virtual hitting point and time is less accurate due to unobserved spin and an underestimated initial velocity of the ball. These predictions are updated frequently and the trajectory generation is adapted. Nevertheless, the robot manages to hit the ball reliably. The main problem for missing balls and underestimating the velocity of the ball up to now is the limited field of view of the camera setup. See Figure 3 for the trajectories of the racket and the ball of the real system, Figure 4 for trajectories of individual joints and Figure 2 for snapshots of the movement.

5 Conclusion

Using knowledge on human table tennis, we have created a biomimetic model for striking movements. This model is realized in a computational form. We have shown that the resulting model can be used as an explicit policy for returning incoming table tennis balls to the opponent's court using a real seven DoF Barrett WAM. Our setup, with an anthropomorphic arm and a cluttered environment, is significantly more challenging than the tailored ones of previous robot table tennis players. The biomimetic model with its four stages of the stroke and the goal parameterization using virtual hitting points and pre-shaping of the orientation has proven to be successful in operation.

Our future work will concentrate on improving the precision in returning the ball to a desired point on the table and to improve the transition between fore- and backhand. Furthermore, we plan to replace the spline based trajectory for movement generation by dynamic systems motor primitives [11] for each of the four stages suggested by Ramanantsoa.

References

1. Acosta, L., Rodrigo, J., Mendez, J., Marchial, G., Sigut, M.: Ping-pong player prototype. Robotics and Automation Magazine 10, 44–52 (2003)
2. Andersson, R.L.: A robot ping-pong player: experiment in real-time intelligent control. MIT Press, Cambridge (1988)
3. Billingsley, J.: Robot ping pong. Practical Computing (1983)
4. Bootsma, R., van Wieringen, P.: Visual control of an attacking forehand drive in table tennis. In: Complex Movement Behaviour: The Motor-Action Controversy, pp. 189–199. North-Holland, Amsterdam (1988)
5. Bootsma, R., van Wieringen, P.: Timing an attacking forehand drive in table tennis. Journal of Experimental Psychology: Human Perception and Performance 16, 21–29 (1990)
6. Cruse, H., Brüwer, M., Brockfeld, P., Dress, A.: On the cost functions for the control of the human arm movement. Biological Cybernetics 62, 519–528 (1990)
7. Dennis, J., Schnabel, R.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice-Hall, Englewood Cliffs (1983)
8. Fässler, H., Beyer, H.A., Wen, J.T.: A robot ping pong player: optimized mechanics, high performance 3d vision, and intelligent sensor control. Robotersysteme 6, 161–170 (1990)
9. Hartley, J.: Toshiba porgress towards sensory control in real time. The Industrial Robot 14-1, 50–52 (1987)
10. Hashimoto, H., Ozaki, F., Asano, K., Osuka, K.: Development of a ping pong robot system using 7 degrees of freedom direct drive. In: Industrial Applications of Robotics and Machine Vision (IECON), pp. 608–615 (1987)
11. Ijspeert, A.J., Nakanishi, J., Schaal, S.: Learning attractor landscapes for learning motor primitives. In: Advances in Neural Information Processing Systems 16 (NIPS), vol. 15, pp. 1547–1554. MIT Press, Cambridge (2003)
12. Knight, J., Lowery, D.: Pingpong-playing robot controlled by a microcomputer. Microprocessors and Microsystems 10(6), 332–335 (1986)
13. Lee, D., Young, D.: Visual timing of interceptive action, pp. 1–30. Martinus Nijhoff, Dordrecht (1985)
14. Matsushima, M., Hashimoto, T., Takeuchi, M., Miyazaki, F.: A learning approach to robotic table tennis. IEEE Trans. on Robotics 21, 767–771 (2005)
15. Miyazaki, F., Matsushima, M., Takeuchi, M.: Learning to dynamically manipulate: A table tennis robot controls a ball and rallies with a human being. In: Advances in Robot Control, pp. 3137–3141. Springer, Heidelberg (2005)
16. Ramanantsoa, M., Durey, A.: Towards a stroke construction model. International Journal of Table Tennis Science 2, 97–114 (1994)
17. Schaal, S.: The SL simulation and real-time control software package. Technical report, University of Southern California (2009)
18. Schmidt, R., Wrisberg, C.: Motor Learning and Performance, 2nd edn. Human Kinetics (2000)
19. Todorov, E.: Optimality principles in sensorimotor control. Nature Neuroscience 7, 907–915 (2004)
20. Tyldesley, D., Whiting, H.: Operational timing. Journal of Human Movement Studies 1, 172–177 (1975)
21. Wolpert, D., Miall, C., Kawato, M.: Internal models in the cerebellum. Trends in Cognitive Science 2, 338–347 (1998)

Attentional Modulation of Mutually Dependent Behaviors

Ernesto Burattini, Silvia Rossi, Alberto Finzi, and Mariacarla Staffa

Dipartimento di Scienze Fisiche[⋆,⋆⋆]
University of Naples "Federico II" – Naples, Italy
{ernb,srossi,finzi}@na.infn.it, mariacarla.staffa@unina.it

Abstract. In this paper, we investigate simple attentional mechanisms suitable for sensing rate regulation and action coordination in the presence of mutually dependent behaviors. We present our architecture along with a case study where a real robotic system is to manage and harmonize conflicting tasks. This research focuses on attentional mechanisms for regulating the frequencies of sensor readings and action activations in a behavior-based robotic system. Such mechanisms are to direct sensors toward the most salient sources of information and filter the available sensory data to prevent unnecessary information processing.

1 Introduction

The main goal of our current research is the design of a behavior-based robotic architecture that has the capability of adapting behaviors activations both to the rate of change of the environment and to changes of its internal states. For this purpose, attentional mechanisms, balancing sensors elaboration and actions execution, can play a crucial role. In particular, attentional processes play two main roles: direct sensors towards the most salient sources of information; filter the available sensory data to prevent unnecessary information processing. As a result of the application of these mechanisms, the robot should react faster and more effectively to task-related or safety critical stimuli because processing resources are focused on relevant stimuli. Attentional mechanisms applied to autonomous robotic systems have been proposed elsewhere, mainly for vision based robotics. However, attentional mechanisms are necessary not only to focus the attention on salient regions of the space, but also to distribute resources and activities in time [7,10,12]. Also in neuroscience researchers started to investigate the temporal domain of neural activity (for example neural synchrony [11]), and relate such activity to different cognitive processes such as binding, sensory motor-coordination, attentional selection and executive functions.

⋆ The research leading to these results has received funding from the European Community's 7th Framework Programme (FP7/2007-2013) under grant agreement no.216239.

⋆⋆ The Authors want to thank MariaRosaria Ambrosino for the support provided in the creation of this work.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 283–292, 2010.

In this direction, our working hypothesis is that attentive and adaptive behaviors can be achieved starting from self-regulated periodic mechanisms, which modulate sensory readings and behaviors activations. In previous papers [4,5], we introduced the Adaptive Innate Releasing Mechanisms (AIRMs) to speed up or slow down the period of behaviors activation and thereby the reading frequency of the sensors, according to both the robot-environment interactions and the interactions that may arise within the robots itself [13].

In the AIRMs, each behavior is endowed with an independent regulation mechanism directly depending on internal and external stimuli, while the mutual influence, among the parallel behaviors, is left as a consequence of the overall self-regulating emergent behavior. However, the notion of divided attention [14] suggests that a limited amount of attention is allocated to tasks, when resources are shared in multi-task behavior, and attention can be available in graded quantity for each task. Indeed, the activations of some behaviors may directly require the activation or the inhibition of other behaviors: two behaviors may not be able to activate themselves as frequently as they need without a degrade of performances (e.g. cognitive load and interference [8]); otherwise, the activations of one behavior may directly induce the activation or synchronization of other behaviors (e.g. synchrony in attentional selection [11]).

The human behavior provides several examples of tasks that, while apparently conflicting, are simultaneously carried on. For example, some research analyzed the human behavior while driving and achieving a parallel task, such as talking over a mobile phone [15,8]. Driving a car is a complex behavior that requires the extraction and integration of information from multiple sources. Most of the information relevant for driving are taken by the view, so every change in the visual exploration can be significant for a safe driving. For example, in [8] the authors tried to experimentally assess the effects of cognitive load caused by a secondary task, simultaneously executed. Their results have shown that drivers, under a high cognitive load, execute less saccadic movements consistently with an increase of fixation time and a smaller exploration of the visual field. These experiments show that subjects are able to complete tasks in parallel, but the resources allocated to each task must dynamically adapt themselves to environmental conditions and to cognitive and physical capabilities of the subject.

In this paper, we describe our cognitive/attentive general framework obtained as an extension of the AIRM architecture [4,5] that integrates mechanisms for mutual influence among attentive behaviors. For this purpose, we introduce simple constraints among the behaviors' sampling rates. This mutual influence can work both as an inhibitory or facilitatory process [6]. To assess our framework, inspired by the studies on cognitive distraction [15,8], we define a case study where a real robot is to achieve two conflicting goals. In this context, we compare performances of this architecture with respect to non attentive versions of the same system. The empirical evaluation shows that the proposed framework is capable of harmonizing conflicting goals and distracting activities while maintaining the features of adaptability and reactivity of the previous architecture.

2 Adaptive Periodic Behaviors Modulation

In previous papers [4,5], we introduced the AIRM (Adaptive Innate Releasing Mechanisms) architecture. In the AIRM framework, the robotic system is controlled by a behavior-based executive, where each behavior can be described by a schema theory model [1]. Each behavior is characterized by a Perceptual Schema (PS), which elaborates sensor data, a Motor Schema (MS), producing the pattern of motor actions, and a control mechanism, based on a combination of a clock and a releaser (see Fig.1). The releaser enables/disables the activation of the MS, according to the sensor data. The adaptive clock controls the PS, it is active with a base period and it enables/disables data flow from sensors to PS. When the activation is disabled, sensor data are not processed (yielding to sensory readings reduction). Furthermore, the clock regulates its period p (ranging the values in the interval $[p_{min}, p_{max}]$), hence the frequency of data processing, using a feedback mechanism.

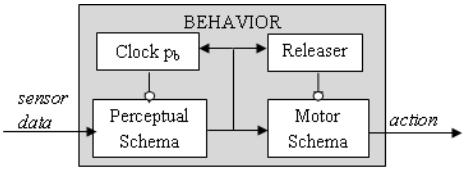


Fig. 1. Schema theory representation of AIRMs

3 Attentive Monitoring and Mutual Influence

Our goal is to develop attentional mechanisms supporting a supervisory attentional system [12] and providing a kind of divided attention [14], which focuses sensory resources and modulates tasks activations taking into account mutual influences and constraints among the behaviors.

Attentive Monitoring. Our framework combines the following design principles:

- *Behavior-based control:* the executive control is obtained from the interaction of a set of parallel behaviors working at different level of abstraction;
- *Attention filtering and focusing:* Attention modulation mechanisms consist in focusing monitoring and control activities on relevant behaviors and external stimuli. For each behavior, the process of changing the rate of sensory readings may be interpreted as an increase or decrease of attention towards a particular aspect of the environment the robotic system is interacting with; The higher is the frequency of the sampling rate associated with one behavior, the higher is the resolution at which this is monitored and controlled.
- *Internal and external sources of saliency:* The sources of salience are behavior and task dependent, and these can depend on either internal states or external stimuli;

Mutual Influence. In our attentive framework, the attention modulation strategies should be suitably regulated not only with respect to the internal or external saliency, but also with respect to attentional demand of other behaviors. To account of the problem of mutual influence among attentional behaviors, in this work, we propose an extension of the AIRM architecture endowed with explicit constraints among the internal clocks and regulation mechanisms able to respect these constraints. The aim is to capture mutual dependencies in terms of interrelations among the clocks' sampling rates and then to regulate the clocks' frequencies according to the presence of conflicting or synergetic behaviors. For example, given two mutually exclusive processes, since these are to be interleaved, the associated clock periods should be opportunely changed to allow their alternated execution; on the other hand, for two concurrent behaviors, the associated clocks are to be aligned: when the frequency of one clock increases/decreases the other clock should be accelerated/decelerated and vice versa. However, we want to add this simple mechanism while maintaining the main features of the AIRM model: the periodic activation of behaviors should provide both a relative decrease in the computational burden and the ability to monitor the internal/external environment (e.g. the robot reads sensors more often if there is a dangerous or salient situation and less often in cases of a safe operational situation).

In this new setting, for each set of clocks $p_1, ..., p_n$, we can introduce a relationship $R(p_1, ..., p_n)$ that specifies the mutual influence. However, in this paper, we mainly focus on the relationships between couple of behaviors. In particular, we consider binary constraints $R(p_A, p_B)$ like mutex or synchronization constraints. In this case, the frequencies of the clocks p_A and p_B, associated with the two behaviors, depend not only on the salience of the tasks, but also on the joint frequencies. Examples of these constraints will be provided in the case study presented in the following section.

Related Work. The problem of mutual influence among behaviors were tackled in different approaches. For example, in [9] the author presents a homeostatic system where couples of behaviors are connected through "successor" or "conflicter links" to inhibit or activate each other. These links play a role which is analogous to that of our mutual constraints; however, our regulation mechanisms are different because they are based on attentional modulation of clocks sampling rates. Moreover, our focus is not on the constraint *per se*, but on the effects of constraints on our architecture.

Concurrent tasks interacting with the attentive processes are considered in [16] where a robot architecture integrates active vision and tasks execution. However, mutual influence is not considered while attentive and goal-directed behaviors are integrated and coordinated using a perceptual memory.

Our attentive sampling can be also related to flexible scheduling for periodic tasks in real-time systems. In [3], period modulation is exploited only to keep the system load balanced. Similar techniques can be incorporated in our framework, however, in our case sampling rate and interaction among behaviors depend not only on the computational load, but also on saliencies due to environmental changes, internal states, and goals.

4 Case Study

Inspired by the studies on cognitive distraction during driving activities [15,8], we designed a case study with two conflicting goals. In a hallway there are some clusters of green blobs distributed on the left and on the right wall. The robot has the task of running across the hallway in the shortest time possible, while counting all the green blobs (see Fig.2(a)(b)). The two tasks conflict on the speed of the robot. In fact, the first task would require a high speed, while the second, in order to effectively count all the blobs, would require a slow one.

Environment. The hallway is straight, without obstacles, $14m$ long and has a width of $1.60m$ (see Fig.2(a)). All along the walls there are 27 green blobs arranged in 3 clusters of 9 blobs each, symmetrically disposed as a 3x3 grid (see Fig.2(b)). The three grids are randomly distributed along the walls.

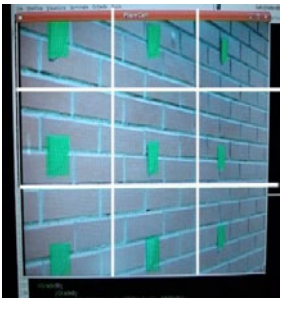
(a)  (b) 

Fig. 2. (a) A snapshot of the robot in the environment. (b) A snapshot of the robot field of view with a superimposed grid to identify different areas.

Behaviors. In order to accomplish the two tasks we implemented three behaviors: `RUN`, `SEARCH` and `SCAN` (see Fig. 3).

`SEARCH` looks for green blobs on the left and right wall. When the behavior is activated, it causes a random movement of the pan-tilt camera. This behavior is activated every machine cycle until it detects a green area. In this case, the period is increased from 1 to 9 machine cycles proportionally to the amount of green color detected in the wall (i.e. the minimum time to allow to the `SCAN` behavior to identify the 9 blobs composing the cluster) (see Sec. 4.1).

`SCAN` counts the blobs once a salient area is identified. According to active vision [2] approach, in order to count an object on the wall, the camera has to center the object in its field of view, simulating a saccadic movement. For this purpose, we split the field of view of the camera into nine areas (see Fig.2(b)).

`RUN` sets the speed of the robot. Differently from the previous behaviors, the effect of the activation of such behavior continues even when the behavior is off. In fact, after the behavior has sent a command to the motors, the controller of the robotic system will keep such speed until a new command will arrive. The

value of the speed is in inverse proportion to the value of its period. The range of allowed speeds is from $0.01m/s$ to $0.24m/s$.

The system starts with a medium speed, looking for green objects on the walls of the corridor. Its behavior will change according to the visual percept. When the system detects a green object, the `SCAN` behavior period decreases, allowing the robot to slow down its speed and to count the objects it detects. Similarly, if no green objects are detected, `SEARCH` and `RUN` periods become smaller, allowing a more accurate exploration (moving several times right and left the camera looking for objects), and increasing the system speed in order to reach the end of the corridor as fast as possible.

4.1 Mutual Influence Rules

The regulation of mutual influence of two clocks, with periods p_A and p_B, depends on the statical and the dynamical priorities between behaviors and the relationship $R(p_A, p_B)$.

Priorities. Priorities in changing periods depend on the importance of the behavior in accomplishing the task and in ensuring the safety of the robot. Behaviors that are safety critical have the maximal priority, hence the other behaviors will be activated consequently. In the case of behaviors with the same priority, the policy for updating the value of the period is "the first takes all", i.e., at each machine cycle, the first behavior that changes its period has to notify to other behaviors such variation. The other behaviors have to modulate their periods accordingly.

Relationship. If two behaviors A and B, with respectively p_A and p_B periods and with ranges $[p_{Amin}, p_{Amax}]$ and $[p_{Bmin}, p_{Bmax}]$, share the same resources and are potentially in conflict, we have to define a relationship between these two values. To better understand, we consider what happens in the frequency domain in which a low pass filter prevents the passage of frequencies below a particular cutoff frequency. If K is this cutoff frequency (i.e. in some way the maximum bandwidth available) representing in our case the maximum rate of behavior activation, and $f_A = \frac{1}{p_A}$ and $f_B = \frac{1}{p_B}$ respectively represent the activation frequencies of two conflicting behavior, with the relation: $f_A + f_B \leq K$, we indicate that each frequency will benefit from the breadth bandwidth not used by the other and vice versa. Likewise, if the activation period p_A assumes a particular value within its allowed range $[p_{Amin}, p_{Amax}]$, the period p_B can only assume a value within $[p_{Bmin}, p_{Bmax}]$, limited to the remaining bandwidth. However, if two behaviors need to be executed simultaneously in order to realize a macro behavior, or if their outputs may be summed and are not in conflict, we may assume the following synchrony relationship: $|p_A - p_B| = 0$. In our architecture we have that the `SCAN` behavior (with period p_{Sc}) and the `SEARCH` behavior (with period p_{Sr}) cooperate on the achievement of one of the tasks, but conflicts on the use of the pan/tilt camera. On the contrary, `RUN` (with period p_R) conflicts with `SCAN` on tasks. Indeed, the first one has the goal to reach

the end of the corridor as soon as possible, while the second one needs to slow down as much as possible the speed of robot in order to optimize the counting phase. Finally, the RUN behavior and the SEARCH behavior can cooperate in the achievement of their own task. In fact, both require a high speed. Let α, β, and γ be constants equal to $\alpha = p_{Scmax} + p_{Srmin}$, $\beta = 0$ and $\gamma = p_{Scmax} + p_{Rmin}$, in this test the relationships among the periods of these behaviors can be formalized as $\alpha \geq p_{Sc} + p_{Sr}$ (or $\alpha = p_{Sc} + p_{Sr}$ if we want a strong dependence), $\beta = |p_{Sr} - p_R|$ and $\gamma \geq p_{Sc} + p_R$ (or $\alpha = p_{Sc} + p_{Sr}$ if we want a strong dependence).

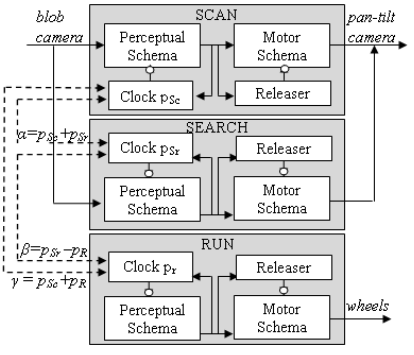


Fig. 3. Control architecture for the mobile robot

4.2 Results

In order to evaluate the performances of our system, we compared three different architectures, each with different behaviors settings, implemented on a Pioneer 3DX, equipped with a pan/tilt camera and range sonar (see Fig.2(a)), and defined as follows:

- **AIRM**: all behaviors are equipped with adaptive clocks;
- **AIRM v max**: adaptive clocks only in SCAN and SEARCH; the speed of RUN is kept constant at the highest value (0.24m/s);
- **AIRM v med**: adaptive clocks only in SCAN and SEARCH; the speed of RUN is set to a medium value (0.11m/s);
- S_C2S_R8 **v med**: the behavior activation is periodic (for SCAN $p_{S_c} = 2$, for SEARCH $p_{S_R} = 8$ and for RUN $p_R = 1$), while the speed of the system is kept constant to a medium value (0.11m/s);
- S_C5S_R5 **v med**: the same as the previous case with different periods (for SCAN $p_{S_c} = 5$, for SEARCH $p_{S_R} = 5$ and for RUN $p_R = 1$);
- **Sub v max**: the behaviors are active at every machine cycle and they are coordinated by a subsumption architecture (i.e., SCAN subsumes SEARCH). The speed is equal to 0.24m/s;
- **Sub v med**: the behaviors are active at every machine cycle as in *Sub v max*, but the speed is equal to 0.11m/s.

In Figure 4(a), we summarize the results collected during the tests, considering the number of counted blobs and the time spent to complete the task. For each setting, we performed 10 tests. The *AIRM* architecture performed well in terms of number of blobs counted. In fact, the *AIRM* implementation counts an average of 17.8 blobs. In the case of the *AIRM* architecture with adaptive clocks only for `SCAN` and `SEARCH`, the speed of the `RUN` behavior is kept constant during the tests. In these two cases, the number of counted up blobs is smaller than the *AIRM* case. However, for the *AIRM v max*, time performances are better, while in the case of medium speed the average time ($127.7s$) is comparable with the *AIRM* case ($123s$). Another important thing to highlight is that, while the average speeds in the medium case and in the *AIRM v med* case are comparable, the number of counted blobs is better in the *AIRM* case. This is because the system will adapt itself to the surrounding environment speeding up or slowing down, taking advantages of empty areas to accelerate, while decelerating when it perceives blobs to count.

In the cases of periodic (not adaptive) activation of behaviors (S_C5S_R5 and S_C2S_R8), the performances with respect to the number of counted blobs are worst than in the *AIRM* case. The case S_C5S_R5 presents the worst results in terms of counted blobs. We experienced a little improvement in the case of more frequent activation of `SCAN` (S_C2S_R8). However, we have to highlight that the periodic activation of behaviors in the case of S_C2S_R8 determines a higher number of activation of perceptual schemas (i.e. wasting more resources) with respect to the *AIRM* case, elaborating camera data even during the exploration of empty areas.

In the last set of tests, we evaluated the performances of a subsumption architecture (*Sub*). In this implementation `SCAN` subsumes `SEARCH`. The speed of `RUN` is kept constant at 0.24m/s and 0.11m/s. The performances of *Sub v med* result to be better with respect to the other cases except for the *AIRM* that performs the best. Indeed, the subsumption architecture resolves potential conflicts on resources (i.e. the pan/tilt of the camera) while, without an arbitrator module, such conflicts may reduce performances. However, in this case, similarly to the periodic activation of behaviors, we have a higher number of activations of the `SCAN` perceptual schema that elaborates camera data at each machine cycle. These results make us foresee that in the case of a higher elaboration load an adaptive architecture may significantly improve performances.

In Figure 4(c), we plotted the cost/benefit (time/counted blobs) evaluation. Also from this point of view, the *AIRM* implementation performs better than the others. However, this plot shows that the AIRM case presents a greater standard deviation in the time performances. A high standard deviation implies a high variability of the test results. This variability is caused by the adaptability of the system with respect to the environment and, consequently, to the changes of the system speed.

Finally, in Figure 4(b) we evaluated the error on the number of counted blobs for units of time. This error is evaluated as $(nB - nCB)/t$, where nCB is the

(a)

	n of blobs	time (s)
AIRM	17.8 ± 3.22	123 ± 11.25
AIRM v med	7.6 ± 2.76	127.7 ± 1.64
AIRM v max	5.9 ± 1.45	60.4 ± 1.84
S_C5S_R5 v med	4.6 ± 1.07	127.5 ± 1.96
S_C2S_R8 v med	9.9 ± 2.08	128.7 ± 3.37
Sub v med	13.8 ± 2.49	127.4 ± 2.01
Sub v max	7.4 ± 1.17	61.7 ± 2.83

(b)

	error
AIRM	0.07
AIRM v med	0.15
AIRM v max	0.35
S_C5S_R5 v med	0.18
S_C2S_R8 v med	0.13
Sub v med	0.10
Sub v max	0.32

(c)

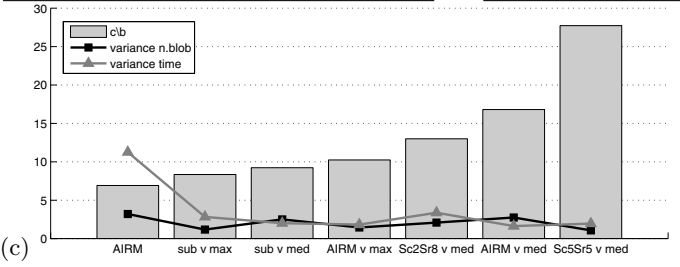


Fig. 4. (a) Performances and standard deviations in term of number of counted blobs and time spent to accomplish the task. (b) Error on the number of counted blobs for units of time. (c) Plot of costs/benefits of the tests. C/b is evaluated as time/counted blobs.

number of counted blobs, nB is the total number of blobs in the environment and t is the time spent to accomplish the task.

5 Conclusion

In this paper, we investigated simple attentional mechanisms for coordinating competitive and cooperative behaviors in a behavior-based robotic system. The results show that the AIRM mechanisms are effective in adapting the frequency of behaviors activations according to the particular circumstances, incrementing or decreasing the attention toward salient aspects of the robot environment or the internal state. We compared our architecture w.r.t. different architectures not endowed with attentional mechanisms. In summary, we observe that the proposed architecture performs better than the others in terms of: number of detected blobs (effectiveness); tradeoff between time and counted blobs (cost/benefit); error of detection (precision); less activations of the perceptual schema (efficiency). Basically, the system can modulate the activation frequencies on the basis of the available resources and external conditions. Indeed, by using the adaptive clocks, the number of behaviors activations substantially decreases compared to the case in which the control system enables the robot's behaviors at each machine cycle, and this results in a substantial gain in performances.

References

1. Arbib, M.A.: Schema theory. In: The Handbook of Brain Theory and Neural Networks, pp. 830–834. MIT Press, Cambridge (1998)
2. Ballard, D.H.: Animate vision. Artif. Intell. 48(1), 57–86 (1991)
3. Beccari, G., Caselli, S., Zanichelli, F.: A technique for adaptive scheduling of soft real-time tasks. Real-Time Syst. 30(3), 187–215 (2005)
4. Burattini, E., Rossi, S.: Periodic adaptive activation of behaviors in robotic system. Int. J. Pattern Recognition and Artificial Intelligence 22(5), 987–999 (2008)
5. Burattini, E., Rossi, S.: Periodic activations of behaviours and emotional adaptation in behaviour-based robotics. Connection Science 22 (in press, 2010)
6. Cooper, R., Shallice, T.: Contention scheduling and the control of routine activities. Cognitive Neuropsychology 17, 297–338 (2000)
7. Coull, J., Nobre, A.: Where and when to pay attention: The neural systems for directing attention to spatial locations and to time intervals as revealed by both pet and fmri. Journal of Neuroscience 18(18), 7426–7435 (1998)
8. Harbluk, J.L., Noy, Y.I., Eizenmann, M.: Impact of cognitive distraction on driver visual behavior and vehicle control. Tech. Rep., 81st Annual Meeting of the Transportation Research Board, Washington, DC (January 2002)
9. Maes, P.: A bottom-up mechanism for behavior selection in an artificial creature. In: Proc. of the First Intern. Conf. on Simulation of Adaptive Behavior on From Animals to Animats, pp. 238–246. MIT Press, Cambridge (1990)
10. Miniussi, C., Wilding, E., Coull, J., Nobre, A.: Orienting attention in time: Modulation of brain potentials. Brain (122), 1507–1518 (1999)
11. Niebur, E., Hsiao, S.S., Johnson, K.O.: Synchrony: a neuronal mechanism for attentional selection? Curr. Opin. Neurobiol. 12(2), 190–194 (2002)
12. Norman, D.A., Shallice, T.: Attention to action: willed and automatic control of behaviour. In: Davidson, R.J., Schwartz, G.E., Shapiro, D. (eds.) Consciousness and Self-regulation: Advances in Research, vol. IV, pp. 1–18. Plenum Press, New York (1986)
13. Parisi, D.: Internal robotics. Connect. Sci. 16(4), 325–338 (2004)
14. Pashler, H.E.: Divided Attention. In: The Psychology of Attention, pp. 101–166. MIT Press, MA (1998)
15. Patten, C.J.D., Kircher, A., Ostlund, J., Nilsson, L.: Using mobile telephones: cognitive workload and attention resource allocation. Accid. Anal. Prev. 36(3), 341–350 (2004)
16. Wasson, G., Kortenkamp, D., Huber, E.: Integrating active perception with an autonomous robot architecture. Rob. Autom. Syst. 26, 325–331 (1999)

An Empirical Evidence of Braitenberg Vehicle 2*b* Behaving as a Billiard Ball

Inaki Rañó

Computer Sciences and Systems Engineering Dept.
University of Zaragoza,
Spain

Abstract. Braitenberg vehicles have been used for decades on an empirical basis to implement different tasks. Grounded on a mathematical model of vehicle 2*b*, this paper presents empirical evidence that some theoretical results from billiard ball dynamics can be used to analyse the behaviour of the vehicle. The possibility to use such vehicles to cover a region is presented as a consequence of the its behaviour.

1 Introduction

Braitenberg vehicles have been used for decades in robotics on an empirical and intuitive basis. Each vehicle displays a different behaviour according to a thought experiment presented in [3]. Complexity on their behaviour emerges from the interaction of the vehicle with an external stimulus, not from the internal mechanisms that generate the behaviour. They were used as a justification of the simplicity of the behaviour based approach to robotics [4]. Braitenberg vehicles were also successfully used in earlier works of artificial life [14] [17], but they can be used to model animal behaviour as well.

Different kinds of Braitenberg vehicles have been used to provide robots with several abilities like obstacle avoidance [2], local navigation [18], chemical source seeking [6] [9], and even complex tasks like football playing [10], yet always based on empirical evidence. However, since the original work is both qualitative and very intuitive there has been no effort in trying to build a quantitative framework for a further development. In fact, besides purely empirical implementations, Braitenberg vehicles are used as test-bed for evolutive behaviour generation [15] [7], where usually neural networks act as controllers of the vehicles [8]. Neural networks are selected as controllers so they can be evolutively adjusted to perform well, but also because of their properties to deal with noise and approximate any function with no need of design knowledge.

As we will see later, one of the applications of Braitenberg 2*b* vehicles is coverage, a well studied problem on robotics with many applications. In [5] a classification of coverage techniques as heuristic or complete is performed. Complete techniques treat coverage as a multiple sequential planning problem, the execution of a sequence of planned trajectories, which turns it into a time expensive and complexity growing problem. On the other hand, heuristic techniques

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 293–302, 2010.

are based on sequencing behaviours or random motions, making hard to treat them formally or to measure its performance. The use of Braitenberg vehicle 2*b* for coverage could be included in the heuristic category, though it can be treated theoretically under some circumstances. This paper will show that a single mechanism, the Braitenberg vehicle 2b, can be used to generate covering trajectories in simple scenarios. Although the simplicity of the mechanism, the resulting behaviour is difficult to analyse, in the most general case, since it is modelled as a nonlinear differential equation with no equilibrium point. However, we will see experimentally, that mathematical theory of billiard can be used to explain its behaviour under some circumstances. This allows the application of theoretical results to the behaviour of Braitenberg vehicles.

The rest of the paper is organised as follows: Section 2 reviews Braitenberg vehicle 2*b* and presents some results of its behaviour under the influence of a bounded stimulus with a single point maximum. When the stimulus maxima can be approximated by a region a new behaviour appears as presented in Section 3. Conclusions and further work lines are drawn in Section 4. Each section includes its corresponding simulated results.

2 Braitenberg Vehicle 2b in Bounded Stimuli with Point-Like Maximum

The Braitenberg vehicle 2*b* consists on a dual-drive wheeled vehicle with two point-like frontal sensors arranged symmetrically. The connections between the sensors and the motors are shown in Figure 1. Each sensor is linked to the wheel on the opposite side in an increasing way. The sensors capture some scalar stimulus $S(\mathbf{x})$ from the environment D. The stronger the stimulus is perceived on the sensor, the faster the opposite wheel spins, that is what the '+' sign on the figure means. The overall effect is making the robot turn in the direction of the stimulus. Moreover, as the stimulus intensity grows, the linear velocity of the vehicle increases, since both wheels turn faster. Therefore, the vehicle heads the stimulus while increases its velocity. If the stimulus is generated by a point source, the vehicle will eventually reach the source with a maximal speed, this is the reason why the behaviour was originally named aggression. Before drawing any conclusion of the behaviour, we will present the formal model of the vehicle and derive some theoretical results.

Let $D \subset \Re^2$ denote an open set such that $D \cup \partial D$ is connected and compact, this will represent the workspace of the vehicle and its boundary. A stimulus function can be modelled as a C^2 function $S : D \cup \partial D \rightarrow \Re^+ \cup \{0\}$ (a function with second order continuous derivatives), a single absolute maximum and such that $S(\partial D) = 0$ but $S(D) \neq 0$. This represents a scalar stimulus from a point source with a bounded effect, vanishing as we move away from the source. The stimulus is time independent, i.e., even there is spatial variation, it remains constant in time. A Braitenberg 2*b* connection function is a function $F(s)$, $F : \Re^+ \cup \{0\} \rightarrow \Re^+ \cup \{0\}$, with second order continuous derivatives such that $\frac{dF(s)}{ds} > 0 \forall s \in \Re^+ \cup \{0\}$ and $F(0) = 0$. This function models the increasing connection between the

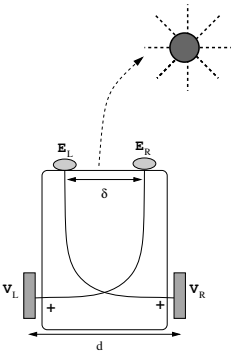


Fig. 1. Internal structure of the Braitenberg vehicle 2*b*

sensors and the motors. The fact that it cannot take negative values implies the vehicle cannot move backward. This has a biological justification since animals usually do not move backward. The condition $F(0) = 0$ can be replaced by $F(S_{min}) = 0$, which makes the vehicle move until it reaches a stimulus threshold. Under these conditions ∂D will be replaced by a closed curve Γ defined by the level set on $S(\Gamma) = S_{min}$. These two functions are the basis for modelling the differential equation governing the behaviour of the Braitenberg vehicle 2*b*. The actual state space of the vehicle is $\Re^2 \times S^1$ since its heading has to be considered. We will denote the Cartesian coordinates $\mathbf{x} = (x, y)$ and θ will be the vehicle heading. Therefore, the whole state will be $(\mathbf{x}, \theta) = (x, y, \theta)$.

Assuming the wheel radius is a multiplying factor included in $F(s)$, the velocities of the right and left wheels of the vehicle are $v_r = F(S(\mathbf{x}_l))$ and $v_l = F(S(\mathbf{x}_r))$, where $\mathbf{x}_r$ and $\mathbf{x}_l$ are the positions of the right and left sensors respectively. Since both, functions $S(\mathbf{x})$ and $F(s)$ are C^2, the velocities can be approximated as a first order Taylor series around the midpoint between the sensors $\mathbf{x}$ as:

$$v_r \approx F(S(\mathbf{x})) + \frac{\delta}{2}\nabla F(S(\mathbf{x})) \cdot \hat{e}_p \tag{1}$$

$$v_l \approx F(S(\mathbf{x})) - \frac{\delta}{2}\nabla F(S(\mathbf{x})) \cdot \hat{e}_p \tag{2}$$

where δ is the distance between the sensors, $\nabla F(S(\mathbf{x}))$ is the gradient of the compound connection-stimulus function and $\hat{e}_p = [-\sin\theta \cos\theta]^T$ is a unitary vector orthogonal to the vehicle's head direction pointing to its left. Assuming a point like differential-drive vehicle, its state evolves according to the following system of differential equations:

$$\dot{x} = F(S(\mathbf{x}))\cos\theta \tag{3}$$

$$\dot{y} = F(S(\mathbf{x}))\sin\theta \tag{4}$$

$$\dot{\theta} = \frac{\delta}{d}\nabla F(S(\mathbf{x})) \cdot \hat{e}_p \tag{5}$$

where d is the wheelbase of the vehicle (see [13] for more details). It is worth noting that the dynamical system describing the behaviour of the vehicle has no stability point in D since equations (3) and (4) never vanish simultaneously.

Intuitively the vehicle will never stop but it will slow down as it approaches ∂D since $F(s) \in C^2$ and $F(S(\partial D)) = 0$. It can be proved that the vehicle will not move outside D, i.e. that the solution $\mathbf{x}(t)$, $\theta(t)$ of the Cauchy problem (3), (4), (5), $\mathbf{x}_0 \in D$ and $\theta_0 \in (-\pi, \pi]$ stays in D for all t. In fact, the solution of the Cauchy problem will stay on the configuration space $D \times S^1$. For a solution going outside the configuration space, the flow defined by the dynamical system (3), (4) and (5) should have a component pointing outside the surface $\partial D \times S^1$. Since the first two components of the flow are zero at the boundary ($F(S(\partial D)) = 0$), and the normal to the boundary has no angular component, the dot product of the flow and the normal vector to $\partial D \times S^1$ is zero, and therefore either the flow vanishes in $\partial D \times S^1$ or is tangent to $\partial D \times S^1$. From this, we can deduce that there can be stability points of the dynamical system on the boundary of D where the gradient $\nabla F(S(\mathbf{x}))$ is orthogonal to $\hat{e}_p$. This means that the vehicle can, under some circumstances, move towards some point of ∂D, though it can be shown that these points are unstable equilibria of the dynamical system. Formally, the points $(\mathbf{x}, \theta) \in \partial D \times S^1$ where $\nabla F(\mathbf{x}) \cdot \hat{e}_p = 0$ are unstable equilibria of the dynamical system.

A special case with analytic solution to the Braitenberg vehicle 2b model appears when the stimulus function has circular symmetry. This is common for some real stimulus and is treated in [12]. It can be shown that if a positive real value r_0 such that $F(S(r)) + r\frac{\delta}{d}\frac{\partial F(S(r))}{\partial r} = 0$ exists, two circular trajectories of radius r_0 are solutions of the differential equations with the proper initial conditions. If equations (3), (4) and (5) are converted to polar coordinates, they can be linearised around the trajectory $r_0(t) = r_0$, $\psi_0(t) = \omega_0 t + \psi_0$ and $\theta_0(t) = \omega_0 t + \psi_0 - \pi/2$, where (r, ψ) are the polar coordinates. This leads to the system of linear differential equations

$$\begin{bmatrix} \dot{\tilde{r}} \\ \dot{\tilde{\psi}} \\ \dot{\tilde{\theta}} \end{bmatrix} = \begin{bmatrix} 0 & -v_0 & v_0 \\ \left(1 + \frac{d}{\delta}\right)\frac{\omega_0}{r_0} & 0 & 0 \\ \frac{d^2F(r_0)}{dr^2} & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{r} \\ \tilde{\psi} \\ \tilde{\theta} \end{bmatrix} \tag{6}$$

where the matrix has constant coefficients, $v_0 = F(S(r_0))$, $\omega_0 = \frac{F(S(r_0))}{r_0}$ and $\tilde{r}(t) = r(t) - r_0$, $\tilde{\psi}(t) = \psi(t) - \psi_0(t)$ and $\tilde{\theta}(t) = \theta(t) - \theta_0(t)$ are the incremental variables. The eigenvalues $\lambda_i = \left\{0, \mp\sqrt{v_0\left(\frac{\delta}{d}\frac{d^2F(r_0)}{dr^2} - \left(1 + \frac{d}{\delta}\right)\frac{\omega_0}{r_0}\right)}\right\}$ of this matrix establish the behaviour of trajectories close to the equilibrium one. When the factor inside the square root is negative, the linearised solution oscillates around the circular trajectory. Moreover, if the quotient of ω_0 and the eigenvalue is an irrational number, the resulting trajectory will be dense in D. This means that for any point in D there is always a trajectory passing arbitrarily close to it, so the vehicle will cover that part of the workspace.

Algorithm 1. Fixed Point of Poincaré Map $P(x)$

Require: $P(x)$, x_0
Ensure: $x_m = P(x_m)$
  repeat
    $j \leftarrow 1$
    $x_0 \leftarrow x_m$
    $S = \{x_0\}$
    repeat
      $x_j \leftarrow P(x_{j-1})$
      $S \leftarrow S \cup \{x_j\}$
      $j \leftarrow j + 1$
    until points in S form a closed trajectory
    $x_m \leftarrow average(S)$
  until $error > \epsilon$
  return x_m

In the general case of non symmetric stimulus, finding an analytic solution to the differential equation is a hard problem. However, some numerical methods exist to find periodic solutions of nonlinear differential equations [11]. Unfortunately, either they need knowledge about the period of the solution or they only work for attracting limit cycles. This is not the case of the system at hand, since the oscillation has constant amplitude around the circle (eigenvalues are pure imaginary), and therefore the equilibrium is not an attractor. We therefore used an approximated way of computing the periodic solutions.

The trajectories being the superposition of two periodic trajectories means that close to the equilibrium, the trajectory is diffeomorphic to a 2D torus in the state space $(\mathbf{x}, \theta)$, and the periodic solution is contained inside the torus. For such a trajectory, the Poincaré map (the intersection of the 2D torus and a plane on the state space) is diffeomorphic to a circle. Moreover, the closer the initial condition is to the equilibrium, the more the intersection will look like a circle. Taking advantage of this, we implemented Algorithm 1, which averages several points of the Poincaré map. This algorithm works even for unstable periodic solutions provided that the initial point is close to the real solution and the divergent component of the map is small enough. The real map will not be exactly a circle since there will be a divergent component and therefore the trajectory will look more like the sampling of a logarithmic increasing spiral. However, when averaging the mean point will get close to the periodic solution, a fixed point in the corresponding Poincaré map.

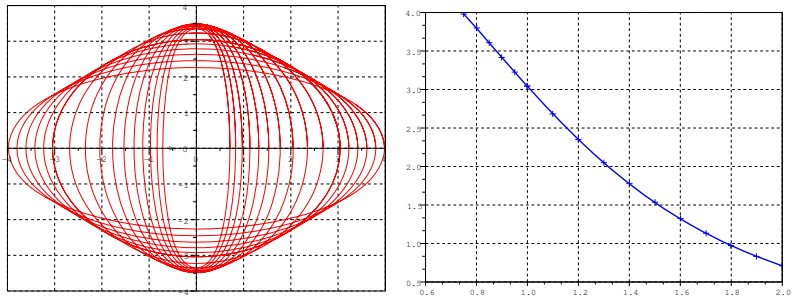
2.1 Simulations for Point Like Stimulus

To test the results of the previous section we simulated the model on a stimulus such that the function $F(S(\mathbf{x}))$ was:

$$F(S(\mathbf{x})) = g_0 - a\mathbf{x}^T \Sigma \mathbf{x} = g_0 - a\mathbf{x}^T \begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix} \mathbf{x} \tag{7}$$

where g_0 and a are positive constants and α is a parameter of the dynamical system. For this function, the domain D is defined by the equation $g_0 > a\mathbf{x}^T\Sigma\mathbf{x}$, and $\partial D = \{\mathbf{x}|g_0 = a\mathbf{x}^T\Sigma\mathbf{x}\}$. When $\alpha = 1$ the stimulus function has circular symmetry and therefore a periodic solution exists for $r_0 = \sqrt{\frac{g_0}{a(1+\delta/d)}}$. For $\alpha \neq 1$ the periodic solution was computed numerically using algorithm 1. In the simulations we used $g_0 = 2.5$, $a = 0.1$ and $\delta/d = 0.85$.

As an initial guess for the position of the periodic solution the closest known periodic solution is used. The algorithm was iterated for different (increasing and decreasing) values of α starting from α close to 1 with initial condition $(3.04, 0, \pi/2)$, and the plane to compute the Poincaré map was $y = 0$. On each iteration the value of the parameter α was slightly changed and the previous solution of the fixed point of the Poincaré map was used as initial guess on the algorithm. The Poincaré map was computed using the time halving algorithm presented in [11].

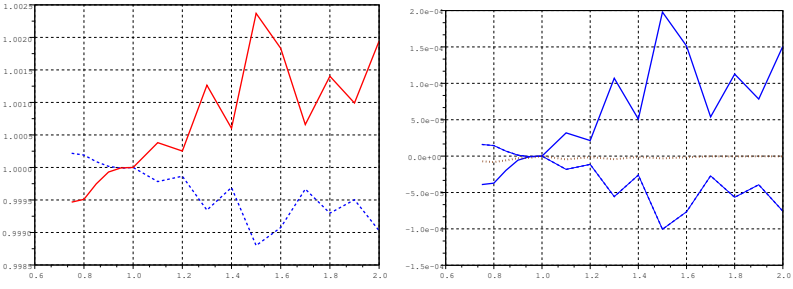


(a) Periodic trajectories. (b) x coordinate of the periodic trajectories.

Fig. 2. Periodic trajectories and initial conditions for different α values

Figure 2(b) shows the resulting x coordinate for the periodic solution. This coordinate is the fixed point of the corresponding Poincaré map, the other coordinates were $\theta = \pi/2$ and $y = 0$. Figure 2(a) shows the simulation of periodic trajectories for different values of the parameter α. The trajectories were computed by integration of the system of differential equations with initial conditions $(x_0, 0, \pi/2)$ using a fixed step Runge-Kutta4 algorithm. As the value of α moves away from 1 the trajectories differ from the circle. We saw that for circular symmetric stimulus the trajectories close to the circle oscillate around it, however nothing can be said for a parabolic shaped stimulus. To obtain information about the trajectories close to the periodic solutions we computed the characteristic multipliers of the solutions. If we integrate the Jacobi field obtained from the derivative of the vector flow along a periodic trajectory the characteristic multipliers are the eigenvalues of the Jacobian matrix. The eigenvalues of this matrix are related to the stability of the corresponding Poincaré map and, therefore, also to the stability of the periodic solution [1]. In our case all the trajectories

had two complex conjugate eigenvalues and one real. Figure 3(a) shows the absolute value of the three characteristic multipliers as a function of the parameter α, the complex ones actually generate the same plot. In the case of circular symmetric stimulus all the eigenvalues fall into the unit circle of the complex plane, this matches the theoretical result of oscillating trajectories around the circular solution. For $\alpha \neq 1$ the absolute value of at least one of the eigenvectors has value larger than one, making the Poincaré map and the corresponding differential equation unstable. In fact, this is the reason why we had to implement Algorithm 1. However, since at least one of the eigenvalues has absolute value smaller than one, the fixed point of the Poincaré map is actually a saddle point.



(a) Characteristic Multipliers. (b) Lyapunov Exponents.

Fig. 3. Characteristic multipliers and Lyapunov exponents of the periodic solution for different values of α

Figure 3(b) shows the Lyapunov exponents for the periodic trajectories as a function of the parameter α. If the sum of all the exponents is positive but some of them are negative, the corresponding dynamical system will be chaotic. The relation between the Lyapunov exponents and the characteristic multipliers is given by the expression $\lambda = \frac{1}{T} \ln \nu$, where λ is the Lyapunov exponent corresponding to the characteristic multiplier ν, and T is the period of the solution trajectory (see [1]). The period can be easily obtained while computing the characteristic multipliers, since it is the simulation time needed to get the Poincaré map. The dotted line represents the sum of all Lyapunov exponents. Since its sum is close to zero we can deduce that the behaviour of the vehicle is not chaotic for the given functions and parameters.

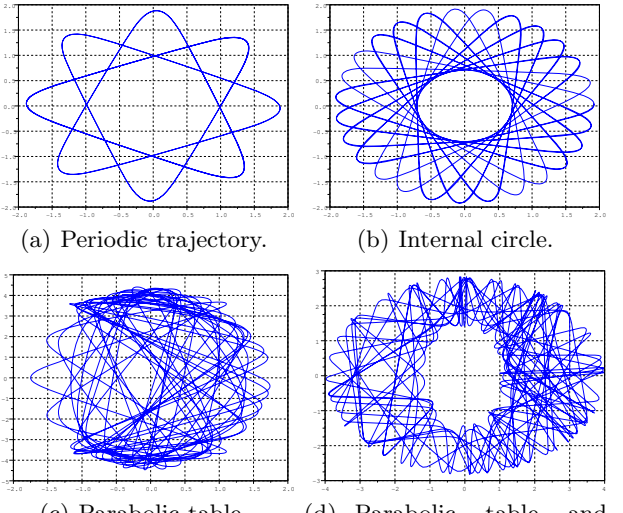
3 Wandering on a Flat Stimulus

So far we saw some results for point like stimulus sources. If the stimulus has no isolated or single maximum the behaviour of the vehicle is different and other analysis techniques need to be used. A special case is a constant stimulus, for which $\nabla F(S(\mathbf{x})) = 0$ in some subset of D. Clearly, since the rotational speed of the vehicle is proportional to this gradient it will not turn, but it will still

move forward since $F(S(\mathbf{x})) > 0$. However, some of the previous results are still valid; the vehicle will move in a bounded area provided $F(S(\mathbf{x})) = 0$ for $\mathbf{x} \in \partial D$. In fact, the area does not need to be simply connected, since the contour of the obstacles can belong to ∂D, and therefore ∂D will be a set of closed curves inside a bigger one. Since $F(S(\mathbf{x})) \in C^2$ and vanishes at ∂D, if there is a region where $\nabla F(S(\mathbf{x})) = \mathbf{0}$, there must also be a region close to the boundary where $\nabla F(S(\mathbf{x})) \neq \mathbf{0}$. Therefore, there will be a region where the vehicle will potentially turn. One way of building a stimulus on a bounded area with obstacles is to generate a function of the distance to the obstacles with continuous derivative. Such function should take a zero value close to the obstacles and the boundary, and a constant value far from them.

To approximate a flat stimulus keeping the function in C^2 we will compute the hyperbolic tangent of the previously chosen function $F(S(\mathbf{x}))$. Obviously this function has also a maximum at the same point as before, however when the vehicle is far enough from the contour, the gradient is so small that it can be considered zero and the trajectory can be approximated by a straight line. Having a close look at the trajectories, we could see they are not straight lines since the derivative of $\tanh(F(S(\mathbf{x})))$ does not vanishes. In sum, the trajectory will almost follow a straight line and will only blend close to ∂D. This behaviour looks like the one of balls on a billiard table with elastic collisions. In fact, the trajectories described by billiard balls are not C^2 functions, and they follow a geometric reflection law on the border of the table.

It is well known that billiard balls on a circular table can follow periodic trajectories after several bounces when the angle between the tangent to the



(a) Periodic trajectory. (b) Internal circle.

(c) Parabolic table. (d) Parabolic table and parabolic obstacle.

Fig. 4. Braitenberg vehicle as a billiard ball

table and the trajectory (ϕ) is a rational multiple of π, otherwise the trajectory is dense [16]. We simulated Braitenberg vehicle 2*b* on a circular stimulus with an almost flat region in the centre and found that it also follows a periodic trajectory when the proper initial conditions are selected. Figure 4(a) shows the trajectory described by the vehicle starting with initial conditions $(1, 0, \pi/2.715)$, which is clearly periodic. Another result for unit radius circular billiard table is that the trajectories are tangent to a circle with radius $r_{int} = \cos\phi$ [16]. Figure 4(b) shows that, even the trajectory of the Braitenberg vehicle 2*b* is not the same as the billiard ball, it is tangent to some circle. Given the initial condition of the vehicle $(\mathbf{x}, \theta) = (1, 0, \pi/4)$ and the size of the circle $r = 2$ we can compute the angle of a bouncing billiard ball $\phi = 65°$ and the radius of the inner circle $r\cos\phi = 0.85$, which roughly matches the simulation results for the Braitenberg vehicle. In fact, the actual radius of the internal circle of the figure is $r = 0.705$ slightly smaller. Neither the apparently straight trajectories far from the circle boundary are straight lines nor the bounces are real bounces, though there is a clear experimental match between the expected results from the ball and the simulation of the Braitenberg vehicle pointing to a new way of looking at them and analysing their behaviour.

Other results from billiard theory also apply to Braitenberg vehicle 2b, like the ones related to trajectories on a elliptic billiard table. However, these results apply only to a certain limit. Figure 4(c) shows the trajectory of a simulated vehicle on an elliptic stimulus with quite different lengths on the principal axis. Theoretically the resulting trajectory should be tangent to an hyperbola or an ellipse, but it is clearly not true in this case. It seems that we moved away from the parametric region where the billiard ball approximation is valid. Figure 4(d) shows the simulation of a Braitenberg vehicle inside an elliptic region with an elliptic obstacle inside. The trajectory is complex to analyse but the simulation points that it is dense on the free space making the control mechanism look suitable for coverage tasks.

4 Conclusions and Further Work

This paper shows that there is empirical evidence to apply the mathematical results of billiard balls to Braitenberg vehicle 2*b*. For simple shapes of the environment the control technique of the vehicle can be used to generate dense trajectories, the ones suited on covering tasks. Furthermore, this technique is clearly much simpler than using algorithms for sequential planning of the trajectories. Obviously these results need a formal support and the conditions under which the billiard theory can be applied need to be identified.

References

1. Anishchenko, V.S., Astakhov, V., Neiman, A., Vadivasova, T., Schimansky-Geier, L.: Nonlinear Dynamics of Chaotic and Stochastic Systems. In: Tutorial and Modern Developments. Springer, Heidelberg (2002)

2. Bicho, E., Schöner, G.: The dynamic approach to autonomous robotics demonstrated on a low-level vehicle platform. Robotics and Autonomous Systems 21, 23–35 (1997)
3. Braitenberg, V.: Vehicles. Experiments in synthetic psycology. The MIT Press, Cambridge (1984)
4. Brooks, R.A.: A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation 2(1), 14–23 (1986)
5. Choset, H.: Coverage for robotics – a survey of recent results. Annals of Mathematics and Artificial Intelligence 31, 113–126 (2001)
6. Farah, A.M., Duckett, T.: Reactive localisation of an odour source by a learning mobile robot. In: Proc. of the 2nd Sweedish Workshop on Autonomous Robotics (2002)
7. Floreano, D., Mondada, F.: Evolution of homing navigation in a real mobile robot. IEEE Transactions on Systems, Man and Cybernetics 26(3), 396–407 (1996)
8. French, R.L.B., Cañamero, L.: Introducing neuromodulation to a Braitenberg vehicle. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 4188–4193 (2005)
9. Lilienthal, A.J., Duckett, T.: Experimental analysis of gas-sensitive Braitenberg vehicles. Advanced Robotics 18(8), 817–834 (2004)
10. Meriçli, T., Akın, H.L.: Soccer without intelligence. In: Proc. of the 2008 IEEE Int. Conf. on Robotics and Biomimetics, pp. 2079–2084 (2008)
11. Parker, T.S., Chua, L.O.: Practical Numerical Algorithms for Chaotic Systems. Springer, Heidelberg (1989)
12. Rañó, I.: Hanging around and wandering on mobile robots with a unique controller. In: Proc. of the 4th European Conf. on Mobile Robots, pp. 135–140 (2009)
13. Rañó, I.: A steering taxis model and the qualitative analysis of its trajectories. Adaptive Behavior 17(3), 197–211 (2009)
14. Resnik, M.: LEGO, Logo and Life. In: Artificial Life. Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems, pp. 397–406 (1987)
15. Salomon, R.: Evolving and optimizing Braitenberg vehicles by means of evolution strategies. Int. J. Smart. Eng. Syst. Design 2, 69–77 (1999)
16. Tabachnikov, S.: Geometry and Billiards. American Mathematical Society, Providence (2005)
17. Travers, M.: Animal construction kits. In: Artificial Life. Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems, pp. 421–442 (1987)
18. Yang, X., Patel, R.V., Moallem, M.: A Fuzzy-Braitenberg Navigation Strategy for Differential Drive Mobile Robots. Journal of Intelligent Robotic Systems 47, 101–124 (2006)

Insectomorphic Robot Maneuvering on a Movable Ball*

Yury F. Golubev and Victor V. Korianov

Keldysh Institute for Applied Mathematics, Russian Academy of Sciences,
Miusskaya pl. 4, Moscow, 125047 Russia
golubev@keldysh.ru, korianov@keldysh.ru

Abstract. The problem of generating motions of a six-legged robot with the help of a ball rolling on a horizontal support plane is investigated. The robot motion is synthesized in order to climb on a ball from a support plane, accelerate or decelerate the ball in the directions of the longitudinal or lateral axes of the body. Robot provides its dynamic stability on the ball. The goal of robot's efforts is to remove the ball to a given place on the support plane. Robots behavior is accomplished with the help of dry friction forces only. The motion of the body and legs is executed by imposing servo constraints in the form of adaptive step cycles of legs and the required geometrical and dynamical properties of the body motion. The results of 3D computer simulation of the controlled robot dynamics are presented.

Keywords: legged robot, adaptive control, stabilization.

Introduction

This paper develops the results of [1,2,3]. The ability of a walking robot to move on a real terrain can be formed by gradual training of a robot to overcome both isolated obstacles and their typical combinations. Some examples of overcoming terrain with sufficiently small unevennesses were presented in [4]. Obstacles of big height were considered in [5,6,7]. Robots using vacuum suckers, electrostatic or other kinds of adhesion are developed [8], since they allow the robot to move along vertical walls. Together with this, walking robots are able to move on structures of considerable height simply using the Coulomb friction forces, as this is frequently done by animals.

In certain extremal situations, the ability of a walking robot to overcome obstacles may considerably depend on the ability of the robot to provide efficiently the dynamic stability [1]. It is interesting to investigate the possibility of providing dynamic stability on support objects being in the state of indifferent equilibrium, such as a cylinder or a ball, free lying on a solid horizontal surface. It is also interesting to learn how to use the dynamic properties of support objects

* This work was supported by the Russian Foundation for Basic Research, project No 10-01-00160.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 303–312, 2010.

of this type in order to transport them together with the robot to the required place. Because the ball has only one support point on the plane it seems to be a more arduous object than a cylinder. A walking robot, getting onto a surface of the ball, heightens the center of mass of the entire mechanical system and turns it into unstable one.

In this paper we present a solution to the problem of the robot climbing onto a massive ball, which can roll freely on a horizontal plane, of providing an acceleration and deceleration of the ball by special motions of the robot on the ball, a passage of the robot from the ball to a shelf. A set of typical methods that allow the robot to execute maneuvers under a given friction coefficient restriction are proposed. Typical motion elements are given using the adaptive trajectories of feet movements and the trajectories of motion of characteristic points of the body. The efficiency of the proposed methods is confirmed by means of three-dimensional computer simulation [9] of the complete dynamics of the robot having 24 degrees of freedom, together with the ball of six degrees of freedom. The results of computer simulations of the operation of the described algorithms for motion design are presented.

1 The Problem Statement

The six-legged robot construction is shown on Fig. 1. The details of robot's kinematics can be found in [1]. Here we note only that legs are numbered from rear edge of robot's parallelepiped-shaped rigid body to its front edge so that the rear legs have the numbers 1 and 2, the middle legs have the numbers 3 and 4, the front legs have the numbers 5 and 6. All right legs have odd numbers. The robot can contact with a support surface and with any other objects only by feet. Robot's legs should not have mutual intersections during all stages of the motion.

Dimensions of the robot meet the following condition:

$$a : b : c : l_1 : l_2 : R = 1 : 0.5 : 0.1 : 0.5 : 0.33 : 0.8,$$

where R — radius of the ball, a — side of the body (length), b — front or rear edge (width), c — thickness of the body. Six identical insectomorphic legs are symmetrically attached to the sides of the body. Points in which the legs are attached (legs attachment points) are located uniformly along the sides. Each leg consists of two links: hip, length l_1 and shank, length l_2 (Fig. 1). The body and the links of legs have some volume and mass by gravity.

We assume that robot is equipped with the electromechanical drives in joints and has full access to the following information: the geometry of obstacles, an own position relatively to the obstacles, joint angles and velocities. The programmed values of the joint angles are generated by the algorithm of control. The algorithm is not strictly fixed, the information about the actual robot configuration during the motion essentially used. For realization of programmed values of joint angles the servo-control method is used [10]. Required motion is generated as a servo-constraint, which robot will be aimed to realize by setting the control voltage on the drives [1].

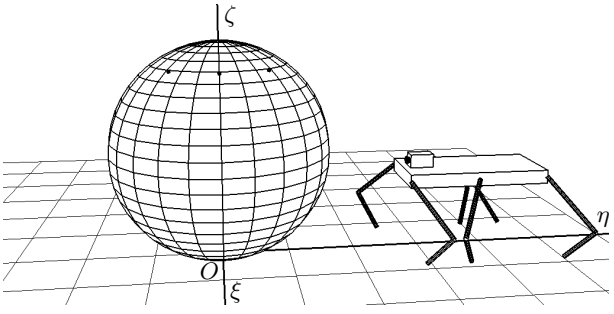


Fig. 1. The robot and the ball

The solving problem is as follows. Initially robot and ball dispose on horizontal support plane $O\xi\eta$. Axis $O\zeta$ is directed up to vertical. Point O is the initial contact point of the ball and the support plane. Reference frame $O\xi\eta\zeta$ is immovable. Robot have to approach to the ball, climb onto the top of it and make the ball to move in the prescribed direction. At final position of this motion there is a shelf. Robot have to approach the ball to the vertical wall of the shelf and get onto the horizontal top of the shelf from the ball. The problem is not very simple, because the robot don't know forces in contact point of the ball and support plane as well as forces in contacts of feet with the ball.

2 Climbing onto the Ball

We suppose that initial position of the robot is symmetrical relative to a vertical plane getting through the center of the ball and there is no slippage in contact points. The motions of the ball and the robot's body are designed as plane-parallel with that vertical plane. Let ω_0 is an initial angular velocity of the ball and $\dot{\varphi}$ is a constant angular velocity of the robot's body relative to the ball center. We need that the ball should stop when robot will get onto the top of the ball. In accordance with the angular momentum theorem relative to a support point of the ball values ω_0 and $\dot{\varphi}$ are related by the equation

$$\dot{\varphi}^2 PR - \dot{\varphi}\omega_0 R + Pg = 0, \quad P = \ln\frac{1+p}{1+p\sin\varphi_0}, \quad p = \frac{mRr}{J_b + (M+m)R^2}, \quad (1)$$

where M — mass of the ball, J_b — axis inertia moment of the ball in a point of the ball contact with the plane, m — mass of the robot, r — distance between the center of the ball and the robot's center of mass. Under the condition

$$\omega_0^2 \geq \frac{4P^2 g}{R} \longrightarrow v_0 \geq 2P\sqrt{gR}, \quad (2)$$

where v_0 — the initial velocity of the ball center, the equation (1) with regard to $\dot{\varphi}$ has two positive roots. We choose minimal root according to design requirements:

$$\dot{\varphi} = \frac{\omega_0 R - \sqrt{\omega_0^2 R^2 - 4P^2 Rg}}{2PR}. \quad (3)$$

It is evident that robot can't climb the big enough ball if the ball is immovable at start of climbing. So robot has to push the ball in the same direction that it approach to the ball and to give the ball big enough initial velocity. At the start of a push the body of the robot should be orientated along the surface of the ball. Let us suppose that front edge of the body should be higher then rear edge. Then, for reasons of reaching the support plane by feet, it is clear that the rear legs have to engage a push. After separation of all the feet from the support plane front and middle legs must be securely entwine the ball so that the robot could move the body along the ball without slipping feet on the surface of the ball. Due to transfers of feet require finite time, we conclude that by the beginning of push front and middle legs must already be on the ball, and the push should be carried out only by the rear legs. To avoid rotation of the ball around a vertical axis the gallop gait will be used for climbing, with posing the body and support points symmetrically about the vertical plane.

Let feet of some number of legs be on the ball and other part of feet are on the support plane. If the ball can only roll without slippage and robot moves preserving distances between its support points, then the ball can't move. This property allow the robot to prepare initial position before pushing the ball. That position is shown at Fig. 2, a. After push the ball begins to roll and feet of rear legs transfer to ball (Fig. 2, b). The robot climbs the ball slowing it. Robot choose support points on the ball to avoid mutual intersection of legs. When robot comes to the top of the ball the velocity of the ball becomes almost zero (Fig. 2, c).

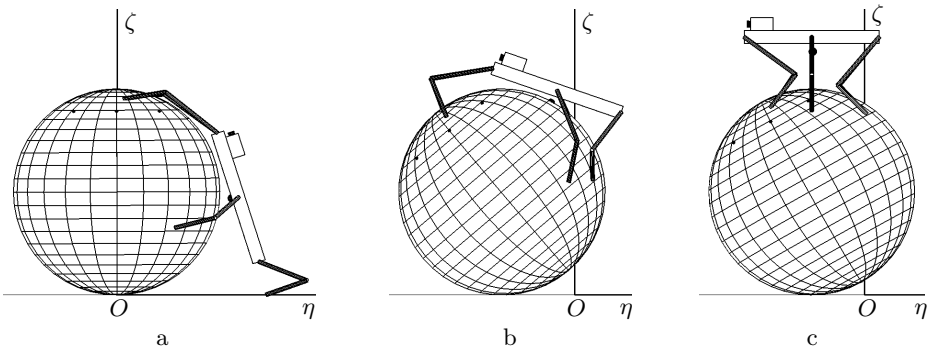


Fig. 2. Climbing onto the ball

3 Stabilization, Speed Up and Slow Down the Ball

Once the robot got to the ball, it may be tasked to achieve, while staying on the ball, some place on support plane. If this place can be reached, forcing the center of the ball to move precisely along the longitudinal axis of robot's body, that to this end, we can apply the algorithm presented in [3]. However, the target point can be located anywhere on the plane, and can be separated by any obstacle from the place where the ball and the robot are.

In this situation, the robot can remaining on the top of the ball, for example, turn itself on the ball in the direction of target point or in the direction of the desired bypass obstacles. However, if the robot will do it, then the ball by theorem of change the angular momentum for the system acquire a vertical component of the angular velocity, which would interfere movement in the desired direction. A corresponding problem of sighting appears. The decision of it will be hampered by the lack of reliable information about the properties of interaction between the ball and the support plane.

Another method is free from these shortcomings and able to provide achivement of the desired location on the plane. This method can be constructed on the basis of implementation of the robot motion with the ball on the grid without body rotation around a vertical axis. It uses the opportunity the robot motion in the lateral direction relative to the longitudinal axis of the body. The triple gait when the legs are moved across the robot's body turns to be most effective in this case.

Firstly let us consider the longitudinal motion. Assume that at the beginning of motion the robot and ball are in rest, and the robot stands on the ball so that its center of mass is projected at the ball support point. The robot body is oriented along the unit vector $\mathbf{e} \perp \mathbf{g}$, specifying the direction of the forthcoming motion of the ball center B. The robot has to accelerate the ball to a required velocity and then to move together with the ball in the direction of the vector $\mathbf{e}$. The regular stage of the ball acceleration in the time interval $[t_0, t_f]$ is performed according to the formula $\delta\mathbf{r}_k^{(1)} = k_v[v_d - v_e(t_0)]\mu(\tau)\mathbf{e}$, where k_v is a coefficient introduced for smoothing the acceleration process, v_d is the magnitude of the required velocity, $v_e = \mathbf{v} \cdot \mathbf{e}$ is the projection of the ball velocity on the direction $\mathbf{e}$, $\tau = t - t_0$, and the function $\mu(\tau)$ is determined by the formula

$$\mu(\tau) = \begin{cases} \dfrac{\bar{u}\tau^2}{2}, & 0 \le \tau < \tau_1, \\ \bar{u}\left[\dfrac{\tau_1^2}{2} - \dfrac{(\tau-\tau_1)^2}{2} + \tau_1(\tau-\tau_1)\right], & \tau_1 \le \tau < \dfrac{T}{2} - d, \\ 2, & \dfrac{T}{2} - d \le \tau < \dfrac{T}{2} + d, \\ 2 - \dfrac{\bar{u}}{2}\left(\tau - \dfrac{T}{2} - d\right)^2, & \dfrac{T}{2} + d \le \tau < \tau_2, \\ 2 - \dfrac{\bar{u}}{2}\left(\tau - \dfrac{T}{2} - d\right)^2 + \bar{u}(\tau-\tau_2)^2, & \tau_2 \le \tau \le T, \end{cases} \tag{4}$$

where

$$\tau_1 = \frac{T}{4} - \frac{d}{2}, \quad \tau_2 = \frac{3T}{4} + \frac{d}{2}, \quad \bar{u} = \frac{32}{(T-2d)^2}$$

and $d = 0$ for this case.

The acceleration stages can be performed both on each gait period and in several periods. For a sufficiently large number of acceleration stages, the ball velocity approaches the value v_d. As an illustration, Fig. 3,a presents the ball

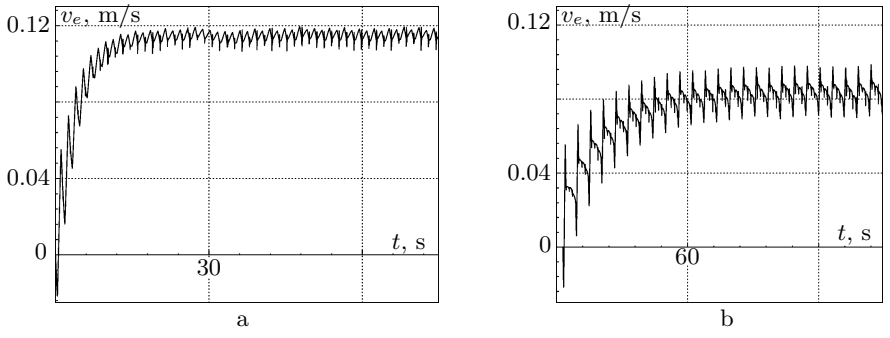


Fig. 3. Acceleration of the ball

acceleration process for the case $v_d = 0.12$m/s. The dependence of the velocity of the ball center on time is shown. The acceleration stages (pushes) are executed at each gait step. The irregularities of velocity variation are caused by collisions of feet with the ball surface. It is clear that the velocity approximates "in average" the required value. Fig. 3,b gives the dependence of the velocity of the ball center on time for the same value $v_d = 0.12$m/s, under the condition that accelerating pushes are executed once for each four steps. It can be seen that the velocity approaches "in average" a value that is lower than the required one. This can completely be explained by the fact that for the time of executing four steps by the robot, the ball velocity decreases because of natural dissipation.

To realize the rectilinear ball motion, the robot walks on the ball by the "triples" gait. To avoid a transversal motion of the ball, the center of mass of the robot has to be projected on the straight line passing through the support point of the ball in the longitudinal direction $\mathbf{e}$. To avoid the deviation of robot's center of mass from pointed line the algorithm was proposed in [3] for calculating corresponding displacement of robot's body

$$\delta\mathbf{r}_k^{(2)} = \chi\frac{\mathbf{g}\times\mathbf{e}}{g},$$

where χ is coefficient depending on legs position.

Because of errors of execution and other errors, the motion of the ball support point may deviate from the straight-line one. Therefore a stabilizing control with a feedback returning the ball support point to the prescribed straight line is necessary both by the coordinates and by the velocity of the ball center. This feedback is provided by the term

$$\delta\mathbf{r}_k^{(3)} = -\left(\mu(\tau)[k_1\mathbf{r}_b(t_{n0}) - k_2\mathbf{r}_b(t_{(n-1)0}) - (k_1-k_2)\mathbf{r}_0]\cdot\frac{\mathbf{g}\times\mathbf{e}}{g}\right)\frac{\mathbf{g}\times\mathbf{e}}{g}, \quad (5)$$

where $k_1 > k_2$ are the feedback coefficients, $\mathbf{r}_0$ is the position of the ball center before speeding up, $\mathbf{r}_b(t_{n0})$, $\mathbf{r}_b(t_{(n-1)0})$ are the positions of the ball center at the beginning of the current and at the beginning of the preceding gait period. The

filter in the square brackets takes into account the transversal deviations both in the coordinate and in the velocity from zero values. Finally the program motion law of the body during the ball longitudinal motion is expressed by the formula

$$\delta \mathbf{r}_k = \delta \mathbf{r}_k^{(1)} + \delta \mathbf{r}_k^{(2)} + \delta \mathbf{r}_k^{(3)}.$$

The formed body motion is realized by joint control torques of the supporting legs [5]. The support points are chosen in the coordinate system $B\mathbf{e}'_1\mathbf{e}'_2\mathbf{e}'_3$, turned relative to the motion direction by the angle $\alpha = Tv_b/(2R)$, where T is the gait half period. This period is also the period of controlling the body motion. The matter is that the support points, being immovable on the ball, move as the ball rotates. The angle α, chosen in the specified way, provides the symmetry of positions of the support points at the time instant of putting the legs on the surface, and the position of the corresponding support points at the time instants of lifting the same legs with respect to the vertical line passing through the ball center.

The lateral motion is organized by analogous to longitudinal motion. Only difference is that the function $\mu(\tau)$ in (4) is calculated at $d = T/4$, with reduced value of coefficient k_v and $k_1 = k_2$ in (5). This is done because during the lateral motion there is the dangerous for support points to appear under the robot's body.

4 Robot Passage between the Ball and the Shelf

Assume that at the initial position the six-legged robot situated at the top area of a shelf with a vertical edge, and a ball of a radius R is located on the horizontal bottom ground and is leaned against the shelf wall. The shelf height is $\approx 1.8R$. The robot has to move to the ball into the position from which it can start the acceleration of the ball in the direction perpendicular to the shelf edge. The main difficulty of the problem consists in the fact that in the course of the passage, the ball may roll out from the shelf (releasing constraint) or it may start rolling along the shelf under an insufficient friction force between the ball and the vertical wall of the shelf. If the robot press the ball actively to the shelf, then the friction force, arising at the point of a contact of the ball with the vertical wall of the shelf and bounded by the friction cone, counteracts the roll of the ball along the vertical wall of the shelf. This circumstance can be used by the robot in order to get safely to the ball top. The pressing of the ball to the shelf is regulated by the fact that the projection of the center of mass of the robot on the horizontal plane during the passage from the shelf to the ball is moved in the required way towards the shelf from the point at which the ball touches the bottom ground. To provide that in the passage of the robot from the shelf to the ball the pressure of legs on the ball is, if possible, symmetric with respect to the vertical plane passing through the ball center and containing the direction of the future ball motion, the "gallop" gait is used. The same reasons are used to organize motion from ball to the shelf by reverse doings.

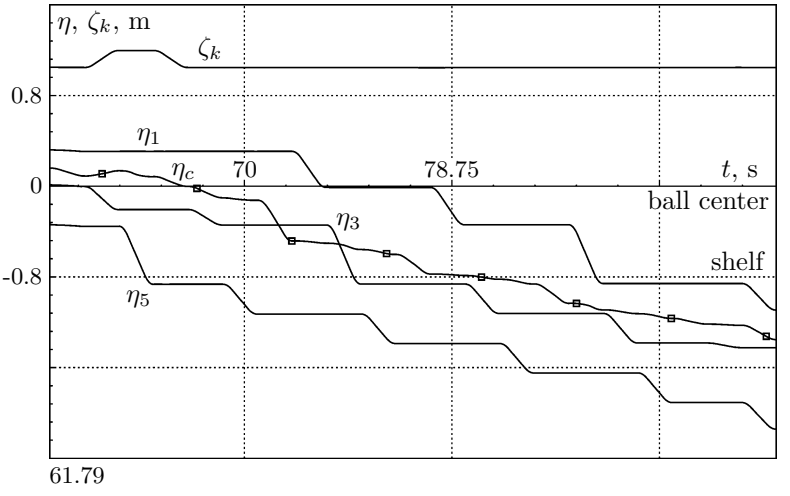


Fig. 4. Transfer from the ball to the shelf

Fig. 4 presents plots of the dependences of the longitudinal coordinates of feet of legs and the center of mass of the robot on time during the passage of the robot from the ball to the shelf. The maneuver is executed by the "gallop" gait, the dependences for the legs only with odd numbers are shown. We can notice that in the course of the passage, the center of mass of the robot is always projected within the supporting polygon and is situated between the shelf edge and the ball center. As a result of this, the robot in the passage to the shelf presses the ball to the shelf and thus provides comfort conditions for executing the maneuver. Additionally, the dependence of the vertical coordinate of the body center is presented. It shows that at a certain time instant the body raises itself in order to avoid mutual crossing legs. The maneuver of passing from the ball to the shelf can be executed successfully both in the case when the ball stops at a small distance to the shelf and in the case when the ball has a small velocity in approaching the shelf. In a considerable collision of the ball and the shelf, the ball gains the reverse velocity, and the robot goes on the ball in the reverse direction and can execute a new attempt in order to approach the shelf.

5 Computer Simulation

Computer simulation was performed similarly to that in [1,3]. The control was implemented in the form of a DLL-library, appended to the software complex "Universal mechanism" [9]. In combination, the software complex "Universal mechanism" and the DLL-library generate a virtual robot environment. The support surface in this environment is represented by a horizontal plane with shelves located on them. Particular objects, such as a ball and a robot, interact with the support surface and with each other according to the model of the

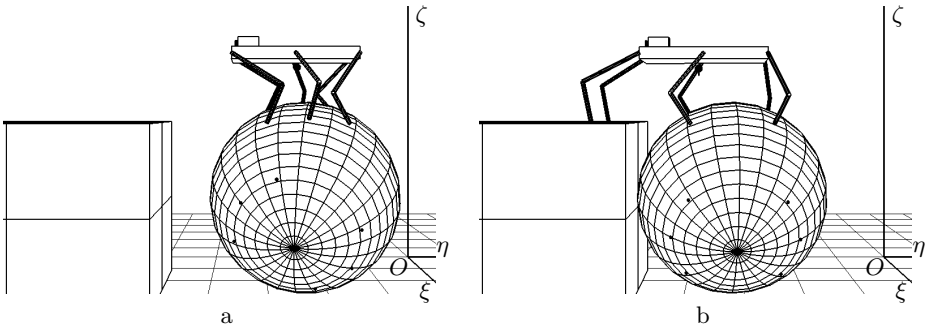


Fig. 5. Climbing to the shelf

Coulomb friction based on viscous–elastic interaction of bodies at the contact points. The interaction of the ball with the support surface is determined by the complex UM [9] and takes into account the viscous–elastic component of the model of the Coulomb friction. The moment of rolling friction was taken into account.

It is supposed that the robot has electromechanical drives of joint angles. The required motion is realized as in reality only by supplying a controlling electric voltage to the electromechanical drives.

Information about the motion of the robot and other elements of the virtual environment arise as a result of computer integration of the differential equations of 3D robot dynamics together with the surrounding objects involved in the system of 30 degrees of freedom. Simulation results are displayed on the monitor in the form of motion of images of the investigated material objects. Fig. 5 shows fragments of the solution of the problem of robot getting from the ball to the shelf. In motion the ball is subject to rolling friction and gains the velocity asymptotically, and the robot moves on the ball stably. When the velocity becomes sufficiently large, the robot by reverse maneuvers decelerates the ball practically to stopping in front of the required shelf. The final stop of the ball occurs because of the rolling friction inherent in the ball motion. As computer experiments have shown, any, even very small motion of the robot on the ball affects proportionally the position and motion of the ball. The stabilization system operates continually even in the apparent rest state of the ball. In fact, stable motion of the robot and ball takes place in a small neighborhood of both a rest position and the required motion.

Conclusion

A solution of new problems on constructing a dynamically stable motion of an insectomorphic robot on a ball, staying in a state of indifferent equilibrium on a horizontal support plane, was obtained. Standard methods for robot climbing the ball from support plane and stabilizing the position of the robot in the upper unstable position on the ball were developed, as well as methods for damping the

ball velocity and providing a required velocity of the ball in the case when the robot is completely situated on the ball and does not touch the support plane.

A stable motions of the robot in order to pass from a shelf to a ball leaned against the shelf and from the ball to the shelf were generated. The proposed algorithms for stabilizing and organization of the task-oriented motion of the robot on a movable ball were worked out by means of computer simulation in a program environment, realizing the interaction of a 3D complete dynamic robot model and objects surrounding it with account of Coulomb friction forces and models of electromechanical drives of robot joints. It was shown that the proposed program motion, providing active stabilization of the motion of the entire system, can be implemented for the friction coefficient no more than 1. The control that provides stabilization of robot motion is adaptive and is constructed in the form of a feedback.

References

1. Golubev, Y.F., Korianov, V.V.: Computer Simulation of a Climbing Insectomorphic Robot. In: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J.C.T., Marocco, D., Meyer, J.-A., Miglino, O., Parisi, D. (eds.) SAB 2006. LNCS (LNAI), vol. 4095, pp. 126–137. Springer, Heidelberg (2006)
2. Golubev, Y.F., Koryanov, V.V.: Motion Design for an Insectomorphic Robot on Unstable Obstacles. In: Proceedings of 11th International Conference CLAWAR 2008, Coimbra, Portugal, pp. 654–661. World Scientific Publishing Co. Pte, Singapore (2008)
3. Golubev, Y.F., Korianov, V.V.: Motion Control for an Insectomorphic Robot on a Movable Ball. J. of Computer and Systems Sciences Intl. 48(5), 801–813 (2009)
4. Pugh, D.R., Ribble, E.A., Vohnout, V.J., Bihari, T.E., Walliser, T.M., Patterson, M.R., Waldron, K.J.: Technical Description of the Adaptive Suspension Vehicle. International Journal of Robotics Research 9(2), 24–42 (1990)
5. Golubev, Y.F., Koryanov, V.V.: Construction of Motions of an Insectomorphic Robot that Overcomes a Combination of Obstacles with the Help of Coulomb Friction Forces. J. of Computer and Systems Sciences Intl. 44(3), 460–472 (2005)
6. Golubev, Y.F., Koryanov, V.V.: A Control for an Insectomorphic Robot in Motion along a Vertical Corner and a Horizontal Beam. J. of Computer and Systems Sciences Intl. 45(1), 144–152 (2006)
7. Golubev, Y.F., Koryanov, V.V.: A Control for an Insectomorphic Robot in Climbing to the Top of a Vertical Corner and in Moving on a Step Ladder. J. of Computer and Systems Sciences Intl. 47(1), 139–148 (2008)
8. Longo, D., Muscato, G.: Adhesion Techniques for Climbing Robots: State of the Art and Experimental Considerations. In: Proceedings of 11th International Conference CLAWAR 2008, Coimbra, Portugal, pp. 6–28. World Scientific Publishing Co. Pte, Singapore (2008)
9. http://www.umlab.ru
10. Golubev, Y.F.: Mechanical Systems with Servoconstraints. J. Appl. Math. Mechs. 65(2), 205–217 (2001)

A Study of Adaptive Locomotive Behaviors of a Biped Robot: Patterns Generation and Classification

John Nassour[1,3], Patrick Hénaff[2], Fathi Ben Ouezdou[1], and Gordon Cheng[3]

[1] Versailles Saint Quentin University - France
[2] University of Cergy Pontoise, ENSEA, CNRS-F95000 Cergy Pontoise
[3] Institute for Cognitive Systems, Technical University Munich
{ouezdou,nassour}@lisv.uvsq.fr, patrick.henaff@u-cergy.fr, gordon@tum.de

Abstract. Neurobiological studies showed the important role of Centeral Pattern Generators for spinal cord in the control and sensory feedback of animals' locomotion. In this paper, this role is taken into account in modeling bipedal locomotion of a robot. Indeed, as a rhythm generator, a non-classical model of a neuron that can generate oscillatory as well as diverse motor patterns is presented. This allows different motion patterns on the joints to be generated easily. Complex tasks, like walking, running, and obstacle avoidance require more than just oscillatory movements. Our model provides the ability to switch between intrinsic behaviors, to enable the robot to react against environmental changes quickly. To achieve complex tasks while handling external perturbations, a new space for joints' patterns is introduced. Patterns are generated by our learning mechanism based on success and failure with the concept of vigilance. This allows the robot to be prudent at the beginning and adventurous at the end of the learning process, inducing a more efficient exploration for new patterns. Motion patterns of the joint are classified into classes according to a metric, which reflects the kinetic energy of the limb. Due to the classification metric, high-level control for action learning is introduced. For instance, an adaptive behavior of the rhythm generator neurons in the hip and the knee joints against external perturbation are shown to demonstrate the effectiveness of the proposed learning approach.

1 Introduction

Biological studies of animals suggest that animals' locomotion is mainly generated at the spinal cord, by a combination of a central pattern generator (CPG) and reflexes receiving adjustment signals from a cerebrum, cerebellum and the brain [1], [2], [3]. These studies were taken into account in robot's locomotion gait in order to implement such mechanism, especially on legged robots [4], [5], [6], [7], [8]. Biologically inspired walking mechanism for legged robot does not require a perfect knowledge of the robot's dynamics. Different models of neural oscillators are widely used to generate rhythmic motion [9], [10], [11], [12], [13].

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 313–324, 2010.

Such oscillations generated by two mutually inhibiting neurons are described in a set of differential equations (e.g. Matsuoka [9]). Rowat and Selverston [14] proposed a new model of rhythmic neuron that can generate different types of patterns such as oscillatory ones. The different behaviors in the activity of these neurons can be used in robot's locomotion to achieve different tasks as well as walking. Complex task, like walking, hopping, running, and obstacle avoidance, require correct synchronization and switching between patterns [15]. In action learning approach, where learning always occurs in the space of parameters, there is a limitation to learn complex tasks, due to the dimension of this space which can drastically increases. This issue can solved by looking for a new representation of patterns. Instead of learning in the space of parameters, learning can be occur inside a new space called patterns' space. (e.g. in case of one dimensional patterns space, patterns will be represented only on one axis). Our work aims to produce a biological inspired neural controller for biped walking, based on CPG with a rhythmic neuron proposed by Rowat and Selverston [14]. According to the environment changes, the adaptation of the neurons behavior will be shown. Therefore, a new space for patterns allowing intrinsic behaviors of a joint motion will be proposed.

This paper is organized as following. Section 2 presents the principles of the neural controller based on the model of rhythmic neurons, which is able to generate CPG-like patterns. The three layers of the CPG used in bipedal control will be presented. A coupling circuitry for walking will be proposed. Next, the walk learning phase based on previous experience with a threshold of vigilance to allow extensive patterns search within a large space of parameters will be detailed in section 3. In the fourth section, a new representation of successful and failure walking patterns is proposed. This approach allows a high level control in space of patterns instead of space of parameters. The effectiveness of our learning scheme, which allows switching between bipedal patterns to achieve different locomotive tasks will be demonstrated. Moreover, an example on the adaptation behavior of the rhythm generator neurons in the hip and the knee joints against external perturbation will be shown. The last section gives a conclusion and details of further developments.

2 Neural Control of Locomotion

Physiological studies suggest that rhythmic movements in animal's locomotion system are produced by a neural network called CPG [16]. It can generate a locomotive rhythmic behaviors with neither sensory nor central inputs [17]. Sensory inputs shape the output of this locomotion system, and allow the animal to adapt its locomotion patterns to external or internal changes. Genetic studies on newborn rat and mice suggest that rhythmic limb movements during locomotion are generated by neuronal networks located within the spinal cord [18]. Matsuoka and McMillen neural oscillators are widely used as mathematical models for non-linear oscillators [9], [10]. These half-centre oscillators consist of two neurons that individually have no rhythmic behavior, but which produce rhythmic outputs when they are reciprocally coupled. This paper present another

model of non-linear rhythm generator. This model is based on the fact that one neuron can generate oscillatory as well as different motor patterns [14].

2.1 Cell Model

The cell model introduced by Rowat and Selverston to modulate the gastric mill CPG in the lobster is interesting due to its ability to generate different patterns by controlling only two parameters [14]. Furthermore, such patterns can be generated with only one neuron without need for another coupled neuron as used in classical models [9], [10]. In the adopted model, the membrane currents of the neuron are separated into two classes, fast and slow, according to their time responses. The sum of all fast currents is modeled by a single fast one, and a single slow current is used to model the sum of all slow ones. This model cell has two differential equations, one for membrane potential V, derived from current's conservation, and one for lumped slow current q, derived from current's activation, see eq.(1).

$$\tau_m.\frac{dV}{dt} = -(fast(V,\sigma_f) + q - i_{inj}) \qquad \tau_s.\frac{dq}{dt} = -q + q_\infty(V) \tag{1}$$

While the fast current is supposed to activate immediately, the membrane time constant τ_m is assumed to be smaller than the slow current's time constant for activation τ_s. We have taken the ratio of τ_s to τ_m to be about 20 as in [14], $\tau_m = 0.05$, and $\tau_s = 1$ for all rhythmic neurons. The injected current is i_{inj}. An idealized current-voltage curve for the lumped fast current is given by: $fast(V,\sigma_f) = V - A_f.tanh((\sigma_f/A_f)V)$. The fast current can represent the sum of a leak current and an inward Ca^{++}. The dimensionless shape parameter for current-voltage curve is given by: $\sigma_f = \frac{g_{Ca}}{g_L}$. Where g_L is a leak conductance and g_{Ca} is the calcium conductance. $q_\infty(V)$ is the steady state value of the lumped slow current, which is given by: $q_\infty(V) = \sigma_s(V - E_s)$. $q_\infty(V)$ is linear in V with a reversal potential E_s. σ_s is the potassium conductance g_K normalized to g_L. σ_s is given by: $\sigma_s = \frac{g_K}{g_L}$. q and i_{inj} have the dimension of an electrical potential. A true current is obtained by multiplying the model current by a leak conductance g_L. V, E_s, i_{inj}, and q are given in millivolts while τ_s and τ_f are expressed in milliseconds. With different values of the cell parameters, different intrinsic behaviors can be achieved : quiescence (Q), almost an oscillator (A), endogenous oscillator (O), depolarization (D), hyperpolarization (H), and plateau (P), as shown in Fig.1. In bio-inspired locomotion, a pair of neurons with mutual inhibition can be used to generate rhythmic motion in extension and flexion. A bio-inspired model for locomotion is proposed in the next section.

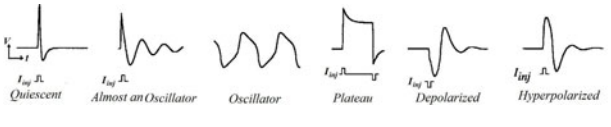


Fig. 1. The six intrinsic behaviors of the cell's model, Rowat and Selverston [14]

2.2 Locomotion Model

Studies of rhythmic movement in the animal show that local circuits in the spinal cord are able to control the timing and coordination of complex motion patterns [19]. The locomotion and rhythmic movements in mammals are organized by oscillatory spinal cord circuits called CPGs. Experimental studies show that the rhythmic patterns in cat limbs can be generated in the absence of descending control from higher centers and sensory feedback [3]. Each joint appears to have its own CPG, which can be coupled to the CPG of another joint in order to achieve complex movements such as walking, running, swimming, flying, etc. These CPGs controlling such behaviors in animals locomotion can be responsible of rhythmic movements in human locomotion [20]. Several schemes for the spinal CPG have been proposed to generate rhythmic movements: "half-center CPG" proposed by Brown [21], "half-center CPG" with more complex patterns of motorneuron activity introduced by Perret et al. [22] and "half-center CPG" with sensory input proposed by Orlovsky et al. [1]. One drawbacks of these models is the direct excitatory connection between the rhythm generator interneurons and motorneurons. Any change in the interneurons layer will affect simultaneously the motorneurons layer. A more complicated architecture is required to face the adaptation with the environment changes. Two and three levels CPGs with rhythm generation and pattern formation circuitry have been proposed by [2] and [23]. This model separates cycle timing and motoneurons activation. In order to achieve a rhythmic movement such as walking, the CPG model was implemented on a simulated biped robot using MATLAB software. Fig.2(a) shows the wiring diagram for one biped robot's joint. It can be separated into three layers: Rhythm Generation neurons (RG), Pattern Formation neurons (PF) and MotorNeurons (MN). Sensory feedback shapes the activity of these neurons. This paper focuses on the effect of descending control on the rhythm generators neurons in order to control the behavior of these neurons when external perturbation occurs during walking. In the analytical study, after observing the phase diagram of a joint and changing σ_s and σ_f in the rhythm generators neurons, different motion behaviors were observed on the joint. Fig.2(b) shows the distribution of motion patterns in space of σ_s and σ_f. Varying σ_s and σ_f in RG of a joint will change its motion pattern. The four detected basic motion patterns can lead the robot to achieve some complex tasks like walking, running, and jumping depending on synaptic circuits between joint CPGs.

2.3 Control Architecture for a Biped Robot

Previously, the basic motion patterns obtained for one joint was shown. To achieve a complex movement like walking, synchronization between joints is needed. The complex patterns like walking and running are always composed of synchronized basic patterns. The synchronization between patterns is ensured by coupling the CPGs for the joints. Fig.3 shows the proposed coupling circuits between the rhythm generator neurons for the hip, the knee, and the ankle joints of a simulated biped robot. Each joint is driven by a simulated servo motor. With

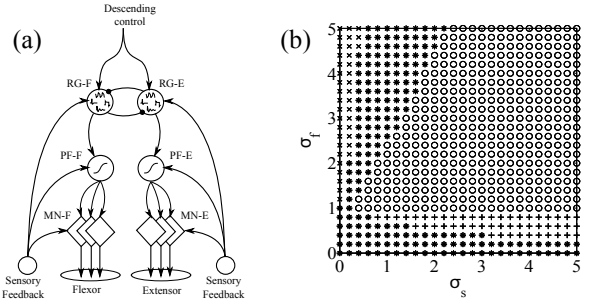


Fig. 2. Model of one joint controller and its motion patterns. (a)The model's scheme, CPG with three levels: Rhythm Generator, Pattern Formation, and MotorNeuron level. (b)The different behaviours observed on the joint for the same injected current. (x): Plateau , (*): Quiescence; (+): Almost an oscillator, and (o): Oscillatory behavior.

such simple coupling, the robot can carry out walking task from basic oscillatory patterns. With different coupling circuits, another task can be achieved. In some complex circuits, the robot can walk with different gaits. A desired task can be accomplished by defining basic patterns and special coupling circuit. The principle of our proposed circuit for walking (see fig.3) is described by the activity between the CPGs which is regulated by excitatory synaptic connections. For inter-limb circuitry, rhythm generator neuron extensor in the left hip (RG-E-hipL) excites rhythm generator neuron flexor in the right hip (RG-F-hipR). Rhythm generator neuron flexor in the left hip (RG-F-hipL) excites rhythm generator neuron extensor in the right hip (RG-E-hipR). The same synaptic excitation is proposed from the right hip to the left hip. For one leg, rhythm generator extensor neuron in the hip (RG-E-hip) excites rhythm generator extensor neuron in the knee (RG-E-knee) and rhythm generator extensor neuron in the ankle (RG-E-ankle) of the same leg. Rhythm generator flexor neuron in the hip joint (RG-F-hip) excites rhythm generator flexor neuron in the knee one (RG-F-knee) and rhythm generator flexor neuron in the ankle joint (RG-F-ankle) of the same leg. As described before, the locomotion is the interaction between CPG, sensory feedback, and descending control. Sensory information

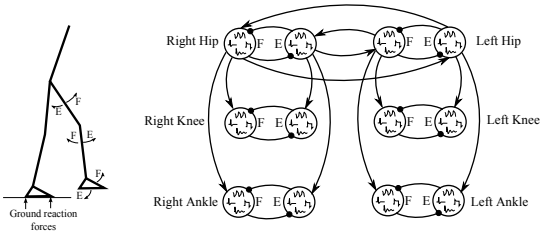


Fig. 3. Planar Biped model and proposed coupling circuitry between rhythm generator neurons for all joints. E and F are extension and flexion.

is used to shape the motion and manage some perturbations and balance control [24]. Thanks to the interaction with sensory feedback, the robot can walk without a perfect knowledge of its dynamics. A static model of sensory neuron proposed by Ekberg [25] is described in eq.(2). ρ_i is the activity of sensory neuron, α is a positive constant that denotes the dynamics of the neuron, θ is the amplitude and ϕ is the input on the neuron. ϕ can be an angular position, or a contact force [26].

$$\rho_i = (1 + e^{\alpha(\theta-\phi)})^{-1} \tag{2}$$

The extension and flexion sensory neurons in each joint inhibit the corresponding motorneuron for this joint. This circuitry is referred as articular reflex. Equilibrium control is achieved by the difference between the center of pressure and the projection of the center of mass. In our model, the parameter of equilibrium used as input of two neurons: falling forward and falling backward neurons. The activity of both neurons is injected in pattern formation layer at the ankle CPG. If the robot may fall forward, the corresponding neuron becomes active to excite the pattern formation neuron extensor for the ankle of stance leg. The flexor pattern formation neuron will be excited if the falling backward neuron becomes active. Once the control architecture was proposed and the model of rhythmic neurons is determined, it is time to show how the simulated biped is learning to walk on a flat terrain. As the desired task is the walking and the coupling circuit is already defined, the biped will learn basic patterns, in space of σ_s and σ_f, that lead to successful walking.

3 Success and Failure Learning

The objectives of the learning mechanism is to detect in the space of σ_s and σ_f the basic patterns which lead to successful walk. Our previous work in experience-based learning mechanism with the vigilance concept has been used here to detect successful and failure walking patterns, see [27] for more details. Walking trial occurs inside a time window of ten seconds. Successful walking is defined when the simulated biped did not fall during the time window and achieved two steps at least.

This mechanism is composed of two phases: evaluation and decision, see Fig.4. In the evaluation phase, two independent neural networks based on well-known

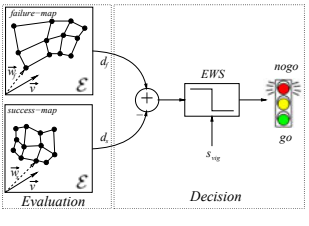


Fig. 4. Learning mechanism with evaluation and decision phases

Self Organizing Maps, proposed by Kohonen [28], are used to represent the knowledge in success and in failure. Success map learns in case of success trials, and failure map learns in case of failure ones. During learning, the two maps will be self-organized in the space of parameters that will be therefore divided into three zones: a zone of success represented by success map, a zone of failure represented by failure map, and a zone of conflict that corresponds to the interference between the two maps. The evaluation of any vector $\overrightarrow{v}$ from space $\mathcal{E}$ belonging to success or failure is defined by the distance between $\overrightarrow{v}$ and each map. The distance of a vector with a map is the distance between this vector and the closest neuron in the state space (the winner neuron). For each $\overrightarrow{v}$, two distances therefore exist: one to success map called d_s, and another to failure map called d_f. In the decision phase, the comparison between the distance with success map d_s and the one with failure map d_f leads to an expected result in the case where the vector $\overrightarrow{v}$ is applied on the controller (trial). According to expected result, if it may lead to failure, then an Early Warning Signal (EWS) becomes active to avoid the trial, and the decision will be "nogo". When EWS is inactive, the decision called "go"is taken. The decision mechanism is affected by the threshold of vigilance s_{vig}, which represents the tolerance to risk. The vigilance is related to human learning approaches and decision making [29].

In order to increase the reflectivity of the vigilance threshold model proposed in our previous work [27], a modulation of the above mentioned threshold s_{vig} is introduced. This lead to get different values of it for each trial. Hence, this model increases the learning mechanism efficiency by extending the learning process to sectors of space of parameters. As an important issue, the risk behavior will change from prudence at the beginning of learning to adventure at the end. An example of vigilance threshold modulation is given as following (see Fig.5(c)):

$$y_1 \leq s_{vig} \leq y_2 \qquad \begin{cases} y_1 = a_1 - b_1 * log((x + c_1)^2) \\ y_2 = a_2 - b_2 * log((x + c_2)^2) \end{cases} \tag{3}$$

The coefficients values are $(a_1 = 0.9, a_2 = 1.47, b_1 = b_2 = 0.15, c_1 = c_2 = 20)$ and were chosen after several attempts. y_1 and y_2 chosen curves ensure smooth change between the prudence and adventure above mentioned behaviors. Walking patterns are presented by success map and falling patterns are presented by failure one. With such learning mechanism, learning failure map is as important as learning success map, since falling patterns stored in failure map can be used in an adaptation approach where walking patterns are limited (ex: in case of external perturbation). Fig.5 shows success and failure maps after learning 200 trials based on the new model of the vigilance threshold. The state space is normalized between 0 and 1 and each map has 25 neurons. Weights of neuron (w_1,w_2) denote the parameters of the rhythmic neuron ($w_1 = \sigma_s$, $w_2 = \sigma_f$). Therefore, there are 25 different configurations in each map that match 25 successful walking gaits stored in success map, and 25 unsuccessful walking patterns stored in failure map. Because of the topological properties of the Self Organizing Maps, three neurons in failure map are situated in the success zone and show oscillatory behaviors ((0.39, 0.57),(0.46, 0.33),(0.17, 0.23)), see Fig.5(a). As

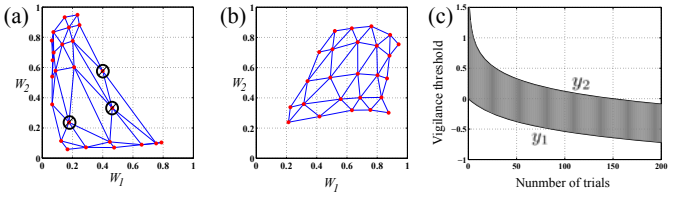


Fig. 5. Success and failure maps after learning walk on flat terrain. (a)Failure map after learning unsuccessful walking patterns. Three neurons was eliminated from the map, because they did not represent any input vector. (b)Success map after learning walking patterns. (c)New vigilance Model related to learning iterations, $y_1 \leq s_{vig} \leq y_2$. The risk behavior will change from prudence at the beginning of learning to adventure at the end.

these neurons did not represent any failure pattern, they are eliminated from the failure map.

4 Adaptive Behavior for Perturbation

As shown in the previous section, the walking task was achieved in the success map zone for the proposed coupling circuits. Because of the synaptic connection between rhythmic generator neurons for all joints, patterns cannot be independent. Then, the same pattern in all joints exist whenever the coupling circuitry is active. To have different patterns on different joints at a time, the synaptic connection between the CPGs must be inhibited. By having independent patterns in the hip, the knee and the ankle joints, the biped can achieve some complex behaviors. In this section, how the robot reacts to an external perturbation force is detailed.

As switching between success map neurons during walking will change the walking pattern and thus walking gait, it can also be interesting to switch between these neurons against external perturbation. The limitation of this algorithm will appear for a large perturbation force. This can be solved by switching toward failure patterns stored in failure map neurons. Inhibit the synaptic connection between CPGs is necessary to get different patterns in different joints.

The space of parameters in such case will be augmented, with a pair (σ_f, σ_s) for each joint. It increases from 2 dimensions in case of existing of coupling circuitry to 12 dimensions in case of independent patterns. To reduce dimensionality, we propose to represent all the patterns of a joint in one axis only. This will reduce the dimension by two and facilitates classification and visualization of high-dimensional data. To do so, a metric $\mathcal{E}$ which reflects the kinetic energy of one limb is introduced (eq. 4). Based on this metric, an energy based classification of the patterns can be carried out.

$$\mathcal{E} = \int_{t_0}^{t_f} \dot{\theta}^2 \, dt \tag{4}$$

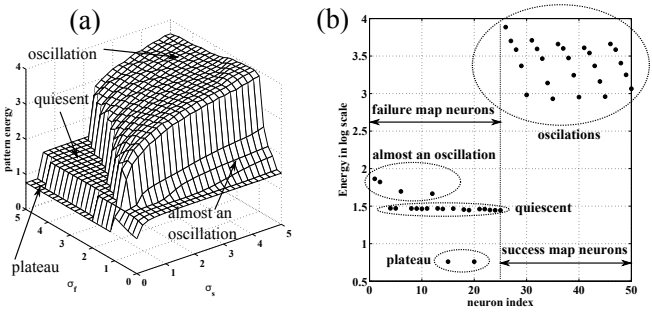


Fig. 6. The energy based metric patterns for the space of σ_s and σ_f (a), and for success and failure neurons represented on the horizontal axis (b). Neurons of success map represent oscillatory patterns with different frequency. Each neuron represent a pattern, but neurons are separated into four classes of patterns according to the energy based metric.

Fig.6(a) shows the logarithmic scale of the energy based metric for all the motion patterns of Fig.2. Fig.6(b) shows the logarithmic scale of the energy based metric of all neurons of failure and success maps given in Fig.5. First 25 neurons belong to failure map, and last 25 neurons belong to success map. The different behaviors are separated according to the energy based metric of motion patterns. Two neurons with Plateau have the lower values for the energy based metric, then 16 neurons with Quiescent behaviors, then four neurons with Almost an oscillator, then all the neurons of success map according to the Oscillation frequency. Patterns can be classified on a new axis according to the logarithmic scale of the energy based metric. As shown in Fig.6(b) patterns can be positioned on this axis in the following order: Plateau, Quiescent, Almost an oscillator, and Oscillatory patterns from low to high oscillation frequency. All neurons in success and failure maps can be placed on the new axis according to their rhythm. Therefore, two dimensional space (σ_s, σ_f) can be represented in only one dimension axis. One axis is obviously needed for each joint. In the first step of the study, only synapses between CPGs of the hip and the knee joints are inhibited. While the connection between CPGs of the ankle and the hip joints are kept. Fig.7 shows two dimensional space of patterns for the hip and the knee joints. Walking zone in Fig.7(b) corresponds to oscillatory patterns in the hip and the knee joints. In case of external perturbation force, pattern manipulation is necessary to avoid falling. The figure shows the group of patterns in the hip and the knee joints by which the robot can react against the perturbation. An example for walking and reaction phases is shown in Fig.8. First, it presents the normal walking on a flat terrain without any perturbation. Next, it illustrates the fall because of external perturbation force of $45N$ applied on the back of the robot (the simulated robot mass is about 22 kg and the walking speed is almost $0.2m/s$). Fig.8(c) shows how the biped robot react correctly against the external force by adapting the behavior of the rhythm generators neurons.

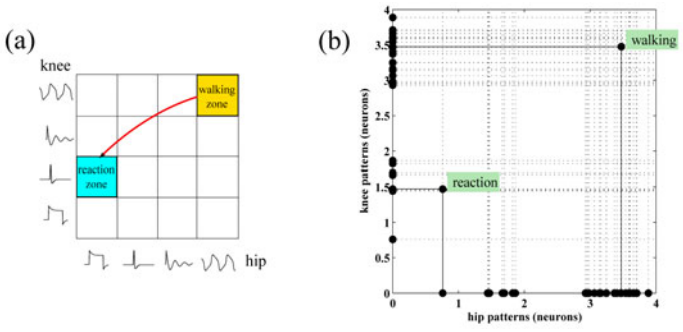


Fig. 7. The space of patterns is for hip and knee joints, with an example of switching against perturbation. (a)Patterns switch from walking by oscillatory patterns to quiescent pattern for knee and plateau for hip. (b)Neurons switch from walking zone to other neurons that represent quiescent pattern for knee and plateau for hip. Each neuron represents one pattern.

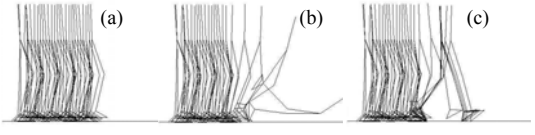


Fig. 8. Effects of adaptation mechanism on the biped to avoid falling. (a)Walking without perturbation.(b)Falling due to the perturbation. (c)Successful walking with adapation to the perturbation.

5 Conclusion

In this paper a neurobiological inspired controller for biped walking is presented. We showed how the behavior in rhythm generator neurons brings adaptation to face external perturbations. The switching between patterns was simplified by using a simple method to classify success and failure. Moreover, a technique for dimensionality reduction depending on the energy based metric patterns leads greater benefits, since the classification can be carried out over one axis only in relation to the motion patterns. Hence, establishing a space of patterns for the hip, and the knee joints. This space allows high-level control for goal directed action, thus, learning to achieve more complicated reactions. It also permits other rhythmic movement, where learning patterns replaces learning parameters. This was done by our experience based learning mechanism with this new model for vigilance threshold; we are able to explore in more efficient manner the space of parameters for new motion patterns. This mechanism was implemented on a simulated planar biped and allowed the robot to learn to walk and to react to perturbation without supervision. Our future work shall address goal directed action learning and adaptation to further changes in the environment, as well as changes in the physical parameters of the biped. This important issue will be

addressed in order to apply the proposed adaptation mechanism to a humanoid prototype under development.

References

1. Orlovsky, G.N., Deliagina, T., Grillner, S.: Neuronal control of locomotion: from mollusc to man, Anonymous. Oxford University Press, New York (1999)
2. McCrea, D.A., Rybak, I.A.: Organization of mammalian locomotor rhythm and pattern generation. Brain Research Reviews 57(1), 134–146 (2008)
3. Graham-Brown, T.: The intrinsic factors in the act of progression in the mammal. Proc. R. Soc. Lond. B. Biol. Sci. 84, 308–319 (1911)
4. Ijspeert, A.J.: Central pattern generators for locomotion control in animals and robots: a review. Neural Networks 21(4), 642–653 (2008)
5. Taga, G., Yamaguchi, Y., Shimizu, H.: Self-organized control of bipedal locomotion by neural oscillators. Biological Cybernetic. 65, 147–159 (1991)
6. Kimura, H., Akiyama, S., Sakurama, K.: Realization of dynamic walking and running of the quadruped using neural oscillator. Auton. Robots. 7, 247–258 (1999)
7. Endo, J., Morimoto, J., Matsubara, T., Nakanishi, J., Cheng, G.: Learning CPG-based Biped Locomotion with a Policy Gradient Method: Application to a Humanoid Robot. The International Journal of Robotics Research, Special Issue on Machine Learning in Robotics 27(2), 213–228 (2008)
8. Morimoto, J., Endo, J., Nakanishi, J., Cheng, G.: A Biologically Inspired Biped Locomotion Strategy for Humanoid Robots: Modulation of Sinusoidal Patterns by a Coupled Oscillator Model. IEEE Transaction on Robotics 24(1), 185–191 (2008)
9. Matsuoka, K.: Sustained oscillations generated by mutually inhibiting neurons with adaptation. Biological Cybernetics 52(6), 367–376 (1985)
10. McMillen, D.R., D'Eleuterio, G.M.T.: Simple Central Pattern Generator model using phasic analog neurons. Physical Review 59(6), 6994–6999 (1999)
11. Righetti, L., Buchli, J., Ijspeert, A.J.: Adaptive Frequency Oscillators and Applications. The Open Cybernetics and Systemics Journal 3, 64–69 (2009)
12. Righetti, L., Buchli, J., Ijspeert, A.J.: Dynamic hebbian learning in adaptive frequency oscillators. Physica D 216(2), 269–281 (2006)
13. Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., Kawato, M.: Learning from demonstration and adaptation of biped locomotion. Robotics and Autonomous Systems Journal 47(2-3), 79–91 (2004)
14. Rowat, P.F., Selverston, A.I.: Learning algorithms for oscillatory networks with gap junctions and membrane currents. Network 2(1), 17–41 (1991)
15. Ivanenko, Y.P., Cappellini, G., Dominici, N., Oppele, R.E., Lacquaniti, F.: Modular Control of Limb Movements during Human Locomotion. The Journal of Neuroscience 27(41), 11149–11161 (2007)
16. Marder, E., Calabrese, R.L.: Principles of rhythmic motor pattern generation. Physiol. Rev. 76, 687–717 (1996)
17. Kuo, A.D.: The relative roles of feedforward and feedback in the control of rhythmic movements. Motor. Control. 6, 129–145 (2002)
18. Kiehn, O., Butt, S.J.: Physiological, anatomical and genetic identification of CPG neurons in the developing mammalian spinal cord. Prog. Neurobiol. 70, 347–361 (2003)
19. Williams, S.M., Purves, D.: Neuroscience, 2nd edn., January 15. Sinauer Associates (2001)

20. Taga, G.: Adaptive Motion of Animals and Machines. In: Nonlinear Dynamics of Human Locomotion: from Real-Time Adaptation to Development, pp. 189–204. Springer, Tokyo (2006)
21. Graham-Brown, T.G.: On the fundamental activity of the nervous centres: together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system. J. Physiol. 48, 18–41 (1914)
22. Perret, C., Cabelguen, J.M., Orsal, D.: Analysis of the pattern of activity in knee flexor motoneurons during locomotion in the cat. In: Gurfinkle, V.S., Ioffe, M.E., Massion, J., Roll, J.P. (eds.) Stance and Motion: Facts and Concepts, pp. 133–141. Plenum Press, New York (1988)
23. Koshland, G.F., Smith, J.L.: Mutable and immutable features of paw-shake responses after hindlimb deafferentation in the cat. J. Neurophysiol. 62, 162–173 (1989)
24. Taga, G.: A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance. Biological Cybernetics 78(1), 9–17 (1998)
25. Wadden, T., Ekeberg, O.: A neuro-mechanical model of legged locomotion: single leg control. Biological Cybernetics 79(2), 161–173 (1998)
26. Geng, T., Porr, B., Wörgötter, F.: Fast Biped Walking with a Sensor-driven Neuronal Controller and Real-time Online Learning. The International Journal of Robotics Research 25(3), 243–259 (2006)
27. Nassour, J., Hénaff, P., Ben Ouezdou, F., Cheng, G.: Experience-based learning mechanism for neural controller adaptation: Application to walking biped robots. In: The IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, pp. 2616–2621 (2009)
28. Kohonen, T.: Self-Organizing Maps, 3rd Extended edn. Springer Series in Information Sciences, 501 pages. Springer, Heidelberg (1995/1997/2001), ISBN 3-540-67921-9, ISSN 0720-678X
29. Ahn, H., Picard, R.: Affective-Cognitive Learning and Decision Making: A Motivational Reward Framework For Affective Agent. In: The 1st International Conference on Affective Computing and Intelligent Interaction, Beijing, China, pp. 22–24 (2005)

Predicting Affordances from Gist

Pedro Santana[1], Cristina Santos[2], David Chaínho[3], Luís Correia[1], and José Barata[3]

[1] LabMAg, University of Lisbon, Portugal
[2] Industrial Electronics Department, University of Minho, Portugal
[3] UNINOVA, New University of Lisbon, Portugal

Abstract. This paper presents an incremental learning mechanism to create associations between the affordances provided by the environment and its gist. The proposed model aims at helping the agent on the prioritisation of its perceptual resources, and consequently on visual attention. The focus on affordances, rather than on objects, enables a self-supervised learning mechanism without assuming the existence of symbolic object representations, thus facilitating its integration on a developmental framework. The focus on affordances also contributes to our understanding on the role of sensorimotor coordination on the organisation of adaptive behaviour. Promising results are obtained with a physical experiment on a natural environment, where a camera was handled as if it was being carried by an actual robot performing obstacle avoidance, trail following and wandering behaviours.

1 Introduction

Environmental context is known to modulate several aspects of the animal's behaviour, such as its locomotion [1]. The importance of context to the animal's survival is so strong that, in the case of humans, there are situations where it is not even possible to consciously suppress its effects altogether [2]. Robustness and parsimony in visual search is also known to be strongly correlated with contextual cues [3], in line with active vision research [4]. Computational models in this case focus learning the statistics describing objects and typical scenes co-occurrence [5]. The acquisition of this knowledge, according to these models, is thus based on the existence of a mechanism capable of determining whether a given object is present in the scene, which is ultimately used to supervise the learning process. However, a global isomorphic representation of the object [6], is unlikely to exist in an embodied agent whose autonomous development occurs bottom-up, in interaction with the environment. Conversely, representations are distributed and purpose-oriented [7], thus making a signal to supervise the learning process hard to define.

Under the embodied cognition framework [8], perception can only be understood in terms of behaviour, and thus body, nervous system and environment must be seen in an holistic way [9,10,11]. As a consequence, sensorimotor coordination plays a key role on adaptive behaviour [12,8,13], and in particular to shape sensory information so as to facilitate perception [14]. In fact, representations may well be defined themselves in terms of sensorimotor dynamical states [15,16,17,18]. This further complicates the definition of a well localised and steady-state signal to supervise the learning process.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 325–334, 2010.

Focused on embodied agents, and not on general purpose vision systems, this paper solves the learning supervision problem by using context to predict affordances [19,20], rather than objects. A by-product of not assuming explicit object representations is the ability to operate even when the behaviours composing the control structure are yet not fully matured.

Nevertheless, the model starts by assuming that the agent is already capable of exploiting the environment's affordances. That is, in the presence of a given object, i.e. the aggregate of a given set of perceptual features, the agent knows which behaviour from its repertoire is better applied to it. An example is the *follow* behaviour, which can be effectively applied in the presence of a *trail*. Hence, in the case of a trail, the affordance is *to be followable*. This object-centred knowledge can be evolved [21] or learnt [22,23] by having the agent exploring its behavioural repertoire in encountered objects. The model is thus operating on a more advanced developmental stage, exploiting the knowledge obtained so far.

The model's next assumption is that the learnt affordances are used to trigger the corresponding behaviours according to a layered behavioural hierarchy [24]. The winning behaviour at each moment is then associated to the current visual context and stored in the agent's short-term associative memory. Latter on, this memory can be consulted to predict which behaviours are the most appropriate given the visual context at the recalling moment, and by consequence, which affordances are more likely to be present in the environment. Given the likelihood of a given affordance to occur, the agent should be capable of parsimoniously, and in a context-dependent way, determining how much relevant is to search for a given object, and consequently, how much perceptual resources must be allocated to it.

In this paper visual context is captured through the gist of the scene. Being a global descriptor, gist is highly fast and robust to local environment variations [3]. This is particularly interesting as it enables the agent to exploit contextual cues robustly and parsimoniously. In addition to reduce sensitivity to varying agent's posture changes, where the scene is observed from different perspectives, the gist provides highly generalisable contextual cues, and so enabling their reuse in new environments.

The use of gist in the robotics domain has been mostly limited to learning of places [25] and scene categories [26] for localisation and mapping purposes. In these works, learning is done off-line and supervised by an external signal (e.g. a symbolic label of the scene). Conversely, our model operates fully online and learning is self-supervised, i.e. the teaching signal is provided directly from the behavioural repertoire of the robot.

The paper is organised as follows. Section 2 presents the proposed model and describes each of the model building blocks. In Section 3, the results are shown and discussed, depicting the system's prediction and generalisation capabilities. The article ends by discussing the obtained results and future issues.

2 Model Overview

Fig. 1 illustrates the main building blocks composing the proposed model. As mentioned, the model assumes a bottom layer where a behaviour-based architecture [24] is responsible for the selection of the affordance to be exploited at each moment, i.e. the

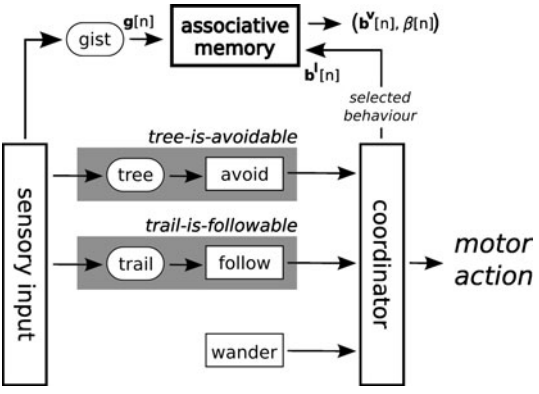


Fig. 1. Model's Building Blocks. Ovals correspond to object percepts, whose labels are simple descriptions and not symbolic representations. The coordinator basically selects which affordance is exploited at each moment, i.e. whose corresponding behaviour is activated. Gray shadows represent the system's built-in affordances, i.e. the link between a given object and an agent's behaviour.

behaviour having access to the agent's actuators. The selection among the q possible behaviours is done by a coordinator node, which arbitrates according to a set of fixed priorities. The output of the coordinator node is a binary q-dimensional behaviour selection vector, $\mathbf{b}^{\mathbf{l}}[n]$, whose non-zero element corresponds to the selected behaviour at frame n. This behaviour is selected for actual control of the agent.

On the top of this behavioural architecture, an associative memory grows incrementally so as to learn the mapping between the behaviour selection vector, $\mathbf{b}^{\mathbf{l}}[n]$, and the current visual context, given by the scene's gist, $\mathbf{g}[n]$. The associative memory can be queried at any time for the most likely behavioural selection vector, $\mathbf{b}^{\mathbf{v}}[n]$, given the gist of the current scene, $\mathbf{g}[n]$. Given the global nature of the gist, this prediction is quite often affected by environmental information located in the agent's far field-of-view. This makes the prediction highly useful to modulate the agent's behaviour. One possible exploitation of the associative memory is that, given the current visual context, a behaviour which it is known to be likely to become active could be predisposed. Another possibility, is the allocation of perceptual resources to the detection of this behaviour's associated affordance. With particular utility for the behavioural modulation aspect, a confidence level on the prediction, $\beta[n]$, is also provided. This enables to access whether predictions are likely to be accurate, and should consequently be considered for the action selection modulation.

For the purpose of the current study, a set of two affordances are considered, namely *to be avoidable* and *to be followable*. An example of an object category affording *to be avoidable* is *tree* and of an object category affording *to be followable* is *trail*. A found trail is thus assumed to be followed by the agent. Avoidable objects are more relevant for the agent's survival and so their presence right in front of the agent subsume the other affordance. In the absence of any of these in the environment, the agents starts wandering. When wandering, the presence of any avoidable object, be it on the front of the agent or not, will activate the avoidance behaviour.

2.1 Gist Calculation

In this study, the environmental context is defined in terms of the scene's visual gist. Although more complex and accurate methods exist [5,25,26], in this study the gist of the scene is represented by a simple and fast to compute histogram over the whole agent's visual input. Being a global descriptor, the gist is not sensitive to local variations on the environment. This in turn results in good generalisation capabilities in categorising the scene, which as experimental results will show, help the agent when facing new environments.

Concretely, the gist descriptor, $\mathbf{g}[n]$, is a three dimensional histogram obtained from the whole image in the HSV colour space. To reduce sensitivity to illumination effects, the saturation (S) and value (V) channels are represented by only 4 bins, whereas the hue (H) is represented by 16 bins. This descriptor is consequently a vector of 256 numerals whose combined values are representative of a given type of environment, such as *forested*. Note that no label is associated to the descriptor. As it will be shown, the learning process just associates this non-symbolic descriptor to behaviour selections taken by the agent.

2.2 Incremental Learning

Once the gist is computed, it can be associated to the selected behaviour being engaged by the agent. This association can then be exploited to know which behaviour should be acted given the current gist, or in other words, which affordance that is more likely to be found in the environment should be attended first.

Most gist-related research focuses on off-line learning [5,25,26], which is not adequate for a truly autonomous agent. In this study, the learning procedure follows the lazy learning paradigm, where the training examples are stored until they are necessary, i.e. when recalling is taking place. The biggest advantage of lazy learning is the possibility of locally approximating the learnt function according to the stored training examples. In the limit, a single example is necessary to generate a classification. This enables one-shot learning and consequently fast adaptation. In turn, the biggest disadvantage of lazy learning is the large memory requirements to store all training examples. However, as sensory flow in an embodied agent is highly correlated in time, a large redundancy is observed.

We exploit the existing redundancy on the sensory flow by creating segments of sequential frames, whose first element's gist is similar to the gist of the remaining ones. That is, a segment is created by accumulating frames until the gist descriptor of the current frame is too dissimilar from the one of the first segment's frame, or until an upper bound of η frames is reached. Two gist vectors are assumed to be dissimilar if the Chi-Square distance between them is above δ. To reduce sensitivity to noise, a newly created segment is rejected from further processing if represented by less than ζ frames.

The average gist, $\mathbf{s}(m) = \sum_{j=a(m)}^{n}(\mathbf{g}[j]/(n - a(m)))$, of the newly created segment m, is associated to the histogram of behavioural selections occurred during the segment's composing frames, $\mathbf{h}(m) = \sum_{j=a(m)}^{n} \mathbf{b^1}[j]$, where $a(m)$ is the index of the segment's first frame and n the index of the current and consequently segment's last frame. The tuple $\langle \mathbf{s}(m), \mathbf{h}(m) \rangle$ is introduced to the associative memory M as follows.

If the average gist of the new segment, $\mathbf{s}(m)$, is significantly similar to the most similar segment already present in the associative memory, $\mathcal{X}^2(\mathbf{s}(m), \mathbf{s}(o)) < \rho$ with $o = \arg\min_{b \in M}(\mathcal{X}^2(\mathbf{s}(m), \mathbf{s}(b)))$, then both are blended, where $\mathcal{X}^2(\cdot)$ is the Chi-Square distance. Otherwise m is simply appended to the memory. Merging occurs by averaging both gist descriptors, $\left(\frac{n_m}{n_m+n_o}\mathbf{s}(m) + \frac{n_o}{n_m+n_o}\mathbf{s}(o)\right)$, weighted by their number of supporting frames, n_m and n_o. The behaviour selection histograms are also blended via a simple summation, $(\mathbf{h}(m) + \mathbf{h}(o))$. The resulting merged segment is then compared to the second most similar segment to m, $p = \arg\min_{b \in M \setminus \{o\}}(\mathcal{X}^2(\mathbf{s}(m), \mathbf{s}(b)))$, and if the merging conditions are met (see above), both segments are merged. This two-step merging procedure is an attempt to avoid the associative memory from growing unbounded, without incurring in excessive processing.

2.3 Gist Classification

Every time a new image frame is obtained, the associative memory can be queried for the most likely behaviours given the gist descriptor, $\mathbf{g}[n]$, of the current scene. This is done according to an adaptation to the weighted k nearest-neighbour method, where $k = 4$ has shown to provide the best results for the tested data-set, i.e. trade-off between accuracy and generalisation capabilities.

In more detail, given the query $\mathbf{g}[n]$, the associative memory is searched for the closest k segments, which are said to compose the ordered set $K = \{m_0, \dots m_k\}$. The order is given by the Chi-Square distance to the query, at the gist descriptor level, i.e. $\mathcal{X}^2(m_i, \mathbf{g}[n]) > \mathcal{X}^2(m_j, \mathbf{g}[n]), \forall i > j$. The return to the query, i.e. the classification, is a normalised behaviour selection histogram resulting from the weighted sum of the behaviour selection histograms of the segments in K, $\mathbf{b^v}[n] = \sum_{l=0}^{k} \mathbf{h}(m_l) w(m_l)$. The weight of a segment $m_l \in K$ is as large as the Chi-Square distance to the query gist descriptor is small, and as high as it is its order in K, $w(m_l) = \frac{2 - \mathcal{X}^2(\mathbf{s}(m_l), \mathbf{g}[n])}{2l}$. The magnitude of the elements composing $\mathbf{b^v}[n]$ represent the likelihood of each behaviour to occur, given the current gist, and consequently the possibility of finding their associated affordances.

2.4 Gist Classification Confidence Level

Aside the estimate behaviour selection histogram $\mathbf{b^v}[n]$, the associative memory also returns a confidence level, $\beta[n]$, on the classification. $\beta[n]$ varies according to: 1) the confidence the system has on the visual context, given by ξ; 2) the discrepancy between the predicted and current behavioural context, $d(\mathbf{b^v}[n], \mathbf{b^l}[n])$, which controls the value of γ; and 3) the rate of variation of ξ, i.e. $\dot{\xi}$, in case it decreases. To account for these aspects, β is modelled as, $\dot{\beta} = (\xi - \beta) \cdot \alpha_1 + \left(\mathcal{H}(-\dot{\xi})\dot{\xi}\beta\right) \cdot \alpha_2 - (\gamma\beta) \cdot \alpha_3$, where $\alpha_1 + \alpha_2 + \alpha_3 = 1$, and $\mathcal{H}(\cdot)$ is the Heaviside step function. The use of dynamical systems to calculate β assures robustness in the final system.

The higher the number of frames supporting $\mathbf{b^v}[n]$, $f(\mathbf{b^v}[n])$, the more confident is the system on its prediction. This confidence, ξ, is given by $\xi = \mathcal{G}(f(b^v)[n])$, where $\mathcal{G}(x) = e^{-5e^{-\lambda x}}$ is the Gompertz function such that $\xi \in [0, 1]$. This function makes ξ

converge faster towards near 1 in face of reliable information from the associative memory, but more slowly in reaching the final value of 1, which would mean that confidence in the predicted behaviour selection vector is maximum.

The more discrepancies exist between the predicted and current behavioural contexts, the less confident the system is on the former, and β should approach zero. This discrepancy, $d(\mathbf{b}^{\mathbf{v}}[n], \mathbf{b}^{\mathbf{l}}[n])$, is defined in terms of the Euclidean distance between both vectors $\mathbf{b}^{\mathbf{v}}[n]$ and $\mathbf{b}^{\mathbf{l}}[n]$. The following dynamical system takes into account these discrepancies, $\dot{\gamma} = (1 - \gamma) \cdot d(\mathbf{b}^{\mathbf{v}}[n], \mathbf{b}^{\mathbf{l}}[n]) \cdot \kappa_1 - \kappa_2 \gamma$.

Discrepancies, $d(\mathbf{b}^{\mathbf{v}}, \mathbf{b}^{\mathbf{l}})$, are accumulated in γ at a rate of $k_1 \cdot (1 - \gamma)$. γ tends to zero in case there are no discrepancies, meaning one should increase the confidence in the visual context and β should approach γ, that is 1. Similarly, γ tends towards one in case discrepancies are maximum, meaning one should reduce the confidence in the prediction, and β should approach 0.

3 Experimental Results

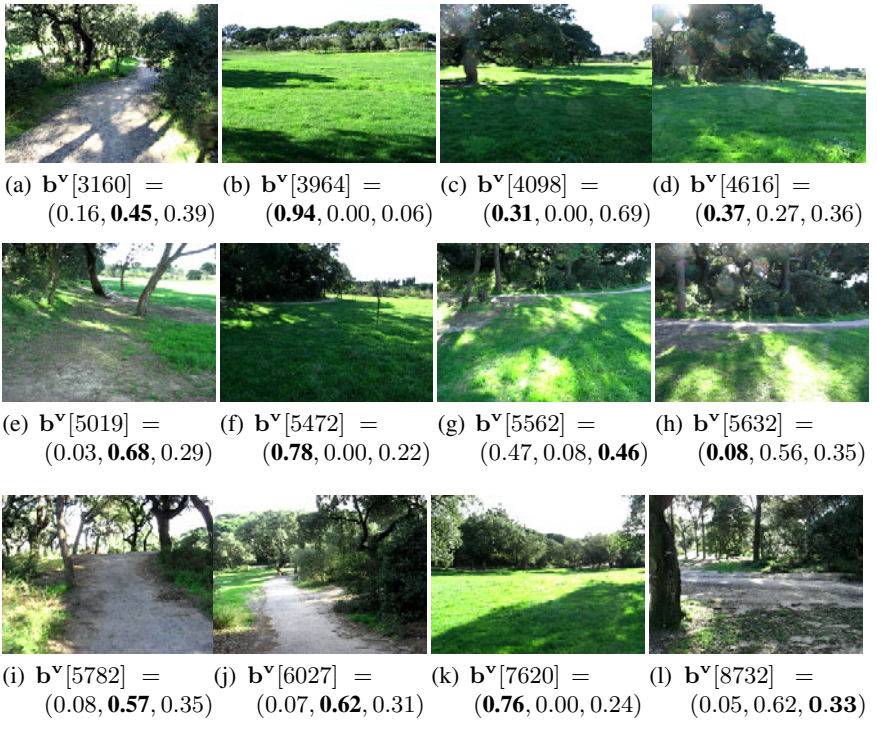
To validate the proposed model, an experiment was prepared, where the system was parametrised as follows, $\zeta = 20$, $\delta = 0.2$, $\lambda = 0.03$, $\alpha_1 = 0.6$, $\alpha_2 = 0.3$, $\alpha_3 = 0.1$, $k_1 = 0.8$, $k_2 = 0.2$, $\rho = 0.4$ and $\eta = 50$. The goal of the experiment is to demonstrate the ability of the associative memory to learn a generalisable gist-affordance mapping.

For the experiment, a video composed of 9000 frames was obtained by a person walking through a predefined course in a natural park (see Fig. 2) with a hand-held camera at the shoulder's height. The camera felt a considerable level of oscillations, typical in off-road robots, which result in sudden viewpoint changes and induced blur. The camera was moved as similar as possible as it would be if mounted on a mobile robot acting according to the behavioural hierarchy presented in Section 2. That is, when the person selected to follow a trail, the camera was pointed towards its vanishing point. Any obstacle faced by the person was circumnavigated, thus emulating the avoidance behaviour. In the absence of a trail and nearby obstacles, the person engaged on a wandering behaviour. The video was then hand-labelled with respect to which behaviour was being emulated by the person at each frame. That is, the signal that would be output by the behavioural architecture, $\mathbf{b}^{\mathbf{l}}[n]$, was manually defined according to the emulated behaviour. The system was then evaluated as if the video was being obtained on-line and $\mathbf{b}^{\mathbf{l}}[n]$ was being generated by the behavioural hierarchy.

The one-shot learning capability of the system can be appreciated at location `1` (see Fig. 2), i.e. soon after the onset of the first trail following. At this time, the associative memory was already able to recognise the scene as containing elements *to be followable* (see in Fig. 4). Another sign of this property is the fast stabilisation of the associative memory, roughly at half of the run, on an amount of 24 segments. This small quantity of segments shows that the model generates bounded/parsimonious representations of the environment. This is a demonstration of the model avoiding to over-fit the environment, which is in turn the cause for its good generalisation ability.

Fig. 3 illustrates a set of key frames of the run, where the system's prediction and generalisation capabilities are evident. An example of generalisation is the one depicted in Fig. 3(a), where the associative memory confirms the behavioural hierarchy in what

Fig. 2. Experimental environment. The line corresponds to the motion path, whose direction is cued by the arrows. Letters are key locations, whose associated frames are exhibited in Fig. 3.



(a) $\mathbf{b}^{\mathbf{v}}[3160] = (0.16, \mathbf{0.45}, 0.39)$ (b) $\mathbf{b}^{\mathbf{v}}[3964] = (\mathbf{0.94}, 0.00, 0.06)$ (c) $\mathbf{b}^{\mathbf{v}}[4098] = (\mathbf{0.31}, 0.00, 0.69)$ (d) $\mathbf{b}^{\mathbf{v}}[4616] = (\mathbf{0.37}, 0.27, 0.36)$

(e) $\mathbf{b}^{\mathbf{v}}[5019] = (0.03, \mathbf{0.68}, 0.29)$ (f) $\mathbf{b}^{\mathbf{v}}[5472] = (\mathbf{0.78}, 0.00, 0.22)$ (g) $\mathbf{b}^{\mathbf{v}}[5562] = (0.47, 0.08, \mathbf{0.46})$ (h) $\mathbf{b}^{\mathbf{v}}[5632] = (\mathbf{0.08}, 0.56, 0.35)$

(i) $\mathbf{b}^{\mathbf{v}}[5782] = (0.08, \mathbf{0.57}, 0.35)$ (j) $\mathbf{b}^{\mathbf{v}}[6027] = (0.07, \mathbf{0.62}, 0.31)$ (k) $\mathbf{b}^{\mathbf{v}}[7620] = (\mathbf{0.76}, 0.00, 0.24)$ (l) $\mathbf{b}^{\mathbf{v}}[8732] = (0.05, 0.62, \mathbf{0.33})$

Fig. 3. Key situations along the run (see Fig. 2 for their localisation). The first, second and third elements of the classification vector $\mathbf{b}^{\mathbf{v}}[n]$ correspond to the likelihood of occurrence of *wander*, *follow* and *avoid* behaviours, respectively. The bold element corresponds to the behaviour selected by the behavioural hierarchy, i.e. the non-zero element of $\mathbf{b}^{\mathbf{l}}[n]$, which also operates as the teaching signal.

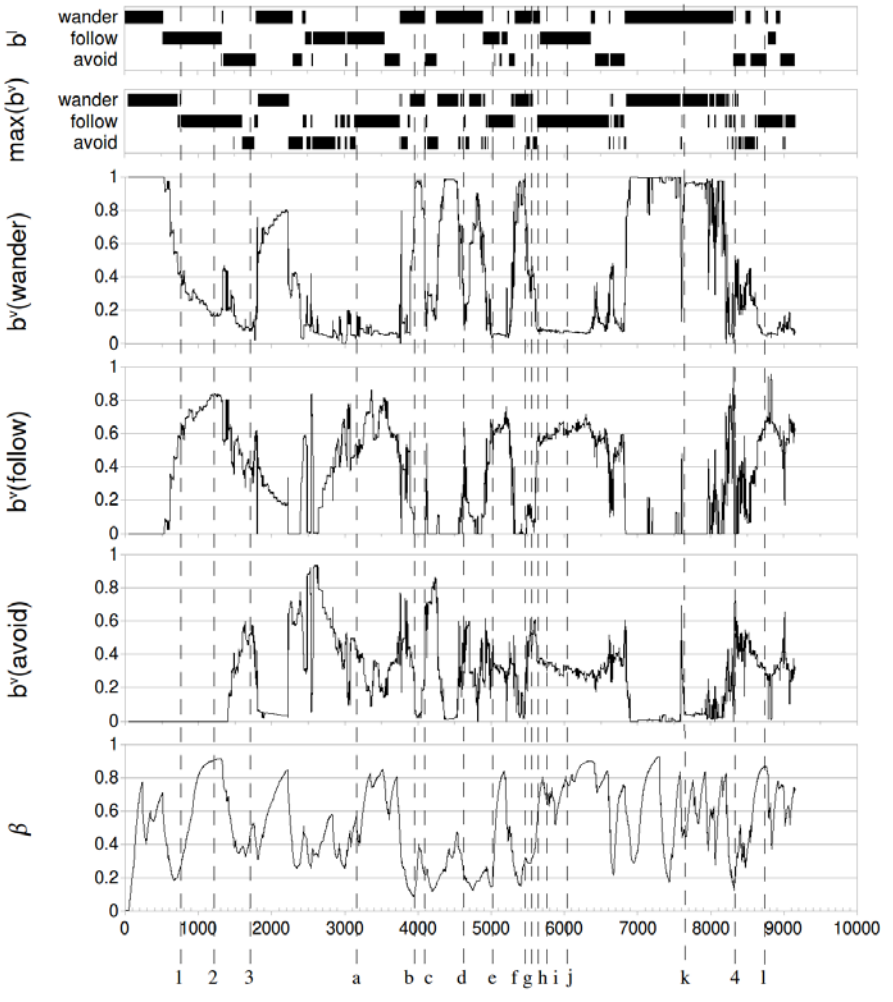


Fig. 4. Plots of experimental results. The black bars on the $\mathbf{b}^{\mathbf{l}}$ plot correspond to selection made by the behavioural hierarchy at each frame. The bars on the plot underneath, i.e. $\max(\mathbf{b}^{\mathbf{v}})$, refer to the behaviour whose activity is predicted as the strongest one. The three plots bellow are the predicted activity of each behaviour. Labels 1-4 and a-l indicate key locations.

regards the presence of the *to be followable* affordance, and further generalises it by also predicting the occurrence of the *to be avoidable* affordance. This generalisation is boosted by the similarity of the environment in question with the previously experienced one at location 3 (see Fig. 2), where the dense presence of trees took the behavioural hierarchy to select the *to be avoidable* as the affordance to be exploited (see Fig. 4).

The frame sequence from Fig. 3(f) to Fig. 3(i) illustrates a situation where the agent switches from wander behaviour to trail following. Interestingly, prior to the behavioural hierarchy selecting the *to be followable* affordance, the associative memory was already

predicting it (see Fig. 3(h)), as well as nearby trees (see Fig. 3(g)). We speculate that the system uses the presence of trees and shadows to predict the occurrence of trails.

The plot in Fig. 4 shows that when the prediction is stable and it matches the current behaviour selection vector, such as at location 2 (see Fig. 2), β is high. Conversely, when the prediction changes often in a short time and consequently mismatches the current behaviour selection vector, as at location 4, β decreases considerably. As a consequence, β shows to be a good indicator of how much certain are the predictions generated by the associative memory.

4 Conclusions and Future Work

An incremental learning mechanism used to create associations between the affordances provided by the environment and its gist was presented. With the proposed model, embodied agents are endowed with a mechanism to prioritise their perceptual resources on those aspects of the environment more likely to occur. The focus on affordances, rather than on objects, enables a self-supervised learning mechanism without assuming the existence of symbolic object representations. This is essential to enable context modulation on a developmental setup.

This paper is thus an additional account on the centrality of sensorimotor coordination, which is what affordances are about, in the understanding of adaptive behaviour. Although context-based visual attention might seem to be a perceptual problem, this paper supports the idea that higher levels of autonomy are more easily obtained if it is seen instead as a sensorimotor problem. This also shows that, for embodied agents, explicit isomorphic representations are less required than previously expected.

Further experiments are required to better understand the impacts of the model. First, the model should be tested on a real robot. Second, the actual benefits of modulating visual attention with the output of the proposed model must be thoroughly assessed. Third, the model should be tested on a developmental framework, where affordances are being discovered, exploited and refined, at the same time context is being taken into account. Environmental context is not limited to visual information. Future work will cover other sensory modalities, and temporal information inclusion.

References

1. McVea, D., Pearson, K.: Contextual learning and obstacle memory in the walking cat. Integrative and Comparative Biology (2007)
2. Reynolds, R., Bronstein, A.: The moving platform aftereffect: limited generalization of a locomotor adaptation. Journal of Neurophysiology 91(1), 92 (2004)
3. Oliva, A., Torralba, A.: The role of context in object recognition. Trends in Cognitive Sciences 11(12), 520–527 (2007)
4. Ballard, D.H.: Animate vision. Artificial Intelligence 48(1), 57–86 (1991)
5. Torralba, A., Murphy, K.P., Freeman, W.T., Rubin, M.A.: Context-based vision system for place and object recognition. In: Proc. of the IEEE ICCV, pp. 273–280 (2003)
6. Marr, D.: Vision: A computational investigation into the human representation and processing of visual information. Henry Holt and Co., Inc., New York (1982)

7. Goodale, M.A.: Action without perception in human vision. Cognitive Neuropsychology 25(7), 891–919 (2008)
8. Pfeifer, R., Scheier, C.: Understanding intelligence. MIT Press, Massachussetts (1999)
9. Ashby, w.R.: Design for a Brain. Chapman&Hall, London (1952)
10. Beer, R.D.: A dynamical systems perspective on agent-environment interaction. Artificial Intelligence 72(1-2), 173–215 (1995)
11. Thelen, E., Smith, L.B.: A dynamic systems approach to the development of cognition and action. The MIT Press, Cambridge (1996)
12. Ballard, D.H., Hayhoe, M.M., Pook, P.K., Rao, R.P.N.: Deictic codes for the embodiment of cognition. Behavioral and Brain Sciences 20, 723–767 (1997)
13. Mossio, M., Taraborelli, D.: Action-dependent perceptual invariants: From ecological to sensorimotor approaches. Consciousness and Cognition 17(4), 1324–1340 (2008)
14. Sporns, O., Lungarella, M.: Evolving coordinated behavior by maximizing information structure. In: Proc. of ALife X, pp. 3–7 (2006)
15. Scheier, C., Pfeifer, R., Kuniyoshi, Y.: Embedded neural networks: exploiting constraints. Neural Networks (11), 1551–1596 (1998)
16. Beer, R.D.: The dynamics of active categorical perception in an evolved model agent. Adaptive Behavior 11(4), 209–243 (2003)
17. Floreano, D., Toshifumi, K., Marocco, D., Sauser, E.: Coevolution of active vision and feature selection. Biological Cybernetics 90(3), 218–228 (2004)
18. Nolfi, S.: Categories Formation in Self-Organizing Embodied Agents. In: Handbook of Categorization in Cognitive Science, pp. 869–889. Elsevier, Amsterdam (2005)
19. Gibson, J.: The Ecological Approach to Visual Perception. Lawrence Erlbaum Ass., Mahwah (1979)
20. Chemero, A.: An outline of a theory of affordances. Ecological Psychology 15(2) (2003)
21. Slocum, A., Downey, D., Beer, R.: Further experiments in the evolution of minimally cognitive behavior: From perceiving affordances to selective attention. In: Proc. of the Intl. Conf. on Simulation of Adaptive Behavior (SAB), vol. 6, pp. 430–439 (2000)
22. Fritz, G., Paletta, L., Kumar, M., Dorffner, G., Breithaupt, R., Rome, E.: Visual learning of affordance based cues. In: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J.C.T., Marocco, D., Meyer, J.-A., Miglino, O., Parisi, D. (eds.) SAB 2006. LNCS (LNAI), vol. 4095, pp. 52–64. Springer, Heidelberg (2006)
23. Kim, D., Sun, J., Oh, S.M., Rehg, J.M., Bobick, A.: Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. In: Proc. of the IEEE ICRA, pp. 518–525 (2006)
24. Arkin, R.C.: Behavior-Based Robotics, May 1998. The MIT Press, Cambridge (May 1998)
25. Siagian, C., Itti, L.: Rapid biologically-inspired scene classification using features shared with visual attention. IEEE Transactions on PAMI, 300–312 (2007)
26. Collier, J., Ramirez-Serrano, A.: Environment classification for indoor/outdoor robotic mapping. In: Proc. of the IEEE Canadian Conf. on Computer and Robot Vision, pp. 276–283 (2009)

Analyzing Interactions between Cue-Guided and Place-Based Navigation with a Computational Model of Action Selection: Influence of Sensory Cues and Training

Laurent Dollé[1,⋆], Denis Sheynikhovich[1,2], Benoît Girard[1], Balázs Ujfalussy[3], Ricardo Chavarriaga[4], and Agnès Guillot[1]

[1] Institut des Systèmes Intelligents et de Robotique UPMC-Paris 6, CNRS UMR 7222, 4 place Jussieu, 75252 Paris cedex 05-France
[2] Laboratoire de Neurobiologie des Processus Adaptatifs UPMC-Paris 6, CNRS UMR 7102, 9 quai St. Bernard, 75005 Paris-France
[3] KFKI Research Institute for Particle and Nuclear Physics of the Hungarian Academy of Sciences, Konkoly-Thege Miklós út 29 – 33. Budapest, H-1121, Hungary
[4] CNBI, Center for Neuroprostethics, EPFL, Station 11 CH-1015 Lausanne, Switzerland
laurent.dolle@isir.upmc.fr

Abstract. The hypothesis of multiple memory systems involved in different learning of navigation strategies has gained strong arguments through biological experiments. However, it remains difficult for experimentalists to understand how these systems interact. We propose a new computational model of selection between parallel systems involving cue-guided and place-based navigation strategies that allows analyses of selection switches between both control systems, while providing information that is not directly accessible in experiments with animals. Contrary to existing models of navigation, its module of selection is adaptive and uses a criterion which allows the comparison of strategies having different learning processes. Moreover, the spatial representation used by the place-based strategy is based on a recent hippocampus model. We illustrate the ability of this navigation model to analyze animal behavior in experiments in which the availability of sensory cues, together with the amount of training, influence the competitive or cooperative nature of their interactions.

1 Introduction

Animal experiments demonstrate that parallel memory systems, assumed to support the learning of cue-guided and place-based navigation strategies, favour separate sets of sensory cues [1]. According to several studies, the first system, mediated by dorsolateral striatum (DLS), mostly uses proximal cues and the second system, mediated by hippocampus (Hc) and prefrontal cortex (PFC),

⋆ Corresponding author.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 335–346, 2010.

encodes configurations of distal cues [2]. The spatial representation built from these distal cues is often termed "cognitive map" [3]. Both systems are supposed to interact competitively or cooperatively, depending on the circumstances. Competition between the systems happens if the inactivation of one enhances the learning of the other, and cooperation if the learning of one compensates the lack of the other (e.g., [4]). Such interactions are influenced by both external and internal factors [5]. Although behavioral and neurophysiological studies provide valuable information about interactions between strategies and their potential biological substrates, the mechanisms underlying these interactions are not clear [6]. This is mainly due to the difficulty of knowing exactly which strategy is chosen at specific moments of the experiments.

We present here a computational model of navigation that provides direct information on rats' behavior in a Morris maze paradigm in which interactions between cue-guided and place-based strategies were shown to be influenced by the type of landmarks and the amount of training [7]. It supplies possible explanations of strategy selection mechanisms that can produce competition or cooperation and makes it possible to estimate the influence of sensory cues on strategy selection. The model is based on the assumptions (i) that both strategies are mediated by separate navigation "experts" that learn in parallel (as proposed by previous computational models [8,9,10]); (ii) that the selection mechanism continuously updates its estimation of the efficiency of both strategies (as in [8,9,10]); (iii) that the learning of both systems are of different nature: cue-guided strategies rely on a procedural "stimulus-response" learning implemented as a TD algorithm, while place-based strategies rely on a graph-search algorithm – not dependent on the reinforcement learning framework [10] – that is more flexible and faster to relocate a goal. Point (iii) constitutes the novelty of the model, which led to a major issue: finding the relevant "common currency" allowing to compare the efficiency of strategies having different learning processes. Another novelty is the integration of a recent hippocampus model [11] that computes the spatial representation used by the place-based strategy.

Section 2 describes the model of strategy selection; Section 3 the experimental protocol and the simulation procedure; Section 4 reports the results of computer simulations reproducing the animal data; Section 5 discusses the results in relation to other experimental and computational works and outlooks future work.

2 The Model of Strategy Selection

In our model "Taxon" and "Planning" experts represent DLS and Hc-PFC dependent memory systems, respectively. During navigation, they propose a direction for the next movement according to either visual input (Sensory Cells for the Taxon expert) or the estimated location (Planning graph, built from Place Cells, for the Planning expert). Sensory Cells are learned from the vision of the intra-maze landmark, Place Cells from the vision of the extra-maze landmarks. In addition, a third expert, Exploration, proposes a direction of movement randomly chosen between 0 and 2π (Fig. 1a). The movement actually performed by

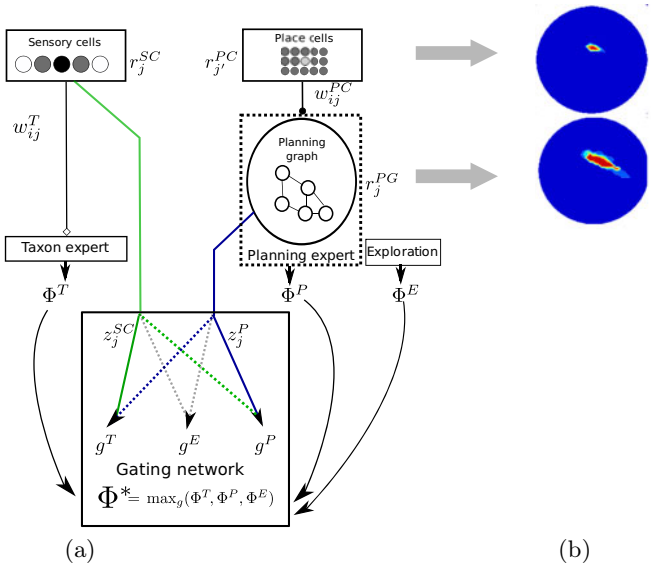


Fig. 1. (a) Model overview (see text for details). SC: Sensory Cells, PC: Place Cells, PG: Planning Graph, T: Taxon expert, P: Planning expert, E: Exploration Expert. Φ^* is the final direction of movement. The Gating network learns to choose the right expert, according to their performance. (b) Example place fields of Place Cells (top) and Graph nodes (bottom), the brighter the more the cell or node are active.

the animat is determined by a gating network which selects one of the experts to take control over behavior on the basis of previous performance (Fig. 1a). At each timestep, all the experts learn from the previously executed action, irrespective of which expert was responsible for it (Taxon, Planning or Exploration).

Taxon expert. For the purpose of the protocol reproduced below, we design here the Taxon expert as a "guidance" strategy – approaching a hidden target marked by a landmark located on a certain distance from it [7] – but other forms of Taxon (e.g., beacon approach) can be encoded. The visual input is encoded in a population of N_{SC}=100 Sensory Cells (SC) in which the activity of cell i signals presence or absence of the landmark in the direction ϕ^i from the animat. The motor response to the visual stimulus is encoded by $N_{AC} = 36$ Action Cells (AC), so that one action codes for a direction every $2\pi/36$. The learning is done by a Q-learning algorithm adaptation [12], so that the action space is continuous: in the update rule, instead of reinforcing the only action that is chosen, a Gaussian activity profile tuned around the selected action allows the closer actions to update their weights in the same direction. This activity profile is possible since all actions are direction movements, and therefore comparable. The activity in the whole AC population is interpreted as a population code for the direction ϕ^T of the next movement of the animat, proposed by the Taxon expert. Details of the computations are given in [9].

Planning expert. The Planning expert, inspired by the model of [13], first learns a topological representation of the environment in a reward-independent manner (*map building phase*) and then uses this representation to remember the goal location and to plan an optimal path towards it (*goal planning phase*). The *map building phase* is run during pretraining sessions. The map takes as input the activity of a population of Place Cells. This population is provided by the hippocampus model of [11] consisting of two different neural networks, simulating the enthorinal cortex (EC) and the dentate gyrus (DG). EC modelling is based on the recently discovered grid cells, which are receptive to specific spatial frequency and orientation, and therefore are good candidates for implementing path integration and other navigation strategies like, e.g., praxis. The grid cells are appended to a vector of 100 gray units representing the sight of the extra-maze landmarks. EC cells are then fed to the DG, which produces Place Cells by means of a Hebbian learning. Then a sparse representation is computed with a filter function that only keeps a few cells active (Fig. 1b, top) and sets the others to zero. Detailed computations can be found in [11]. The final activation is then normalized and processed by the Planning expert to build the nodes of the planning graph (PG) (Fig. 1b, bottom). For that, a pool of 100 nodes is connected to the DG at random synaptic weights, and the sparse learning used in the DG is replicated during the map building phase.

First the firing rate of a PG node j is computed as follows:

$$r_j^P = f_j(\sum_i W_{ij}^{(DG,PG)} DG_i \ , \ s^{PG}), \tag{1}$$

where W_{ij} is the synaptic weight linking the DG place cell i to the PG node j and $f_j(x, s^{PG})$ is the same non-linear function as in the DG, returning a sparse encoding of x, with a sparseness level of s^{PG}.

The synaptic weights are learned following a Hebbian rule, similar to the one used for learning the DG output:

$$\Delta W_{i,j}^{(DG,PG)} = \alpha^{(DG,PG)} r_j^P (DG_i - W_{i,j}^{(DG,PG)}) \tag{2}$$

A link between nodes N_i and N_j stores the allocentric direction of movement required to move from one node to the other. The *goal planning phase* begins when the goal position is found, the closest node being set to the delivered reward value. Then, given the PG, the optimal path to the goal is determined by the bio-inspired activation-diffusion mechanism [14] based on Dijkstra's algorithm for finding the shortest path between two nodes in a graph [15]. At each timestep, the Planning expert proposes the corresponding direction. If the goal position is not known, a random direction is proposed.

Strategy selection. Each expert computes at the same time its own proposition of movement. The time spent by the Planning expert to compute the path is not taken into account as a potential cost in our selection mechanism. The gating network selects at each timestep which of the Taxon, Planning or Exploration experts (T, P and E) will control the future movement, on the basis of

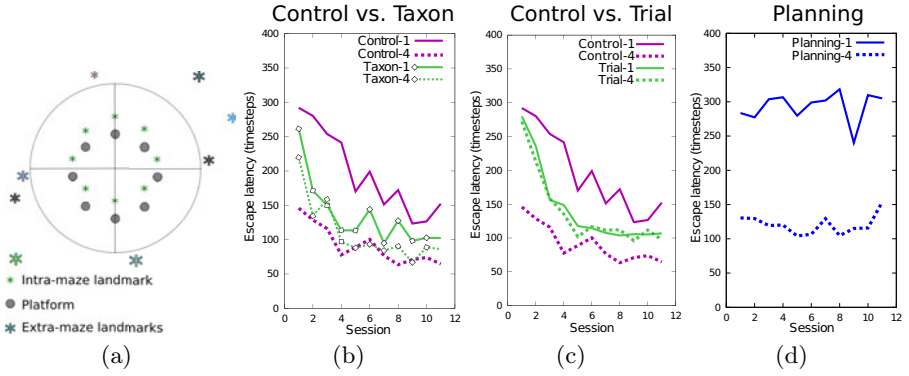


Fig. 2. (a) Simulated environment with one intra- and eight extra-maze landmarks. Simulated results of (b) Control vs Taxon groups. (c) Control vs Trial groups. (d) Planning group only, in the same protocol as the Control group.

candidate directions Φ^k of movement. It consists of three units $k \in \{T; P; E\}$, each corresponding to an expert. The activity g^k of expert k is its "gating value". They are computed as the weighted sum of both SC and PG inputs by synaptic weights z^k:

$$g^k(t) = \sum_{j=1}^{N_{SC}} z_j^k(t) r_j^{SC}(t) + \sum_{j=N_{SC}+1}^{N_{SC}+N_P} z_j^k(t) r_j^P(t), \tag{3}$$

where z_j^k is the connection weight between the unit k of the gating network and input unit j of the experts. A winner-take-all scheme then chooses the next movement direction $\phi^{k'}$:

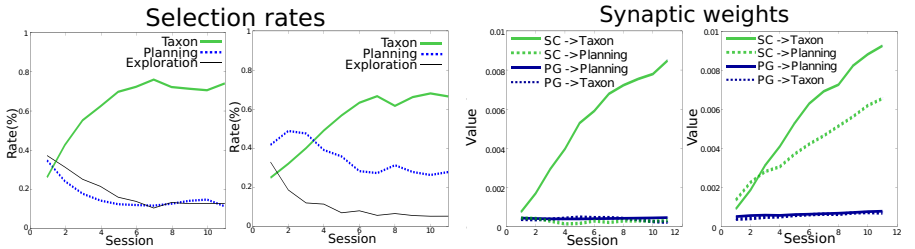
$$\phi^{k'}(t); k' = \text{argmax}_i(g^i(t)) \tag{4}$$

The gating values connection weights are adjusted using the same Q-learning algorithm as the Taxon expert, except that the update is modulated by the angular difference between the proposed orientation and the one actually chosen, so that the closer an orientation is from the chosen one, the stronger is its update.

We then evaluate different versions of the model – some corresponding to the simulation of lesioned animals – in the experimental paradigm described below.

3 Experimental Paradigm and Simulation Procedure

In the experiment of [7], two groups of rats – intact (Control) and Hc-lesioned – learned to find the location of a hidden platform in a Morris water maze surrounded by several room landmarks. The platform was cued by a visible landmark located in the pool at a certain distance northward to the platform. In a first experiment, for both Control and Hc groups, the platform and landmark were moved to one of eight predefined locations at the start of eleven sessions, where they stayed for four trials. In a second experiment, intact rats were tested



(a) Trial (b)Control (c) Trial (d)Control

Fig. 3. Trial selection rates of experts averaged over four trials (one session) for (a) Trial and (b) Control groups. Evolutions of synaptic weights in the gating network across sessions for (c) Trial and (d) Control groups show the relative influence of all sensory cues on the selection.

in the same apparatus, but the platform and landmark were moved from trial to trial rather than from session to session (Trial group). The same sensory cues were available for both tasks. Then the only difference was the possibility or the incapacity of training one or the other strategy system within each session.

Considering their results (Fig. 3 and 5 in their paper), the authors hypothesized (i) that, contrary to Hc group which could only learn and use a cue-guided strategy, Control and Trial groups would be able to acquire both strategies, thanks to the presence of cues inside and outside the pool; (ii) that all groups would be able to achieve the tasks (i.e., with one or both strategies available); (iii) that moving the platform from trial to trial would have the same effect on intact rats as a hippocampal lesion: rats of Trial group would indeed exclusively rely on a cue-guided strategy, as they would not be able to sufficiently train to refresh their cognitive map between two successive platform displacements. Our model will test each of these hypotheses by analyzing the interactions between these control systems.

The simulated water maze, rat, reward location and landmark are represented by circles of 200 cm, 15 cm, 10 cm, and 20 cm in diameter, respectively. The reward location is always 20 cm south from the landmark. Eight extra-maze landmarks are placed at different distance of the walls (30 cm to 50 cm), simulated by Gaussian visual stimuli of different widths (Fig. 2a). Three groups are simulated, corresponding to those tested in the actual experiment: the Control and Trial groups of intact rats are simulated with Taxon, Planning and Exploration experts; the Hc-lesioned group, henceforth called Taxon group, is simulated with Taxon and Exploration experts only. The same training protocol as in [7] is applied. We add a fourth group (Planning group, which could correspond to DLS-lesioned animals), not present in the original experiment, which is simulated with Planning and Exploration experts only, tested with the same protocol as the Control group. For each group, 100 sets of experiments were performed.

The performance of the Control, Trial and Taxon groups is statistically assessed by comparison of their mean escape latencies (number of timesteps to reach

the goal), within and across sessions, in the 1^{st} and 4^{th} trials of a session, using signed-rank Wilcoxon test for matched-paired samples. Between-group comparison is performed using a Mann-Whitney test for non matched-paired samples. Two measures quantize the animat's behavior: *goal occupancy rate* (number of times the animat visits a rewarded zone divided by the total trajectory length); *trial selection rate* of an expert (averaged number of times the expert is selected over the total length of the trajectory). The influence of sensory inputs on the selection of the strategies is assessed by comparing synaptic weights from the SC and from the PG nodes to the units of the gating network, which correspond to the relevance the strategies.

4 Results

Test of performance of separate experts. Separately trained on a simple task in which the animat had to escape to a visible and fixed platform, both experts differ in their learning processes. The Taxon expert learns slower than the Planning expert (mean of escape latencies from the 1^{st} to 4^{th} trials for Taxon= 235.04; for Planning = 160.82) but has a better performance when the training becomes intensive (mean of escape latencies from the 40^{th} to 44^{th} trials for Taxon= 21.08; for Planning = 83.18). These differences are due to the sparsity of the Planning Graph nodes, which allows this expert to quickly locate the new goal location (as encoded by existing nodes), but no further learning is done.

All groups are able to learn the tasks. As Fig. 2b and c attest, the model reproduces the main characteristics of the original experimental results. Indeed all groups achieve the tasks, as all escape latencies decrease across sessions ($p<0.001$). In accordance with [7], the cue-guided strategy is mainly responsible of this improvement: in Control and Trial groups, the Taxon selection rate dominates the others and increases throughout the sessions (Fig. 3a,b). In all gating networks, the evolution of weights between sensory cues and experts also reflects the growing influence of SC on Taxon expert (Fig. 3c,d).

No interaction between experts in Trial group. The similar performance of Taxon and Trial groups ($p = 0.08$; learning across, but not within sessions) suggests that the Taxon expert mainly controls the behavior of Trial animats (Fig. 2b, c). This is confirmed by Fig. 3a, showing the rapid and huge increase of the selection rate of this expert during the experiment, while the others remain very low. This is also illustrated by typical trajectories, showing that the Taxon expert mainly leads the animat near the platform (Fig. 4a, b). The results of the Taxon group indicate that its learning process is not flexible enough to quickly improve within session. Thus this explains why Trial group behaves similarly: the possibility of training or not during four trials does not change the performance.

The occupancy rates near the current platform (Fig 5a, 1^{st} and 4^{th} trials) are significantly lower in the Trial than in the Taxon groups. This may suggest a

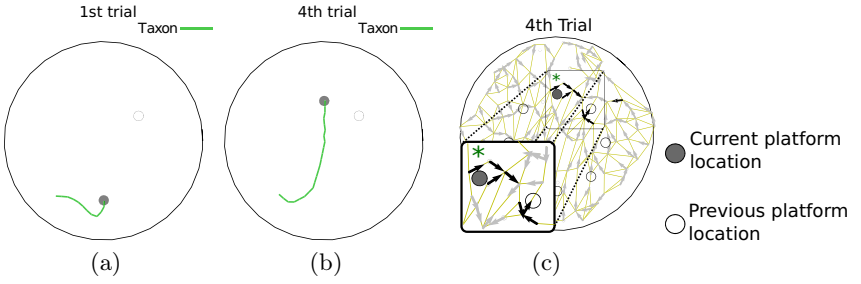


Fig. 4. (a-b) Typical trajectories of Trial animats, corresponding to beginning (a) and end (b) of session; (c) Navigational map of Planning expert in the Trial group (i.e., learned directions of movement for each spatial node) at the end of the same session. The zoom depicts that, during a new trial, the planning experts keeps the memory of the previous platform location.

negative influence of the Planning expert on Trial animats. Yet the evolution of weights in the gating network, with increasing links from SC to the Taxon expert only (Fig. 3c), minimizes the importance of this result: it clearly indicates that the Planning expert does not play any role in this task. Nevertheless, this expert continuously builds a navigational map in which it stores the successive platform locations (Fig. 4c), then the animat could use it if required by a new situation.

Training within session produces both competition and cooperation. As Fig. 2b shows, the possibility to train within session modifies significantly the behavior of Control animats, both at the beginning and at the end of sessions. Similarly to Control animals in the original experiment, they perform worse at the beginning and better at the end of one session than Trial and Taxon groups. The role of Taxon in Trial group proves that these differences should come from the Planning expert. Fig. 3b confirms that Control animats select it more often than Trial ones (especially at the beginning of the experiment). The fast learning of this expert – also favored by the parallel learning of the efficient Taxon expert (see the performance of Planning alone, Fig. 2d) – indeed allows the Control group to quickly memorize the position of the platform in the cognitive map. During each session, both experts cooperate and enhance the overall performance: the escape latencies are lower (Fig. 2b) and the occupancy rates near the current location higher (Fig. 5a, 4^{th} trials) than Trial group.

The gating network reflects this synergistic interaction by increasing the weighting of SC to both Taxon and Planning experts (Fig. 3d). The nature of their cooperation could be deduced from the navigational maps of both experts at the end of a session showing that the Taxon expert drives the animat southward the landmark (Fig. 6a) whereas the Planning expert leads it to the platform location (Fig. 6b). A typical trajectory illustrates this hypothesis (Fig. 6c).

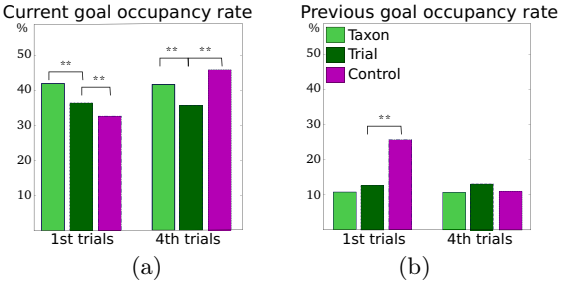


Fig. 5. Occupancy rates near the (a) current and (b) previous platform locations

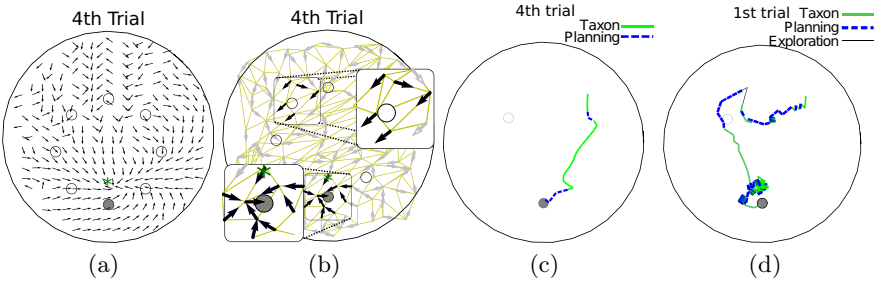


Fig. 6. Control group. Example of navigational maps of Taxon (a) and Planning (b) experts (learned directions of movement for each sample location or each spatial node) at the end of sessions. All possible platform locations are positioned, the current one is in grey. Taxon map is pointing the current landmark; Zooms in Planning map depict that the map is pointing the current platform position, but not the previous one. (c) Typical trajectory at the end of sessions (4^{th} trial). (d) Typical trajectory at the beginning of sessions (1^{st} trial) showing the planning expert leading the animat towards the wrong location.

At the beginning of a new session, as the gating network grants more confidence to the Planning, this expert has a stronger tendency to be selected. As a consequence, both Planning and Taxon experts compete, driving the animat respectively towards the previously memorized (thus wrong) platform location or the current one (Fig. 6d). Fig. 5b shows the significant differences of occupancy rates near the previous platform location between Control, Trial and Taxon groups during the 1^{st} trials, explaining the worse performance of the Control group. The selection rates of Planning and Taxon experts in Fig. 3b however show that this competition progressively decreases across sessions. Indeed the Taxon expert takes more and more control over Planning (negative correlation between Taxon and Planning selection rates: $r = -0.91$). Exploration is used until both experts have switched (i.e., until Taxon becomes sufficiently relevant, Fig 3b).

5 Discussion and Conclusion

The model of selection between Taxon and Planning navigation strategies presented here is based on the theory of parallel control systems in the rat brain. The place-based strategy uses a graph-search algorithm using the propagation of the reward signal to find the shortest path to the goal. The graph is learned on-line in pretraining sessions using the activities of learned place cells, and positions the location of the hidden goal. The cue-guided strategy uses a TD learning rule to approach the hidden goal marked by a landmark. The strategy selection is performed by a gating network that learns to predict, also using a TD-learning rule, the most successful strategy given current sensory input. Previous computational models of navigation rely on similar learning modes (e.g., dependent on the RL framework) for different navigation strategies [8,9,10], and the strategy selection mechanism is usually non-adaptive [16,17]. In contrast, our model allows for adaptive selection among different learning modes (including topological representations), exploiting a general and simple "common currency".

The model was evaluated in two simulated water maze tasks, in which the same sensory cues (intra-maze and extra-maze landmarks) were available. In one task, the cued hidden platform moves at each trial; in the other, the goal stays at the same place during a four trials session before moving to another location (Trial and Session tasks in [7]). Then what differs is the possibility, or not, to learn each position of the platform. Due to a separation between cooperative interactions (during action learning) and competitive interactions (during action selection), the model was able to assess the relative contribution of different strategies to the observed behavior. In accordance with the results of the original experiment, the selection mechanisms of the model – which did not change over the experiment – could explain why both place-based and cue-guided strategies did not interact in the trial condition, and why they cooperated or competed when they (particularly the place-based strategy) could learn the same platform position during several trials. A substantial contribution of the model concerned the analysis of the influence of different types of sensory cues on strategy selection. From the evolution of synaptic weights between sensory inputs and gating units in our simulations, it made it possible to assess that intra-maze landmark information was predominant for strategy selection in both tasks and moreover that it contributed to the selection of the Planning when this expert was allowed to be trained (see evolution of SC→Planning, Fig. 3d). Indeed, we remind that the gating network updates its selection on the basis of *both* sensory inputs .

We showed the ability of our model to efficiently select navigation strategies in two experiments in which the effects of environmental cues and training on navigation system interactions were rather simple. In particular conditions, these interactions may be more complex. For example, under the hypothesis that all spatial cues compete for predicting reward, numerous experiments supporting associative theory emphasize blocking (i.e., when a well learned cue predicting reward prevents learning of a novel cue predicting the same reward) or overshadowing effects (i.e., when one cue predicting the reward detracts the learning of

another present cue able to predict the same reward) in navigation [18]. Some of these effects are challenged by experiments supporting the cognitive mapping theory (e.g., [3]). Our model is a potential tool for investigating such contradictions.

Besides, the effect of practice on the selection of navigation strategies is far less investigated than the influence of sensory cues. This was particularly done in the experiments of [19], which stressed the complexity of this factor in studying the influence of intensity of training experience during one session on the later use of a specific strategy. In contrast to previous conclusions (e.g., [20]), short or long training, but also various periods of pre-exposure to the task, critically determine when and how a particular strategy emerges from interactions between both memory systems. In the future, our model could be applied to analyze such influences.

However we have to notice that the validation of the model was made in experiments using Morris water mazes, in which every orientation is left opened and not constrained by corridors. A recent paper raises the hypothesis that, in this kind of device, expression of strategy switching may be different from T- or Star-mazes [21]. Then we need to verify if the selection mechanism of our model supports such eventuality. More importantly, the same paper questions the role of DLS, with the assumption that it is not especially involved in egocentric/response strategy, but in the selection itself, in case of several available strategies. This requires analyzing, in similar protocols as theirs, which bias the specific role attributed to our expert Taxon is susceptible to entail, and in which extent it should be questioned.

Interaction among several spatial memory systems may improve the performance of animals either by speeding learning through cooperation of different strategies, or competitive processes that prevents sub-optimal strategies to be applied. Better understanding of these interactions by computational modelling may also provide a good basis for the design of robots able to cope with a wider range of behavioral tasks.

Acknowledgment. This research was granted by the EC Integrated Project ICEA (Integrating Cognition, Emotion and Autonomy, IST 027819).

References

1. White, N., McDonald, R.: Multiple parallel memory systems in the brain of the rat. Neurobiol. Learn Mem. 77, 125–184 (2002)
2. Doeller, C.F., King, J.A., Burgess, N.: Parallel striatal and hippocampal systems for landmarks and boundaries in spatial memory. Proc. Natl. Acad. Sci. USA 105(15), 5915–5920 (2008)
3. O'Keefe, J., Nadel, L.: The hippocampus as a cognitive map. Oxford University Press, Oxford (1978)
4. Hartley, T., Burgess, N.: Complementary memory systems: competition, cooperation and compensation. Trends Neurosci. 28(4), 169–170 (2005)

5. Carrillo-Mora, P., Magda, G., Abel, S.: Spatial memory: Theoretical basis and comparative review on experimental methods in rodents. Behav. Brain Res. 203(2), 151–164 (2009)
6. Rich, E., Shapiro, M.: Rat Prefrontal Cortical Neurons Selectively Code Strategy Switches. J. Neurosci. 29(22), 7208–7219 (2009)
7. Pearce, J., Roberts, A., Good, M.: Hippocampal lesions disrupt navigation based on cognitive maps but not heading vectors. Nature 396(6706), 75–77 (1998)
8. Dolle, L., Khamassi, M., Girard, B., Guillot, A., Chavarriaga, R.: Analyzing interactions between navigation strategies using a computational model of action selection. In: Freksa, C., Newcombe, N.S., Gärdenfors, P., Wölfl, S. (eds.) Spatial Cognition VI. LNCS (LNAI), vol. 5248, pp. 71–86. Springer, Heidelberg (2008)
9. Chavarriaga, R., Strosslin, T., Sheynikhovich, D., Gerstner, W.: A computational model of parallel navigation systems in rodents. Neuroinformatics 3(3), 223–242 (2005)
10. Uchibe, E., Doya, K.: Reinforcement learning with multiple heterogeneous modules: A framework for developmental robot learning. In: The 4th International Conference on Development and Learning, pp. 87–92 (2005)
11. Ujfalussy, B., Eros, P., Somogyvari, Z., Kiss, T.: Episodes in space: A modelling study of hippocampal place representation. In: Asada, M., Hallam, J.C.T., Meyer, J.-A., Tani, J. (eds.) SAB 2008. LNCS (LNAI), vol. 5040, pp. 123–136. Springer, Heidelberg (2008)
12. Stroesslin, T., Sheynikhovich, D., Chavarriaga, R., Gerstner, W.: Robust self-localisation and navigation based on hippocampal place cells. Neural. Net. 18(9), 1125–1140 (2005)
13. Martinet, L.E., Passot, J.B., Fouque, B., Meyer, J.A., Arleo, A.: Map-based spatial navigation: A cortical column model for action planning. In: Freksa, C., Newcombe, N.S., Gärdenfors, P., Wölfl, S. (eds.) Spatial Cognition VI. LNCS (LNAI), vol. 5248, pp. 39–55. Springer, Heidelberg (2008)
14. Burnod, Y.: Organizational levels of the cerebral cortex: an integrated model. Acta Biotheor. 39(3-4), 351–361 (1991)
15. Dijkstra, E.: A note on two problems in connection with graphs. Numer. Math. 1, 269–271, 269-270 (1959)
16. Girard, B., Filliat, D., Meyer, J., Berthoz, A., Guillot, A.: Integration of navigation and action selection functionalities in a computational model of cortico-basal-thalamo-cortical loops. Adapt. Behav. 13(2), 115–130 (2005)
17. Guazzelli, A., Corbacho, F., Bota, M., Arbib, M.: Affordances, motivation, and the world graph theory. Adapt. Behav. 6(3), 435–471 (1998)
18. Leising, K., Blaisdell, A.: Associative basis of landmark learning and integration in vertebrates. Comp. Cogn. Behav. Rev. 4, 80–102 (2009)
19. Martel, G., Blanchard, J., Mons, N., Gastambide, F., Micheau, J., Guillou, J.: Dynamic interplays between memory systems depend on practice: The hippocampus is not always the first to provide solution. Neuroscience 150(4), 743–753 (2007)
20. Chang, Q., Gold, P.E.: Switching memory systems during learning: changes in patterns of brain acetylcholine release in the hippocampus and striatum in rats. J. Neurosci. 23(7), 3001 (2003)
21. Botreau, F., Gisquet-Verrier, P.: Re-thinking the role of the dorsal striatum in egocentric/response strategy. Front. Behav. Neurosci. 4, 1–12 (2010)

A Cortical Column Model for Multiscale Spatial Planning

Louis-Emmanuel Martinet[1,2] and Angelo Arleo[1]

[1] CNRS - UPMC Univ Paris 6, UMR 7102, F-75005, Paris, France
[2] CNRS - UPMC Univ Paris 6, UMR 7222, F-75005, Paris, France
louis-emmanuel.martinet@upmc.fr

Abstract. An important issue in spatial memory is the learning of abstract representations suitable for navigation planning. To address this problem, we have already developed a planning system inspired by the columnar organization of the mammalian cortex [1]. This model provides a neuromimetic architecture capable of learning topological spatial representations and planning goal-directed actions. The work presented here deals with the ability to encode multiscale representations of the environment, in order to solve large maze tasks. This is shown by validating the model on a multiscale version of the Tolman & Honzik's detour task [2]. Simulation results demonstrate that the performances of the planning system are invariant with respect to the scale of the maze. A series of statistical analyses is provided to characterise the neural activities subserving spatial planning. It is shown that the structural properties of the environment are encoded by the discharges of the location-selective neurones of the model. Complementing this purely spatial coding, the activity of another class of neurones in the model integrates both spatial and reward-dependent information suitable for navigation planning.

1 Introduction

According to experimental evidence, spatial navigation planning is likely to rely on a distributed neural network spanning limbic and cortical brain structures. This network includes the hippocampus, which mediates spatial representations, and neocortical structures, such as the prefrontal cortex, which participate to the elaboration of abstract contextual descriptions (e.g., accounting for motivation-dependent memories and action cost/risk constraints). We have built a columnar cortical model [1] to provide a neuromimetic architecture suitable for spatial navigation planning, and based on the interaction between the hippocampus and the prefrontal cortex. The planning process is based on an activation-diffusion mechanism, propagating reward-related information from the goal position through the entire topological network [1]. This propagation enables the system to plan action sequences (i.e., trajectories) from the current position towards the goal. The activation-diffusion mechanism produces an exponential decrease of the intensity of the goal signal that propagates along the topological graph [1]. To prevent the system from planning failures in the presence of large scale environments (where locations exist in which the propagating signal is likely to reach the noise level

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 347–358, 2010.

and decision making becomes random) the current model also learns topological representations whose resolution is adapted to the complexity of the environment (to account for structural regularities as corridors).

A review of theoretical discussions on hierarchical cognitive maps can be found in [3]. McNamara et al. [4] have suggested that human can solve difficult spatial problems by building a hierarchical cognitive map including multiple representations of the same environment at different spatial scales. Moreover, animals may be able to chunk available information and build hierarchical representations to facilitate learning [5,6,7,8,9]. Recently, multiscale spatial representations have been identified at the neural level. For example in the entorhinal cortex, Hafting et al. [10] have shown that grid cells have spatial fields forming a grid of variable resolution. Kjelstrup et al. [11] have provided neural recordings of place cell activities in a large maze, supporting the same multiscale coding property in the hippocampus. In our model, we suggest that this kind of multiscale representations should also be found in the neocortical areas such as the prefrontal cortex, commonly associated with high-level cognitive processes.

2 Methods

2.1 Topological Map Learning and Action Planning with a Column Model

Existing cortical column models (from earlier, e.g. [12,13], to most recent, e.g. [14]) focus on either the cytoarchitecture of the column or the functional aspect of columnar computation. Our model lies between these two extremes, i.e. it attempts to relate the columnar organization to the behavioral responses based on a bioinspired (highly simplified) neural network model. The basic components of our column model and its learning principles have been previously presented [1]. To summarise, an unsupervised learning scheme is employed to make each column encode a specific spatial location $s \in S$. Within a column, a set of minicolumns are selective for all the state-action pairs $(s, a_{1\cdots N}) \in S \times A$ experienced by the animat at location s. During navigation planning, all the minicolumns of a column compete with each other to locally infer the most appropriate goal-directed action.

Compared to our previous model [1], the columnar structure has been refined in order to provide a better understanding of the dynamics of the planning system and to improve its biological plausibility. In the model presented here (Fig. 1A), a column consists of three computational units S, P and V and a set of minicolumns, each of which consists of two units Q and D. S neurones are meant to encode a compact state-space representation from the location-selective activities of hippocampal place cells [15]. The simulated place cells provide the system with a continuous distributed and redundant allocentric state-space representation S [16,17,18]. Q and V neurones are responsible for encoding respectively the quality (i.e. the efficiency) of an action given a state and the value of a state regarding its distance to the goal. D neurones integrate spatial and reward-related information to code for the best local decision in their discharges. P neurones are used to propagate the path signal encoding the plan from a given position to the goal. The discharge of these units simulates the mean firing activity of a population of cortical neurones either in supragranular layers II-III (for S, P, V and Q neurones), or in infragranular layers V-VI (for D neurones).

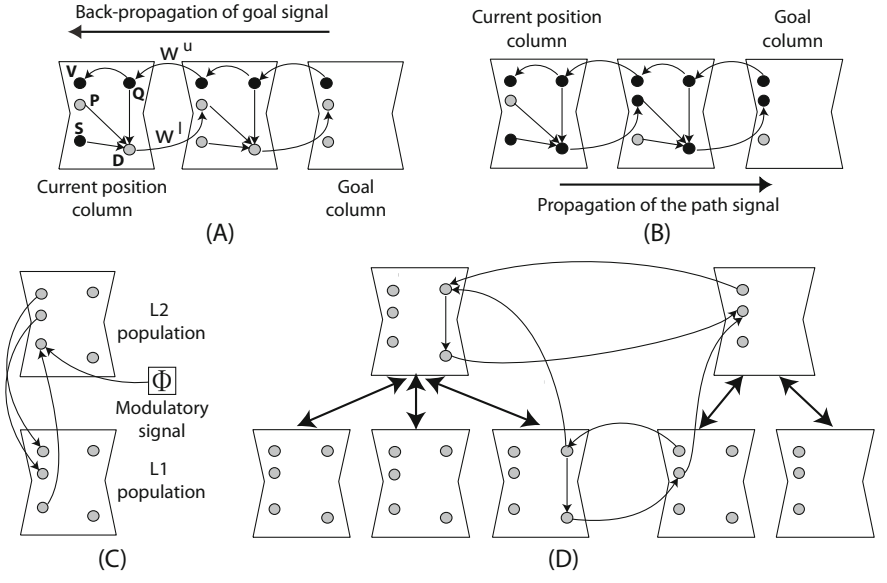


Fig. 1. (A-B) The cortical model and the implementation of the activation-diffusion process. Columns consist of three supragranular layer units (S, P, V) and a set of minicolumns containing a supragranular (Q) and an infragranular (D) layer unit. Black neurones are firing whereas gray ones are silent. **(A)** back-propagation of the motivational signal through the network of Q and V neurones. **(B)** forward-propagation of the goal-directed action signal through the P and D neurones. **(C)** Top-down and bottom-up connections between a L_1 column (bottom) and a L_2 column (top). Φ is a modulatory signal indicating variation in the high level context. **(D)** Topological connections are also learnt in the L_2 network (on this picture, connections detailed in (C) are summarized by a bidirectional arrow).

The planning process mediated by the columnar network (see example in Fig. 1A-B) is inspired by Burnod's activation-diffusion mechanism [19]. During trajectory planning, the unit V of the column corresponding to the goal location is activated via a motivational signal. Then, this reward-related activity is back-propagated through the network via the V and Q units (Fig. 1A). Q neurones convey this goal-related information to D units, where it is integrated with the spatial information coming from S and P units. When the back-propagated goal signal reaches the column selective for the current position s, the D unit becomes active and triggers the forward propagation of a goal-directed signal through projections $\mathbf{w}^l$ (Fig. 1B).

Notice that each $\mathbf{w}^u$ synapse attenuates the back-propagated goal signal. Thus, the smaller the number of synaptic relays, the stronger the goal signal received by the Q neurones of the column corresponding to the current location s. Since the receptive fields of the model columns are distributed uniformly over the environment by the unsupervised learning scheme [1], the intensity of the goal signal at a location s is roughly proportional to the distance of the target. Thus, goal-related metrical information is encoded implicitly by the network, which is is fundamental in order to select the shortest pathway to the target.

2.2 Dealing with Large Scale Environments

Let us denote population L_1 the previous cortical column population receiving spatial inputs from the hippocampus. A second population L_2 of columns is learnt by the current model to encode a large scale map adapted to the size of the environment. The learning algorithm is based on a measure that can define the boundaries between the high scale states. Here, we use a very simple mechanism suited for mazes with corridors, but the overall principle remains the same. A signal Φ is introduced to encode a change in the egocentric locomotion: $\Phi = 1$ when the animat is going straight and $\Phi = 0$ when it turns. This signal conveys relevant information to extract subpart of corridors in a maze. L_2 columns and minicolumns are the same generic computational units as in L_1 network but they are receiving afferents from L_1 columns modulated by the gating signal Φ (Fig. 1C). This "boundary" signal introduces a locomotion-dependent bias in the spatial selectivity of S neurones, such that the morphological properties of the environment (e.g., alleys in a maze) can be encoded by the L_2 topological map explicitly. An unsupervised growing network scheme is being employed to recruit L_2 columns similarly to the L_1 population. Additional top-down connections are created from L_2 to L_1 so that the former population can exert a top-down modulation on the P and V neurones of the L_1 population (Fig. 1C), enabling the planning process at the level of L_1 to cope with the decreasing back-propagating signal. This is achieved simply by enhancing the transfer function of P and V units in L_1 with a positive factor.

Because the size of high scale states will not be homogeneous as opposed to the state representation in L_1, a more flexible topological learning must be employed to account for the distance between any state and the goal. To solve this issue, two sets of bottom-up weights are used to convey the goal-distance information estimated at the level of the L_1 network by the activity of its Q and P units (Fig. 1D). This input is used to learn the lateral weights $\mathbf{w}^l$ and $\mathbf{w}^u$ in the population L_2, so that the activity of a V unit in L_2 is correctly correlated with the distance of the high scale state to the goal thanks to the information encoded in L_1. In other words, planning computations propagated at the level of the L_1 network are available in the L_2 network which uses them to estimate correct goal-distance information. Thus there is a bi-directional (bottom-up and top-down) flow of information between the two populations of columns of the model, making it possible to encode the environment at multiple scales and to solve large maze planning tasks.

3 Results

3.1 Spatial Behaviour in a Detour Task

In order to validate our multiscale navigation planning system, we chose the classical experimental task proposed by Tolman & Honzik [2], as in our previous work [1]. The main objective of this behavioural protocol was to demonstrate that rodents undergoing a navigation test were able to show some "insights", e.g. to predict the outcome of alternative trajectories leading to a goal location in the presence of blocked pathways. The original Tolman & Honzik's maze and protocol are shown in Fig. 2A. Here we extended its principle by using multiple size of the same maze to test the ability of the model to

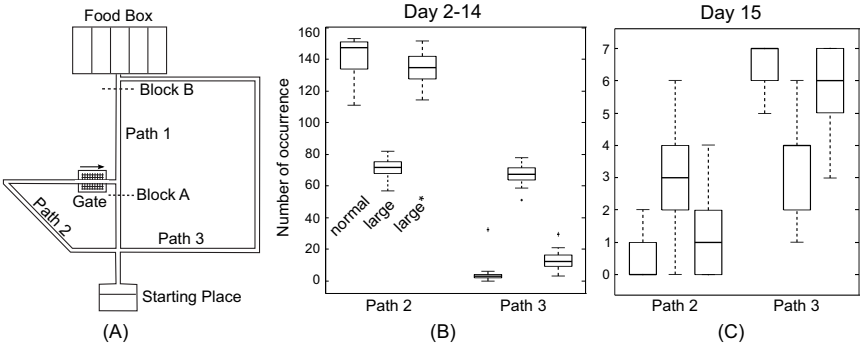


Fig. 2. **(A)** Tolman & Honzik's maze (adapted from [2]). The gate near the second intersection prevented rats from going from right to left. **(B-C)** Behavioural results for the Tolman & Honzik's maze represented as the mean number (averaged over 40 animats) of transits through P2 and P3 during Day 2-14 (B) or Day 15 (C). Several sizes of the maze are used: normal and large (four time bigger). In the normal and large conditions, no multiscale learning was used, unlike in the large* condition.

produce multi-scale topological maps and to solve detour tasks in increasingly larger mazes. Two versions of the Tolman & Honzik's maze were thus used: the classical one and a large one which was four times bigger than the original. For their experiments, Tolman & Honzik used 10 rats with no previous training. In our simulations, we examined a set of 40 simulated animats for each experimental condition. In the *classical* and the *large* conditions, the top-down influence of the L_2 population was discarded to show how the size of the maze progressively impaired the performance of animats in the absence of a compensatory neural adaptation. We also ran a set of 40 experiments in the large maze allowing the top-down influence of the high-scale cognitive map over the planning process (*large* condition*). Here we focus on the multiscale aspect of the task, because we have already shown in [1] that the cortical column could reproduce the original results in Tolman & Honzik's normal maze. We assessed the statistical significance of the results by means of an ANOVA analysis (the significant threshold was set at 10^{-2}, i.e. $p < 0.01$ was considered significant).

Day 1. During the first 12 training trials, the animats learnt the topology of the maze and planned their navigation trajectory in the absence of both block A and B. Similar to Tolman & Honzik's findings, our results in all conditions (normal, large and large*) show that the model learnt to select the shortest goal-directed pathway P1 significantly more frequently than the alternative trajectories P2, P3 (ANOVA, $p < 0.0001$ for all mazes). However, for the large condition (but not for large*), the size of this maze began to induce few mistakes, as indicated by a lower median value of Path 1 selection.

Days 2-14. During this training phase (consisting of 156 trials), a block was introduced at location A, which forced the animats to update their topological maps dynamically, and to plan a *detour* to the goal. P1 was ignored in this analysis (similarly to Tolman & Honzik's analysis) because blocked. The results reported by Tolman & Honzik provided strong evidence for a preference for the shortest *detour* path P2. Consistently, in our simulations (Fig. 2B) we observed a significantly larger number of

transits through P2 compared to P3 for normal and large* cases (ANOVA, $p < 0.0001$), but this was hardly significant for the large condition with a mean number of selected P3 very closed to P2 (ANOVA, $p < 0.0082$). This low performance was very closed to the behavior of an animat turning randomly toward Path 2 or Path 3.

Day 15. Seven probe trials were performed during the 15th day of the simulated protocol, by removing the block A and adding a new block at location B. This manipulation aimed at testing the "insight" working hypothesis: after a first run through the shortest path P1 and after having encountered the unexpected block B, will animats try P2 (wrong behaviour) or will they go directly through P3 (correct behaviour)? According to Tolman & Honzik's results, rats behaved as predicted by the insight hypothesis, i.e. they tended to select the longer but effective P3. Our probe test simulation results are shown in Fig. 2C. Similar to rats, the animats exhibited a significant preference for P3 compared to P2 (ANOVA, $p < 0.0001$) for normal and large* cases. However this probe test was a failure for the large condition, where the number of P3 choices was not significantly different from the number of P2 choices (ANOVA, $p = 0.6750$).

Taken together, these results clearly show an impaired performance proportional to the size of the maze, which can be overcome thanks to an adaptive multiscale representation fitting the structure of the maze and providing a top-down modulation of the activation-diffusion mechanism.

3.2 Analyses of Neural Activities

For all the simulations, we used pools of 600 units for each type of neurones (S, P, D, V, Q and HP). A series of analyses, partially based on the same theoretical tools as in [1], was done to characterise the neural activities subserving the behavioural responses of the system. The set of stimuli S consisted of the places visited by the animat. For the analyses, the continuous two-dimensional input space was discretized, with each location $s \in S$ defined as a 5 x 5 cm square region of the environment.

First, spatial neural activities from three populations were recorded during the large Tolman & Honzik's task: HP cells and S units from L_1 population as well as from L_2 population. In our previous work, we have shown that the cortical column model (i.e., the L_1 population) was able to build a more compact spatial representation storing the main part of the spatial information [1]. Here we focus on the spatial properties of the L_2 population compared to L_1 and HP neurones. Fig. 3 conveys a clear information on the population distinction between S units of L_1 and L_2: *(i)* fewer units of L_2 are necessary to represent the same environment (Fig. 3A, ANOVA, $p < 0.0001$), *(ii)* according to their spatial density measure [1,20], their receptive fields are less redundant (Fig. 3B, ANOVA, $p < 0.0001$), and *(iii)* the distribution of L_2 population responses is sparser than for L_1 (Fig. 3C, ANOVA, $p < 0.0001$) as shown by its kurtosis value [1,20], i.e. fewer neurones of population L_2 were, on average, responding to a given stimulus s simultaneously. These results suggest that the L_2 cortical column network was able to provide an even sparser state-space population coding than L_1 population. Note that the density is strictly greater than 0 for L_2, thus every place of the maze are represented by the neural network.

In a second series of analyses, we focused on the activity of single cells, and we recorded the receptive fields of the three types of units. Fig. 3E displays some samples

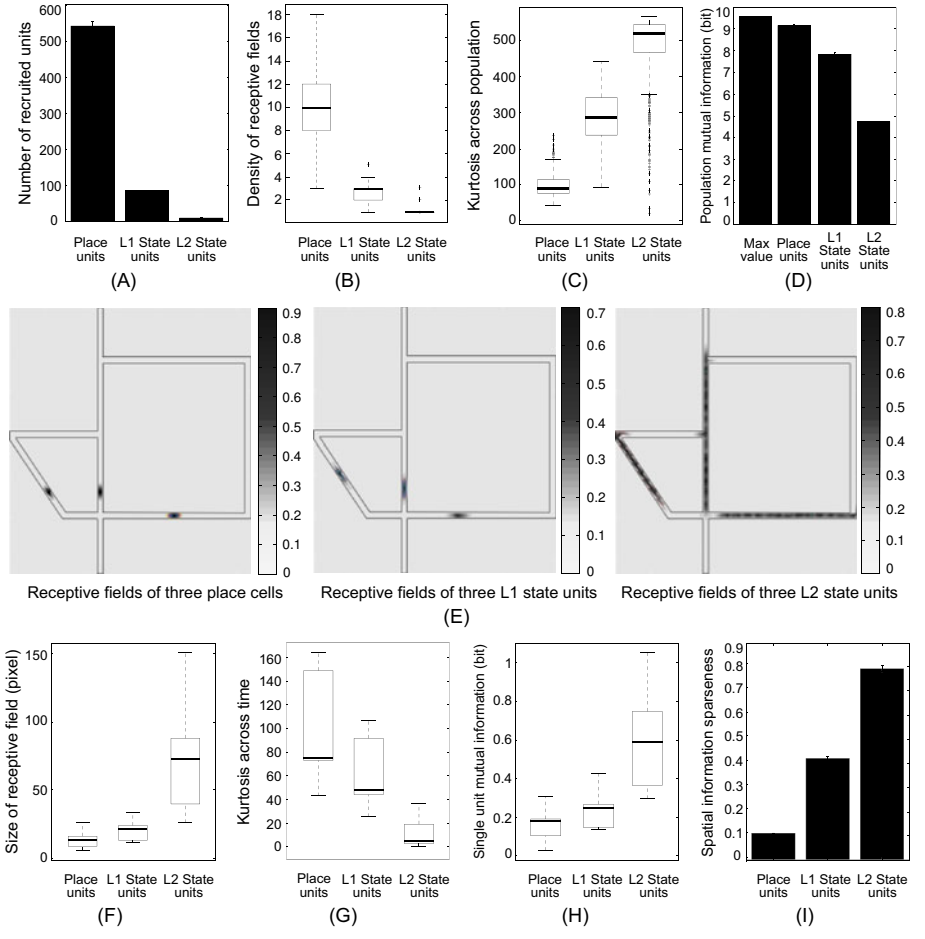


Fig. 3. Spatial correlates **(A-D)** Population spatial properties for HP cells, L_1 cortical S units and L_1 cortical S units: number of recruited units during the task, spatial density of the receptive fields, population kurtosis and population spatial mutual information. **(E)** Samples of receptive fields of three units from each population studied here. **(F-I)** Single cell spatial properties for HP cells, L_1 cortical S units and L_1 cortical S units: size of receptive fields, lifetime kurtosis, single unit spatial mutual information and information sparseness computed as the ratio between population information and the sum of single cell information.

of place fields for the three populations. What is mostly remarkable here is the firing properties of L_2 state neurones: after learning, the activity of these units capture some structural properties of the environment (i.e., corridors organization). A quantitative analysis was performed: the mean size of place fields (computed as the number of contiguous pixels with the firing rate above the grand mean rate plus the standard deviation [21]) was indeed significantly bigger than for L_2 units (Fig. 3F, ANOVA, $p < 0.0001$). Coherently their responses were the least sparse ones across the time (Fig. 3G, ANOVA, $p < 0.0001$) according to their low lifetime kurtosis [1,20]. Here the results tend to differ from our previous study [1]: the mean size of place cell receptive fields was indeed significantly lower than for L_1 cortical units (ANOVA, $p < 0.0001$), and coherently their responses were the sparsest ones across the time (ANOVA, $p < 0.0001$). The difference with our previous results (no significant difference) can be simply explain by the change of synaptic input function for the cortical neurones, which tends to broader the range of responses for L_1 S units. In fact, these new results receive support from experimental data by [21]: these authors measured the field size of place cells and prefrontal neurones in rats solving a navigational task, showing that the former were significantly smaller than the latter.

We also used an information theoretic analysis [22]: the *mutual information* $\mathrm{MI}(S;R)$ between neural responses R and spatial locations S allowed us to quantify the spatial information content of a neural code, i.e. how much could be learnt about the animat's position s by observing the neural responses r. It was evaluated for single units as well as for a whole population of neurones (in that case, r was a vector of firing rates), and the ratio between these two values was used to assess the level of sparseness of spatial information. The results of our information theoretic analysis are consistent with the properties described in the previous paragraph). Indeed, L_2 state units responding to a broader range of spatial stimuli, their single neurone mutual information is much higher (Fig. 3H, ANOVA, $p < 0.0001$). The spatial mutual information computed for the whole population of place cells, L_1 and L_2 state units (Fig. 3D) demonstrates a larger information content for the HP population (ANOVA, $p < 0.0001$), which may look in contradiction with the single cell mutual information. However they quantify quite different properties. As mentioned, the latter is related to the range of stimuli that make a cell fire, as well as the variability of this discharge. The population spatial information indicates that, for the binning procedure applied in this analysis, the place cell population is far more precise to encode a position (because of its high redundancy). In comparison, state neurones in L_2 population encode a very coarse spatial information. This is coherent with our initial goal of building a more compact representation accounting only for the main properties of the environment (here the corridors). Finally, when computing the information sparseness (i.e. the ratio between population information and the sum of single cell information), it appears that the information content was more redundant for place cells (Fig. 3I, ANOVA $p < 0.0001$), meaning that many of them encoded the same information. Although loosing a part of the population spatial information, the cortical population achieved a better coding scheme, maximizing the coding role of each units, particularly for the L_2 population.

Our second objective here is to show how information relevant for planning are encoded in the neural network. It is first necessary to demonstrate that the V population of

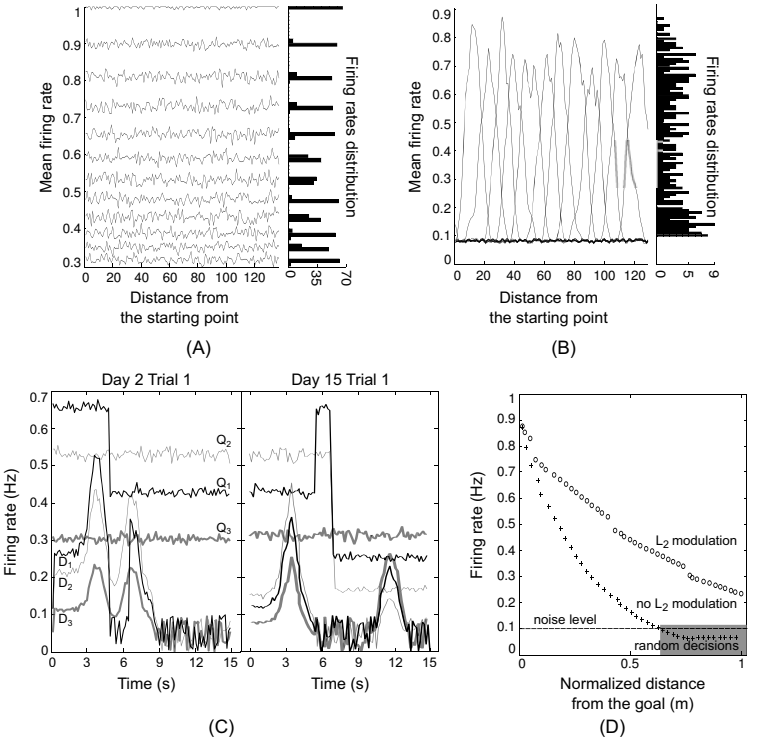


Fig. 4. **(A)** Activities of 12 V units (left: spatial activity, right: histogram of firing rates. **(B)** Activities of 12 S units. **(C)** Activities of 3 pairs of (Q,D) units belonging to the same minicolumn for two phase of the protocol where a block is introduced in the maze. **(D)** Effect of the top-down modulation exert by population L_2 V units over the propagating activity at the level of L_1 V unit population.

the cortical model encodes a measure of distance to the goal. As such, we need to show that responses of V units belonging to columns selective for places situated at different distances of the goal are not ambiguous and are anti-correlated with the distance. The first point is demonstrated in Fig. 4, in comparison with S units. Finding the selectivity of the latter is easy in the spatial domain (see previous paragraph), however that is not the case for V units. Instead, it is necessary to observe their firing frequency preferences to see that their responses are not overlapping (Fig. 4A), unlike S units (Fig. 4B). This is confirmed by a redundancy and a population kurtosis analysis showing the much sparser responses of V units in the frequency domain compared to the spatial domain (ANOVA, $p < 0.0001$), with an opposite effect for S units (ANOVA, $p < 0.0001$). The second property of goal distance anti-correlation is shown on the Fig. 4D, with or without the effect of the high-scale cortical population. This study observed the effect of the top-down modulation exert by population L_2 V units over the propagating activity at the level of L_1. Indeed, we remind that one motivation for this extension of the model was the possibility to deal with large environments. We have shown behaviourally that the model was able to adapt to them. Fig. 4D is a direct evidence of the neural effect of this

top-down modulation. Without any modulation, the strength of V units discharge fall exponentially with the distance of the column from the goal position. At a given point, this fast decreasing propagating activity will reach the neural noise level. From that point, only random decisions will be made because there will not be any correlation left between the firing activity and the real distance to the goal (e.g., the low performance on Day 15 for the *large* condition). When a top-down modulation is present, the decreasing effect becomes piecewise linear, each subpart corresponding to a high scale zone encoded by a L_2 column.

Reward-related V units and location-selective S units convey their information into the D neurones which integrate them into activities reflecting the selection of action (Fig. 4C). We remind that each minicolumn of the model is supposed to encode a specific state-action pair (s, a). As such, Q units encode the distance to the goal if a is selected at s, and D units integrate spatial information indicating the current position with this reward information. It can be seen on Fig. 4C at $t = 6s$ of the Day 2 Trial 1 that the animat has updated its connectivity in the cortical network to represent the presence of the block A. Thus, the previous best choice Path 1, represented by the best pair (Q_1, D_1) at $t \approx 4s$ is not correct anymore at $t \approx 7s$: Path 2 is now the best alternative as shown by the best pair (Q_2, D_2). The same mechanism occurs on Day 15 Trial 1, with Path 3 represented by (Q_3, D_3) becoming the best choice. Taken together, all these analyses demonstrate that the network encodes enough behavioural information, including distance-to-goal and best alternative information, to solve a planning navigation task.

3.3 Conclusion

We presented a multiscale extension of our previous cortical column model for spatial navigation. It enables the encoding of cognitive maps whose resolution fits the structure of the environment (e.g., corridors). As a consequence, the model is provided with a better adaptability in large mazes (e.g., in the presence of a maze four times larger than the original Tolman & Honzik's one), thanks to a top-down modulation regulating the activation-diffusion process. It should be noted that encoding multiscale maps is not the only solution to solve behavioral tasks in large mazes. However this approach is useful to address the issue of learning multiscale spatial representations, as found in the brain (e.g., [10,11]). Moreover, the model unravels the possible links between the single unit level and the behavioural level relevant to the learning of the task (e.g., to the selection of the shortest path to the reward). Our neural response analysis suggests how the interplay between the simulated hippocampus and prefrontal cortex can yield to the encoding of manifold information pertinent to the spatial planning function, including for example distance-to-goal correlates.

The model is currently being validated by comparing simulated neural response patterns against those obtained by in vivo electrophysiological recordings from the hippocampus and the prefrontal cortex of freely moving rats [23]. This comparative study aims at providing new insights on the interaction between the hippocampus and the prefrontal cortex. In addition, an ongoing work in coordination with experimentalists [23] attempts to study the learning processes related to spatial memory, such as

declarative memory consolidation occurring during sleep. This will possibly lead to testable predictions about the formation of memory traces relevant to spatial behaviour.

Acknowledgments. Funded by the EC Integrated Project ICEA, IST-027819-IP.

References

1. Martinet, L.E., Passot, J.B., Fouque, B., Meyer, J.A., Arleo, A.: Map-based spatial navigation: A cortical column model for action planning. In: Freksa, C., Newcombe, N.S., Gärdenfors, P., Wölfl, S. (eds.) Spatial Cognition VI. LNCS (LNAI), vol. 5248, pp. 39–55. Springer, Heidelberg (2008)
2. Tolman, E.C., Honzik, C.H.: "Insight" in rats. Univ. Calif. Publ. Psychol. 4(14), 215–232 (1930)
3. Schmajuk, N., Voicu, H.: Exploration and navigation using hierarchical cognitive maps. In: Animal Spatial Cognition: Comparative, Neural, and Computational Approaches (2006)
4. McNamara, T.P., Hardy, J.K., Hirtle, S.C.: Subjective hierarchies in spatial memory. J. Exp. Psychol. Learn. Mem. Cogn. 15(2), 211–227 (1989)
5. Dallal, N.L., Meck, W.H.: Hierarchical structures: chunking by food type facilitates spatial memory. J. Exp. Psychol. Learn. Mem. Cogn. 16(1), 69–84 (1990)
6. Fountain, S.B., Rowan, J.D.: Coding of hierarchical versus linear pattern structure in rats and humans. J. Exp. Psychol. Learn. Mem. Cogn. 21(3), 187–202 (1995)
7. Macuda, T., Roberts, W.A.: Further evidence for hierarchical chunking in rat spatial memory. J. Exp. Psychol. Learn. Mem. Cogn. 21(1), 20–32 (1995)
8. Meck, W.H., Williams, C.L.: Perinatal choline supplementation increases the threshold for chunking in spatial memory. Neuroreport 8(14), 3053–3059 (1997)
9. Roberts, W.: Spatial memory in the rat on a hierarchical maze. Learn. Motiv. 10, 117–140 (1979)
10. Hafting, T., Fyhn, M., Molden, S., Moser, M.B., Moser, E.I.: Microstructure of a spatial map in the entorhinal cortex. Nature 436(7052), 801–806 (2005)
11. Kjelstrup, K.B., Solstad, T., Brun, V.H., Hafting, T., Leutgeb, S., Witter, M.P., Moser, E.I., Moser, M.B.: Finite scale of spatial representation in the hippocampus. Science 321, 140–143 (2008)
12. Szentágothai, J.: The 'module-concept' in cerebral cortex architecture. Brain Res. 95, 475–496 (1975)
13. Eccles, J.C.: The modular operation of the cerebral neocortex considered as the material basis of mental events. Neurosci. 6, 1839–1856 (1981)
14. Markram, H.: The blue brain project. Nat. Rev. Neurosci. 7, 153–160 (2006)
15. O'Keefe, J., Nadel, L.: The Hippocampus as a Cognitive Map. Oxford Univ. Press, Oxford (1978)
16. Arleo, A., Gerstner, W.: Spatial orientation in navigating agents: modeling head-direction cells. Neurocomput. 38(40), 1059–1065 (2001)
17. Arleo, A., Smeraldi, F., Gerstner, W.: Cognitive navigation based on nonuniform gabor space sampling, unsupervised growing networks, and reinforcement learning. IEEE Trans. Neural. Netw. 15(3), 639–651 (2004)
18. Sheynikhovich, D., Chavarriaga, R., Osslin, T.S., Arleo, A., Gerstner, W.: Is there a geometric module for spatial orientation? insights from a rodent navigation model. Psychol. Rev. 116, 540–566 (2009)
19. Burnod, Y.: Organizational levels of the cerebral cortex: an integrated model. Acta Biotheoretica 39, 351–361 (1991)

20. Willmore, B., Tolhurst, D.J.: Characterizing the sparseness of neural codes. Netw. Comput. Neural Syst. 12, 255 (2001)
21. Hok, V., Save, E., Lenck-Santini, P.P., Poucet, B.: Coding for spatial goals in the prelimbic/infralimbic area of the rat frontal cortex. Proc. Natl. Acad. Sci. USA 102(12), 4602–4607 (2005)
22. Bialek, W., Rieke, F., de Ruyter van Steveninck, R., Warland, D.: Reading a neural code. Science 252(5014), 1854–1857 (1991)
23. Peyrache, A., Khamassi, M., Benchenane, K., Wiener, S.I., Battaglia, F.P.: Replay of rule-learning related neural patterns in the prefrontal cortex during sleep. Nat. Neurosci. 12, 919–926 (2009)

Why and How Hippocampal Transition Cells Can Be Used in Reinforcement Learning

Julien Hirel, Philippe Gaussier, Mathias Quoy, and Jean-Paul Banquet

Neurocybernetic team, ETIS, CNRS - ENSEA - University of Cergy-Pontoise, 95000 Cergy-Pontoise, France
julien.hirel@ensea.fr

Abstract. In this paper we present a model of *reinforcement learning* (RL) which can be used to solve goal-oriented navigation tasks. Our model supposes that *transitions between places* are learned in the hippocampus (CA pyramidal cells) and associated with information coming from path-integration. The RL neural network acts as a bias on these transitions to perform action selection. RL originates in the basal ganglia and matches observations of reward-based activity in dopaminergic neurons. Experiments were conducted in a simulated environment. We show that our model using transitions and inspired by Q-learning performs more efficiently than traditional actor-critic models of the basal ganglia based on temporal difference (TD) learning and using static states.

Keywords: hippocampus, basal ganglia, navigation, reinforcement learning, Q-learning.

1 Introduction

In previous papers, we proposed a model in which *"place cells"* [1] are not primary located in the hippocampus proper but in the entorhinal cortex. The activity recorded in the CA pyramidal cells would not primarily originate from *"place cells"* but from *"transition cells"* coding for the transient states from one place to the next [2,3]. The reason for this proposal arose from two experimental findings. First, experimental recording of our EC artificial visual place cells displayed large place fields allowing to reach a goal without the need to store a lot of places in the environment [4]. The merging of "What" and "Where" information about surrounding landmarks was sufficient to build a robust place code that could be simply recognized in order to build place cells. Hence the need for a dense mapping of the environment was not justified in simple sensori-motor navigation tasks. Second, we faced the impossibility to connect directly a cognitive map made of place cells and coding for multiple goals and motivations with a motor control system [2]. As a matter of fact, an homonculus was necessary to read the gradient activity on the cognitive map in order to deduce that moving in a particular direction would induce a better satisfaction than taking another direction. It was then always necessary to simulate at each time step these back and forth movements between the current place and the next possible places.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 359–369, 2010.

The building of a cognitive map linking transition cells suppressed this problem since one transition is always associated with a single movement. Action selection would take place in the nucleus accumbens (ACC) where planning activity coming from the cognitive map, linked to the prefrontal and/or parietal cortices, could be used as a bias to select from the current static state the most interesting transition. In our model, we used the dentate gyrus and its granular cells as a way to store past activities using a spectral timing model [5]. Area CA3 of the hippocampus received information about current and past states from the entorhinal cortex and dentate gyrus respectively. An associative memory allowed the learning of existing transitions between places. According to our model, CA3 pyramidal cells should predict the next possible transitions. Recording such cells should induce a strong spatial activity correlated with the animal place (the reason why they are called place cells) but somehow anticipating the animal next place. New neurobiological results are in agreement with such a prediction [6] but it is not sufficient to convince all the neurobiologists to move from a place cell model to a transition cell model. The cognitive map uses latent learning and constitutes an efficient system for dealing with dynamically changing environments with multiple goals. Yet there is no proof that the rat builds a cognitive map. Most of the hippocampal models used for navigation are based on place-action associations through RL and succeed to display interesting navigation performances [7,8,9,10].

In this paper, we show how the learning of transitions in the hippocampus, required by the cognitive map for complex planning tasks, can also form the perfect basis for a RL model based on Q-learning, as transitions are analogous to state/action couples. RL can easily be added to allow both backward planning with latent learning using the cognitive map and motivations, and forward planning using reinforcement hints to select the current action. Moreover the model can account for anatomical and physiological data in both the hippocampus and basal ganglia. This work is part of a project aiming at modeling the interaction between the hippocampus, the prefrontal cortex and the basal ganglia. We will show how our model can be more efficient than actor-critic models based on TD learning in tasks with several goals and motivations. Finally we will demonstrate the performances of the model in goal-oriented tasks in a simulated environment.

2 Model

In RL the environment is usually described as a Markov Decision Process (MDP). The agent can be in a certain number of states in which it can choose between a certain set of actions to perform. Experiments have been made in simulation where the agent switched between finite states based on its location in a grid world [11] or relative to prominent landmarks [9,10]. Place cells, with their *place fields* defining particular locations of the environment, can be used to characterize the state of the agent in RL [7,8].

The *Temporal Difference* (TD) learning algorithm [12] aims at maximizing the sum of expected rewards. While in TD learning an estimation of that sum is

learned as a function of states, Q-learning [13] creates an estimate as a function $Q(s, a)$ of state and action. After performing action a_1 to move from state s to state s', the Q value is adjusted with the following equation:

$$Q(s, a_1) \leftarrow Q(s, a_1) + \alpha(r + \gamma \max_a Q(s', a) - Q(s, a_1)) \quad (1)$$

where r is the reward obtained when in s', α is the learning rate and γ is a discount factor. The pair (s, a_1) can also be represented as a transition $s \rightarrow s'$

The discovery of the response of dopaminergic neurons in the substantia nigra pars compacta (SNc) and the ventral tegmental area (VTA) with their modulation of the basal ganglia neuronal activity, suggested the strong involvement of these structures in RL [14]. These neurons exhibit short bursts of firing just after the occurrence of an unexpected reward and go through a short period of depression when an expected reward is not received. The similarity of this behavior with the computation of the error on the prediction of expected rewards in TD learning has lead many researchers to build models of RL associated with the basal ganglia [15,16]. In the models the computation of the TD error made in the SNc matches the neurophysiological observations of dopaminergic neurons.

How the neural differentiator used to compute the difference between subsequent predictions for the TD error signal works is subject of debate. A hypothesis is that it originates from the direct and indirect connections between the striatum and the substantia nigra pars compacta (SNc)[15]. Direct inhibitory connections and indirect excitatory connections through the subthalamic sideloop would provide the desired signal. This model supposes different timings of spike propagation in the direct and indirect pathway. It is limited because of its reliance on the internal dynamics of synapses and neurons to account for the acceptable delay between subsequent predictions. Moreover the use of the temporal characteristics of the direct/indirect pathway as the neural substrate for the TD error computation seems to be inconsistent with the known neuroanatomy [17].

In addition, several RL models use delayed synaptic learning with an eligibility mechanism [18,8,7,9]. This mechanism assumes that a memory trace of past activity is present at the synaptic level. The current reward expectation is used to modify the synaptic weights selected by the eligibility trace corresponding to the last actions. The biological plausibility of the eligibility trace remains unclear. Houk and colleagues [15] gave an hypothesis as to how this learning can happen in real synapses. Their model involves the spiny neurons in the striosomal compartments of the striatum. The properties of a protein (CaM PK II) and a cascade of intracellular signaling mechanisms are used to account for the delay of the synaptic strengthening. However, once again, the timing of the reward is highly dependent on the properties of the internal dynamics of the neuron. It cannot account for a large variability in the delay between the action and the occurrence of the reward signal.

The need for both temporal mechanisms arises from the unavailability of the corrected reward prediction $(r + \gamma \max_a Q(s', a)$ in eq. 1) when the action is performed. This value is available in the following moments when the agent is in the new state s' and has received an optional reward r. However the previous

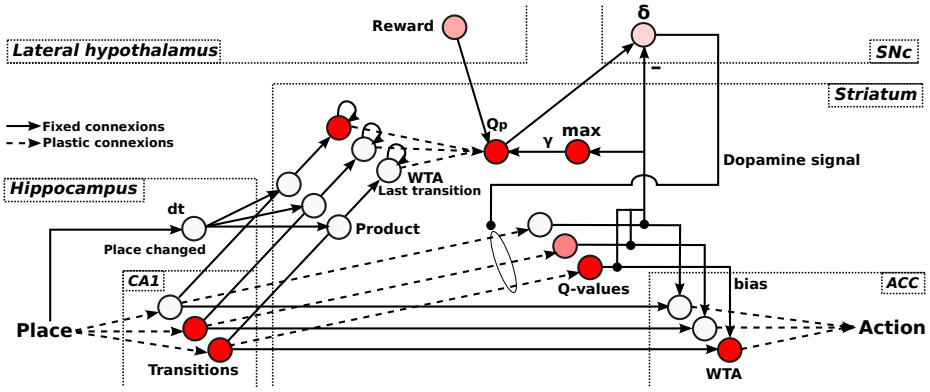


Fig. 1. Model of the Q-learning using the 2-step learning. The mapping of the various functions to cerebral structures is shown.

state is no more active and cannot be directly associated with the corrected estimation of its reward expectation value.

Taking inspiration from the actor-critic model, a neural implementation of the TD learning algorithm [18], we designed a neural network model of the Q-learning algorithm (Fig. 1). The model addresses the issues discussed in the previous paragraphs by the use of a 2-step learning mechanism, suppressing the need for both an eligibility trace and specific temporal dynamics in direct/indirect pathways between the striatum and the SNc.

Step 1: A working memory in the striatum stores information about the last transition performed. When the representation of the new state is stabilized and reward predictions about available transitions arise, their maximum value is learned and associated with the value in the working memory. If a primary reward signal is received, it will also be learned. For any transition $s \rightarrow s'$ we learn to predict what the value of $t_j = r + \gamma \max_a Q(s', a)$ will be. The learning is made by a simple conditioning using the Widrow-Hoff Delta rule:

$$w_{ji} \leftarrow w_{ji} + \alpha(t_j - x_j^{Q_p}).x_i^{WTA} \tag{2}$$

where α is the learning rate. $x_j^{Q_p}$ and x_i^{WTA} are the activities of post- and pre-synaptic neurons respectively. All activities are rate-coded.

Step 2: Q values are learned in synaptic weights with transition cells as pre-synaptic neurons. Connections from the hippocampus (area CA1) to the striatum allow the propagation of transition activity to the RL system. When the agent starts to explore a new place, it begins to predict all the available transitions along with their Q values. The TD error signal, computed from the difference of current and predicted reward expectations, acts as a dopaminergic modulation of synaptic learning for transition Q-values. The learning equation used is :

$$w_{ji} \leftarrow w_{ji} + \alpha.\delta.x_i^{CA1} \tag{3}$$

where δ is the TD error signal and x_i^{CA1} the activity of the currently performed transition. Transition activity is as follows: if a transition is being performed (i.e. the agent switches from place A to B) then the only active transition is AB ($x_{AB} = 1$), otherwise if the agent explores place A (i.e. the place cell coding for A has the strongest activity) then all predicted transitions are active ($x_{AB} = 1, x_{AC} = 1$, etc.).

This system allows the simultaneous availability of the $Q(s, a_1)$ value learned in step 2 and the $r + \gamma \max_a Q(s', a)$ value learned in step 1. Hence the computation of the TD error signal does not require input pathways with different temporal properties. Only simple inhibitory and excitatory pathways are used. The trade-off for the absence of time-dependent local synaptic rules is a convergence speed for the neural network divided by 2.

The Q values for each predicted transition are used to bias the original activity of the transition cells. A WTA competition results in the optimal transition to be selected. The output of the competition is not a direct motor action but rather a motor transition, as opposed to hippocampal transitions which are perceptual. The transition then activates its corresponding learned action, which could range from complex behaviors to simple motor commands. Even though we chose to only take the optimal transition into account to select an action, secondary transitions are still predicted and provide their reward expectancies and actions as possible alternatives. In a model where actions are chosen from static states rather than transitions, a single state can correctly give a choice of actions along with their order of preference only if all the actions are coded in orthogonal patterns. If actions are coded as overlapping patterns, the connectivity of each transition with the action neurons allows the coding of well separate actions for each transition. Moreover in our case the learning of associations between actions and transitions is latent. It can happen at any time when navigating in the environment, even during an exploratory phase without any reward. Here the actions are represented by a direction to take and coded in a *neural field* [19]. Path integration information from the last place, computed from odometric input, is used to associate a direction with every transition performed. In the model, the only actions considered in each state are based on what was learned to be possible, not a set of pre-programmed actions (e.g Go east, Go west, etc.) as it is often used in actor-critic models [7,8]. The architecture also distinguishes itself by merging the learning of state and action reward prediction into a single learning of state+action values.

The synaptic learning of predictions is modulated by particular events triggering transitory neuro-modulatory signals. The learning of Q-values through conditioning (3) happens when a transition is performed (i.e when the most active place cell changes). The learning of future predictions and rewards (2) is modulated by the delivery of the reward. A fixed time interval between place entry and reward delivery is fixed at the beginning of the experiments to allow extinction. The fixed delay is needed to provide the timing of expected rewards and produce negative reinforcement values in case an expected reward is not

delivered. Future work will involve the use of a time spectrum architecture to learn reward timings and allow the delivery of rewards at any time.

3 Improvements over Actor-Critic Models

In a simple experiment where the environment contains only one reward location, place and transition-based systems work in similar ways. In computational terms, in addition to the N place cells coding for states, the transition architecture requires the use of between $4N$ and $6N$ neurons in average to learn the transitions [3].

However the transition architecture shows its strength in complex tasks with multiple goals and motivations. The motivations could range from basic drives (e.g. hunger, thirst) to the need to satisfy various goals and sub-goals. Let us consider a case where several types of resources (food, water etc.) are present in the environment. The corresponding K drives indicate the need for a particular resource. In TD learning, as a direct state-action association is created, the original model cannot learn to associate different actions to a particular state depending on the motivational context. A direct connexion from motivations to actions would indeed guide the behavior of the agent towards making always the same action when motivated, independently of the place it is in. An intermediate layer of $K * N$ neurons would need to be created to learn the association of state/drive couples with actions [7]. Actions learned in a state for one motivation would need to be learned again for other motivations event if they lead to the same direction (Fig. 2). If the action is coded as a direction vector, the learning of the movement between two place fields can take some time to converge to the vector between the two centroids (e.g. by averaging the directions taken each time to move from one place field to the other). On the other hand the Q-learning network would only need to associate the drives with

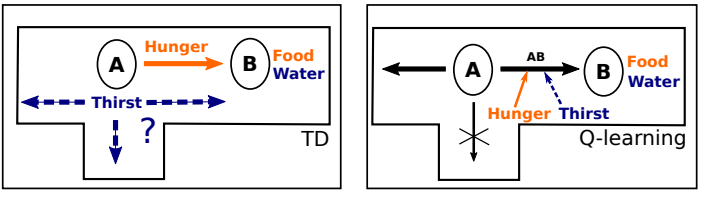


Fig. 2. Scenario with a food and water source intermittently located at place B. When the food source is found, the TD solution associates the A+hunger state with the action *Go east*. Further discovery of the water source eventually leads to the renewed slow learning of the action *Go east*, this time associated with the state A+thirst. With the transition solution, the agent learns the AB transition with the corresponding action *Go east*, independently of resource discovery. Further discovery of the food and water sources leads to the fast association of hunger and thirst with the existing AB transition. Moreover dead-end recognition could lead to a lower prediction value for the transition leading south, hence promoting the other transitions by default.

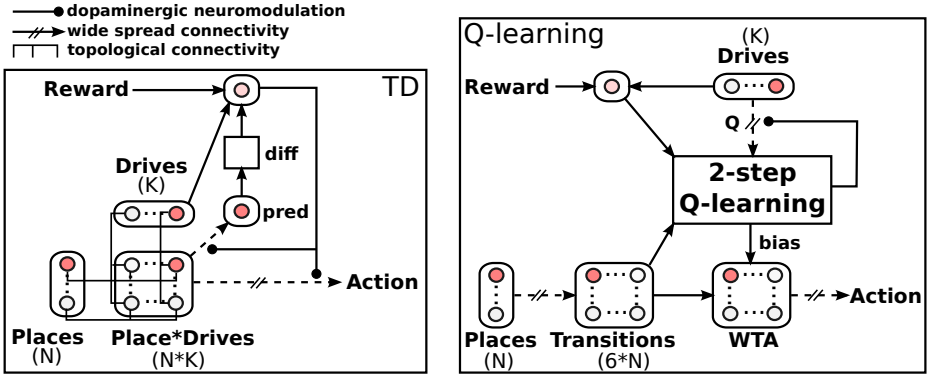


Fig. 3. Comparison of neural implementations of TD and Q-learning for a multiple drives scenario. When the number of places and drives increases, the transition based solution becomes less and less expensive as compared to the TD solution.

existing transitions. The action associated with a transition is learned whenever the transition is made, independently from the motivational context. Transition prediction activity would however have to be initiated by the co-activation of place recognition in the hippocampus and learned drive associations. Figure 3 shows a comparison of the two architectures. The Q-learning system is the one shown in fig. 1 with a few modifications to allow multiple drives. Rewards are associated with a drive to detect different types of goals, the resulting signal is given as input to the 2-step learning RL system. In place of the modulation described in section 2, the dopaminergic neuromodulatory signal is used to modulate the learning of the Q values in synapses originating from the drive neurons. The bias used to select the next action is thus combined from current transition and drive activities.

In our model, the number of neurons needed to encode transitions, states and actions is independent from the number of motivations. Due to the 2-step learning, a lot more neurons are needed for simple tasks with few motivations than in simple actor-critic networks. However these neurons can work with any number of motivations. The transitions model can use a direct bias of every new motivation on transition activity whereas actor-critic models would have to add extra place/drive neurons. In complex tasks with many goals and sub-goals this could lead to significant improvements in information compression, meaning more ecologically viable architectures. The trade-off is the need of wide-spread connectivity between places, drives, transitions and actions.

4 Experiments

The neural network has been tested in a simulated environment using the *Promethe* NN simulator [20]. The simulated environment is an open square environment with 20 perfectly identifiable landmarks equally spaced along the walls

to simulate visual input. One food source is placed in the upper left corner. The speed of the agent is constant throughout the experiment except when avoiding walls. The passing of time in the simulation is discretized into a series of time steps. However the functioning of the architecture is not dependent on the fineness of this discretization. The simulation works with any time step (e.g. 50ms, 100ms, 500ms etc.), however too large time steps would lead to the agent "teleporting" itself and missing sensory input on the way, leading to a less reactive system and decreased performances. The results were obtained using 100ms time steps.

First the agent performs an exploratory session in order to map its environment. During this phase of the experiment, navigation is guided by a random exploration strategy. The direction of the agent is periodically changed, based on a Gaussian probability function centered on the current direction. Simulated ultrasound obstacle detection allows the agent to avoid hitting the walls. Place cells are learned based on a minimum activity threshold. As the agent moves from place to place, transitions between place cells are learned and associated with a direction. During this random exploratory phase, the agent is able to discover the food source and build its representation of optimal paths using transitions and RL (Fig. 4).

During the second phase of the experiment, the exploratory/exploitation phases are modulated by an internal motivational signal. When motivated, the agent will use the learned transition bias and corresponding actions to reach the food source. The delivery of the food reward then inhibits the motivation signal and an exploratory phase begins. The motivation is triggered again when the agent reaches an area comprising the eastern and southern extremities of the environment. A

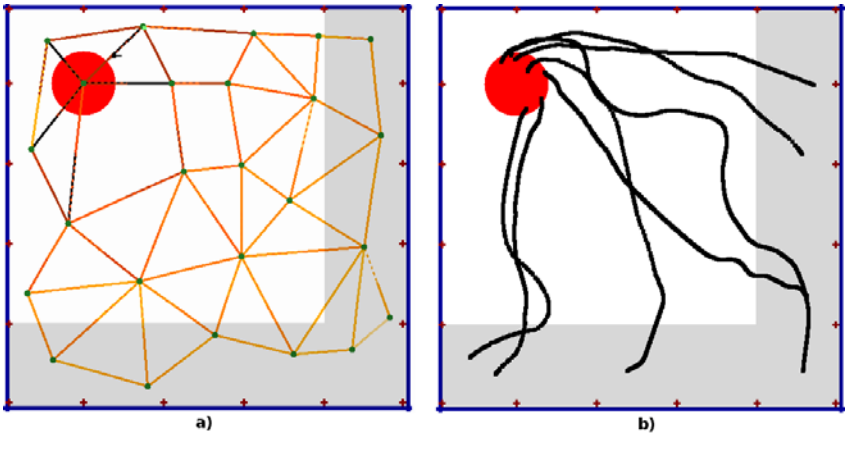


Fig. 4. a) Graph of all learned transitions in the simulated environment. Darker colors mean higher Q values for the corresponding transitions. **b)** Trajectories taken by the agent during goal-directed navigation. The goal location is represented by a disk in the upper left corner of the environment. All starting points for goal-oriented navigation trials are located in the gray area.

Table 1. Mean escape latency and standard deviation in seconds for transition Q-learning and random exploration. The values are computed from a set of 50 trials for each strategy. The parameters of the simulation are : learning rate $\alpha = 0.5$, discount factor $\gamma = 0.8$, reward value $r = 1$.

	Mean	Standard deviation
Transition Q-learning	36.7	14.5
Random exploration	115.1	90.1

good level of performance in this task requires the ability to quickly reach the goal location from any starting position in this area.

Figure 4 shows example trajectories of motivated navigation using RL. As the agent follows the path given by a single winning transition, the trajectories roughly follow the edges of the transition graph and are thus not straight lines to the goal. Smoother trajectories could be obtained using a soft competition when selecting transitions and their associated actions. Mean escape latencies and standard deviations are given in table 4 for both the transition Q-learning and a random exploration strategy. They express the time needed by the agent to reach the goal when motivated, with starting points randomly spread in the motivation trigger area. The transition Q-learning architecture performs 3 times better than random exploration with obstacle avoidance. A soft competition for transition and action selection could be used to further increase the performances of the algorithm.

5 Discussion

In addition to being consistent with neurobiological observations [6], the transition learning architecture could serve as a basis for several navigation strategies. The prediction of available transitions at any given time provides the system with a repertoire of possible actions. The transition-action association is learned autonomously and is dissociated from navigation strategies such as path planning or RL. As opposed to usual actor-critic models of TD learning where the motor action is the output of the RL network, a Q-learning based model can work with transitions as its sole representation of the environment and be more efficient in complex scenarios.

By using transitions as a common representation, one can easily integrate several navigation strategies in the same architecture. We previously used a cognitive map to solve navigation tasks. This strategy also provided a bias to the competition between predicted transitions. The competition leading to the selection of the next action can accept several such biases, given by different strategies working in parallel. The parallel use of the cognitive map planning and RL will bring to light the advantages/disadvantages of one system over the other and show the way for an integrated architecture with the 2 cooperating systems. More transition-based strategies, such as timed sequences of actions, could

eventually be added. In this case transitions would have to be able to learn both spatial and temporal properties. Future work will involve the implementation of a system capable of modulating these concurrent strategies. The modulation could be based on a performance criterion, thus selecting the best strategy for a particular task. Internal signals could also be monitored by a meta-controller capable of detecting whether a strategy is dysfunctional or not.

We have recently built an architecture which used transitions with both spatial and temporal components to build a cognitive map and solve planning tasks involving navigation and the precise timing of particular actions. The integration of timed transitions into the present RL model would help reproduce the precise time-dependent prediction capabilities of dopaminergic neurons in the basal ganglia. This is necessary to be able to select an appropriate behavior depending on the timing of a reward. One particular case in which we are interested is the autonomous learning of a precisely timed waiting period requiring movement inhibition from the animat.

Acknowledgments. This work is supported by the CNRS, as part of a PEPS project on neuroinformatics, and the DGA. We thank B. Poucet, S. Wiener and E. Save for useful discussions.

References

1. O'Keefe, J., Dostrovsky, J.: The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. Brain Res. 34(1), 171–175 (1971)
2. Gaussier, P., Revel, A., Banquet, J.P., Babeau, V.: From view cells and place cells to cognitive map learning: processing stages of the hippocampal system. Biol. Cybern. 86(1), 15–28 (2002)
3. Cuperlier, N., Quoy, M., Gaussier, P.: Neurobiologically inspired mobile robot navigation and planning. Front Neurorobotics 1, 3 (2007)
4. Giovannangeli, C., Gaussier, P., Banquet, J.-P.: Robustness of visual place cells in dynamic indoor and outdoor environment. International Journal of Advanced Robotic Systems 3(2), 115–124 (2006)
5. Grossberg, S., Merrill, J.W.: A neural network model of adaptively timed reinforcement learning and hippocampal dynamics. Brain Res. Cogn. Brain Res. 1(1), 3–38 (1992)
6. Alvernhe, A., Van Cauter, T., Save, E., Poucet, B.: Different ca1 and ca3 representations of novel routes in a shortcut situation. J. Neurosci. 28(29), 7324–7333 (2008)
7. Arleo, A., Gerstner, W.: Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. Biol. Cybern. 83(3), 287–299 (2000)
8. Foster, D.J., Morris, R.G., Dayan, P.: A model of hippocampally dependent navigation, using the temporal difference learning rule. Hippocampus 10(1), 1–16 (2000)
9. Khamassi, M., Lachèze, L., Girard, B., Berthoz, A., Guillot, A.: Actor-critic models of reinforcement learning in the basal ganglia: From natural to artificial rats. Adaptive Behavior 13(2), 131–148 (2005)
10. Mannella, F., Baldassarre, G.: A neural-network reinforcement-learning model of domestic chicks that learn to localize the centre of closed arenas. Philos. Trans. R. Soc. Lond. B. Biol. Sci. 362(1479), 383–401 (2007)

11. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. Journal of Artificial Intelligence Research 4, 237–285 (1996)
12. Sutton, R.S.: Learning to predict by the methods of temporal differences. Machine Learning 3, 9–44 (1988)
13. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning 8(3), 279–292 (1992)
14. Schultz, W.: Predictive reward signal of dopamine neurons. J. Neurophysiol. 80(1), 1–27 (1998)
15. Houk, J.C., Adams, J.L., Barto, A.G.: A model of how the basal ganglia generate and use neural signals that predict reinforcement. In: Models of Information Processing in the Basal Ganglia, pp. 215–232. MIT Press, Cambridge (1995)
16. Montague, P.R., Dayan, P., Sejnowski, T.J.: A framework for mesencephalic dopamine systems based on predictive hebbian learning. J. Neurosci. 16(5), 1936–1947 (1996)
17. Joel, D., Niv, Y., Ruppin, E.: Actor-critic models of the basal ganglia: new anatomical and computational perspectives. Neural. Netw. 15(4-6), 535–547 (2002)
18. Barto, A.G.: Adaptive critics and the basal ganglia. In: Models of Information Processing in the Basal Ganglia, pp. 215–232. MIT Press, Cambridge (1995)
19. Amari, S.I.: Dynamics of pattern formation in lateral-inhibition type neural fields. Biological Cybernetics 27(2), 77–87 (1977)
20. Lagarde, M., Andry, P., Gaussier, P.: Distributed real time neural networks in interactive complex systems. In: CSTST 2008, pp. 95–100. ACM, New York (2008)

The Complementary Roles of Allostatic and Contextual Control Systems in Foraging Tasks

Encarni Marcos[1], Martí Sánchez-Fibla[1], and Paul F.M.J. Verschure[1,2]

[1] SPECS, Technology Department, Universitat Pompeu Fabra, Carrer de Roc Boronat 138, 08018 Barcelona, Spain
[2] ICREA Institució Catalana de Recerca i Estudis Avançats, Passeig Lluís Companys 23, 08010 Barcelona, Spain
encarnacion.marcos@upf.edu, marti.sanchez@upf.edu, paul.verschure@upf.edu
http://specs.upf.edu

Abstract. To survive in an unknown environment an animal has to learn how to reach specific goal states. The animal is firstly guided by its reactive behavior motivated by its internal needs. After exploring the environment, contextual information can be used to optimally fulfill these internal needs. However, how a reactive and a contextual control system complement each other is still a fundamental question. Here, we address this problem from the perspective of the Distributed Adaptive Control architecture (DAC). We extend DAC's reactive layer with an allostatic control system and integrate it with its contextual control layer. Through robot foraging tasks we test the properties of the allostatic and contextual control systems and their interaction. We assess how they scale with task complexity. In particular, we show that the behavior generated by the contextual control layer is of particular importance when the system is facing conflict situations.

Keywords: Self-Regulation; Allostatic Control System; Contextual Control System; Cognitive Architecture; Robot Behavior; Bio-Inspired Robotics.

1 Introduction

One of the main challenges an animal faces when exploring a novel environment is how to learn about it and exploit it. Firstly, reactive behaviors drive animal exploration motivated by the animal's internal needs. Reactive behaviors also allows the exploration and acquisition of the state space. Once the states of the environment are learned and appropriate behaviors shaped, the animal is able to use this information to reach goal states, e.g. food. However, how these low and high level systems complement each other when the task difficulty increases is not clear yet. Here, we exploit this question in the basis of a combined robotics and computational neuroscience approach.

We investigate a robot model of self-regulatory processes based on the behavior of rodents. Rodents are optimal real-world foragers that can smoothly regulate complex sets of behaviors [1] based on their internal motivation, maintaining a dynamic stability with the environment while learning about it. Our

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 370–379, 2010.

model tackles exactly these two issues: self-regulation and learning about the environment. Self-regulation is provided by a reactive layer that is based on the concept of *allostasis* [2]. This reactive layer drives the robot behavior while information about the environment is acquired and retained in a long-term memory. This memory is part of the contextual control (CC) system which will be capable of driving the robot's behavior based on the robot previous experience.

The allostatic control system (AC) of our model allows the robot to not only explore the environment but also to acquire its salient states. The robot locally senses different reward gradients present in the environment and can reach its desired values in the gradient by performing instantaneous reactive motor actions. The CC system picks up information about the environment. Both systems need each other and are fundamental for solving navigation tasks. The CC system will be able to solve tasks when the cues, e.g. gradients, that guide the behavior of the reactive layer is incomplete or contains conflicts. In addition, it can optimize the content of the long-term memory system thanks to the states that are classified and the behaviors triggered by the AC system.

Many models deal with the problem of realizing an artificial rodent [3,4]. Generally, it is usually tackled in a bottom up approach solving the navigation problem using a cognitive map. Our approach differs from these models in that our first building block integrates many regulatory subsystems and on top of it we add a cognitive system able to learn about the environment.

The AC and CC systems integration will be made in the context of the biomimetic Distributed Adaptive Control architecture (DAC) [5,6,7]. In this paper we make two new contributions. Firstly, we will augment the DAC reactive layer in order to support self-regulation on the basis of the physiological principle of *allostasis*. The AC orchestrates different homeostatic subsystems achieving stability at a meta-level (see Section 2.1). However, during this study, we will use only one homeostatic subsystem at a time thus bypassing the question of how multiple homeostatic subsystems affect optimal performance. The second contribution of this study is that we extend the contextual layer to be able to exploit the different internal states of the agent in its memory structures. The long-term memory stores sequences that belong to different goal states, i. e. desired values in the gradient. In the recall phase, information is retrieved based on the internal motivation of the robot, e.g. hunger (see Section 2.2). We integrate these two control systems and investigate the main implications of their integration (see Section 2.3). Our results show a successful integration which gives rise to realistic foraging in a variety of benchmark tasks using a simulated robot (see Section 4). In addition, it also indicates how low level predefined behavior control systems of the brain can be integrated with more advanced neuronal systems.

2 Methods

To understand how AC and CC systems complement each other we work with the DAC architecture. DAC distinguishes three coupled layers that interact

between each other: reactive, adaptive and contextual. The reactive layer contains a pre-wired repertoire of reflexes, which creates a behavior that allows an interaction with the environment. Originally, this reactive layer implements collision avoidance and light appetitive mechanisms. We will extend this layer to provide it with an AC system. The adaptive layer processes and classifies the sensory input. This classification together with the actions executed by the robot are sequentially stored in the contextual layer which is equipped with a short and a long-term memory. These representations are used to plan ongoing behavior, and have been shown to be compatible with formal Bayesian models of decision making [5]. In the original DAC, the contextual layer stores positive or negative sequences that lead to goal states defined by reward or punishment respectively. We extend the contextual layer to equip it with labeled information where the content of the memory can lead to different kinds of rewards or punishments.

We will test our model in different foraging tasks where gradients are projected into the environment and visual cues are placed on the floor. The gradients are gaussian functions sensed by the robot. The AC system will steer the motors of the robot depending on the desired and actual value of the gradient. Sequences of sensory-motor contingencies are learned by the CC system from the robot's interaction with the environment. We will show that AC system alone is not always sufficient to reach goal states when the gradient has a conflict information.

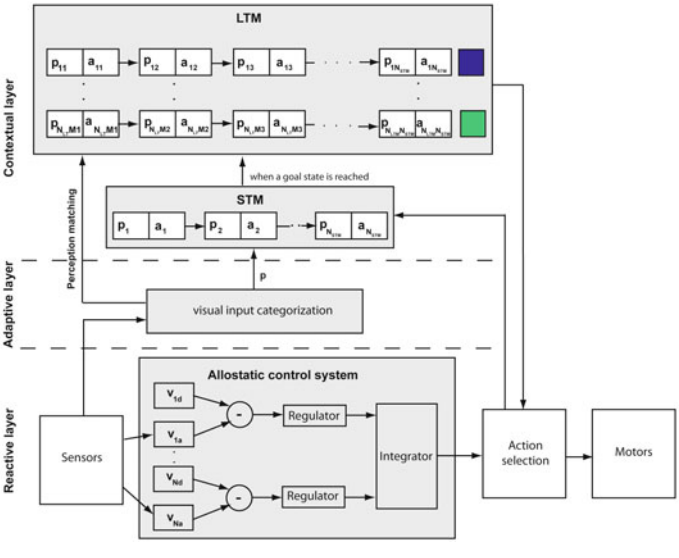


Fig. 1. Contextual and allostatic integration in the framework of DAC. Abbreviations mean: p, perception; a, action; v, value. Blue and green colors at the end of the sequences mean different reward associated with them. Action selection is done by priority: contextual layer actions have more priority than reactive layer actions.

2.1 The Allostatic Control System

In our self-regulation model approach, different simpler homeostatic subsystems coexist. Each homeostatic subsystem is associated to one reward gradient and has access to an actual (V_a) and desired (V_d) value in that gradient. The actual value is determined by the actual position of the robot in the gradient and the desired value represents the goal state in the gradient. The homeostatic subsystem acts in a closed loop trying to bring close the actual to the desired value and thus achieving stability (see Figure 1). An integrator mechanism orchestrates the different homeostatic subsystems. All the subsystems compete to control the robot. In this study, only one homeostatic subsystem is activated at a time.

Imagine, as an example, an animal placing itself in an optimal distance to a heating source in order to achieve a desired temperature [8]. The gradient would correspond to the temperature map which would have its highest peak around the source. The agent is able to sample the temperature gradient locally. The homeostatic subsystem could bring the agent closer to or further away from the heating source depending on the difference between the desired and actual value.

2.2 Short and Long Term Memory

The contextual layer of DAC contains a short-term memory (STM) and a long-term memory (LTM). Sequences of sensory-motor contingencies are learned from the robot's interaction with the environment. Perception-action associations are stored in STM. When a goal state is reached, the sequences of associations are copied into the LTM labeled with the reached goal state (see Figure 1). When the robot is exposed to a perception, it is clasified by the adaptive layer and compared with the perceptions stored in LTM. The stored perceptions that match this comparision, and belong to a sequence with a goal state coincident with its current internal motivation, are selected and an action is executed. For further explanation about the memory structure in DAC see [6].

2.3 Allostatic and Contextual Integration

We integrate AC and CC systems using the framework of DAC (see Figure 1). The reactive layer of DAC is provided with an AC system. This AC system steers the motor of the robot driven by the gradient in the environment and the internal motivation of the robot. This results in egocentric actions executed. These egocentric actions are converted into allocentric ones by the CC system. To do the conversion, we added a path integration computation that calculates the vector between visual perceptions using the head orientation of the robot. In this way, the information in memory contains visual cues and the vector connecting two visual cues or visual cue to a goal state.

3 Foraging Tasks

We test the integrated model in foraging tasks using 3D environment with a mobile agent. The simulated agent is implemented in C++ and wSim [9] using

the Open Graphics Library approximating a Khepera robot[1] widely used for behavioral modeling. The validity of the simulated robot with respect to a real one has been demonstrated in several studies [9]. The robot has a radius of 5.5 cm and it is equipped with three blocks of eight light sensors and eight proximity sensors. The sensors integrate an exponential decay function with respect to the distance to the light sources or to the obstacles respectively. The robot is also equipped with a color camera pointing to the floor (with an angle of 45°). Therefore, the 3rd dimension of the environment is limited to the walls. The action group from the architecture is connected to the motor group of the robot. Each cell of the motor group maps a direction of movement. A winner-take-all (WTA) takes place at the motor map level and selects the neuron with highest activity. The default movement of the robot is to go forward.

Our aim is to test the model in tasks with increasing difficulty to understand how AC and CC systems scale with task complexity, i. e. tasks where gradients and visual cues have coherent information and tasks where this information is contradictory. This will allow a better understanding on when AC system would be enough to fulfill the robot's internal motivation and when CC system would be necessary to optimally fulfill this internal motivation.

We run experiments in three different environments. Every environment contains visual cues and rewards. The visual cues are patches on the floor whereas the rewards are gradients. The internal motivation of the robot is set to the highest value in a way that the desired value is reached at the center of the gradient. The first foraging task is an open field environment with one kind of gradient, a light. We vary the weight of the gradient from 0 to 1 in steps of 0.1 (see Figure 2a). When the robot reaches the reward, i. e. its desired value in the gradient, the gradient is turned off to avoid that the robot stays next to the reward during the whole experiment. It is activated again when the robot is far away from it. For high weights of gradient it covers almost the whole foraging space, therefore we expect a similar performance in both AC and CC systems, since gradient and patches have coherent information. Secondly, we set an environment also with one kind of gradient, a light, with obstacles that do not allow a direct path between the initial position of the robot and the reward (see Figure 2b). This environment is remotely based on the Tolman maze [10] as in [11]. We expect to have a significantly better performance with the CC system, due to its capability of learning sequences of perception-action to reach goal states. In this case, every trial finishes when the robot reaches the reward and starts again from same spatial position and a random orientation selected from a two dimensional normal distribution $N(0, 1)$. As a final step, we set a third environment with three different gradients (see Figures 2c and 2d). The robot searches for one reward or other depending on its internal motivation which is randomly selected at the beginning of every trial. We expect a significantly better performance in the case of the CC system due to the presence of obstacles as in the previous task. To see the influence in performance in the CC system due to the number of visual cues we run experiments with four and eight patches in the environment.

[1] K-Team, Lausanne, Switzerland.

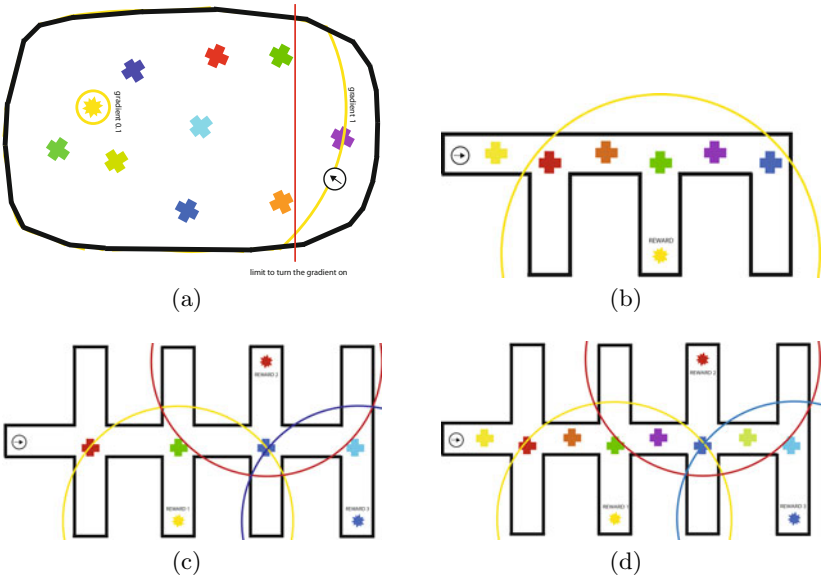


Fig. 2. Foraging tasks environments. (a) Open field task. Yellow lines indicate the lowest and the highest value of the gradient. Red line indicates the limit the robot has to cross so the gradient is activated again after reaching the reward. (b) Maze with one reward. Yellow line indicates the area covered by the gradient of the reward. (c) and (d) Maze with three different rewards. Yellow, red and blue lines indicate the area covered by each of the reward gradients.

In the three tasks, we keep constant the size of the memory, with 40 sequences of a maximum of 120 perception-action associations.

To simulate real conditions, we added 5% of noise to the motors of the robot. For every condition, we run 10 experiments with 20000 cycles each of them.

4 Results

4.1 Open Field Task

To compare performance between AC and CC systems we record data only when the AC system is activated. Later on, we activate the CC system and when the memory is full we record data again. We look at the ratio between the number of targets reached and the distance explored by the robot (see Figure 3). We observe that when the gradient is not present, i.e. the weight of the gradient is 0, the CC system performance is significantly higher than the AC system performance (Wilcoxon rank sum test $p < 0.001$). However, as the weight of the gradient increases the AC system performs better than the CC system. This is expectable since the gradient gradually occupies the whole environment and

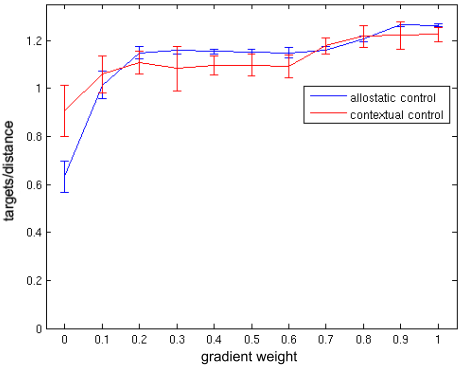
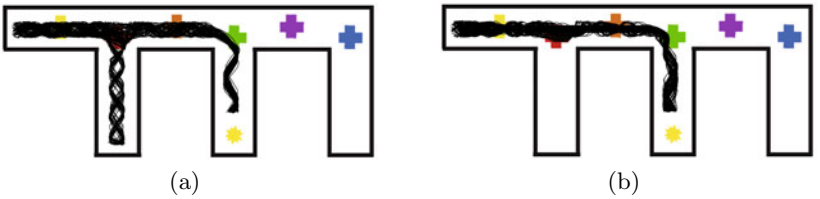


Fig. 3. Ratio targets/distance for the allostatic and contextual control systems for different gradient values

the AC system can optimally calculate next action to perform. Thus, the CC system improves performance for a low value of the gradient but for values higher than 0.3 the AC system performance is significantly better than the CC system performance (Wilcoxon rank sum test $p < 0.01$). Therefore, when there is not conflict in the gradient information the CC system is not fundamental and the AC system can properly perform the task.

4.2 Maze One Gradient Task

Firstly, ee test the AC system performance. We observe that the gradient of the reward drives the robot to a wrong path in some occasions (see Figure 4a). When the CC system is also activated, we observe that the robot is not driven to the wrong path (see Figure 4b). This difference in the paths selected in both cases is translated to a significant increase in the number of cycles needed by the robot to reach the reward with the AC system in comparison to the CC system (mean in the AC system is 222 cycles Vs. 141 in the CC system).



(a) (b)

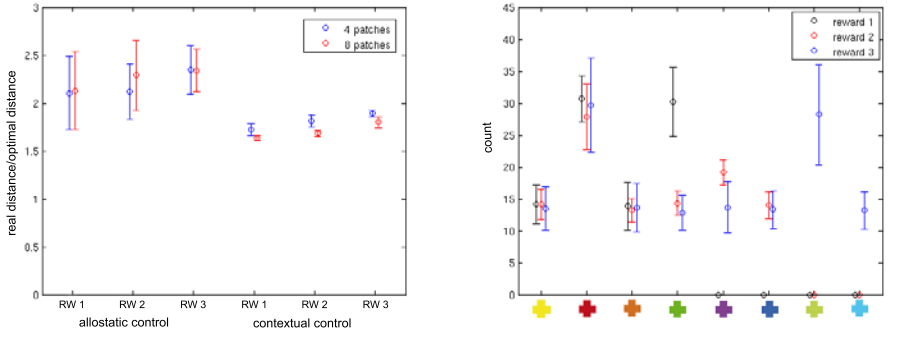
Fig. 4. Trajectory plots. (a) Trajectories of the robot with allostatic behavior. (b) Trajectories of the robot for contextual behavior.

4.3 Maze Three Gradients Task

We start testing the integration model with 4 patches in the environment. As in previous tasks, we first record data with the AC system alone and then we activate the CC system. We calculated the optimal distance the robot should cover to reach each of the rewards (see Figure 2c and 2d). In Figure 5a, we plot the ratio between the real distance covered by the robot and the optimal one for each of the rewards with the AC and CC systems. We observe that the CC system performance is significantly better for the three rewards (Wilcoxon rank sum test $p < 0.001$). The results also show that the performance decreases as the distance to the reward increases. This is to be expected since the probability to leave the optimal path increases. However, we observe that the slope of the performance for the three rewards is lower in the CC system than in the AC system suggesting that the performance will decrease faster in the AC system as the distance to rewards increases.

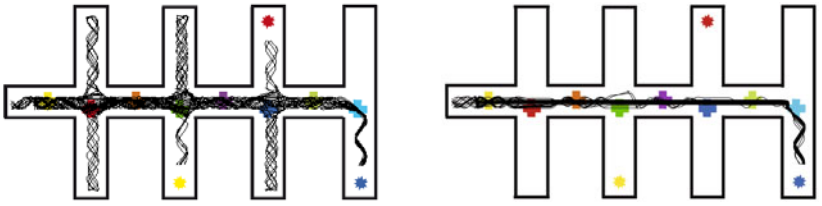
To test the influence of the number of patches in the CC system performance we increase the number of the patches in the environment, from 4 patches to 8. We observe in Figure 5a that the performance at the AC system level keeps constant since the patches are not used by it. However, we observe an increase in performance in the CC system. Due to the increase in the number of patches the CC system has a more accurate knowledge of the environment, i. e. greater number of internal representations leading to a greater number of actions that the memory might trigger.

To better understand how the CC system picks up information about visual cues, we look at the content of the memory. In Figure 5b we plot the number of times each visual cue appears in memory for the different rewards. We observe that the last two patches are used only for reaching the third reward whereas the rest of patches are part of the sequences corresponding to the three different



(a) (b)

Fig. 5. (a) Ratio real distance/optimal distance for allostatic and contextual control systems for each of the three rewards present in the environment: RW 1, reward 1; RW 2, reward 2, RW 3, reward 3. (b) Number of times that each patch is stored in memory for each of the three rewards.



(a) (b)

Fig. 6. Trajectory plots. (a) Trajectories of the robot with allostatic behavior. (b) Trajectories of the robot for contextual behavior.

rewards. As an example, in Figure 6 we see the trajectories followed by the robot with the AC and CC systems when the internal motivation of the robot is to reach the reward 3. The AC behavior guides the robot to a wrong path in some occasions. The CC system trajectories are more accurate than the AC system ones.

5 Conclusions

We investigated how allostatic and contextual control systems complement each other. In order to do that, we extended the DAC architecture. On the one hand, we equipped its reactive layer with an allostatic control capability. On the other hand, we extended the memory content allowing the storage and retrieval of information related to different goal states. Furthermore, we integrated both systems converting the egocentric actions from the allostatic control system in allocentric ones for the contextual control system. We tested the model in a variety of foraging tasks. We show that allostatic control system is able to succeed fulfilling the robot's needs when the information from the reward does not contain conflicts. When obstacles in the environment do not allow a direct path to the reward a contextual control system is needed. Our results show that in these cases the contextual control system is able to pick up salient information from the environment and use it to fulfill the robot's needs.

The model integration proposed here can be reminiscent of taxon Vs. route navigation strategies [12]. On the one hand, the allostatic control system is capable of solving taxon task, i. e. tasks where the goal is visible. On the other hand, as the complexity of the task increases, i.e. the goal is not visible, the contextual control system is capable of solving it through route navigation, i. e. chaining taxon strategies. A number of robotics and artificial intelligence algorithms have been proposed to solve the taxon chaining problem [13,14]. Our approach differs from them in that our first building block is self-regulation and on top of it we add a contextual control system able to learn about the environment. Different internal states leading to different goal states can be handled achieving self-regulation.

Further experiments would need to be done in order to better understand how allostatic and contextual systems interact. For instance, the possibility of

multiple homeostatic subsystems activated at the same time might influence the information learned by the contextual control system. The implications of this influence in the memory content should be tested.

To integrate allostatic and contextual systems in a model allow us to better understand how they complement each other and how they scale with task complexity. Moreover, we propose that these implications might be extended to the biological brain and its multi-level architecture.

References

1. Drai, D., Benjamini, Y., Golani, I.: Statistical discrimination of natural modes of motion in rat exploratory behavior. Journal of Neuroscience Methods 96, 119–131 (2000)
2. McEwena, B.S., Wingfield, J.C.: The concept of allostasis in biology and biomedicine. Hormones and Behavior 2, 2–15 (2003)
3. Meyer, J.-A., Guillot, A., Girard, B., Khamassi, M., Pirim, P., Berthoz, A.: The psikharpax project: towards building an artificial rat. Robotics and Autonomous Systems 50(4), 211–223 (2005)
4. Sheynikhovich, D., Chavarriaga, R., Strosslin, T., Arleo, A., Gerstner, W.: Is there a geometric module for spatial orientation? insights from a rodent navigation model. Psychological Review 116(3), 540–566 (2009)
5. Verschure, P., Althaus, P.: A real-world rational agent: unifying old and new ai. Cognitive Science 27, 561–590 (2003)
6. Verschure, P., Voegtlin, T., Douglas, R.J.: Environmentally mediated synergy between perception and behavior robots. Nature 425, 620–624 (2003)
7. Duff, A., Rennó Costa, C., Marcos, E., Luvizotto, A., Giovannucci, A., Sánchez Fibla, M., Bernardet, U., Verschure, P.: Distributed adaptive control: A proposal on the neuronal organization of adaptive goal oriented behavior. In: Sigaud, O., Peters, J. (eds.) From Motor Learning to Interaction Learning in Robots (2010)
8. Iizuka, H., Di Paolo, E.A.: Extended homeostatic adaptation: Improving the link between internal and behavioural stability. pp. 1–11 (2008)
9. Wyss, R., König, P., Verschure, P.: A model of the ventral visual system based on temporal stability and local memory. PLoS Biol. 4 (2006)
10. Tolman, E., Honzik, C.: Insights in rats. University of California Publications in Psychology 4(14), 215–232 (1930)
11. Hartland, C., Bredechem, N., Sebag, M.: Memory-enhanced evolutionary robotics: the echo state network approach. In: Proceedings of the Eleventh Conference on Congress on Evolutionary Computation, pp. 2788–2795 (2009)
12. Redish, A.D. (ed.): Beyond the Cognitive Map: From Place Cells to Episodic Memory. MIT Press, Cambridge (1999)
13. Kuipers, B.: The spatial semantic hierarchy. Artificial Intelligence, 191–233 (2000)
14. Mallot, H.A., Basten, K.: Embodied spatial cognition: Biological and artificial systems (Cognitive Systems: Perception, Action, Learning). Image and Vision Computing 27(11), 1658–1670 (2009)

Path Integration Working Memory for Multi Tasks Dead Reckoning and Visual Navigation

Cyril Hasson and Philippe Gaussier

Université de Cergy-Pontoise, CNRS, ENSEA
ETIS laboratory UMR 8051
F-95000 Cergy Cedex, France

Abstract. Biologically inspired models for navigation use mechanisms like path integration or sensori-motor learning. This paper describes the use of a proprioceptive working memory to give path integration the potential to store several goals. Then we coupled the path integration working memory to place cell sensori-motor learning to test the potential autonomy this gives to the robot. This navigation architecture intends to combine the benefits of both strategies in order to overcome their drawbacks. The robot uses a low level motivational system based on a simulated physiology. Experimental evaluation is done with a robot in a real environment performing a multi goal navigation task.

1 Introduction

Researchs in the field of navigation robotics have used biologicaly inspired mechanisms like path integration based on odometric information [1,2,3] (return vector computing) or sensori-motor learning based on place cell recognition [4,5,6,7,8]. These navigation strategies are very good solutions to homing problems. Thus, the robot must look for different resources to fullfill its various needs. Path integration doesn't need learning, but, by itself, it is not able to store several goals. And errors, coming from measure imprecision, cumulate to the point where it is too inaccurate to allow the robot to find its goal. Sensori-motor learning is robust but needs learning (generaly man supervised). However, study about insects navigation [9,10] have shown their ability to manage several homing vectors allowing them to go back to a secondary goal when their first one is not available (indicating a memory).

In this paper, we describe a proprioceptive working memory giving path integration abilities the potential to allow robots to display the same kind of behaviors. The robot control architecture use a low level motivational, or *drive* system that reacts to the simulated physiological state and computes the different drives levels (hunger and thirst). To satisfy its drives the robot has to gather the corresponding simulated resource (food and water). From the principles presented in [1], we have designed a proprioceptive navigation strategy using this working memory to store several goals on several path integration neural fields. The robot uses hebbian learning to associate each goal to the corresponding drive.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 380–389, 2010.

Furthermore we then coupled place cell sensori-motor learning to the path integration working memory and the drive system. Place-drive-action associations are used to autonomously build a visual attraction bassin around each goal and allowing to bypass the limitations of path integration when the robot is lost or has been kidnapped.

Section 2 describres the proprioceptive navigation architecture we used. The visual place cell architecture is described in section 3. Section 4 shows experimental results with the robot. And section 5 contains the conclusions. Figure 1 shows the robot and its environment.

Fig. 1. The robot in its environment (equiped with a color detector placed under it). Colored squares on the ground are simulated resources.

2 Proprioceptive Navigation

Path integration is the ability to use proprioceptive information about the movements being done in order to determine the direct movement to any given interesting point of the robot trajectory in the environment. Principles of path integration using dynamical neural fields are described in [1]. Figure 2 is an illustrated example of this computation. This strategy is said to be autonomous because the robot can learn and use it without any supervision. Using Dynamical neural field to represent information in the path integration process allows to use directly the output (the return vector) as the control signal for the robot rotational speed. In order to use path integration to build navigation abilities able to succeed to classical multigoal survival tasks (as presented in introduction), the robot has to be able to come back to several interesting places of its environment (vital resources locations). Thus, instead of only one integration field, the robot must dispose of several integration fields. But because the number of path integration field has to be limited (defining a more realistic kind of working memory), the robot cannot simply recruit another integration field each times it detects a resource. The number of parallel integration fields (nb_{goals}) is a representation of the system working memory span i.e. the number of elements that can be maintained in working memory). Activity of each neuron of these parallel fields at time t is $P_{ij}(t)$:

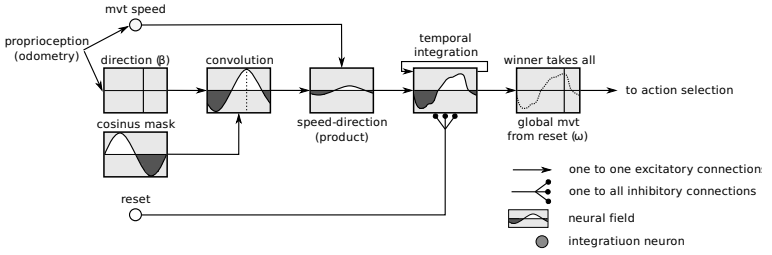


Fig. 2. Path integration : speed is coded as the activity of one neuron and orientation as the most active neuron of a neural field. At every time step, the integrator takes as input the activity of the orientation neural fiels (convoluted by a cosine shape) multiplied by the activity of the speed neuron. This input represents the orientation and distance traveled since the last time step. Summing this input with its own activity, the integration neural field computes the return vector.

$$P_{ij}(t) = \sum_{t_{rj}}^{t} (S(t) \ . \ cos(\frac{d_w(t) - i}{n})) \ . \ (1 - r_j(t))$$

n is the size of the neural fields, $i \in [1 : \mathrm{n}]$, j is the number of path integration fields ($j \in [1 : nb_{goals}]$), t_{rj} is t at the last field j reset, $S(t)$ is the activity of the speed coding neuron at time t, $d_w(t)$ is the active direction neuron position in the field at time t and $r_j(t)$ is the reset signal for the field j at time t (1 during reset, 0 otherwise). A cosine function has been used, but can be replaced by any bell curve activity i.e. (a gaussian).

When the robot finds a new resource, it must be able to recruit a new integration field. And when it is motivated by its simulated physiological needs, the robot must be able to select among the different integration fields the best to lead it to the desired resource. We will describe how this can be done using a modified version of the simulated neural networks used for simple path integration.

New goal / known goal discrimination

To exploit optimally this multiple field path integration architecture, it is important that the robot discriminates new resource locations from known ones. Every time a resource is detected, one of the integration fields must be reset (all neurons in the field have a null activity) : a new integration field when it is a new resource (recruitment) and the corresponding integration field when it is a known resource (recognition). To discriminate new from known goals, we use the distance coding property of the integration fields. The neural field maximum activity is proportional to the goal distance. A group of neurons coding for goals proximity (size = nb_{goals}) receives activation from a constant input and each of the neurons is inhibitied by its corresponding positive activity in the integration field. As the robot gets closer to a known goal, activity of the corresponding goals proximity neuron gets higher. If we use neurons with a non-linear transfert function (here a simple thresold just below 1), a goal proximity neuron will only be active when a known resource is near. This activity could be seen as a goal

prediction or expectation. Activity of each goal prediction neuron at time t is Goal Prediction$_j(t)$:

$$Goal\ Prediction_j(t) = \begin{cases} 1 & if\ (1\ -\ w_{j'j}\ .\ \sum_{i=1}^{i=n} |(P_{ij}(t))|)\ >\ T \\ 0 & otherwise \end{cases}$$

$j \in [1 : nb_{goals}]$, $j' \in [1 : nb_{goals}]$, $w_{j'j}$ is the weight of the path integration field$_{j'}$ - $goal\ proximity_j$ (small negative value), 1 is the constant input, n is the size of the neural integration fields and T is a definite threshold of the form (1 - ϵ).

Figure 3 shows how this goal prediction is used to discriminate new from known goals. Resource detection both activates the new goal and the known goal neurons but goal prediction neurons inhibit the new goal neuron and the new goal neuron inhibits the know goal neuron. Thus when a resource is detected, if no goal prediction is made, this resource is considered as a new goal (a known goal otherwise).

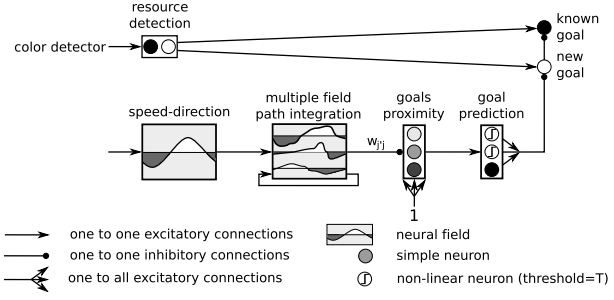


Fig. 3. New goal/known goal discrimination. As the robot gets closer to a known goal, the corresponding goal proximity neuron activity gets closer to 1. As shown here, above a definite threshold (T = 1 - ϵ), a goal prediction is made and resource detection will then be considered as detection of a known goal rather than of a new goal.

Integration field recruitment and goal recognition

When a new goal is detected, a new integration field must be recruited. This is done by resetting one of the integration fields. Figure 4 (upper part) shows how fields to be recruited are selected.The main idea is to take an unused field or at least the field associated to the least used goal. The "most used goals" group of neurons (size = nb_{goals}) receives one to one connections from the recruitment reset group of neurons (same size) and has recurrent one to one connections with a weight slightly under 1. Each time a new integration field is recruited, the corresponding "most used goals" neuron receives activation and the reccurent connections act as a decay function. Thus, the neuron of the "most used goal" group corresponding to a newly recruited field will be more active than the one of an old goal. The "less used goal" group of neurons is a winner takes all groupe of neurons that receives a constant activation input (one to all connections) and is inhibited by the "most used goals" group of neurons (one to one connections).

Its single active neuron thus corresponds to the integration field to inhibit when a new goal is detected. The recruitment reset group simply makes the product of the "less used goal" activities (one to one connections) and the new goal detection neuron activity (one to all connections). Only one neuron of the recruitment reset group can be active at a given time, and only when a new goal is detected. Each of its neurons inhibits an entire field of the multiple path integration fields group. When a known goal is detected, the corresponding integration field should have no activity. However, because the resources are represented by square surfaces the robot might detect a known resource from a position different from the reset point and a little activity might still be found on the corresponding integration field. Furthermore, a residual activity on the field might be caused by integrations errors due to discretisation or even by the sliding of the robot wheels on the floor. To avoid the cumulative effect of these errors, when a known goal is recognized, the corresponding integration field is also reset inducing a recalibration effect similar to [6]. Figure 4 (lower part) shows how fields to be reset because of goal recognition are selected. The nearest goal group is a winner takes all group of neurons that receives one to one connections from the goals proximity group. The only active neuron corresponds to the closest goal. The recognition reset group simply makes the product of the nearest goal group activities (one to one connections) and of the known goal detection neuron activity (one to all connections). Only one neuron of the recognition reset group can be active at a given time and only when a known goal is detected. Each of its neurons inhibits an entire field of the multiple path integration fields group. The recognition reset group projects one to one activation connections to the most used goals group. A goal is thus considered as used when it is recruited as well as when it is recognized.

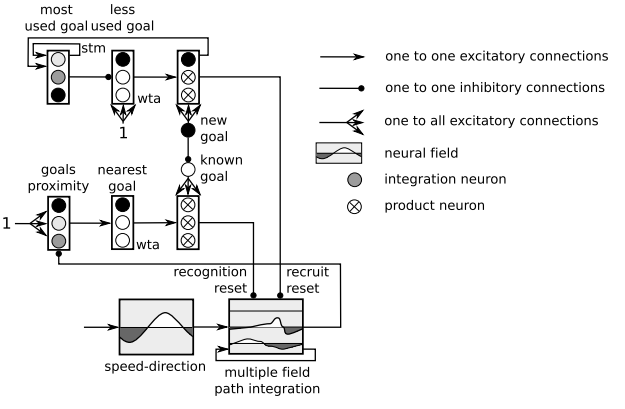


Fig. 4. Integration field recruitment and goal recognition. As shown in this example, when a new goal is detected, the integration field corresponding to the less used goal is reset (recruitment). When a know goal is recognized, the integration field corresponding to the nearest known goal is reset (recognition).

Goals competition and integration field selection
Once several integrations field have been recruited, it is important to be able to select the right one to reach the desired resource. The architecture can only work if it is able to learn the association between the goals (and their corresponding integration fields) and the drive they satisfy. Furthermore, one resource can be present in several different locations of the environment. It is then necessary to select one of these locations (e.g. according to their distances). Figure 5 shows the neural network used to achieve this. The goal-drive association group receives one to one connections from the recruit reset group and the recognition reset group and one to all plastic connections from the active drive group. The one to all plastic connexions from the active drive group will then adapt their wieghts (conditional connexions) according to the recruit and recognition reset activity (unconditional connexions). Following its hebbian learning rule, weights of the plastic connections will adapt so that when a drive is active, the goal-drive association group will have activity on the neurons corresponding the goals that satisfy this drive. Every time a goal is detected, the corresponding goal-drive association is reinforced. Activity of each goal-drive neuron at time t is $GD_i(t)$:

$$GD_i(t) = \sum_{1}^{j} D_j(t) \ . \ w_{ji}(t)$$

Weights adaptation :

$$\Delta w_{ji}(t) = \lambda(t) \ . \ (recruitR_i(t) \ + \ recogR_i(t)) \ . \ D_j(t)$$

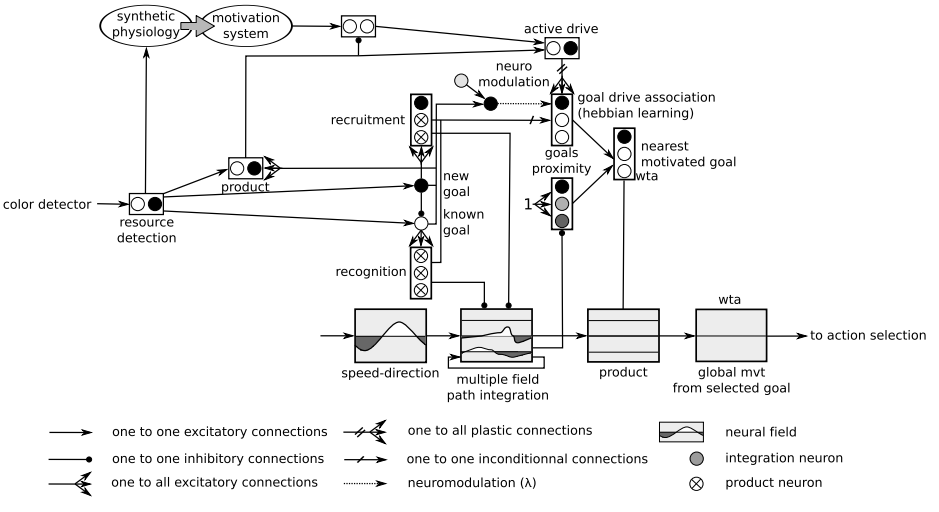


Fig. 5. Goals competition and integration field selection. The goal drive association group learns which goal satisfies which drive and the nearest motivated goal group select the nearest motivated goal. The corresponding integration field (and thus the corresponding action) is selected via a neuronal matricial product.

i is the working memory span size (nb_{goals}), j is the number of drive, $D_j(t)$ is the active drive neuron j activity at time t, $w_{ji}(t)$ is the weight of the D_j - GD_i connexion at time t, $\lambda(t)$ is the learning rate at time t, $recruitR_i(t)$ and $recogR_i(t)$ are the recruitment reset and recognition reset signals for goal i at time t (reset of a neural field when a new goal and a know goal are detected). Neuromodulation of this hebbian learning group of neurons (λ) is high when a goal is detected allowing fast learning of the goal-drive association. It is low otherwise, letting slowly forget the drive associations of goals that could satisfy the active drive but are not detected).

To take into account selection by the drive as well as selection by goal distance, the nearest motivated goal group of neurons receives one to one activations connections from the goal-drive association as well as from the goal distance group and sums its inputs. Using the winner takes all rule, its single active neuron corresponds to the closest goal which satisfy the active drive. Selection of the corresponding integration field is done by a matrix product between the multiple integration fields group and the nearest motivated group. This matrix product is done in two steps. First, a group of neurons the same size as the multiple integration fields group makes the product between the activity it receives from the multiple integration fields group (one to one connections) and from the nearest motivated goal group (one to a field connections i.e. horizontal projections). The field corresponding to the selected goal is the only one to sustain activity (the other fields have an activity which is product by 0). Finally, this almost empty neurons matrix is projected through vertical connections (vertical projection) to a group with a single field of neurons which can then be used just like in the simple path integration model to compute the direction of the return vector.

Path integration benefits and drawbacks
Because the robot only needs to detect a resource once to be able to store and compute dynamically its return vector, the actions that lead to the goal don't need to be learned. When it is not motivated, the robot explores randomly its environment. If a drive is high enough, it will then be able to reach the resources locations it has discovered. However, path integration has a major drawback. It is not precise over long periodes of time. Cumulative errors come from the direction discretization and from the slidings of the robot wheels on the floor. Studies of path integration on different animals [11] have shown that these cumulative errors are structural limitations.

3 Visual Navigation

The visual system is able to learn to characterize (and thus recognize) different "places" of the environment. Inspired by visual navigation models issued from neurobiology [12], the visual system, a simulated neural network, learns place cells. Each place cell codes information about a constellation of local views (visual cues) and their azimuths from a specific place in that environment [4,13]. Activities of the different place cells depend on the recognition levels of these

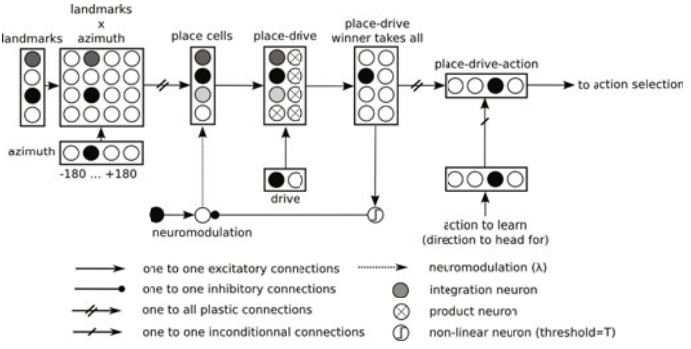


Fig. 6. Sensorimotor visual navigation : a visual place cell is constructed from recognition of a specific landmarks-azimuths pattern and an action is associated with this place cell. The action to learn is usually given through supervised learning.

visual cues and of their locations. A place cell will then be more and more active as the robot gets closer to its learning location. The area where a given place cell is the more active is called its place field. When the maximum recognition level of place cells is below a given threshold T, another place is learned. The higher is T, the more place cells are learned in a given environment. An associative learning group of neurons allows sensorimotor learning (the place-drive-action group on figure 6). Place-drive neurons are associated with the return vector of the corresponding goals to autonomously build a visual attraction bassin around each goal. Figure 6 shows how this navigation strategy works and how it allows different responses according to the active drive. The main advantage of this navigation strategy over path integration is that it is not sensitive to cumulative errors. Learning precision can be maintained over long periodes of time. However, this strategy is long to learn autonomously. [6,7] have used TD lambda and Q learning to learn by trial and errors to reach the goal. But this learning process is long (at least compared to the immediat usability of path integration).

4 Robotic Experiments

The main goal of this proprio-visual navigation architecture is to take advantage of both strategies in order to be immediately functional (path integration benefit) and robust over time (sensorimotor learning benefit). Furthermore, the coupling between these two strategies allows to quickly learn and stabilize the sensorimotor associations. The task is a multiple resource problem : the robot needs two different resources (water and food) and each resource is present in two different places of the environment (see figure 1). In the first experiment, we only used the proprioceptive strategy to learn the task. Figure 7 shows the robot trajectories after learning is made (the resource are discovered by random navigation).

In the second experiment, both strategies were coupled in order for the visual strategy to use information coming from the proprioceptive strategy. During

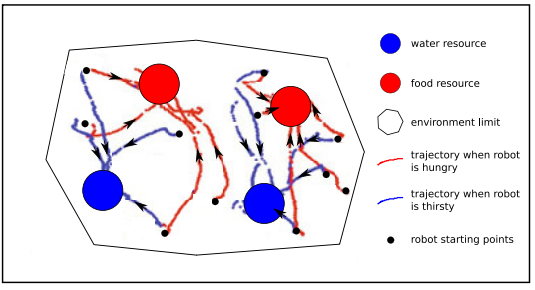


Fig. 7. Proprioceptive navigation trajectories : when motivated, the robot heads for the closest corresponding resource

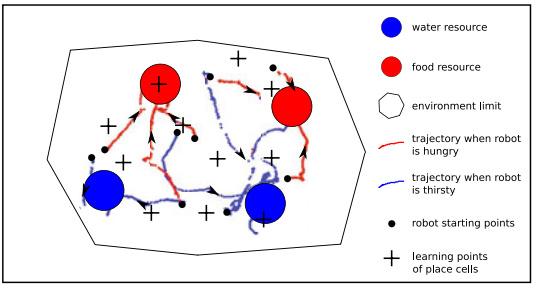


Fig. 8. Visual navigation trajectories : after 10 minutes of proprioceptive navigation, visual navigation has learned enough to produce a converging behavior. When motivated, the robot heads for the corresponding resource.

latent place-action learning, the visual strategy associates place cell recognition to the return vector computed by the proprioceptive strategy. Figure 8 shows the robot trajectories using the visual strategy coupled to path integration (10 minutes of proprioceptive navigation).

5 Conclusion and Perspective

Complementary aspects of a motor working memory based on neural field associated to visual place recognition give a quick access to autonomous abilities for navigation. Proprioceptive navigation using a working memory doesn't need to be learned. The actions that lead to the goal are the result of a computation and not of a learning process. But it is not robust and unless the robot constantly navigate beetween short distance goals, it will inevitably become less and less precise until it is not usable anymore. Visual strategy has the strong advantage of being robust over time allowing the recalibration of the path integration fields. The coupling of these two strategies allows to boostrap learning using the proprioceptive strategy and then to use the proprioceptive strategy output (an

action) as input for the visual strategy sensori-motor learning. This coupling allows a rapid learning of the stable sensorimotor associations.

However, further developments of this navigation architecture should be focused on the need to design a mechanism to select which strategy has to be used. This selection mechanism could rely on very different but nonetheless equally important parameters. The strategy to use could be selected according to its propension to satisfy the underlying drive (or motivation). In other words, a frustration mechanism based on prediction of the drive satisfaction could be very efficient to regulate autonomously the strategy to use and thus the robot behavior. Furthermore, each strategy being based on distinct information sources, strategy selection could rely on perceptive context.

References

1. Gaussier, P., Banquet, J.P., Sargolini, F., Giovannangeli, C., Save, E., Pousset, B.: A model of grid cells involving extra hippocampal path integration, and the hippocampal loop. Journal of Integrative Neuroscience 6(3), 447–476 (2007)
2. Vickerstaff, R.J., Paolo, E.A.D.: Evolving neural models of path integration. Journal of Experimental Biology (208), 3349–3366 (2005)
3. Mittelstadt, M., Mittelstadt, H.: Homing by path integration in a mammal. Naturwissenschaften 67(11), 566–567 (1980)
4. Gaussier, P., Zrehen, S.: Perac: A neural architecture to control artificial animals. Robotics and Autonomous System 16(2-4), 275 (1995)
5. Gaussier, P., Joulain, C., Banquet, J., Leprtre, S., Revel, A.: The visual homing problem: an example of robotics/biology cross fertilization. Robotics and Autonomous System 30(1), 155–180 (2000)
6. Arleo, A., Gerstner, W.: Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. Biological Cybernetics 83(3), 287–299 (2000)
7. Foster, D., Morris, R., Dayan, P.: Models of hippocampally dependent navigation using the temporal difference learning rule. Hippocampus 10, 369–379 (2000)
8. Wiener, S., Berthoz, A., Zugaro, M.: Multisensory processing for the elaboration of place and head direction responses in the limbic system. Cognitive Brain Research 14(1), 75–90 (2002)
9. Gallistel, C.: Symbolic processes in the brain: The case of insect navigation. In: An Invitation to Cognitive Science: Methods, Models, and Conceptual Issues. The MIT Press, Cambridge (1998)
10. Collett, M., Collett, S., Srinivasan, M.: Insect navigation: Measuring travel distance across ground and through air. Current Biology 16(20) (2006)
11. Etienne, A., Maurer, R., Seguinot, V.: Path integration in mammals and its interaction with visual landmarks. Journal of Experimental Biology 199(1), 201–209 (1996)
12. O'Keefe, J., Nadel, L.: The Hippocampus as a Cognitive Map. Oxford University Press, Oxford (1978)
13. Gaussier, P., Lepretre, S., Quoy, M., Revel, A., Joulain, C., Banquet, J.: Experiments and models about cognitive map learning for motivated navigation. In: Interdisciplinary Approaches to Robot Learning. Robotics and Intelligent Systems Series, vol. 24, pp. 53–94. World Scientific, Singapore (2000)

Minimal Model of Strategy Switching in the Plus-Maze Navigation Task

Denis Sheynikhovich[1], Laurent Dollé[2], Ricardo Chavarriaga[3], and Angelo Arleo[4]

[1] Laboratoire de Neurobiologie des Processus Adaptatifs UPMC-Paris 6 CNRS UMR 7102, 9 quai St. Bernard, 75005 Paris-France
[2] Institut des Systèmes Intelligents et de Robotique UPMC-Paris 6 CNRS UMR 7222, 4 place Jussieu, 75252 Paris cedex 05-France
[3] CNBI-EPFL, CH-1015 Lausanne, Switzerland

Abstract. Prefrontal cortex (PFC) has been implicated in the ability to switch behavioral strategies in response to changes in reward contingencies. A recent experimental study has shown that separate subpopulations of neurons in the prefrontal cortex were activated when rats switched between allocentric place strategies and egocentric response strategies in the plus maze. In this paper we propose a simple neural-network model of strategy switching, in which the learning of the two strategies as well as learning to select between those strategies is governed by the same temporal-difference (TD) learning algorithm. We show that the model reproduces the experimental data on both behavioral and neural levels. On the basis of our results we derive testable prediction concerning a spatial dynamics of the phasic dopamine signal in the PFC, which is thought to encode reward-prediction error in the TD-learning theory.

1 Introduction

The ability to switch between different navigational strategies for reaching the goal is crucial for adaptive spatial behavior. Large body of animal studies suggest a large variety of navigational strategies, which depend on sensory input of different modalities [1]. Existing lines of experimental research focus on *(i)* how the different strategies are implemented in the brain and what memory systems support them [2,3,4,5]; and *(ii)* what is the mechanism of selection between different strategies and corresponding memory systems during ongoing behavior [6,7,8]. In particular, a series of animal studies in the plus maze (Fig. 1A) provided an insight into the role of hippocampus (HPC), dorsolateral striatum (DLS) and prefrontal cortex (PFC) in learning the tasks in which changing reward contingency forced the animals to use either a place strategy or a response strategy. In the plus maze, the place strategies can be defined as approaching a particular place associated with reward, e.g. the end of the East arm in the case of strategy 'go East' (Fig. 1C). The response strategies can be defined as making a particular egocentric turn at the center, e.g. 'turn left' strategy (Fig. 1D).

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 390–401, 2010.

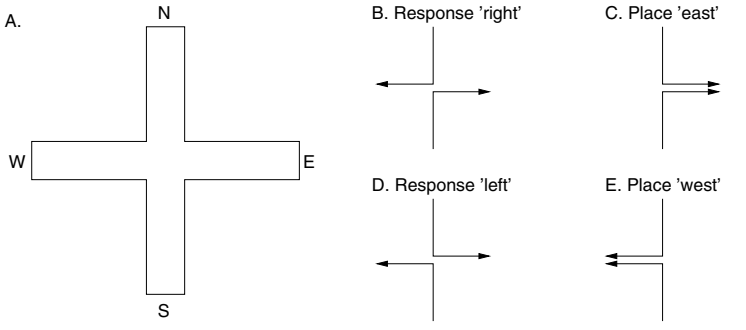


Fig. 1. Experimental setup in the plus-maze task [8]. A. Plus maze. B-E. Learning tasks testing response strategies (B,D) and place strategies (C,E). In (B,C) and (D,E) lower paths are consistent, while upper paths are inconsistent.

Note that if animals start their trial in the North position, both strategies lead to the same location. In contrast, the two strategies lead to different locations if the starting position is South, permitting the dissociation between the two strategies.

Experimental studies in the plus maze have shown that rats with inactivated HPC were strongly biased towards response strategies, while they were biased towards place strategies when DLS was inactivated [4,5]. In other experimental environments a double dissociation between HPC and DLS has been demonstrated in place-learning and response-learning tasks, respectively [2,3]. Moreover, recent electrophysiological recordings of PFC neurons while rats were navigating in the plus maze have demonstrated that different neuronal populations were active depending on which strategy is currently in use [8]. When rats had learned that the correct strategy was a place-based one, a subpopulation of neurons were highly active. When the reward contingency changed so that the response-based behavior was the only valid strategy, another subpopulation became active, while neurons in the first subpopulation became silent. In addition to being a suitable experimental setup for studying strategy learning and switching, the plus maze is also attractive from the modeling point of view. This is because the analysis of strategy switching and its neural correlates can be performed at a well defined location (the cross point) where behavior is crucial for successful performance (in contrast to, e.g., water maze where no such location can be identified).

The experimental data reviewed above suggest a two-level hierarchical organization of spatial behavior in the plus maze. On the first level, different neuronal networks (located in e.g. HPC or DLS) learn corresponding navigational strategies taking a particular sensory modality as an input. On the second level, a neural network (located in the PFC) learns to choose which strategy is more successful and gives the control of the behavior to that strategy [9]. In a simple way, a behavioral strategy can be considered as a rule that determines which of the available motor actions should be performed when a particular sensory input is observed. Similarly, the strategy switching can be described as the choice

of one of the available strategies, given the sensory input. From a theoretical point of view, these two learning tasks are identical, assuming that available motor actions (first level) or available strategies (second level) are encoded by the activities of neural populations.

Here we were interested to see how far we can go in explaining the behavioral and neural data by suggesting the learning algorithm is identical on both levels. The learning algorithm we use is a standard temporal-difference (TD) learning rule called Q-learning [10]. At the core of TD-learning algorithms is the minimization of a so-called reward-prediction error which is proposed to be coded by the phasic activity of dopaminergic (DA) neurons in the ventral tegmental area (VTA) [11]. We have analyzed the learning dynamics of the reward-prediction error in our model and compared the error propagation on the two levels. Our results suggest that a two-level organization of behavioral control with TD-learning on each level is compatible with principal experimental results [4,8]. In addition, our model generated testable predictions concerning differences in DA signaling between HPC and PFC.

2 Model

Our neural network model of navigation consists of three interconnected subnetworks responsible for response strategy, place strategy and strategy selection, respectively (Fig. 2). The response-strategy subnetwork learns egocentric movements such as, e.g., turning left, while the place-strategy subnetwork learns to enter a particular arm. The strategy-selection network learns to choose which strategy will take control of behavior on each time step. In our model, the three subnetworks functionally correspond to different neural structures in the rat brain that are implicated in spatial learning. Namely, the response subnetwork corresponds to the DLS, the place network corresponds to the HPC, while the selection network corresponds to the PFC.

Each of the three subnetworks has the same simple architecture of a single-layer perceptron in which learning occurs according to a TD learning rule. The input layer encodes sensory input to the network, while the output layer represents actions that this network generates. The subnetworks are hierarchically organized in two levels: the two subnetworks of the lower level learn the place and response strategies, while a single subnetwork of the upper level (strategy selection network) learns to select which of the two subnetworks of the lower level will take control over behavior. The selection is assumed to occur by inhibiting the motor output of the non-winner strategy. Below we describe the inputs, outputs, and learning in the different subnetworks.

2.1 Place Learning

Place strategy was implemented as follows. Input to the network was represented by the activities of N_{pc} place cells with Gaussian spatial receptive fields, centers of which were distributed uniformly over the environment. More precisely, the

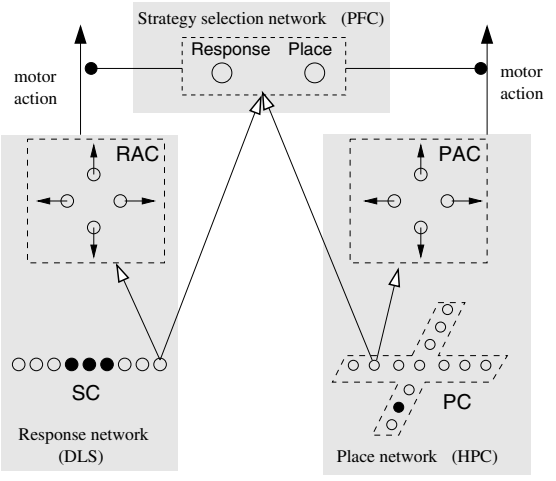


Fig. 2. The architecture of the model. SC – sensory cells, PC – place cells, RAC – response-action cells, PAC – place-action cells. Open (filled) circles denote inactive (active) cells; the large open arrows denote all to all feed-forward projections between corresponding neural structures; the large filled arrows denote motor output of the model; small filled arrows denote preferred direction of the action cells in these structures; ball-arrows denote inhibition.

activity of place cell j was calculated as $r_j^{\mathrm{P}} = \exp(-d_j^2/2\sigma^2)$, where σ is the width of animal in the maze and the center of the j-th receptive field. Place cells projected to place-action cells a_i^{P} with weights w_{ij}^{P}. Four place-action cells coded for movements in allocentric directions East, West, North and South. Activities of the place-action cells were calculated according to Eq. 1, and the weights were updated using Eq. 2 on each time step. An allocentric movement, proposed to be performed on the next time step by the place strategy, was chosen on the basis of response-action cell activities according to Eq. 4 on each time step.

2.2 Response Learning

In the response strategy network, input cells were N_{sc} sensory cells that coded for the presence of walls around the simulated rat. All sensory cells were divided into four subpopulations coding for the space in front of the rat, left from the rat, right from the rat and behind the rat, respectively. Presence or absence of a wall at a specified egocentric direction from the rat (and hence the ability to move in this direction) was encoded by setting activities r_j^{sc} of the corresponding sensory cells to 0 or 1, respectively. Sensory cells projected to response-action cells a_i^{R} with weights w_{ij}^{R}. Four response-action cells coded for movements in egocentric directions Forward, Leftward, Rightward and Backward. Activities of the response-action cells were calculated according to Eq. 1, and weights were updated using Eq. 2 on each time step. An egocentric movement, proposed to be

performed on the next time step by the place strategy, was chosen on the basis of response-action cell activities according to Eq. 4 on each time step.

2.3 Strategy-Selection Learning

In the strategy selection network, input was represented by the combined activities of $N_{\mathrm{pc}} + N_{\mathrm{sc}}$ input cells from place and response strategy networks. An intuition for this particular choice of input is the following: given all available information at the current state, the selection network decides which strategy should be chosen to take control over behavior. Input cells projected to strategy-selective (action) cells a_i^{S} with weights w_{ij}^{S}. Two strategy-selective cells coded for place and response strategies, respectively. Activities of the strategy-selective cells were calculated according to Eq. 1, and the weights were updated using Eq. 2 on each time step. One of the strategies (the winner strategy) was selected on the basis of activities of the strategy-selective cells according to Eq. 4 on each time step. The next movement actually performed by the simulated rat was the one proposed by the winner strategy.

2.4 Experimental Setup and Simulation Procedure

All simulations were performed in a simulated plus-maze environment (Fig. 1A) with length L between the ends of the opposite arms (in arbitrary units, see all parameter values in Table 1). All results were averaged over 100 simulated rats. The noise in the model comes from (i) pseudo-random choice in starting position and (ii) stochastic action choice, see Section 2.5 below.

At the start of a trial, each simulated rat was put in one of the starting positions (either N or S in Fig. 1A) and allowed to reach one of the goal arms (either E or W in Fig. 1A). Starting positions were chosen in a pseudo random order, while the goal position on each trial was chosen depending on the reward contingency (see Fig. 1B). For example, if the task was 'response right' and the starting position was S, then the goal position was set to be E. If a simulated rat started from position S (N), the entrance to northern (southern) arm from the center was blocked.

The simulated rats were tested in either strategy switches or strategy reversals. During strategy switches, different simulated rats were first trained in either place or response task (see Fig. 1B). After 200 trials of training the task was changed to the task of the other type (if it was a response task in the first phase, it changed to a place task for the second phase, and vice versa). During strategy reversals, the first phase of training was the same as during strategy switches; however, in the second phase the task was of the same type but with a different reward contingency (if it was, e.g., a 'response left' task in the first phase, it changed to 'response right' for the second phase).

Upon reaching the goal arm, the rat was given reward $R_t = R$ (see Eq. 3 below and Table 1) and a new trial started. All other actions resulted in $R_t = 0$. An attempt to backtrack, or reaching the arm opposite to the goal arm resulted in starting a new trial (backtracks were not counted as trials).

2.5 Learning Equations

Basic learning equations implemented a standard Q-learning algorithm [10] in which states and actions were encoded in the firing rates of activities of artificial neurons [12,13]. More precisely, a state s_t at time t was represented by the activities of input units $r_i^{\text{state}}(t)$, which projected via connections with weights w_{ij} to action units with activities r_i^{action}:

$$r_i^{\text{action}}(t) = \sum_j w_{ij} r_j^{\text{state}}(t) \tag{1}$$

Each action unit i represented an action a_i available in the state s_t. Activity of an action cell (Eq. 1) was interpreted as the Q-value $Q(s_t, a_t)$ of performing action a_t at state s_t. In the present notation, symbol a_i denotes i-th action available at the current state, while a_t denotes the action actually chosen at time t from the available actions, thus $\forall t\, \exists i : a_t = a_i$.

During learning, weights w_{ij} were adjusted on each time step according to a TD-learning rule:

$$\Delta w_{ij}(t) = \eta \delta(t) e_{ij}(t) \tag{2}$$

where η is the learning rate,

$$\delta(t) = R_t + \gamma Q^*(s_t, a_t) - Q(s_{t-1}, a_{t-1}) \tag{3}$$

is the reward prediction error and $e_{ij}(t)$ is the eligibility trace that represents the memory of past actions. The eligibility trace of a synapse [10] was increased each time the synapse had participated in generating a movement, i.e. $e_{ij}(t+1) = e_{ij}(t) + r_j^{\text{state}}$ if action a_i was performed at time t. All eligibility traces decayed with time according to $e_{ij}(t+1) = \gamma\lambda e_{ij}(t)$, where $0 < \gamma, \lambda < 1$. In Eq. 3, $Q^*(s_t, a_t) = \max_i Q(s_t, a_i)$ is the Q-value of the optimal action at state s_t.

Outcomes of different actions at different states need to be explored in order to learn action values. Exploration was ensured by choosing actions stochastically in each state using softmax algorithm, i.e. the probability of choosing action a_i at time t was calculated according to

$$p(a_t = a_i) = \exp(\beta a_i) / \sum_i \exp(\beta a_i) \tag{4}$$

Table 1. Model parameters

Maze length L	7.0
Size of input populations N_{pc}, N_{ac}	13
Q-learning parameters γ, λ	0.9
Reward for reaching the goal R	10.0
Learning rate η	0.05
Softmax parameter β (selection network/strategy networks)	1.0 / 4.0
Place field width σ	0.4

with β as a constant parameter. We also tested ϵ-greedy criterion for action selection, in which an optimal action is chosen with probability 1-ϵ and a random action is chosen otherwise. Results were similar to those reported here (we used $\epsilon = 0.1$).

3 Results

3.1 Learning Switches and Reversals

Learning performance was estimated by calculating a mean number of failures across training trials, for strategy switches and for reversals (Fig. 3). An outcome of a trial was defined as a failure if the simulated rat reached the arm opposite to the goal arm at the end of the trial, and a success when the rewarded arm was reached. Criterion of 80% of correct trials in the last 40 trials [8] was reached within less than 100 trials for switches and reversals (red solid lines in Fig. 3), similarly to rats. For the purposes of subsequent analysis, we defined a 'before' phase as the period between the trial in which the criterion was reached for the first time and the trial where task contingencies changed; an 'after' phase corresponded to the period after the trial when the criterion was reached after the contingency change and until the end of training (see Fig. 3A).

3.2 Activity of Modeled Prefrontal Neurons during Behavior

Next, we analyzed the activity of strategy-selective neurons during behavior. The focus was on the activity at the choice point of the simulated maze, since only at this position a difference in the activities is crucial for successful performance. During switches from a response to a place strategy, the response-strategy neuron was significantly more active than the place-strategy neuron during 'before' phase, while the activity pattern reversed during the 'after' phase (Fig. 4A). In the model, this is a consequence of the fact that for a reward contingency corresponding to response strategy, place strategy is able to correctly predict reward only in approximately half of the trials. Conversely, when the contingency corresponds to the place strategy, reward predictability of the response strategy is only about half of that of the place strategy. The activity of the strategy selective neurons in our model resembled the persistently changing cells recorded by Rich & Shapiro (2009) [8] ($\approx$24% of all cells recorded from PFC during strategy switches). Persistently changing cells were defined as cells that fired at significantly different rates during the 'before' and 'after' phases.

The advantage of the experimental setup used by Rich & Shapiro (2009) is that the selectivity of neural activity to a particular strategy can be dissociated from the selectivity to other behavioral variables. To clarify this idea, consider all trials started in e.g., North position (see Fig. 1). When switching from, e.g., response strategy 'turn left' to place strategy 'go east', the 'north-to-east' path remains to be a valid (or *consistent*) path, in spite of being controlled by another strategy. In contrast, the 'south-to-west' path which was a valid path before the switch, becomes invalid (or *inconsistent*) after the switch. Therefore, in order to show

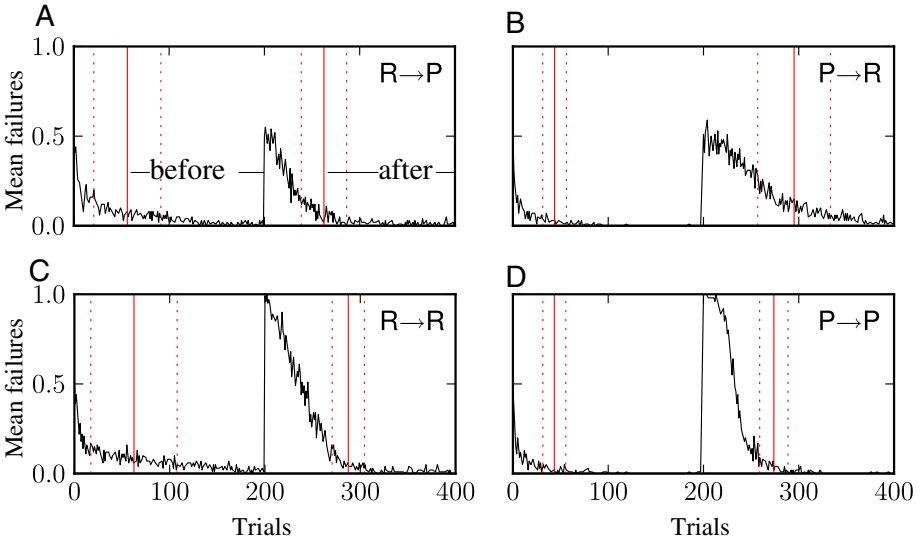


Fig. 3. Model performance during strategy switches (A,B) and reversals (C,D). Black lines show the mean number of failures as a function of trials. Red solid and dotted lines show mean trial number ± SD at which the criterion was reached (see text).

that putative strategy-selective neurons respond specifically to strategy switches, but not to other behavioral and motor variables, one has to demonstrate that these neurons change their activity while the rat runs along the consistent path before and after the switch. This is true for strategy-selective cells in our model. In spite of the fact that on the consistent paths the simulated rats made the same turns before and after the switch, these turns were controlled by the response strategy before the switch and by the place strategy after the switch. The change in strategy is reflected by the corresponding change in the firing rate of strategy-selective cells in our model (Fig. 4B).

Conversely, putative strategy-selective neurons should fire with similar rates when strategy remains the same but behavior changes (e.g. when starting from the opposite arms of the maze). This is so in our model, as demonstrated by similar rates of response-strategy cell before the switch, when starting from different starting positions (Fig. 4C). The same is true for the place-strategy cell after the switch (Fig. 4D).

Finally, putative strategy-selective neurons should be sensitive to changes in strategy, but not to changes in reward contingency when the strategy remains the same (as in reversals, [8]). Thus, if paths and contingency change, but strategy remains the same, as in reversals, strategy-selective neurons should fire with similar rates after the change in contingency. This is indeed so in our model, since the response-strategy cell fired with similar rates during the 'before' and 'after' phases in response reversal (Fig. 4E); similarly, the place-strategy cell fired with similar rates before and after spatial reversal (Fig. 4F). This is in contrast to the case when contingency, paths *and* strategy change, where neurons change

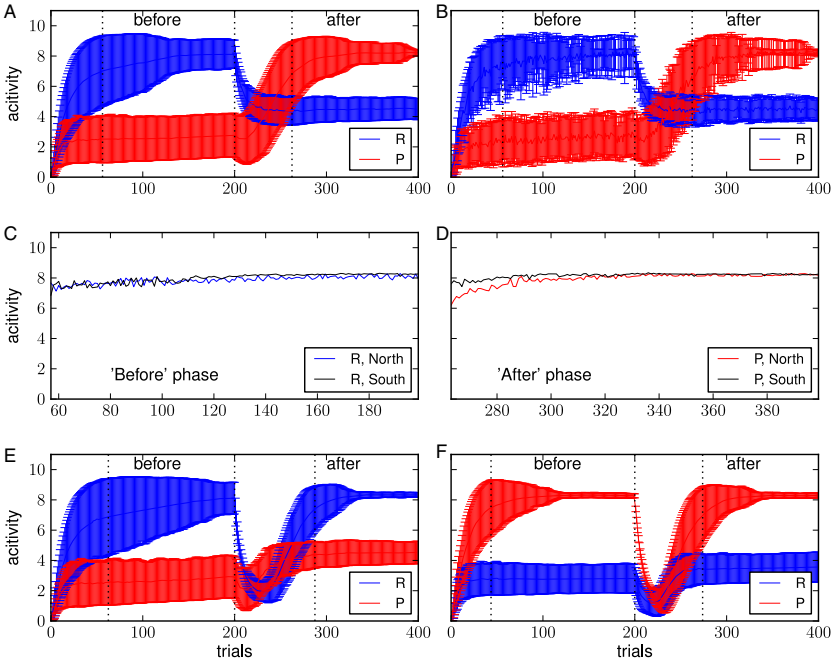


Fig. 4. Activity of the response-strategy (blue) and place-strategy (red) cells during switches (A-D) and reversals (E,F). A. Cell activity during reversal from response ('left') to place ('east') task. B. The same data as in A, but only successful trials from position N are shown (i.e., only for consistent paths). C,D. Activity of the response-strategy neuron during the 'before' phase (C) and of the place strategy neuron during the 'after' phase (D). The neural activities are grouped according to starting positions. E,F. Cell activity during response (E) and place (F) reversal.

significantly their rates as shown in Fig. 4A (the figure is similar in the case of switch from place to response strategy, not shown).

3.3 Prediction about DA Activity in the PFC during Strategy Switches

The results above suggest that different strategies are indeed encoded in the activities of the strategy-selective cells in our model, in a manner resembling the strategy encoding by the rat PFC neurons [8]. It is interesting to see what predictions can be derived from this simple model. The principal feature of the TD learning algorithm that is used in our model to learn strategies and strategy selection is the reliance on the reward-prediction error δ (Eq. 3). It has been hypothesized that activities of dopaminergic neurons in the VTA encode the reward-prediction error [11]. This hypothesis is supported by experimental data from conditioning experiments showing that DA activity is increased immediately after reward delivery in the beginning of learning, while after learning

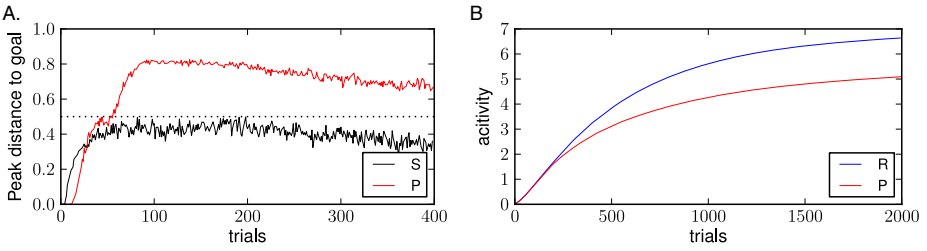


Fig. 5. A. Evolution of the distance between peak δ location and the goal across trials for place-strategy network (red) and strategy-selection network (black). The distance is measured by fractions of total distance L of the maze: 0 corresponds to reward site, while 1 corresponds to start position. The starting positions were either N or S in pseudo-random order, the goal warm was fixed (E). B. Evolution of Q-values of the response strategy (blue) and place strategy (red) across trials. For this simulation, we decreased the learning rate to $\eta = 0.001$ in order to minimize the influence of noise. The starting position (south) and goal location (north) were constant for all trials.

the increase in DA activity occurs upon the presentation of a reward predicting stimulus, i.e. before the reward is received [14]. If this hypothesis is true, then reward propagation dynamics in our model may suggest how DA activity should change during goal learning in our plus-maze navigation task

In order to describe the reward propagation dynamics, we measured the evolution across trials of the mean distance between the location where δ was maximal and the goal location for the place-strategy network and strategy selection network during a stable performance session (i.e., 400 trials with pseudo-random starting positions and a fixed goal arm). Similarly to what has been observed in the conditioning experiments [14], the peak of δ propagated from the reward site to locations closer to the starting position (Fig. 5A). In terms of DA signaling, this corresponds to the shift of a peak in DA activity away from the reward site with training. Moreover, in our simulation the peak of δ shifted further from the goal in the place strategy network than in the strategy selection network (as shown by the difference between the red and black lines in Fig. 5A). Thus, the general testable prediction derived from our model is that propagation of the peak DA signal in the prefrontal cortex will be limited by the choice point of the maze.

4 Discussion

We presented a simple model of strategy switching in the plus maze in which place and response strategies, as well as selection between those strategies, were learned using a standard Q-learning TD algorithm. The architecture of the model can be directly mapped to the architecture of neuronal networks implicated in navigation. In spite of its simplicity, the model has been able to learn successfully spatial reversals, response reversals and strategy switches in the simulated plus maze task. The learning time was comparable to the learning time of real rats in behavioral experiments [8]. Moreover, activities of strategy-selective cells in the selection model were similar to those of strategy-selective biological neurons recorded

from the PFC of behaving rats. In addition, a testable prediction concerning DA signaling in the PFC has been derived from our modeling results.

There are several limitations of the presented model in relation to available experimental data concerning the role of prefrontal cortex in behavior. First, it is known that strategy switches and reversals in the plus maze are mediated by different subareas of the prefrontal cortex, medial PFC (mPFC) and orbitofrontal (OFC) cortex, respectively [6,7]. Moreover, there is experimental evidence suggesting hierarchical organization of these two subareas [15]. In our model, both switches and reversals are learned within a single network which can be considered as a combined model of mPFC/OFC. We will address the differential roles of mPFC and OFC in future versions of the model. Second, it has been shown that when rats were trained to perform multiple strategy switches, lesions to mPFC did not impair switching performance after the third switch. These data suggest that by the third switch rats acquired a mPFC-independent strategy (possibly depending on reactive or habitual mechanisms) which allowed them to immediately switch strategy when reward was not obtained in the predicted location. Our model can not explain these data, since from the point of view of the selection network all switches are identical. Finally, our model is only partially consistent with the data suggesting that place strategy is preferred early in training, while this preference shifts towards response strategy with prolonged training [4]. In Fig. 5B, we plot the evolution of the Q-values for the response and place strategies (which are equal to the activities of strategy-selective cells in the selection network) across trials. The Q-value for the response strategy at the asymptote of training is higher than that for the place strategy, consistent with the preference for the response strategy after overtraining. However, this result is the consequence of the particular model of the sensory input that we chosen for the response-strategy network and hence is not general enough to provide a valid explanation for the experimental data. A more detailed model of sensory input (see, e.g. [13]) may be required to make a stronger claim.

The architecture of the model presented here is similar to a model proposed previously [16,17]. However, the learning algorithm is substantially simplified and is identical in the strategy learning networks and in strategy selection network, greatly simplifying the analysis of the model. In addition, we provide a comparison between strategy selective neurons in our model with new data from experimentally recorded mPFC neurons. Finally, we note that the number of input cells in the model can be increased without decreasing the model's performance. We used a small number of cells to speed up simulations, but any number of input cells can be used.

References

1. Arleo, A., Rondi-Reig, L.: Multimodal sensory integration and concurrent navigation strategies for spatial cognition in real and artificial organisms. J. Integr. Neurosci. 6, 327–366 (2007)
2. White, N.M., McDonald, R.J.: Multiple parallel memory systems in the brain of the rat. Neurobiol. Learn. Mem. 77, 125–184 (2002)

3. Packard, M.G., Hirsh, R., White, N.M.: Differential effects of fornix and caudate nucleus lesions on two radial maze tasks: evidence for multiple memory systems. J. Neurosci. 9, 1465–1472 (1989)
4. Packard, M.G., McGaugh, J.L.: Inactivation of hippocampus or caudate nucleus with lidocaine differentially affects expression of place and response learning. Neurobiol. Learn. Mem. 65, 65–72 (1996)
5. Chang, Q., Gold, P.E.: Intra-hippocampal lidocaine injections impair acquisition of a place task and facilitate acquisition of a response task in rats. Behav. Brain. Res. 144, 19–24 (2003)
6. Ragozzino, M.E., Detrick, S., Kesner, R.P.: Involvement of the Prelimbic-Infralimbic areas of the rodent prefrontal cortex in behavioral flexibility for place and response learning. J. Neurosci. 19, 4585–4594 (1999)
7. Rich, E.L., Shapiro, M.L.: Prelimbic/infralimbic inactivation impairs memory for multiple task switches, but not flexible selection of familiar tasks. J. Neurosci. 27, 4747–4755 (2007)
8. Rich, E.L., Shapiro, M.: Rat prefrontal cortical neurons selectively code strategy switches. J. Neurosci. 29, 7208–7219 (2009)
9. Miller, E.K., Cohen, J.D.: An integrative theory of prefrontal cortex function. Annu. Rev. Neurosci. 24, 167–202 (2001)
10. Sutton, R., Barto, A.G.: Reinforcement Learning - An Introduction. MIT Press, Cambridge (1998)
11. Schultz, W., Dayan, P., Montague, P.R.: A neural substrate of prediction and reward. Science 275, 1593–1599 (1997)
12. Arleo, A., Gerstner, W.: Spatial cognition and neuro-mimetic navigation: A model of hippocampal place cell activity. Biol. Cybern. 83, 287–299 (2000)
13. Sheynikhovich, D., Chavarriaga, R., Strösslin, T., Arleo, A., Gerstner, W.: Is there a geometric module for spatial orientation? Insights from a rodent navigation model. Psychol. Rev. 116, 540–566 (2009)
14. Schultz, W.: Predictive reward signal of dopamine neurons. J. Neurophysiol. 80, 1–27 (1998)
15. Young, J.J., Shapiro, M.L.: Double dissociation and hierarchical organization of strategy switches and reversals in the rat PFC. Behav. Neurosci. 123, 1028–1035 (2009)
16. Chavarriaga, R., Strösslin, T., Sheynikhovich, D., Gerstner, W.: A computational model of parallel navigation systems in rodents. Neuroinformatics 3, 223–242 (2005)
17. Dollé, L., Khamassi, M., Girard, B., Guillot, A., Chavarriaga, R.: Analyzing interactions between navigation strategies using a computational model of action selection. In: Freksa, C., Newcombe, N.S., Gärdenfors, P., Wölfl, S. (eds.) Spatial Cognition VI. LNCS (LNAI), vol. 5248, pp. 71–86. Springer, Heidelberg (2008)

Distributed Online Learning of Central Pattern Generators in Modular Robots

David Johan Christensen[1], Alexander Spröwitz[2], and Auke Jan Ijspeert[2]

[1] The Maersk Mc-Kinney Moller Institute, University of Southern Denmark
david@mmmi.sdu.dk
[2] Biorobotics Laboratory, Ecole Polytechnique Fédérale de Lausanne, Switzerland
{alexander.sproewitz,auke.ijspeert}@epfl.ch

Abstract. In this paper we study distributed online learning of locomotion gaits for modular robots. The learning is based on a stochastic approximation method, SPSA, which optimizes the parameters of coupled oscillators used to generate periodic actuation patterns. The strategy is implemented in a distributed fashion, based on a globally shared reward signal, but otherwise utilizing local communication only. In a physics-based simulation of modular Roombots robots we experiment with online learning of gaits and study the effects of: module failures, different robot morphologies, and rough terrains. The experiments demonstrate fast online learning, typically 5-30 min. for convergence to high performing gaits ($\approx$ 30 cm/sec), despite high numbers of open parameters (45-54). We conclude that the proposed approach is efficient, effective and a promising candidate for online learning on many other robotic platforms.

1 Introduction

Modular robots are made up from a number of interconnected robotic modules. Each module can communicate with neighbor modules, sense its local environment, and control its own actuators. By combining the modules in different configurations robots with different capabilities can be constructed. Since a robot's mobility is highly dependent on the details of its morphology, the flexibility of modular robot's morphology makes them an interesting platform for studying locomotion. However, control and adaptation of locomotion must be implemented in the context of the modular robot's distributed morphology. Further, since modular robots are polymorphic we desire a strategy which is not designed for a specific morphology. The strategy should rather optimize a variable number of control parameters, for a class of morphologies, while the robot is moving in its environment. In this paper we take a distributed control approach to tackle the problem: All modules have individual, identical, and autonomous controllers. Any module optimizes its own set of control parameters based on a global reward signal. The robot's locomotion pattern then emerges from the collective adaptations and behaviors of its modules. We hypothesize that such a distributed strategy may be more robust and flexible since it may be independent to the specific robot's morphology and can adapt online to module failures or morphology changes. Ultimately, we anticipate that by studying such distributed strategies

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 402–412, 2010.

we may gain insights into how adaptive sensory-motor coordination can emerge and self-organize from billions of individual cells in biological organisms.

In this paper we study a distributed learning strategy for online optimization of locomotion gaits. We experiment with two quadruped robots constructed from Roombots modules. Each Roombots module has three actuators which we control using periodic actuation patterns generated by three local oscillators. Neighbor-to-neighbor communication between modules is used to synchronize the module's oscillators. These local connections make the oscillators form a central pattern generator (CPG) network covering the whole robot thereby enabling global synchronization. To enable life-long learning based on noisy fitness measurements we apply the model-less Simultaneous Perturbation Stochastic Approximation (SPSA) method. Each module optimizes its own local CPG parameter set based on a globally shared reward signal. Therefore, both the control and the learning are distributed without any centralized control necessary.

The rest of this paper is organized as follows: In Section 2 we summarize related work. In Section 3 we describe the methods which comprise the online learning strategy. The experimental platform and setup is described in Section 4. A number of experiments with simulated Roombots robots are presented in Section 5. The experiments demonstrate that the proposed strategy finds fitter gaits than random search optimization, works for different morphologies, can adapt to module failures, but converges to suboptimal gaits in rough terrains. We conclude in Section 6 that the proposed approach is efficient, effective and a good candidate for online learning of locomotion on many robotic platforms.

2 Related Work

Here, we review related work on evolutionary adaptation and online learning of modular robots for the task of locomotion. Karl Sims pioneered the field in the early 90's by co-evolving the morphology and control of simulated modular robots [10]. Later work succeeded in transferring similar co-evolved robots from simulation to hardware [6,8]. An example of adaptation by evolution in modular robots was conducted by Kamimura et al., who evolved the coupling parameters of central pattern generators for straight line locomotion of modular M-TRAN robots [5]. By incorporating sensory entrainment in the optimization the authors were able to bridge the reality gap. Although appealing, one challenge with evolutionary approaches is that once transferred the robot is typically no longer able to adapt to major changes in the morphology or environment. To overcome this limitation optimization of locomotion gaits can be performed online. This was studied by Marbach and Ijspeert on the YaMoR modular robotic system [9]. Their strategy was based on Powell's method, which performed a localized search in the space of selected parameters of coupled oscillators. Parameters were manually extracted from the modular robot by exploiting symmetries. Follow-up work by Spröwitz et al. demonstrated online optimization of 6 parameters on a physical robot in roughly 25-40 minutes [14]. We also try to realize simple, robust, fast, model-free, life-long learning on a modular robot. The main difference is that we seek to

automate the controller design further in the sense that no parameters have to be extracted from symmetric properties of the robot. Further, our approach utilizes a form of distributed optimization. A similar approach was taken by Maes and Brooks who performed distributed learning of locomotion on a 6-legged robot [7]. The learning was distributed to the legs themselves. Our strategy is not dependent on the robot's specific morphology. Similarly, Bongard et al. demonstrated learning of locomotion and adaptation to changes in the configuration of a modular robot [1]. They took a self-modeling approach, where the robot developed a model of its own configuration by performing basic motor actions. In a physical simulator a model of the robot configuration was evolved to match the sampled sensor data (from accelerometers). By co-evolving the model with a locomotion gait, the robot could then learn to move with different morphologies. Our work presented here is similar in purpose but different in approach: The strategy is simple, model-less and computationally cheap to allow implementation on small embedded devices, such as modular robots. In previous work we studied distributed, morphology independent, online learning for ATRON and M-TRAN robots [2,3]. This work was based on the same principles but the methods applied were different: instead of SPSA optimization we applied a simple reinforcement learning strategy and instead of coupled oscillators we applied discrete actions and gait-tables.

3 Methods

This section describes the methods for generating periodic actuation patterns for gait implementation and for online optimization of gait parameters. The methods are selected and combined into an online learning strategy with the following design goals in mind:

Morphology Independence: Since a modular robot can take on many different morphologies, the strategy should not be designed for any particular morphology but rather function on a class of different morphologies.

Life-long Learning: The morphology of a modular robot can change over time, either due to module failures, adding or removing of modules, or due to voluntary morphosis, therefore, the strategy must be able to continuously adapt while performing its function.

Noise Tolerance: The gaits must eventually be optimized directly on the physical robot. The interactions between the robot and its environment will be complex and in practice impossible to predict, therefore, the optimization strategy must be tolerant to noisy fitness measurements.

Simple Implementation: Modular robots are embedded devices with limited communication and computation abilities, thus, the strategy must require a minimal amount of resources and ideally be simple to implement on the distributed morphology that modular robots are.

3.1 Central Pattern Generators

Biological CPGs are special neural circuits found in vertebrates, able to produce a rhythmic signal without any external sensory input, where they for example

control muscles during locomotion. We apply a CPG model for gait control because of their ability to generate periodic actuation patterns, ability to self-synchronize in a distributed system, open parameters which are appropriate for optimization, and finally since CPGs are biologically plausible. A review of CPGs and their use in robot control can be found in [4]. The specific CPG model we utilize is a Hopf oscillator in Cartesian space with diffusive coupling [15]. The advantages of this model include its simplicity, stable limit-cycle behavior, and explicit parameters for setting phase, amplitude and frequency. For an oscillator i the coupled differential equations are:

$$\dot{x_i} = \gamma(\mu - r_i^2)x_i - \bar{\omega}y_i \tag{1}$$

$$\dot{y_i} = \gamma(\mu - r_i^2)y_i + \bar{\omega}x_i \tag{2}$$

Where $r_i = \sqrt{x_i^2 + y_i^2}$ and the state variables are x and y. γ is a parameter that affects the speed of convergence towards the oscillators amplitude μ^2. $\bar{\omega}$ is the oscillator's frequency which is a function of a frequency parameter, ω, and is also affected by the sum of couplings to other oscillators. A coupling from oscillator i to oscillator j has a weight parameter, w_{ij}, and a desired phase difference ϕ_{ij}. Then the oscillator may be coupled to other oscillators using:

$$\bar{\omega} = \omega + \sum_{j=1}^{N} \frac{w_{ij}}{r_i} \left[(x_i y_j - x_j y_i)\cos\phi_{ij} - (x_i x_j + y_i y_j)\sin\phi_{ij}\right] \tag{3}$$

We use one oscillator to control the position of an actuator by using x_i as the control set-point for the actuator.

3.2 CPG Network Architecture

To enable a scalable, distributed, and morphology independent control strategy we design the network of CPG couplings so that it is equivalent on each module (homogeneous control). Each Roombots module is programmed with four oscillators: three which are used as set-points for its actuators (C_{m1}, C_{m2} and C_{m3}) and one which acts as a clock (C_c). The architecture is illustrated in Fig. 1. The robot is equipped with a distributed global clock, implemented as a network of in-phase clock oscillators, one per module. Each clock oscillator is coupled with the clock oscillators on its neighbor modules using local communication channels. This architecture is scalable since oscillators are only coupled neighbor-to-neighbor, so the computation/communication load is independent on the number of modules in the robot. Further, the architecture is distributed since the module controls itself based on its local state and local interactions. Finally, the architecture is morphology independent since when adding a new module to the robot new couplings can automatically be established using local communication. Therefore, the individual modules are not aware of the global module configuration.

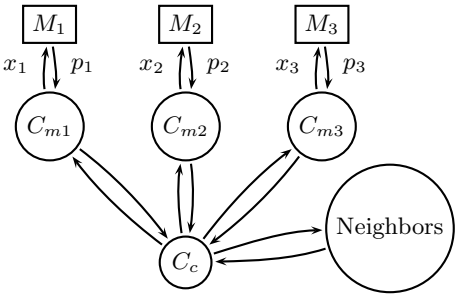


Fig. 1. The CPG network architecture of coupled oscillators within each Roombots. Three oscillators provide the set-points for the three servos (M_1 -M_3). A forth oscillator acts as a clock which is coupled in phase with clock oscillators on neighbor modules. The servo position, p_i, can be used as feedback in the oscillators although we do not use it here.

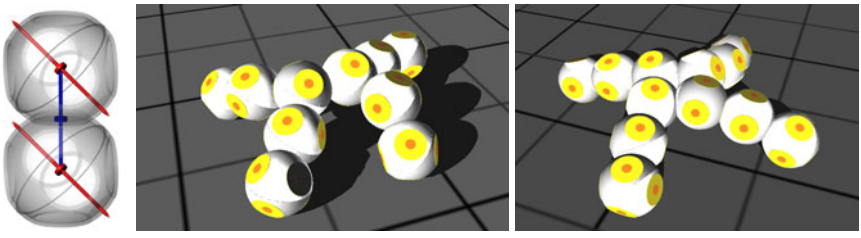
3.3 Learning Algorithm

For online optimization of CPG parameters we select the Simultaneous Perturbation Stochastic Approximation (SPSA) method by Spall [12]. This algorithm requires no explicit gradient and therefore no model of the robot. It is designed to build an approximation of the gradient from direct (generally noisy) measurements of the objective function. Further, SPSA only requires two measurements of the objective function per iteration (i.e. two robot trials with different controllers) independent on the number of adjustable parameters. Also, these measurements are made based on small perturbations of the same parameter set. Hence the robot's behavior only alters slightly while it is learning, unlike optimization based on population-based methods such as evolutionary algorithms. Finally, SPSA is simple to implement in a distributed fashion since each module may independently optimize its own parameters without knowledge of the other modules parameters or the need for any other coordination than simple synchronization of when the parameters are updated.

The SPSA method optimizes the parameter set $\hat{\theta}$ defined by the experimenter. In an iteration, k, it estimates the gradient, $g(\hat{\theta})$, based on two noisy measurements of the objective function $y(\hat{\theta})$:

$$\hat{g}_k(\hat{\theta}_k) = \frac{y(\hat{\theta}_k + c_k\Delta_k) - y(\hat{\theta}_k - c_k\Delta_k)}{2c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ \Delta_{k2}^{-1} \\ \vdots \\ \Delta_{kp}^{-1} \end{bmatrix} \tag{4}$$

Where c_k is a learning parameter and Δ_k is an vector of randomized ± 1. SPSA then updates $\hat{\theta}$ based on $\hat{g}_k(\hat{\theta}_k)$.



(a) (b) Quadruped Type 1 (c) Quadruped Type 2

Fig. 2. (a) An illustration of the Roombots' three degrees of freedom. (b) A robot comprised of five Roombots modules. (c) A robot comprised of six Roombots modules.

$$\Delta\hat{\theta}_k = -a_k \cdot \hat{g}_k(\hat{\theta}_k) \tag{5}$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + sign(\Delta\hat{\theta}_k) \cdot min(|\Delta\hat{\theta}_k|, \epsilon) \tag{6}$$

a_k is a learning parameter, we also added a max step-size, ϵ, to reduce the risk of instability.

4 Experimental Setup

The Roombots is a self-reconfigurable modular robot which is being developed at EPFL [13]. A Roombots module consists of two spherical parts, made up by four hemispheres in total, see Fig. 2(a). The hemispheres can actively be rotated relative to each other, thereby giving a Roombots module two "outer" and one "inner" actuated degree of freedom. The outer and inner hemispheres contain up to three active connectors respectively, which enable a module to connect to other modules. In this paper we experiment with the two different quadrupedal Roombots robots shown in Fig. 2. The Roombots modules are simulated in the commercial Webots robot simulator by Cyberbotics Ltd [16] which relies on Open Dynamic Engine (ODE) for simulation of collisions and rigid body dynamics [11]. The details of the Roombots model used are based on the current prototype of the Roombots as well as expected characteristics of the final design. The characteristics are kept fairly conservative but since module details will vary slightly compared to the final Roombots design, we cannot expect a perfect transfer to the physical modules once ready. The most important module parameters are: mass = 0.975 kg, actuation torque = 5 Nm, and a maximum rotational velocity = 2.62 rad/sec. Other environmental parameters include coefficients of friction and restitution, which are 1.0 and 0.5 respectively.

In the following experiments each module runs identical learning controllers and optimizes their behavior based on a single shared reward signal. For simplicity the reward is velocity computed as the distance traveled by the robot's center of mass in the xy-plane in a fixed length time duration: $y(\hat{\theta}) = \sqrt{\Delta x^2 + \Delta y^2}/T$. Each T seconds a single reward signal is sent to all the modules which corresponds to a measurement of either $y(\hat{\theta}_k + c_k\Delta_k)$ or $y(\hat{\theta}_k - c_k\Delta_k)$. After both

Table 1. Fixed parameters

(a) Coupled Oscillators

Symbol	Description	Value
ω	Frequency	0.8 Hz
γ	Amp. Contraction	1.0
θ_{cc}	Phase Difference	0.0
θ_{mc}	Phase Difference	2π - θ_{cm}
w_{cc}, w_{cm}	Coupling strength	5.0
w_{mc}	Coupling strength	1.5

(b) SPSA-based Learning

Symbol	Description	Value
c_k	Gain parameter	0.025 or 0.05
a_k	Gain parameter	0.00015
ϵ	Max Stepsize	5% of range
T	Time Step	4 sec.

Table 2. Open parameters in the coupled oscillators

Symbol	Description	Init. Val.	Range
μ^2	Amplitude	0.35π	$[0; 0.7\pi]$
θ_{cm}	Phase Difference	0.0	$[-\pi; \pi]$
x_{offset}	Offset	0.0	$[-0.2\pi; 0.2\pi]$

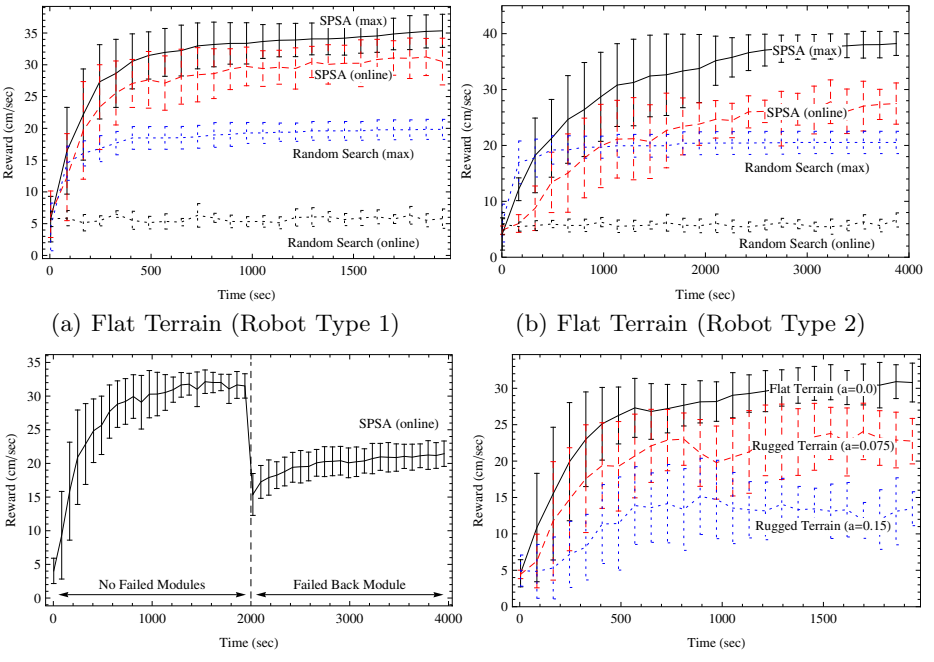
measurements are performed the new $\hat{\theta}_{k+1}$ is computed. Fixed parameters for the SPSA-based learning and the CPG architecture are set as indicated in Table 1. The only parameter which is not the same for the two robots is the learning parameter c_k. For Type 1 $c_k = 0.025$ is appropriate, while it causes divergence in the learning for Type 2. Instead we set $c_k = 0.05$ for Type 2 at the cost of more gait variance during learning. The open parameters which must be optimized by the learning algorithm are shown in Table 2. For SPSA-based learning the open parameters, $\hat{\theta}$, are scaled between 0 and 1 and initialized to 0.5 (midpoint of the valid range). We found that random initialization often produces initial gaits too far from near optimal gaits, which causes the learning system to sometimes get caught in local optima. We plan to experiment with using random search optimization to find a good initial parameter set.

5 Experiments

In this section we present experiments with the proposed SPSA based strategy on simulated Roombots robots.

5.1 Morphology Independent Learning

To study the effects of different morphologies we performed experiments with SPSA-based learning and random search optimization on the Type 1 and Type 2 Quadruped robots. The average result of 10 trials with Type 1 is shown in Fig. 3(a). We observe that both the SPSA-based strategy and random search converge after approximately 10 minutes of trial and error behavior by the robot. This fast convergence gives strength to the claim that the learning could realistically be utilized for life-long adaptation on the physical robot. Further, we observe that



(a) Flat Terrain (Robot Type 1)

(b) Flat Terrain (Robot Type 2)

(c) Module Failure (Robot Type 1)

(d) Different Terrains (Robot Type 1)

Fig. 3. The graphs show the average velocity of the robots as it improves over time. (a) and (b) SPSA-based learning compared to random search optimization. "Max" graphs indicate the average of the highest velocity measured so far in the trial. (c) Adaptation after back module failure. (d) Adaptation in three different terrains. All graphs are the average of 10 independent trials, bars indicate standard deviation.

the online average velocity (30.7 cm/sec), measured from iteration 400 to 500, using the SPSA-based strategy is significantly higher than the maximum gait velocities (20.4 cm/sec) found by random search optimization ($P = 4.88 * 10^{-7}$). This result indicates that the parameter space is too large for random search to find the same solutions within the time given, further it also indicates the existence of gradients in the objective function that the SPSA-based strategy can exploit. For comparison we performed the equivalent experiment, using the same controller with the Type 2 robot. The results are shown in Fig. 3(b). Compared to learning with Type 1 robot we observe that for Type 2 the SPSA-based strategy converges slower (approx. 30 minutes) but still manages to finds high performing gaits. Also here we find that the SPSA-based strategy converges to significantly better gaits (27.6 cm/sec) than those found with random search (20.5 cm/sec) ($P=1.64 * 10^{-6}$).

By inspecting the solutions found by SPSA for the two robots we observe that all found gaits are similar to a trot, where the legs move together in diagonal pairs. Although the solutions found with random search optimization showed greater variability all the gaits were still some variation of a fundamental trot.

These observations indicate that a trot gait is a strong attractor point in the objective function for these particular combinations of robots, environment, and parameterized CPG controller.

In summary the SPSA-based control strategy allows us to perform online learning with two different robot morphologies without changing any part of the strategy. Except that we found it necessary to change the learning parameter, c_k. In future work we will try to remove this limitation by using adaptive learning rates to increase the strategy's morphology independence.

5.2 Adaptation after Morphosis and in Rugged Terrain

To study the effects of involuntary morphosis, i.e. morphological change, we performed experiments with module failures using the Type 1 robot. Initially the five modules comprising the robot are fully functional. After 500 iterations (2000 seconds), a module fails by locking its three actuators in their initial position defined by the starting pose of the robot (the CPG couplings stays intact). We then observe if the robot is able to adapt to this change in morphology by letting the robot learn for additional 500 iterations without resetting the learning parameters or any other part of the control system. For comparison we also performed the equivalent experiments with the learning disabled after module failure. In this case the robot does no longer adapt but keeps performing the same gait as just before the module failure. We performed two experiments: (1) In the first experiment a leg module fails. The module failure event is followed by a minor drop in velocity (from 31.4 cm/sec to 27.6 cm/sec), with no clear later improvement. The results of the equivalent experiment with no adaptation after module failure yields an average velocity of 27.1 cm/sec after module failure. Consistently we find that there is no significant difference in the average velocity between adaptation/no-adaptation after module failure ($P = 0.33$). So adaptation after morphosis seems not important in the case of a failing leg module for this robot structure. (2) In the second experiment the back module fails. In this case the event is followed by a major drop in average velocity, which seems to gradually improve after the event, see Fig. 3(c). The average velocity in the time interval 3000 sec to 4000 sec is 20.9 cm/sec with adaptation and 13.2 cm/sec without adaptation. Statistical analysis confirms that there is a significant difference between adaptation/no-adaptation in this case ($P = 0.00063$). So unlike the case of a failed leg module, in the case of failed back module life-long adaptation is important.

To study the effects of environment parameters on the learning we perform experiments with SPSA-based learning using a Type 1 robot in simulated rugged terrains. In the xy-plane the height of the terrain is defined by: $height(x, y) = a \cdot \cos(b \cdot y) \cdot \sin(b \cdot x)$. We set the parameters to: $b = 2.5$ meters and a to 0, 0.075 and 0.15 meters to create a terrain which vary from completely flat to a hilly terrain with relatively steep slopes and deep valleys. Fig. 3(d) shows the result of learning in these three different terrains. By visually inspecting the found gaits we observe that in all cases the learning finds gaits which are able to move in the given terrain. As for flat terrain, in rough terrain the found gaits were also

trot-like, however, the stride length was generally shorter. In addition, we found two effects on learning in increasingly rough terrain: (1) The first effect is a decrease in the final average velocity: 32.8 cm/sec ($\sigma = 2.59$), 23.2 cm/sec ($\sigma = 3.67$), 15.1 cm/sec ($\sigma = 2.31$) for $a = 0.0$, $a = 0.075$ and $a = 0.150$ respectively (measured by reevaluating the final gaits without adaptation). This effect is not surprising since we expect the robot to move slower in rougher terrain. (2) The second effect is a decrease in the ability to learn near optimal gaits. We observe this by reevaluating the gaits found in flat terrain ($a = 0.0$) in the two non-flat terrains ($a = 0.150$ and $a = 0.075$). Because the gaits are optimized in a different environment we would expect to see a decrease in performance compared to those optimized for the environment. However, instead we observe a drastic relative increase in performance: 30.6 cm/sec ($\sigma = 2.51$) and 24.7 cm/sec ($\sigma = 4.36$) for $a = 0.075$ and $a = 0.150$ respectively. This result indicates that the SPSA-based learning do not find near optimal gaits in the two terrains. The reason for the second effect is likely due to increased noise in the objective measurement (beyond the limits of SPSA's noise tolerance). We have observed a drastic drop in the average signal to noise ratio for the found gaits from $\overline{SNR} = \overline{y(\hat{\theta})}/\sigma = 23.1$, $\overline{SNR} = 10.5$ to $\overline{SNR} = 4.00$ for the three terrains respectively.

6 Conclusion

This paper reported on experiments using a distributed strategy based on the SPSA method for online optimization of a CPG network controlling the locomotion of modular robots. This online learning and control strategy was designed to be independent to the particular robot morphology, simple to implement in a distributed system, and to enable life-long adaptation based on a noisy reward signal. The strategy was evaluated in simulations of different quadrupedal Roombots robots. First, we found that the proposed strategy was appropriate for life-long learning since it could maintain a high performance during learning. Seconds, we found that the strategy could reliably optimize gaits with a considerably higher velocity than those found by random search. A near optimal gait (≈ 30 cm/sec) was typically found in 5-30 minutes. Third, we also found that the strategy enabled the robot to readapt its gait after involuntary morphosis (failed back module). Finally, we found that rough terrains decrease the strategy's effectiveness considerably since it drastically increased the amount of noise in the measured objective function. In conclusion the proposed strategy is efficient and effective on the Roombots robots and is a promising candidate for life-long online learning on many other robotic platforms. However, further work is required to integrate the strategy with appropriate sensor feedback to modulate or change between gaits while learning. In addition, we plan to study the strategy's ability to online co-optimize gait and morphological parameters for a broader class of robot morphologies.

Acknowledgments

This work was performed as part of the "Locomorph" project funded by the EU's Seventh Framework Programme (Future Emerging Technologies, Embodied Intelligence) and as part of the "Assemble and Animate" project funded by the Danish Council for Independent Research (Technology and Production Sciences).

References

1. Bongard, J., Zykov, V., Lipson, H.: Resilient machines through continuous self-modeling. Science 314(5802), 1118–1121 (2006)
2. Christensen, D.J., Bordignon, M., Schultz, U.P., Shaikh, D., Stoy, K.: Morphology independent learning in modular robots. In: Proceedings of International Symposium on Distributed Autonomous Robotic Systems 8 (DARS 2008), pp. 379–391 (2008)
3. Christensen, D.J., Schultz, U.P., Stoy, K.: A distributed strategy for gait adaptation in modular robots. In: Proceedings of the IEEE Int. Conference on Robotics and Automation, ICRA (2010)
4. Ijspeert, A.J.: Central pattern generators for locomotion control in animals and robots: a review. Neural Networks 21(4), 642–653 (2008)
5. Kamimura, A., Kurokawa, H., Yoshida, E., Murata, S., Tomita, K., Kokaji, S.: Automatic locomotion design and experiments for a modular robotic system. IEEE/ASME Transactions on Mechatronics 10(3), 314–325 (2005)
6. Lipson, H., Pollack, J.B.: Automatic design and manufacture of robotic lifeforms. Nature 406, 974–978 (2000)
7. Maes, P., Brooks, R.A.: Learning to coordinate behaviors. In: National Conference on Artificial Intelligence, pp. 796–802 (1990)
8. Marbach, D., Ijspeert, A.J.: Co-evolution of configuration and control for homogenous modular robots. In: Proc. 8th Int. Conf. on Intelligent Autonomous Systems, Amsterdam, Holland, pp. 712–719 (2004)
9. Marbach, D., Ijspeert, A.J.: Online Optimization of Modular Robot Locomotion. In: Proceedings of the IEEE Int. Conference on Mechatronics and Automation (ICMA 2005), pp. 248–253 (2005)
10. Sims, K.: Evolving 3d morphology and behavior by competition. In: Brooks, R., Maes, P. (eds.) Proc. Artificial Life IV, pp. 28–39. MIT Press, Cambridge (1994)
11. Smith, R.: Open dynamics engine (2005), http://www.ode.org
12. Spall, J.C.: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transactions on Automatic Control 37(3), 332–341 (1992)
13. Sproewitz, A., Billard, A., Dillenbourg, P., Ijspeert, A.J.: Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture. In: International Conference on Robotics and Automation (ICRA 2009), Kobe, Japan (May 2009)
14. Sproewitz, A., Moeckel, R., Maye, J., Ijspeert, A.J.: Learning to move in modular robots using central pattern generators and online optimization. Int. J. Rob. Res. 27(3-4), 423–443 (2008)
15. van den Kieboom, J.: Biped locomotion and stability a practical approach. Master's thesis, University of Groningen, The Netherlands (2009)
16. Webots. Commercial Mobile Robot Simulation Software, http://www.cyberbotics.com

Learning New Motion Primitives in the Mirror Neuron System: A Self-organising Computational Model

Serge Thill and Tom Ziemke

Cognition & Interaction Lab
Informatics Research Centre
University of Skövde, PO Box 408, Skövde, Sweden
{serge.thill,tom.ziemke}@his.se

Abstract. Computational models of the mirror (neuron) system are attractive in robotics as they may inspire novel approaches to implement *e.g.* action understanding. Here, we present a simple self-organising map which forms the first part of larger ongoing work in building such a model. We show that minor modifications to the standard implementation of such a map allows it to continuously learn new motor concepts. We find that this learning is facilitated by an initial motor babbling phase, which is in line with an embodied view of cognition. Interestingly, we also find that the map is capable of reproducing neurophysiological data on goal-encoding mirror neurons. Overall, our model thus fulfils the crucial requirement of being able to learn new information throughout its lifetime. Further, although conceptually simple, its behaviour has interesting parallels to both cognitive and neuroscientific evidence.

1 Introduction

Learning by imitation is one way in which an agent can acquire new skills. In order to imitate, one must recognise and understand the actions of others, which is both a trivial insight and a hard problem. One possible approach is to *segment* observed complex actions (*e.g.* drinking from a glass) into constituent parts (*e.g.* reach for glass, grasp glass, bring to mouth, *etc.*). These constituents are commonly referred to as *motion primitives* and there are several approaches within the field of robotics aimed at identifying such primitives. Traditional unsupervised algorithms to this effect typically analyse statistical properties of an observed trajectory to segment this into motion primitives. This can be done, for example, based on joint velocities (*e.g.* Pomplun and Matarić, 2000; Lieberman and Breazeal, 2004, see also Kulić and Nakamura (2008) for a discussion) or based on a measure of variance in the data (*e.g.* Koenig and Matarić, 2006). More recently, Kulić and Nakamura (2008) proposed a Hidden Markov Model strategy to discover primitives in observed motion.

Once a basic repertoire of such primitives has been discovered, they can be used in imitation learning. Yamashita and Tani (2008), for instance, present a

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 413–423, 2010.

humanoid robotic platform capable of learning movements based on a neural network controller. This network exploits multiple activation timescales within its nodes to segment the motions demonstrated to the robot into a number of motion primitives (here, segments which are executed repeatedly during one overall motion) and it can be shown that the motion primitives and the overall action are represented on different timescales in the network. Learning novel compositions of motion primitives can then be achieved by modifying only the slowly activating nodes of the network. The main downside of this approach is that it requires explicit supervised offline learning. Clearly, it would be preferable for robotic agents to be able to self-organise, through so-called *motor babbling* (*e.g.* Meltzoff and Moore, 1997; Der and Martins, 2006), their own repertoire or "vocabulary" of motion primitives autonomously, grounded in on their own embodiment and interaction with the world.

To improve existing robotic approaches, it might be helpful to take inspiration from biology. Within this context, mirror neurons (Rizzolatti et al., 1996) are of particular interest. These neurons have been discovered in Macaque monkeys, but are also widely believed to exist in humans (in the latter case, however, most of the evidence comes from brain imaging techniques rather than single unit recordings - which is why many authors prefer to refer to the *mirror system* rather than individual mirror neurons). The characteristic property of mirror neurons is that they fire both when an agent *executes* an action himself and when it *observes* the same action being executed by another agent. Mirror neurons are widely considered to underlie action understanding and thus imitation in humans, but it is important to note that this is a hypothesis and not a proven fact (Hickok, 2008). Still, the idea of the mirror (neuron) system is an attractive source of inspiration for designing robotic and computational models that are capable of imitation (for a recent review, see Oztop et al., 2006).

A recent computational model of the mirror neuron system, called the *Chain model* (Chersi et al., 2006) is particularly interesting in this context as it is (1) based on neurophysiological data (Fogassi et al., 2005) and (2) explicitly representing motion primitives. In the model, these primitives are *chained* together to form the overall action. The model has found some use in the field of robotics (*e.g.* Erlhagen et al., 2007), but the primitives are typically hard-coded.

The longer-term aim of our work is to produce, for use in humanoid robots, a revised version of the Chain model which is capable of learning and integrating new motion primitives online, and to do so at any point in time. In this paper, we present the first significant step towards this model by creating a self-organising map which is capable of such online learning. The use of a SOM to model a mirror system is not new; a similar approach has been taken by, for instance Wermter et al. (2003). In that work, however, several maps were used and organised hierarchically while we use a single map whose node organisation more closely reflects that hypothesised by the Chain model. Although this is a very simple approach, we find that it reproduces many of the neurophysiological results, which may even suggest new insights into the functional organisation of the mirror neuron system. The rest of this paper is organised as follows: Section 2

presents the methodology used, in particular the self-organising map formalism and the input data organisation. Section 3 presents the simulation results, with a particular focus on the role of motor babbling and context coding. Section 4 presents a brief discussion of the work presented here and future extensions.

2 Methods

2.1 Online SOM

We use a traditional self-organising map (SOM, Kohonen, 1997) with a rectangular grid and no wrapping in the neighbourhood function (*i.e.* the SOM is not mapped onto a torus). However, we make no distinction between training and test data and run the SOM continuously. At every timestep, it is exposed to a different input and the winning node recorded. The evolution of the neighbourhood size n_t and the learning rate α_t over time follows a standard approach:

$$n_t = n_{min} + \lfloor (s - n_{min})\, \tau_t \rfloor \tag{1}$$

$$\alpha_t = \alpha_{min} + (1 - \alpha_{min})\, \tau_t \tag{2}$$

with s being the side length of the map, $\tau_0 = 1$ and $\tau_t = \max(\tau_{t-1} - {}^1/_{t_{inf}}, 0)$, where t_{inf} defines the duration of the *infancy* phase of the network. In this case, the infancy phase is characterised by shrinking neighbourhood sizes and learning rates. At any later point in time, both values are kept constant at their minima (here set to $n_{min} = 1$ and $\alpha_{min} = 0.2$).

2.2 Input Data

Since the present paper is primarily concerned with the organisation of primitives in the mirror neuron system, we are required to simulate input data which is general enough to test the concepts. We therefore require the following minimal

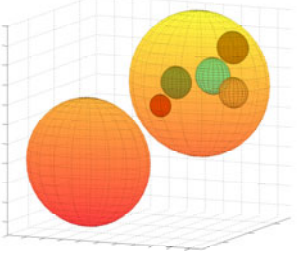


Fig. 1. 3D analogue of input space. Large spheres indicate clusters of datapoints representing all possible motions by whole body parts. Smaller spheres indicate clusters related to specific motion primitives which are a subset of all possible motions of one body part. Example data points for body parts or primitives are randomly sampled from their respective clusters.

properties: (1) the data is sampled from a high-dimensional space (20 dimensions in the present implementation but this is not a limit) in which (2) the overall motions from different body parts are represented by distinct clusters and (3) different primitives related to the same body part are represented by distinct clusters *within* the body part's motion cluster. These subclusters can thus be seen as representing primitives encode meaningful motions. Figure 1 gives a 3 dimensional analogue of the input space. The input data thus presents a certain level of abstraction compared to raw motion data and can represent for instance, some processing based on the spatial coordinates of the end effector as in, for example, the MNS2 model (Bonaiuto et al., 2007). It further removes differences between sensory origins of the data (*e.g.* proprioception v.s vision). However, this is acceptable at this state since we are only concerned with the organisation within the mirror neuron system here and the data received by this system is thought to be pre-processed, as seen for instance in the MNS2 model.

For the purposes of this paper, we limit ourselves to two distinct body parts (which could represent, for instance, arms and legs) and five motion primitives, which takes inspiration from the human analogue (see for instance the motion primitives in Chersi et al., 2006). It should be noted, however, that this does not represent a limitation of the system.

3 Results

In the following section, we describe the behaviour and properties of the SOM throughout the simulation and show that it develops characteristics that are analogue to that of a Mirror Neuron System. For illustrative purposes, we discuss a specific example here but the findings are general and reproducible in repeated runs, even though a SOM has, by nature, random elements that may affect the overall results.

3.1 Motor Babbling

During the infancy phase (which lasts 5000 iterations) of the SOM, no specific primitives are trained. Rather, the SOM is merely exposed to data-points sampled randomly from the two clusters representing all possible motions from the two body parts. The resulting map represents the body parts in roughly equal amounts but, not surprisingly, no motion primitives (Fig. 2A). Of particular interest is the fact that each body part is represented by a single, continuous cluster (rather than, for instance, one of the parts being split into several small and disconnected clusters). Since the neighbourhood size will be small after the initial infancy phase, the implication is that activity related to one body part is not likely to significantly affect nodes encoding the other body part (the exception being nodes that lie at the border). Thus, in order to investigate how the map organises motion primitives, it is an acceptable simplification for the sake of clarity to merely investigate this in detail for one of the body parts.

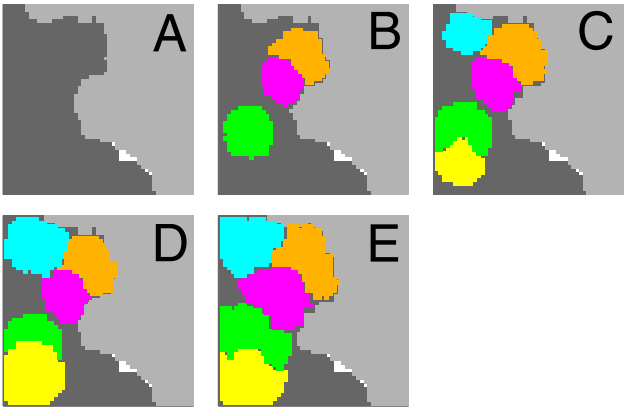


Fig. 2. The SOM modelling the mirror neuron system captured at different points in time. Grey regions indicate neurons encoding general motion of a body part with different greytones representing different body parts. Coloured neurons additionally represent a specific motion primitive involving that body part, with different colours representing different primitives. Figures show the SOM after (**A**) infancy: Nodes have learned to represent the body parts but no specific motion primitives yet; (**B**) exposure to three primitives: Some nodes have specialised to encode those primitives; (**C**) two additional primitives were presented: More nodes specialise to encode those primitives; (**D**) a period in which the first three primitives were no longer observed. The network has nonetheless not "forgotten" these primitives; (**E**) exposure to primitives now shown in different contexts. The additional information had little impact on the overall topology of the network, although some additional nodes now encode specific primitives.

3.2 Learning Motion Primitives

After the infancy phase, the network is exposed to data-points sampled randomly from clusters representing three primitives belonging to one body part. Additionally, the network continues to be exposed to smaller amounts of babbling from the second body part. The state of the map after 5000 data point presentations is presented in Fig. 2B. We first note that there is no radical change in the representation of the two body parts in the map. Second, a subset of nodes within the area representing the first body part have now further specialised into specifically representing the three observed motion primitives. Averaging over 100 runs, we find that the mean area of the map occupied by every primitive is $4.76 \pm 0.90\%$ of the total area.

Next, we include data-points randomly sampled from the input clusters of two additional primitives in the set of points presented to the SOM. The resulting changes in the map after an additional 5000 presentations are shown in Fig. 2C. Again, nodes within the region representing the first body part have specialised to encode the newly observed primitives. Averaging over 100 runs, we find that the mean area representing them is $4.00 \pm 1.38\%$ of the total map, which is only marginally less than for the original primitives. This difference can largely be

explained through the fact that the second set of primitives is presented less often to the map as the first set (since the set from which data-points are sampled is now larger). In the meantime, the average area of each of the original primitives has slightly increased to $5.42 \pm 1.16\%$, since the map continues to be exposed to them. This is expected as the learning rate is always non-zero and the map will thus show some plasticity over time.

Finally, we briefly test the memory of the SOM by removing data-points for the initial three of the five learned primitives from the set of points presented to the map. Figure Fig. 2D illustrates the effects after a further 5000 iterations. It can be seen that none of the primitives were "forgotten", although the mean area representing these maps (again calculated over 100 runs) shrinks slightly to $4.73 \pm 1.83\%$. This is due to some of these primitives sharing a "border" with those that were still actively presented to the map. The area of the active primitives increased slightly (to $7.42 \pm 1.72\%$) and some of this increase reduced the area of the primitives that were no longer observed. Again, this is an expected plasticity of the map and due to the non-zero learning rates.

Overall, these results thus demonstrate that the SOM is capable of learning new primitives throughout its lifetime without a major effect on already learnt primitives. However, the mirror neuron system is also known for encoding the goal of the overall action that a primitive is part of. In the next section, we investigate the self-organisation of our map in an analogous situation.

3.3 Goal-Specific Neurons in the SOM

One of the most interesting findings about mirror neurons is that some of them appear to be specifically encoding the goal of an action (Fogassi et al., 2005). In other words, a "grasping" neuron might fire if the intention of the overall action is eating but not if it is moving an object. Here, we investigate how our SOM performs when actions are presented in the context of different goals. To do so, we attach an additional 3-dimensional vector), chosen from a set of two randomly generated ones, to the existing input (see Fig. 3). The SOM is then run as before, with an infancy phase followed by the presentation of randomly chosen data points from the clusters representing all 5 primitives for 5000

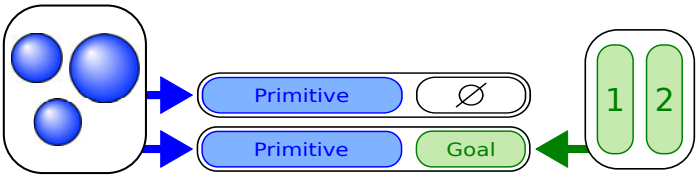


Fig. 3. Input vectors with context. Schematic of input vectors without (top) and with (bottom) goal information. The first component representing motion primitives is sampled from the hyperspheres as detailed in *Methods*. Second component, when used, is chosen from a set of two randomly generated vectors. When not used, it is set to constant values throughout all simulations.

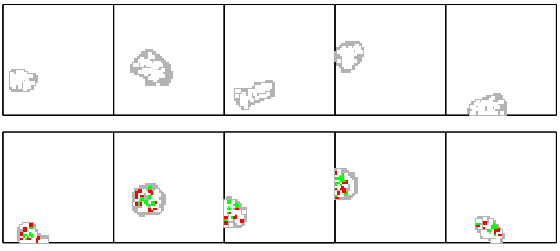


Fig. 4. Neurons reacting to primitives shown in specific goal contexts. Every panel shows SOM response to data from one primitive, with the top (bottom) row obtained from a map that was context unaware (aware). Grey nodes represent all nodes in the SOM that encode the presented primitive. Coloured nodes only respond to a primitive if it is shown in one context (indicated by the colour). They are *specialised*. White nodes respond to a primitive in both contexts. They are *general*. In a context-unaware map, all responding nodes were general. If primitives are shown in different goal contexts, some (but not all) nodes develop a specialised preference for only one of them.

iterations. The additional context vector is chosen randomly for every input. We are first interested in the effect of this additional information on the overall organisation of the SOM and find little change (Fig. 2E). On average over 100 runs, maps that are exposed to five primitives with no contextual information end up allocating $3.92 \pm 1.20\%$ of their overall area to each of the primitives. With contextual information, this changes insignificantly to $3.91 \pm 1.22\%$. The additional information does thus not require more nodes to encode each primitive.

We then test how, if at all, the contextual information is encoded within the SOM. For this, we generate 100 data-points from every motion primitive. This input is given to the map with each of the possible context vectors attached in turn and the responding nodes are determined. We call *specialised* nodes those that only respond if a given primitive is shown in a specific context. If the same node responds to a primitive independently of the context vector, we call it a *general* node. Prior to exposure to primitives in different contexts, all responding nodes are general (top row in Fig. 4). This is important insofar as it shows that nodes are not accidentally responsive to context.

However, if the SOM has in fact been exposed to primitives in the context of different goals, we find that the neurons encoding this primitive have further specialised and some of them only fire if the primitive is observed in a certain context (bottom row Fig. 4). Interestingly, some neurons remain general (white nodes). We repeat the above simulation 100 times and find that on average, about one quarter ($24.44 \pm 6.05\%$) of the nodes that responded to the given stimuli are specialised, whereas the remaining ones are general. The SOM thus reproduces experimental findings in Macaque monkeys, in which both highly goal-specific and merely primitive-specific mirror neurons were found (Fogassi et al., 2005). This is remarkable since the mere addition of contextual information was sufficient to produce this behaviour and may suggest that the organisation observed

in macaque monkeys may be incidental rather than a purposeful encoding of context.

3.4 The Utility of Motor Babbling during Infancy of the Map

Finally, we investigate the effects of the infancy period of the map during which it is exposed to random data-points sampled only from the clusters representing body parts. Although one can see this as an analogue to motor babbling, it is clearly not a requirement for a SOM to be able to learn the motion primitives. It is therefore interesting to evaluate the effect of this design choice on the performance of the map.

To this effect, we run a map for a total of 10000 iterations. During the initial 5000, three primitives from the first body part and random data from the second body part are presented to the map. For the remaining 5000 runs, we add two more primitives from the first body part. We compare two cases. In the first, there is no infancy period at all, meaning that the learning rate and neighbourhood size immediately start at their smallest constant values. The second case has a "normal" infancy period during which the parameters initially shrink as previously. The first scenario thus investigates a map that has neither infancy nor motor babbling while the second scenario deals with a map that merely loses the motor babbling. The final possible scenario - no infancy but an initial phase of motor babbling - is not investigated in detail as it does not differ from the previous setup once the babbling phase is over.

For all maps, we calculate the mean area that nodes encoding (1) the body parts and (2) specific primitives over 100 runs. We compare these results to the previously used setup (in which the 10000 runs are preceded by an infancy phase of motor babbling). The results are summarised in Tab. 1. Maps that have neither infancy nor babbling have the smallest areas encoding primitives and in general terms, the largest amount of the available map space is left unused even

Table 1. Mean area (in % of map size) occupied by nodes representing (1) body parts and (2) primitives presented either from the start or beginning at 5000 iterations for different maps. The table illustrates the benefits of the initial motor babbling during infancy of the map: (1) low variation between runs and (2) facilitation of the learning of new primitives.

		Body parts		Motion primitives	
Iterations	Map Type	First	Second	Initial	Subsequent
5000	No Infancy	9.67 ± 1.74	4.79 ± 1.04	2.79 ± 0.64	-
	Normal	64.87 ± 24.05	33.18 ± 24.00	19.32 ± 14.47	-
	With babbling	50.21 ± 3.07	48.53 ± 2.99	4.76 ± 0.90	-
10000	No Infancy	14.73 ± 2.79	6.28 ± 1.49	2.89 ± 0.83	2.13 ± 0.51
	Normal	65.01 ± 24.03	33.35 ± 23.92	18.09 ± 13.59	2.65 ± 1.09
	With babbling	50.65 ± 3.15	48.17 ± 3.06	5.42 ± 1.16	4.00 ± 1.38

after 10000 iterations. Maps with an infancy phase dedicate almost 60% of the overall area to the first three primitives, but primitives learned at a later point in time are represented by a much lower area. Large standard deviations further indicate that results can vary substantially between runs.

In comparison, our previous setup using maps with babbling during an infancy phase produces results with very small standard deviations, thus ensuring that the results are robust and not likely to differ strongly between runs. In addition, primitives learned at a later time are represented by areas that are only slightly smaller than those representing primitives learned initially (4.00 ± 1.38 vs 5.42 ± 1.16% respectively). It is perhaps more accurate to compare the area of the later primitives with the area occupied by the initially learned primitives after 5000 runs and in this case, the difference is even smaller. In contrast, maps lacking motor babbling encode the initial primitives with an area approximately 7 times larger than that of the primitives learned at a later stage. Maps lacking both infancy and motor babbling end up encoding primitives with areas only half as large as maps with motor babbling and infancy. The initial motor babbling phase thus facilitates the learning of new primitives compared to maps lacking this phase.

4 Discussion

4.1 Online Learning of Motion Primitives

We have presented a simple self-organising map that is capable of learning new knowledge at any point in time. When presented with input intended to represent motion primitives, it has been able to adapt and learn novel primitives as they were perceived. Interestingly, when motion primitives were presented in different contexts, some nodes of the map implicitly began to be sensitive to this context, only responding to a certain primitive if it was presented within a specific context. This replicates neurophysiolgocial findings in the mirror neuron system (Fogassi et al., 2005) and may imply that the goal-specificity of some mirror neurons could merely be incidental.

We made a design choice that the map should initially only be exposed to data akin to motor babble (*e.g.* Meltzoff and Moore, 1997; Der and Martins, 2006). We subsequently investigated the consequences of this choice and found that it both facilitated the learning of new motion primitives in later stages and reduced the variability between results in repeated simulations. This is in line with the view that "embodied sensorimotor coordination is vital for the self-structuring of the sensor space necessary for categorization and higher level cognition" (Der and Martins, 2006).

4.2 Further Work

During the simulations of the map, we observed some expected plasticity depending on the primitives currently being presented to the map. This plasticity

is due to the fact that the map never completely stops learning. However, further research is needed to investigate the effects thereof in more detail and thus to determine the best values for the neighbourhood size and in particular the learning rate of the map.

Finally, the work presented here only addresses the formation of motion primitives but not their connection into chains, as required by the original Chain model (Chersi et al., 2006). The next step in our work will be to extend the map with an associative memory between winning nodes to form this connectivity.

Acknowledgments

This work was supported by the European Commission FP7 project ROSSI (www.rossiproject.eu), Grant agreement no. 216125.

References

Bonaiuto, J., Rosta, E., Arbib, M.A.: Extending the mirror neuron system model, I. Biological Cybernetics 96, 9–38 (2007)

Chersi, F., Mukovskiy, A., Fogassi, L., Ferrari, P.F., Erlhagen, W.: A model of intention understanding based on learned chains of motor acts in the parietal lobe. In: Proceedings of the 15th Annual Computational Neuroscience Meeting, Edinburgh, UK (2006)

Der, R., Martins, G.: From motor babbling to purposive actions: Emerging self-exploration in a dynamical systems approach to early robot development. In: From Animals to Animats, vol. 9, pp. 406–421 (2006)

Erlhagen, W., Mukovskiy, A., Chersi, F., Bicho, E.: On the development of intention understanding for joint action tasks. In: Proceedings of the 6th IEEE International Conference on Development and Learning. Imperial College, London (2007)

Fogassi, L., Ferrari, P.F., Gesierich, B., Rozzi, S., Chersi, F., Rizzolatti, G.: Parietal lobe: from action organization to intention understanding. Science 308, 662–667 (2005)

Hickok, G.: Eight problems for the mirror neuron theory of action understanding in monkeys and humans. Journal of Cognitive Neuroscience 21(7), 1229–1243 (2008)

Koenig, N., Matarić, M.J.: Behavior-based segmentation of demonstrated tasks. In: Proc. of Int. Conf. on Development and Learning (2006)

Kohonen, T.: Self-organizing maps. Springer, Heidelberg (1997)

Kulić, D., Nakamura, Y.: Scaffolding on-line segmentation of fully body human motion patterns. In: IEEE Int. Conf. on Intelligent Robots and Systems, pp. 2860–2866 (2008)

Lieberman, J., Breazeal, C.: Improvements on action parsing and action interpolatin for learning through demonstration. In: IEEE/RAS Int. Conf. on Humanoid Robots, pp. 342–365 (2004)

Meltzoff, A.N., Moore, K.M.: Explaining facial imitation: a theoretical model. Early Development and Parenting 6(2), 179–192 (1997)

Oztop, E., Kawato, M., Arbib, M.A.: Mirror neurons and imitation: A computationally guided review. Neural Networks 19, 254–271 (2006)

Pomplun, M., Matarić, M.J.: Evaluation metrics and results of human arm movement imitation. In: IEEE-RAS Int. Conf. on Humanoid Robotics (2000)

Rizzolatti, G., Fadiga, L., Gallese, V., Fogassi, L.: Premotor cortex and the recognition of motor actions. Cognitive Brain Research 3(2), 131–141 (1996)

Wermter, S., Elshaw, M., Farrand, S.: A modular approach to self-organization of robot control based on language instruction. Connection Science 15(2-3), 73–94 (2003)

Yamashita, Y., Tani, J.: Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. PLoS Computational Biology 4(11), 1–18 (2008)

A Computational Model of Integration between Reinforcement Learning and Task Monitoring in the Prefrontal Cortex

Mehdi Khamassi*, René Quilodran, Pierre Enel,
Emmanuel Procyk, and Peter F. Dominey

INSERM U846 SBRI, Bron, France
mehdi.khamassi@inserm.fr

Abstract. Taking inspiration from neural principles of decision-making is of particular interest to help improve adaptivity of artificial systems. Research at the crossroads of neuroscience and artificial intelligence in the last decade has helped understanding how the brain organizes reinforcement learning (RL) processes (the adaptation of decisions based on feedback from the environment). The current challenge is now to understand how the brain flexibly regulates parameters of RL such as the exploration rate based on the task structure, which is called meta-learning ([1]: Doya, 2002). Here, we propose a computational mechanism of exploration regulation based on real neurophysiological and behavioral data recorded in monkey prefrontal cortex during a visuo-motor task involving a clear distinction between exploratory and exploitative actions. We first fit trial-by-trial choices made by the monkeys with an analytical reinforcement learning model. We find that the model which has the highest likelihood of predicting monkeys' choices reveals different exploration rates at different task phases. In addition, the optimized model has a very high learning rate, and a reset of action values associated to a cue used in the task to signal condition changes. Beyond classical RL mechanisms, these results suggest that the monkey brain extracted task regularities to tune learning parameters in a task-appropriate way. We finally use these principles to develop a neural network model extending a previous cortico-striatal loop model. In our prefrontal cortex component, prediction error signals are extracted to produce feedback categorization signals. The latter are used to boost exploration after errors, and to attenuate it during exploitation, ensuring a lock on the currently rewarded choice. This model performs the task like monkeys, and provides a set of experimental predictions to be tested by future neurophysiological recordings.

1 Introduction

Exploring the environment while searching for resources requires both the ability to generate novel behaviors and to organize them for an optimal efficiency. Besides, these behaviors should be regulated and interrupted when the goals of

* Corresponding author.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 424–434, 2010.

exploration have been reached: a transition towards a behavioral state called exploitation should then be implemented. Previous results on neural bases of these functions in the frontal cortex showed crucial mechanisms that could participate both to reinforcement learning processes [2] and to the auto-regulation of exploration-exploitation behaviors [3]. Several computational and theoretical models have been proposed to describe the collaborative functions of the anterior cingulate cortex (ACC) and the dorsolateral prefrontal cortex (DLPFC) - both belonging to the prefrontal cortex - in adaptive cognition [4, 5, 6]. Most models are based on the hypothesized role for ACC in performance monitoring based on feedbacks and of DLPFC in decision-making. In exploration, challenging, or conflicting situations the output from ACC would trigger increased control by the DLPFC. Besides, several electrophysiological data in non human primates suggest that modulation of this control within the ACC-DLPFC system are subserved by mechanisms that could be modelled with the reinforcement learning (RL) framework [2, 7, 8]. However, it is not clear how these mechanisms integrate within these neural structures, and interact with subcortical structures to produce coherent decision-making under explore-exploit trade-off.

Here we propose a new computational model to formalize these frontal cortical mechanisms. Our model integrates mechanisms based on the reinforcement learning framework and mechanisms of feedback categorization - relevant for task-monitoring - in order to produce a decision-making system consistent with behavioral and electrophysiological data reported in monkeys. We first employ the reinforcement learning framework to reproduce monkeys exploration-exploitation behaviors in a visuo-motor task. In a second step, we extract the main principles of this analysis to implement a neural-network model of fronto-striatal loops in learning through reinforcement to adaptively switch between exploration and exploitation. This model enabled to reproduce monkeys behavior and to draw experimental predictions on the single-unit activity that should occur in ACC and DLPFC during the same task.

2 Problem-Solving Task (PST)

We first use behavioral data recorded in 2 monkeys during 278 sessions (7656 problems $\equiv$ 44219 trials) of a visuo-motor problem-solving task that alternates exploration and exploitation periods (see Fig.1A). In this task, monkeys have to find which of four possible targets on a screen is associated with reward. The task is organized as a sequence of problems. For each problem, one of the targets is the correct choice. Each problem is organized in two succesive groups of trials; starting with search trials where the animal explores the set of targets until finding the rewarded one; Once the correct target is found, a repetition period is imposed so that the animal repeats the correct response at least three times. Finally, a cue is presented on the screen and indicates the end of the current problem and the beginning of a new one. Data used here were recorded during electrophysiological experiments, after animals had experienced a pre-training stage. Thus, monkeys are particularly overtrained and optimal on this

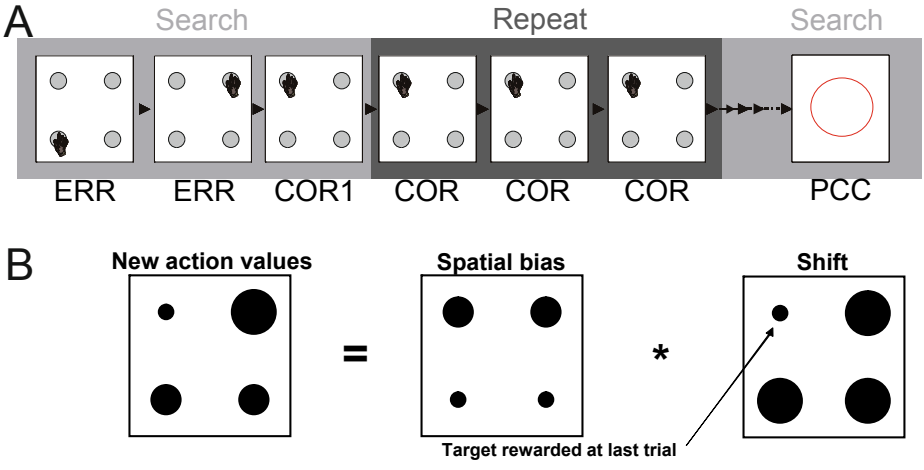


Fig. 1. Monkeys had to find by trial and error which target, presented in a set of four, was rewarded. A) Monkeys performed a set of trials where they chose different targets until the solution was discovered (search period). Each block of trials (or problem) contained a search period and a repetition period during which the correct response was repeated at least three times. A Problem-Changing Cue (PCC) is presented on the screen to indicate the beginning of a new problem. B) Action value reset in the model at the beginning of each new problem.

task. Monkey choice, trial correctness and problem number are extracted and constitute the training data for the reinforcement learning model.

3 Behavior Analysis with the Reinforcement Learning Framework

3.1 Theoretical Model Description

We use the reinforcement learning framework as a model of the way monkeys learn to choose appropriate targets by trial-and-error [9]. The main assumption in such framework is that monkeys try to maximize the amount of reward they will get during the task. This framework assumes that animals keep estimated action values (called Q-values) for each target (i.e. $Q(UL)$, $Q(LL)$, $Q(UR)$ and $Q(LR)$). It also assumes that monkeys decide which action to perform depending on these values, and update these values based on feedbacks (i.e. the presence/absence of reward) at the end of each trial. We used a Boltzmann softmax rule for action selection. The probability of choosing an action a (either UL, LL, UR or LR) is given by

$$P(a) = \frac{exp(\beta Q(a))}{\sum_b exp(\beta Q(b))} \quad (1)$$

where β is an exploration rate ($\beta \geq 0$). In short, when β is low (close to 0), the contrast between action values is decreased, thus increasing the probability

to select a non optimal action (exploration). When β is high, the contrast is high and decision-making becomes more greedy. We differently use β_S and β_R parameters on *search* and *repetition* trials so as to allow different shapes of the Boltzmann function on these two periods. In other words, β_S and β_R were used as two distinct free parameters to see if they would converge on different values, hence indicating meta-learning through the use of two different exploration rates by the animal.

At the end of each trial, the action value is updated by comparing the presence/absence of reward r with the value expected from the performed action according to the following equation

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r - Q(a)) \tag{2}$$

where α is the learning rate of the model ($0 \leq \alpha \leq 1$). Similarly to previous work [2], we generalize reinforcement learning to also update each non chosen action b according to the following equation

$$Q(b) \leftarrow (1 - \kappa) \cdot Q(b) \tag{3}$$

where κ is a forgetting rate ($0 \leq \kappa \leq 1$).

Finally, we add an action-value reset at the beginning of each new problem, when a *PCC* cue appears on the screen. This is based on the observation that monkeys almost never select the previously rewarded target, and have individual spatial biases in their exploration pattern: they often start exploration by choosing the same preferred target (see Fig.1B).

3.2 Simulation of the RL Model on Monkey Behavioral Data

The reinforcement learning model is simulated on monkey data, that is, at each trial, the model chooses a target, we store this choice, then we look at the choice made by the animal, and the model learns as if it had made the same choice. At the next trial, the model makes a new choice, and so on. At the end, we compare the sequence of choices made by the model with the monkey's choices. With this method, the model learns based on the same experience as the monkey. Thus the choice made at trial t becomes comparable to the animal's choice at the same trial because it follows the same trial history $\{1...t-1\}$. For each behavioral session, we optimize the model by finding the set of parameters that provides the highest likelihood of fitting monkeys choices. This optimization leads to an average likelihood of 0.6537 per session corresponding to 77% of the trials where the model predicted the choice the monkeys actually made. Fig.2 shows simulation results on a sample of 100 trials for 1 monkey.

Interestingly, we find that the distribution of each session's β_S used to set the exploration rate during *search* periods is significantly lower than the distribution of β_R used for *repetition* periods (ANOVA test, $p < 0.001$). The mean β_S equals 5.0 while the mean β_R equals 6.8. This reveals a higher exploration rate in monkeys' choices during *search* periods. In addition, we found an average learning rate around 0.9 for the two monkeys and a smaller forgetting rate (mean: 0.45).

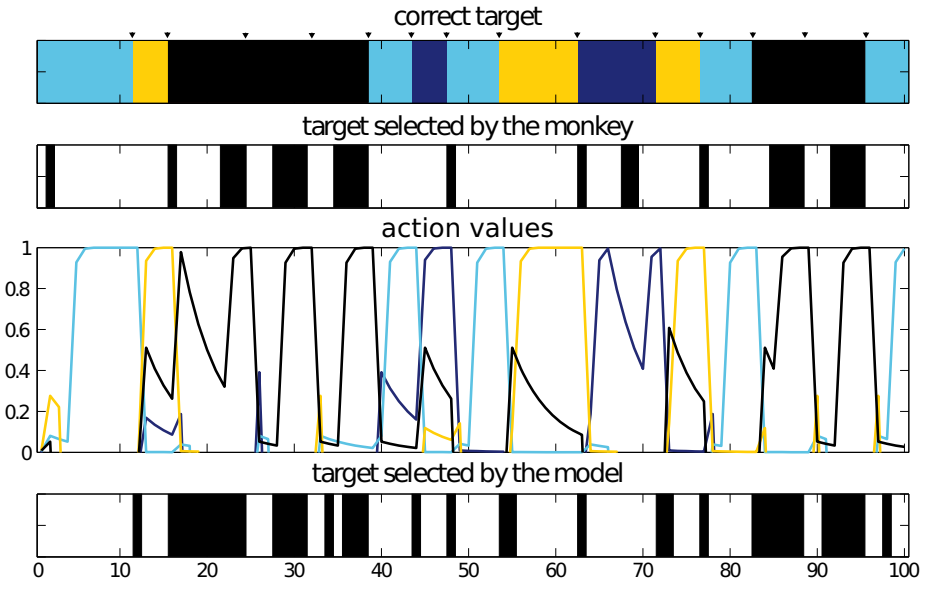


Fig. 2. Simulation of the reinforcement learning model on 100 trials. Each color is associated with a different target (UL, LL, UR, LR). The top line denotes the problem sequence experienced by both the monkey and the model. Black triangles indicate cued problem changes. The second line shows the monkey's choice at each trial. Curves show the temporal evolution of action values in the model. Non selected target have their value decrease according to a forgetting process. These curves also show the action value reset at the beginning of each problem, the decrease of incorrect selected targets value, and the increase of the correct targets value once selected by the animal. The bottom of the figure shows choices made by the model based on these values.

This suggests that reinforcement learning mechanisms in the monkey brain are regulated by parameters that were learned from the task structure. In contrast, raw reinforcement learning algorithms such as Q-learning usually employs a single fixed β value, and need to make errors before abandoning the optimal action and starting a new exploration phase. In the next section, we extract these principles to propose a neural-network model integrating such reinforcement learning and task monitoring mechanisms.

4 Neural Network Model

4.1 Reinforcement Learning Processes

We propose a neural network model in order to propose a computational hypothesis concerning the modular organization of these processes within cortical networks. Our model extends previous models of cortico-striatal loops which are

known to be crucial neural substrates for reward-based learning and decision-making [10, 11]. The principle novelty here is to have the integration of reinforcement learning and task monitoring within the ACC-DLPFC system that produces explore-exploit behaviors. In our neural network, dopaminergic (DA) neurons from the Ventral Tegmental Area (VTA) compute a reward prediction error following equation 2, consistently with DAs supposed role in reinforcement learning [12]. DA signals are used to update action values encoded within the ACC, consistently with previous work [7]. These action values are then sent to DLPFC which makes decision of the target to choose and biases action selection in the striatum. Similarly to classical basal ganglia models but not detailed here, appropriate action in the striatum is selected by desinhibiting the thalamus through the substantia nigra pars reticulata [13, 14]. Finally, the thalamus projects to the motor cortex which drives behavioral output, and which sends efferent copies to the ACC in order to update only relevant action through reinforcement learning (fig.3).

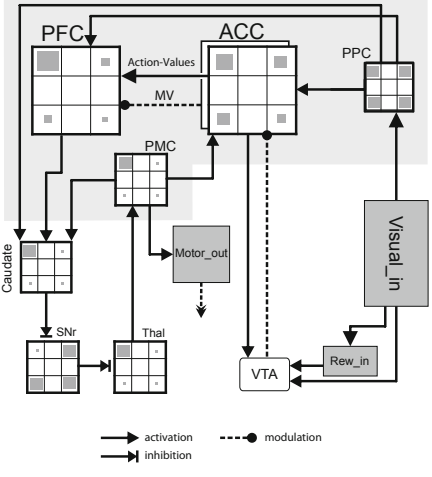


Fig. 3. Neural network model. Visual input (targets presented on the screen) is processed by the parietal cortex and sent to the ACC-DLPFC system (colored in grey) which performs reinforcement learning (RL) to rapidly adapt choices by trial-and-error during the search period. A specific visual signal is used to indicate reward delivery, representing juice obtained by monkeys. According to RL principles, this visual signal is translated by VTA into a reinforcement signal which changes action values within the ACC. In parallel, this reinforcement signal is used to regulate the level of exploration with MV.

With such organization, the system is yet purely dedicated to reinforcement learning. In order to add task monitoring mechanisms, we take inspiration from additional results measured in the PST task. In [8], reaction times were observed to decrease gradually after errors during the search period, to raise sharply after the first correct trial, and to remain high during repetition (fig.4A-B). The exact

opposite pattern was observed at the level of the average activity measured in DLPFC neurons ([15]; fig.4C). These patterns suggest an integration of feedbacks used to update a control or attentional level, and a state change of the system from exploration to exploitation. This resembles the vigilance level employed in [16]'s theoretical model to modulate the level of activation of a global workspace in charge of solving the task. In the next paragraph, we provide a computational hypothesis on the way the ACC could evaluate such kind of vigilance level to modulate the level of control and exploration in DLPFC.

4.2 Integrating Task Monitoring Signals within the Neural Network

In order to regulate exploration based on feedbacks obtained from the environment, we add to our ACC component a second population of neurons dedicated to feedback categorization as described in the monkey ACC in the same task [8]. In our model, these neurons receive the same dopaminergic reward prediction error signals as ACC action value neurons. The difference resides in the influence of such DA signals on feedback categorization neurons. The latter either are inhibited by DA signals and thus produce responses to errors (ERR) or are excited by DA signals and thus produce responses to correct trials (COR). The high learning rate used in the model to fit behavioral data in section 3 results in a strong response of COR neurons to the first reward and in a smaller response to subsequent ones. This produces neurons responding to the first correct trials (COR1) as observed by [8]. Fig.5 shows a simulation of these neurons response patterns. COR and ERR signals are then used to update a modulatory variable (MV) according to the following equation:

$$MV \leftarrow MV + \alpha^{+} \cdot \delta_{+} + \alpha^{-} \cdot \delta_{-} \tag{4}$$

Where δ_{+} and δ_{-} represent the response of correct and error neurons respectively, while α^{+} and α^{-} are synaptic weights set to $-\frac{5}{2}$ and $\frac{1}{4}$ for the considered task. MV is constrained between 0 and 1. This equation makes MV be:

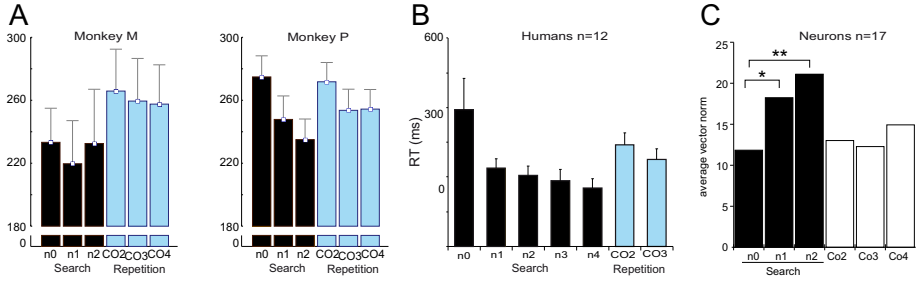


Fig. 4. (A-B) Reaction times during the PST task show a progressive decrease along the search period, and a sharp change during repetition. Adapted from [8]. C) Average activity in the dorsolateral prefrontal cortex show a similar (inversed) pattern. Adapted from [15].

- sharply decreased after a correct trial (COR) ;
- increased after an error (ERR);
- increased after presentation of the problem changing cue (PCC). Although we did not yet study how the model works during pretraining phases of this task (*i.e.* habituation phases preceding electrophysiological recordings), we observed that before the animal learns what the PCC means, the presentation of this cue is very often followed by an error - because the animal persists in repeating the same choice while the problem has changed. Thus we consider here that the PCC has been associated to an error during the pretraining phase and consequently produces an increase of MV each time it occurs during the task.

Importantly, MV is used to modulate the exploration rate and the gain in the DLPFC. The first function is assured in the following manner:

$$\beta_t = \frac{\omega_1}{(1 + exp(\omega_2 * (1 - MV_t) + \omega_3))} \tag{5}$$

Where ω_1, ω_2 and ω_3 are parameters respectively equal to 10, -6 and 1. Such function is a sigmoid which inverses the tendency of MV (see fig.5) and transforms a value between 0 and 1 (for MV) into a value between 0 and 10 (for β) according to table 1.

Table 1. MV effect on β following equation (5) with a = 10, b = -6, c = 4.4

MV	0.00	0.25	0.50	0.75	1.00
β	9.9	9.7	8.8	6.2	2.7

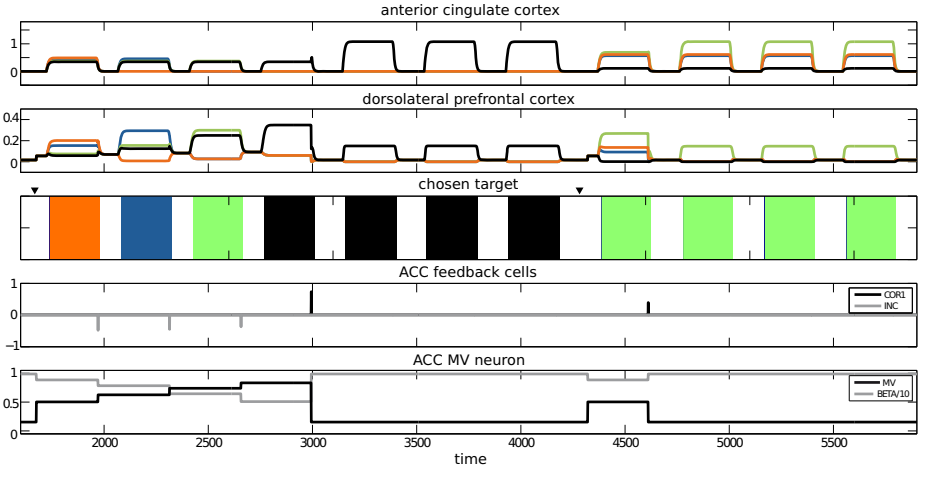


Fig. 5. Neural network model simulation during 2 consecutive problems. Black triangle indicate cued problem changes

The second function is assured by weighting DLPFCs activity by multiplying it by MV (which is always inferior or equal to 1). As a consequence, a low MV produces a high β (almost no exploration) and a low DLPFC activity so as to focus and lock the DLPFC on performing the action with the highest value; whereas a high MV produces a low β (higher stochasticity in decision-making, thus more exploration) and a high activity in DLPFC so as to enable the performance of non optimal actions.

The model can perform the task like monkeys, alternating between search and repetition phases. Fig.5 shows the activation of different neurons in the model during a sample simulation.

5 Discussion and Conclusion

We implemented a reinforcement learning model that can monitor exploration-exploitation trade-off in a monkey visuo-motor task. The model helped us formally describe monkey behavior in a task involving clear distinction between *search* and *repetition* trials. In addition, the model is based on existing anatomical and physiological properties of the monkey brain. Properties of MV modulation in our model are consistent with data in human and in animal showing a higher involvement of ACC-DLPFC when the task is demanding or when it involves conflict resolution [17]. Moreover, our results are consistent with previous electrophysiological work suggesting a role of the ACC-DLPFC system in performance monitoring [5], and in reinforcement learning [2, 7]. Our model enables to draw a list of experimental predictions that have to be tested by simultaneously recording Anterior Cingulate Cortex (ACC) and dorsolateral Prefrontal Cortex (DLPFC) neurons in this task:

1. There should exist MV neurons in ACC. Such MV neurons would have a particular profile of activity: progressive increase of activity during the search period, drop of activity after the first correct response, activity remaining low during the repetition period (as shown on fig.5).
2. MV modulation should effect only on DLPFC action value neurons and not on ACC action value neurons. In the model, we made the choice to keep original action values (that is, not altered by the MV modulation) in the ACC so as to have part of the system properly perform the reinforcement learning algorithm without perturbation, so as to ensure convergence.
3. There should be a higher global spatial selectivity - which reflects the degree to which neurons discriminate choices of spatial targets on the touch screen [15] - in DLPFC than in ACC due to the decision-making process based on the softmax function (which increases contrasts between action values when β is high).
4. There should be an increase of spatial selectivity in DLPFC but not in ACC during the repetition period. Such increase of spatial selectivity in DLPFC neurons in the model is due to the modulatory effect of MV on the β parameter used in the softmax function.

Performance of the neural-network model enables a robotics arm to reproduce monkey behavior in front of a touch screen. Such a pluridisciplinary approach provides tools both for a better understanding of neural mechanisms of decision making and for the design of artificial systems that can autonomously extract regularities from the environment and interpret various types of feedbacks (rewards, feedbacks from humans, etc...) based on these regularities to appropriately adapt their own behaviors.

Future work will consist in modelling how RL parameters are progressively set during familiarization with the environment. Such goal can be achieved by using the model to predict day-by-day behavior observed during monkey pretraining. This will help us understand the dynamics of meta-learning which enable animals in this task to autonomously learn that a high learning rate is relevant and that clear transition between exploration and exploitation are required - based on the extracted structure of task.

Acknowledgments. This work was supported by the French National Research Agency (ANR Amorces) and the European Community Contract FP7-231267 (EU Organic Project).

References

[1] Doya, K.: Metalearning and neuromodulation. Neural Netw. 15(4-6), 495–506 (2002)
[2] Barraclough, D., Conroy, M., Lee, D.: Prefrontal cortex and decision making in a mixed-strategy game. Nat. Neurosci. 7(4), 404–410 (2004)
[3] Procyk, E., Tanaka, Y., Joseph, J.: Anterior cingulate activity during routine and non-routine sequential behaviors in macaques. Nat. Neurosci. 3(5), 502–508 (2000)
[4] Aston-Jones, G., Cohen, J.: Adaptive gain and the role of the locus coeruleus-norepinephrine system in optimal performance. J. Comp. Neurol. 493(1), 99–110 (2005)
[5] Brown, J., Braver, T.: Learned predictions of error likelihood in the anterior cingulate cortex. Science 307, 1118–1121 (2005)
[6] Dosenbach, N.U., Visscher, K.M., Palmer, E.D., Miezin, F.M., Wenger, K.K., Kang, H.C., Burgund, E.D., Grimes, A.L., Schlaggar, B.L., Peterson, S.E.: A core system for the implementation of task sets. Neuron 50, 799–812 (2006)
[7] Matsumoto, M., Matsumoto, K., Abe, H., Tanaka, K.: Medial prefrontal cell activity signaling prediction errors of action values. Nat. Neurosci. 10, 647–656 (2007)
[8] Quilodran, R., Rothe, M., Procyk, E.: Behavioral shifts and action valuation in the anterior cingulate cortex. Neuron 57(2), 314–325 (2008)
[9] Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
[10] Dominey, P., Arbib, M., Joseph, J.: A model of corticostriatal plasticity for learning oculomotor associations and sequences. Journal of Cognitive Neuroscience 7(3), 311–336 (1995)
[11] Khamassi, M., Martinet, L., Guillot, A.: Combining self-organizing maps with mixture of epxerts: Application to an Actor-Critic model of reinforcement learning in the basal ganglia. In: Proceedings of the 9th International Conference on the Simulation of Adaptive Behavior (SAB), Rome, Italy, pp. 394–405. Springer, Heidelberg (2006)

[12] Schultz, W., Dayan, P., Montague, P.: A neural substrate of prediction and reward. Science 275(5306), 1593–1599 (1997)
[13] Gurney, K., Prescott, T., Redgrave, P.: A computational model of action selection in the basal ganglia. I. A new functional anatomy. Biol. Cybern. 84(6), 401–410 (2001)
[14] Girard, B., Cuzin, V., Guillot, A., Gurney, K., Prescott, T.: A basal ganglia inspired model of action selection evaluated in a robotic survival task. Journal of Integrative Neuroscience 2(2), 179–200 (2003)
[15] Procyk, E., Goldman-Rakic, P.: Modulation of dorsolateral prefrontal delay activity during self-organized behavior. J. Neurosci. 26(44), 11313–11323 (2006)
[16] Dehaene, S., Changeux, J.: A neuronal model of a global workspace in effortful cognitive tasks. Proc. Natl. Acad. Sci. USA 95, 14529–14534 (1998)
[17] Cohen, J., Aston-Jones, G., Gilzenut, S.: A systems-level perspective on attention and cognitive control. In: Posner, M. (ed.) Cognitive Neuroscience of Attention, pp. 71–90. Guilford Publications, New York (2004)

Internal Models in the Cerebellum: A Coupling Scheme for Online and Offline Learning in Procedural Tasks

Jean-Baptiste Passot[1], Niceto Luque[2], and Angelo Arleo[1,⋆]

[1] CNRS - UPMC Univ Paris 6, UMR 7102, F75005, Paris, France
angelo.arleo@upmc.fr
[2] Dept of Computer Architecture and Technology, University of Granada, Spain

Abstract. The cerebellum plays a major role in motor control. It is thought to mediate the acquisition of forward and inverse internal models of the body-environment interaction [1]. In this study, the main processing components of the cerebellar microcomplex are modelled as a network of spiking neural populations. The model cerebellar circuit is shown to be suitable for learning both forward and inverse models. A new coupling scheme is put forth to optimise online adaptation and support offline learning. The proposed model is validated on two procedural tasks and the simulation results are consistent with data from human experiments on adaptive motor control and sleep-dependent consolidation [2, 3]. This work corroborates the hypothesis that both forward and inverse internal models can be learnt and stored by the same cerebellar circuit, and that their coupling favours online and offline learning of procedural memories.

1 Introduction

It is largely admitted that the cerebellum plays a major role in motor control (e.g. coordinating movements and making them accurate) by acquiring internal models of the body and the world [1, 4]. In motor control theory, internal models are divided into two groups identified as forward and inverse. The forward model predicts the sensory outcome of an action: it estimates the causal relationship between inputs to the system and its outputs. The inverse model works in the opposite direction, providing a motor command that causes a desired change in state [5]. Both forward and inverse models depend on the dynamics of the motor system and must adapt to new situations and modifications of the motor apparatus [6].

Although Darlot *et al.* (1996) [7] suggested that a forward model could be first formed in the cerebellar cortex and then converted to an inverse model, most of the existing studies on bioinspired control architectures have compared the advantages of one type of internal model against the other, debating on which of them is most likely to be implemented in the cerebellum [8, 9]. Very few works have investigated the benefits of coupling internal models [10, 11], and none has underlied the fact that internal model coupling would endow the system with offline learning capabilities. This is quite surprising, given that sleep is known to contribute to offline consolidation and enhancement of motor adaptation capabilities in humans [12], and that the cerebellum is undoubtedly implied in these adaptation processes [13].

⋆ Corresponding author.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 435–446, 2010.

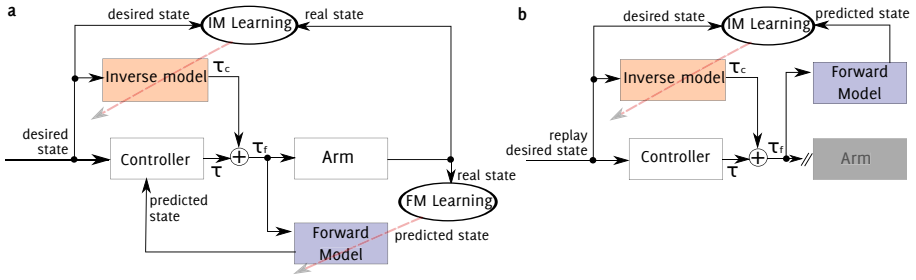


Fig. 1. *Coupling scheme for online and offline motor learning*. (a) Online adaptation. The arm controller receives the desired state and maps it onto a motor command (τ). The desired state is also sent to the inverse model that acts as a feed-forward corrector and calculates the motor correction (τ_c). The resulting command (τ_f) is then sent to the arm actuators. By comparing the desired state against the sensed real state, the inverse model learns to reduce the error between desired and real arm positions. While the motor command τ_f is being sent to the arm, an efference copy of the order is also conveyed to the forward model that learns to predict the consequent future position of the arm. The predicted state is then sent to the arm controller that can recalculate a new trajectory if the expected position in the trajectory differs from the predicted one. Finally, the real state is used to adapt the forward model to mimic the motor apparatus of the arm. (b) Offline adaptation. During offline processing, sensory feedbacks (i.e. the real state signals driving forward and inverse model learning) are not available. Yet, if the forward model is at least partially learnt, the predicted state signals can be used to continue to train the inverse model.

This paper proposes a novel scheme to couple internal cerebellar models. The model is primarily validated on a closed-loop architecture to control the dynamics of a robotic arm. The overall coupling model is depicted in Fig. 1a, whereas the offline functioning of the learning scheme is presented in Fig. 1b, under the assumption that the sequence of actions performed during online training can be replayed offline. This hypothesis relies on earlier animal investigations that have explored the possibility that patterns of brain activity which are elicited during initial task training are replayed during subsequent sleep [14]. The model is prominently assessed on the rotation adaptation task used by Huber *et al.* (2004) [3] to study motor learning (both online and offline) in humans. Our numerical simulations investigate the benefits of using both internal models to improve online learning capabilities, and they evaluate to what extent the proposed coupling scheme can explain the experimental findings on offline learning occurring during sleep [3]. Second, the model is also validated on a significantly different motor adaptation task, proposed by Walker and Stickfold 2004, in which subjects have to type simple numerical sequences on a computer keyboard [2]. Again, both the online and offline learning capabilities of the model are compared to experimental data.

2 Methods

2.1 Cerebellar Microcomplex Model

The cerebellar microcomplex circuit is modelled as a network of populations of spiking neurons (Fig. 2) and simulated using an event-driven simulation scheme [15]. Mossy

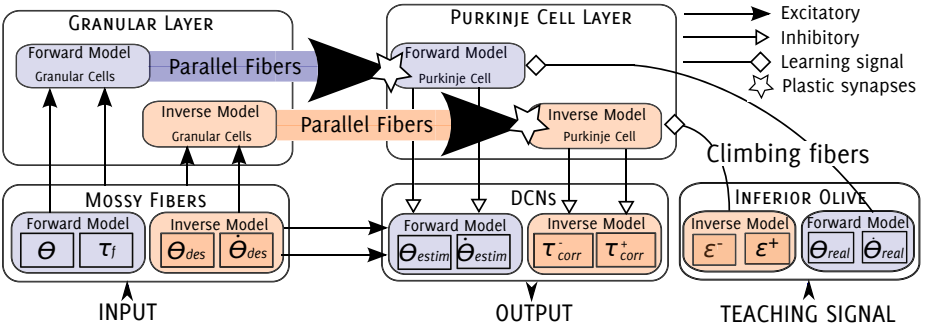


Fig. 2. The cerebellar microcomplex model

fibres (MFs) are implemented as axons of a population of 1600 leaky integrate-and-fire neurones separated in two regions: Forward and Inverse Model, FM and IM, respectively. Their driving currents are determined by using radial basis functions spanning the input space uniformly. The MFs of the forward model carry sensory information (θ) and an efference copy of the motor command τ_f. The MFs of the inverse model convey desired joint position θ_{des} and velocity $\dot{\theta}_{des}$ [16]. Each MF region activates a corresponding population of 200 neurones in the deep cerebellar nuclei (DCN). Also, each MF region projects onto a cluster of 10.000 granule cells (GCs), producing a sparse representation of the input space. Each GC subpopulation activates in turn a population of 200 Purkinje cells (PCs), which send inhibitory projections onto DCN neurones. The firing of DCN provides the outputs of the model, i.e. the forward model estimate the future state of each joints (position θ_{estim} and velocity $\dot{\theta}_{estim}$) and the inverse model correction to be sent to the system (following an agonist-antagonist muscle representation, one population called agonist coding for the positive correction torque τ^{+}_{corr} and one population called antagonist coding for the negative correction torque τ^{-}_{corr}).

The firing rate of DCN units is mainly determined by the inhibitory action of PCs, which in turn are principally driven by the parallel fibre (PF) activity, axons of the GCs. Therefore, modifying the strength of the synapses between PFs and PCs results in changes of the input-output relation characterising the cerebellar system. Bidirectional long-term plasticity (i.e. potentiation, LTP, and depression, LTD) is modelled at the level of PF-PC synapses (see fig. 2, plastic synapses).

The LTP is implemented as a non-associative weight increase triggered by each GC spike, simulating the homosynaptic potentiation rule described by Lev-Ram [17]. As shown in equation 1, the weight of a GC_i -PC_j connection is increased by α every time GC i discharges ($\delta_{GC_i} = 1$):

$$w_{GC_i-PC_j}(t) = w_{GC_i-PC_j} + \alpha\delta_{GC_i}(t) \tag{1}$$

On the other hand, LTD is implemented as an associative weight decrease triggered by a spike from the inferior olive. This is the heterosynaptic rule described in 1982 by Ito [18]. This learning rule is presented in the equation 2 and uses a temporal kernel K

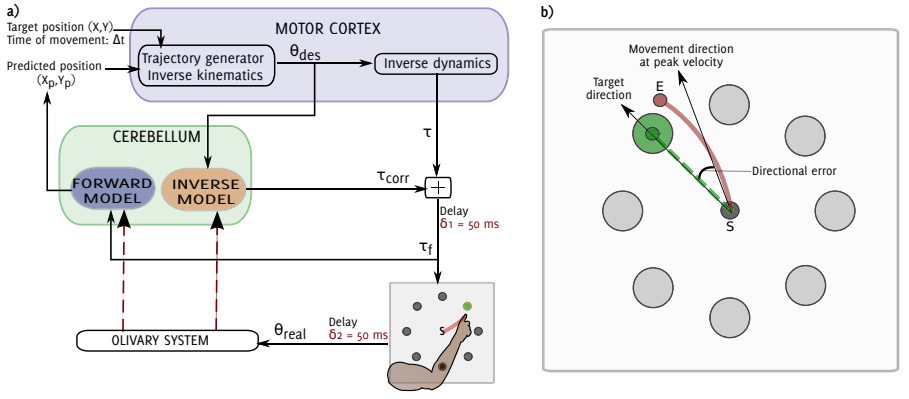


Fig. 3. *Overview of the biomimetic control architecture used to learn the rotation adaption task.* (a) Functional diagram of the controller. A desired trajectory to the highlighted target is computed by the trajectory generator and transformed in the joint-related reference frame via the inverse kinematics model. These desired arm states are used at each time step to compute a crude torque command. The desired state is also sent to the inverse model of the cerebellum, whose output is a corrective command to control arm movements. The cerebellar forward model receives an efference copy of the motor command, and predicts the future state (position and speed) and sends it to the trajectory generator. In the coupling scheme, both internal models drive the system. Trajectory error is sensed at the level of the limb and sent back to the system, which is used to compute the training signal at the olivary system level and conveyed by the climbing fibres to both internal models. (b) Experimental task and calculation of error. S: Starting point; E: Ending; Green dashed line: Ideal movement towards the target; Red line: actual movement.

which correlates each spike from the inferior olive with the past activity of connected GCs (see [19] for more information).

$$w_{GC_i-PC_j}(t_{IO}) = w_{GC_i-PC_j} - \beta \int_{-\infty}^{t_{IO}} K(t - t_{IO})\delta_{GC_i}(t)dt \qquad (2)$$

In my simulations, α and β parameters are set to 0.1 and 0.025, respectively.

The teaching signal is conveyed by a population of 400 inferior olive (IO) neurones, simulated to produce the climbing fibre projections targeting PCs. In the inverse model, the teaching signal relies on the estimated motor errors ε^- and ε^+, which are extracted from the discrepancy between the desired state of each joints and their real state (position θ_{real} and velocity $\dot{\theta}_{real}$). The latter variables are also used to drive the learning of the forward model. Normally, the real state of the joints is calculated in the central nervous system using congruent information from proprioceptive and visual sources. However, in our simulation, the positions and velocities of each joint are known variables, and corrective signals are directly derived algorithmically.

All the neuronal units of the microcomplex (i.e. GCs, PCs and DCNs) are modelled as conductance based, leaky integrate-and-fire units [20]. The irregular discharge of the IO is simulated by means of a Poisson spike-train generation model.

2.2 Global Architecture of the System

The global architecture of the generation of arm movement is illustrated in figure 3. We use the architecture described recently by Carrillo et al. in [19] to control a 2 joints simulated arm in real time. First, a minimum jerk model computes the desired smooth movement of the arm end-point toward the target positioned in (X, Y). The desired trajectory is expressed in Cartesian coordinates for the defined time of movement Δt. This desired movement is then transformed into arm-related coordinates: $\theta_{des}(t) = (\theta_{s,des}, \theta_{e,des})$ are the desired angular position of the shoulder and elbow. Since we use an arm with only two degrees of freedom, there are no redundancy or inversion problems. As there is a small chance that cerebellum could play a major role in dealing with these inconveniences, the model should still be valid for more complex arm devices.

These coordinates are the input of a crude inverse dynamic controller, which extracts a set of torque commands $\tau = (\tau_s, \tau_e)$, then sent to the articulations with a time delay $\delta 1 = 50ms$. All mathematical solutions of minimum jerk, inverse kinematics and dynamics model have been taken from [19]. An error is added to the minimum jerk model, through an added rotation of α degrees at each time step.

Two adaptive internal models encoded by our simulated cerebellum were included to the system, an inverse and a forward model. In the inverse model scheme, the desired angular position for both joints are sent to the cerebellum. The model then calculates a corrective torque signal $\tau_c = (\tau_{s,c}, \tau_{e,c})$ that compensates the rotation error during the realisation of the movement. The torque command applied to each articulation i is the sum of the torque τ_i computed by a basic inverse dynamics model according to the desired kinematic trajectory, and of the cerebellar correction ($\tau_{i,c}$): $\tau_f = \tau + \tau_c$. These two commands are then sent to the limbs with a delay $\delta 1$ = 50 ms. The error in the execution of movement is computed at the level of the arm, and sent back to the system with a delay $\delta 2 = 50$ ms. This error is mainly used to determine the learning signal conveyed by the inferior olive in order to produce anticipative motor corrections.

In the forward model scheme, the simulated cerebellum receives information about the current state of each articulation (the angular position of the elbow and the shoulder $\theta(t) = (\theta_s, \theta_e)$) and an efference copy of the torque command $\tau_f = \tau_{s,f} + \tau_{e,f}$. The model then predicts the future position and velocity of the articulations ($\theta(t) = (\theta_{s,est}, \theta_{e,est})$ and $\dot{\theta}(t) = (\dot{\theta}_{s,est}, \dot{\theta}_{e,est})$). The coordinates are transformed into Cartesian coordinates and sent to the trajectory generator (X_p, Y_p). This prediction is compared to the expected position of the arm. If there is a discrepancy between the two positions, the entire movement from the current estimated place is recomputed by the minimum jerk model. Because this process is supposed to require important neuronal resources, we limited its use at once every 100ms.

We fixed the duration of the motor execution to 0.7s for each movement, followed by a pause period of 0.3s during which joints are reset to their central position, and the activity of the models is allowed to fall back to normal. Because of this short execution time and taking into consideration the delay of the sensory feedback , we assume that high level motor correction (recalculation of the entire trajectory) could not be performed in the absence of a prediction of the sensory feedback signal.

2.3 Main Procedural Adaptation Task

The first simulated task is inspired from the rotation adaptation task realized by Huber *et al.* (2004) [3]. In this task human subjects have to move a handheld cursor on a two-dimensional tablet from a central starting point to one of eight targets displayed on a computer screen together with the cursor position. An opaque shield prevent subjects from seeing their arm. Targets are randomly highlighted at regular 1-s intervals. Unbeknown to the subjects, the cursor position is rotated anticlockwise relative to the hand position by a fixed angle (from 15 to 60°, depending on the trial, see details below).

We simulate the rotation adaptation experiment in order to study the possible role of internal model coupling for online learning and offline consolidation. The global architecture of the simulated arm controller is detailed in Fig. 3a. The ideal trajectory of the arm is computed according to the minimum jerk model ([21]). In order to learn the rotation adaptation task (i.e. to compensate for the unknown anticlockwise bias), two internal models encoded by the modelled cerebellar microcomplex of Fig. 2 form the core of the adaptive closed-loop controller.

The simulated experimental setup consists of a central position S and eight targets evenly distributed on a circle centred at position S (Fig. 3b). A trial is defined as the succession of 90 movements. Each movement starts from S and consists in realising a movement of the arm to one of the eight targets, which is randomly changed every second (1s corresponds to the duration of one target-directed movement in our simulation).

Similar to Huber *et al.* (2004) [3], the experimental protocol involves four incremental steps, for each of which the angular deviation (bias) is increased by 15°, within the range [15°, 60°] (see Fig. 4). Every step is composed of three trials. Three groups (FM, IM, CM_{off}) of ten individuals each are trained on the rotation adaptation task. The FM group uses a pure forward model to solve the task. The IM group employs a pure inverse model to adapt the response to the unknown angular bias. The CM_{off} group uses the coupling scheme.

Following the four training steps, the extent of rotation adaptation of the CM_{off} group is tested using an imposed bias of 60° (Trial 13 in Retest 1). Then, simulated agents are enabled to undergo an offline consolidation process consisting of a series of 48 trials. Subsequently, subjects are retested on a simple trial (Trial 14, retest 2). To assess the benefit of an offline consolidation process against a pure online learning, performances of the CM_{off} group are compared to a group of control subjects (CM_{CTRL}) which do not perform offline consolidation.

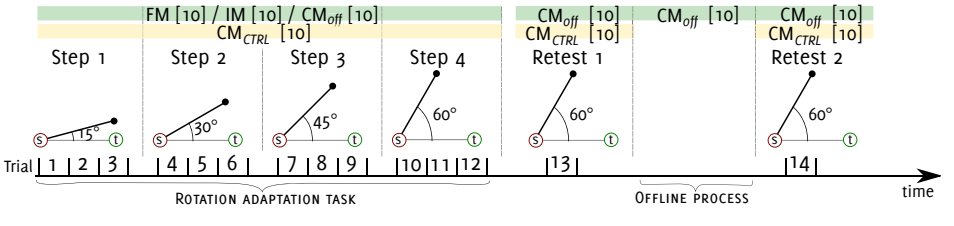


Fig. 4. The protocol of the rotation adaptation task and the offline learning task

Performances are measured by quantifying the directional error (see Fig. 3b), which corresponds to the angle between the line from the initial hand position (S) to the central position of the target (T) (dotted green line) and the line to the position of the hand at the peak outward velocity (solid line).

3 Results

3.1 Rotation Adaptation Task: Online Learning

Figs. 5a,b show the learning performances of the three groups FM, IM, and CM_{off} during the online training sessions (i.e. step 1-4, trial 1-12) of the rotation adaptation task. Fig. 5a displays three examples of arm trajectories towards three different targets. It shows qualitatively that, at the end of the trial 6, subjects using the coupling scheme (CM_{off}, green solid line) tend to perform better than both subjects using the inverse model only (IM, red dashed line) and subjects using the forward model only (FM, blue dotted line).

Fig. 5b quantifies these results for the entire set of training trials by averaging over all subjects. The mean normalised directional error is plotted as a function of training trials. The three groups of subjects learn to solve the rotation adaptation task and cope with the increasing unknown angular bias (from 15° to 60°) over training steps. Forward model subjects (FM, blue dotted curves) adapt quite rapidly but they reach a plateau after the 2^{nd} trial and do not further reduce the error over training. The passage to a new step (i.e. trials 4,7 and 10) does not have a significant impact on the FM performances and leads to a small increase of the directional error (+8% between trial 3 and 4; +6% between trial 6 and 7; and +2% between trial 9 and 10), which reflects the fast learning capabilities of FM subjects. However, subsequent training trials do not significantly decrease the error, which stabilises around 0.45-0.5 until the end of the training process (trial 12).

On the other hand, inverse model subjects (IM, red dashed curves) are slightly slower to adapt than FMs, but they succeed in minimising the directional error within each training session, going beyond the performances of purely FM subjects. Adaptation of IM subjects is rather characteristic and stereotyped during steps 2, 3, and 4 (i.e. for angular deviation ranging from 30° to 60°). Every time the angular bias is increased (i.e. trials 4,7 and 10), the performances of the inverse model are impaired and directional error increases (between 0.43 and 0.47). This result reflects the slow adaptation capability of the inverse model when facing new contexts. Then, during the 2^{nd} and 3^{rd} trials of each step, the inverse model adapts properly and the directional error decreases significantly (converging to accuracy values ranging from 0.25 to 0.3).

Finally, the subjects using the coupled internal models (CM_{off}, green solid curves) perform better than both IM and FM subjects along the entire training period, showing both fast adaptability and error reduction over time. The mean error rises slightly when the angular bias changes (i.e. trials 4,7 and 10) but then it decreases significantly and converges to values ranging from 0.15 to 0.2. Fig. 5b also displays the learning performances of human subjects (yellow data points) as reported by Huber *et al.* (2004) [3]. It is shown that the simulated CM_{off} subjects (green data) have online learning performances comparable to those of real subjects over the entire training process. These results suggest that the proposed coupling scheme, which favours the cooperation

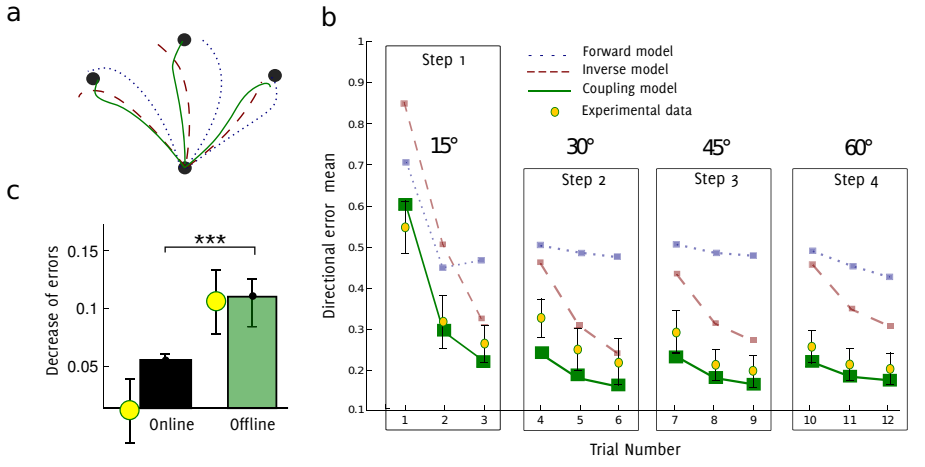


Fig. 5. *Rotation adaptation task. Simulation results for both online and offline learning and comparison with experimental human data.* (a) Example of three target-directed trajectories at the end of trial 6. The system has to adapt its dynamics to compensate for an angular bias of 30°. The blue dotted (resp. red dashed) lines indicate the sample solutions found by purely forward (resp. inverse) model simulated subjects, respectively. The green solid lines denote the trajectories obtained with the coupling scheme model. (b) Results of online learning. The coupling model (green solid curves) provides both rapid adaptation and appropriate convergence levels. Also, it reproduces the experimental data obtained with human subjects undertaking the same rotation adaptation task (yellow data, taken from Huber *et al.* (2004) [3]. (c) Offline learning results. The mean error is significantly reduced in the group of simulated subjects that undergo offline consolidation. The experimental results obtained with real subjects (offline corresponds to sleep-dependent consolidation) are shown in yellow (taken from Huber *et al.* (2004) [3]). ***Significant values, p<0.001.

between internal predictor and corrector models, offers a plausible solution to optimise procedural motor learning.

3.2 Rotation Adaptation Task: Offline Learning and Consolidation

As aforementioned, another potential advantage of the coupling scheme is that it supports offline learning assuming that the sequence of actions executed during online training can be replayed offline [14]. In order to assess whether an offline consolidation process can further increase the system performances reached at the end of the online adaptation protocol, 2 groups of 10 simulated subjects are considered. Both groups consist of subjects adopting the coupling scheme (CM). However, one group (CM_{off}) is allowed to undergo offline learning, whereas the other (CM_{CTRL}) is not.

The Fig. 4 shows the protocol. Both groups (CM_{off} and CM_{CTRL}) undertake the 12 training trials. A first probe test (trial 13) is executed to evaluate the extent of the online rotation adaptation in both groups. Then, subjects from group CM_{off} undergo a simulated offline learning process consisting of a set of 48 trials (4320 trajectories randomly replayed) during which no sensory feedback is provided to the system. Therefore, the learning signal can only be computed based on the prediction provided by the forward

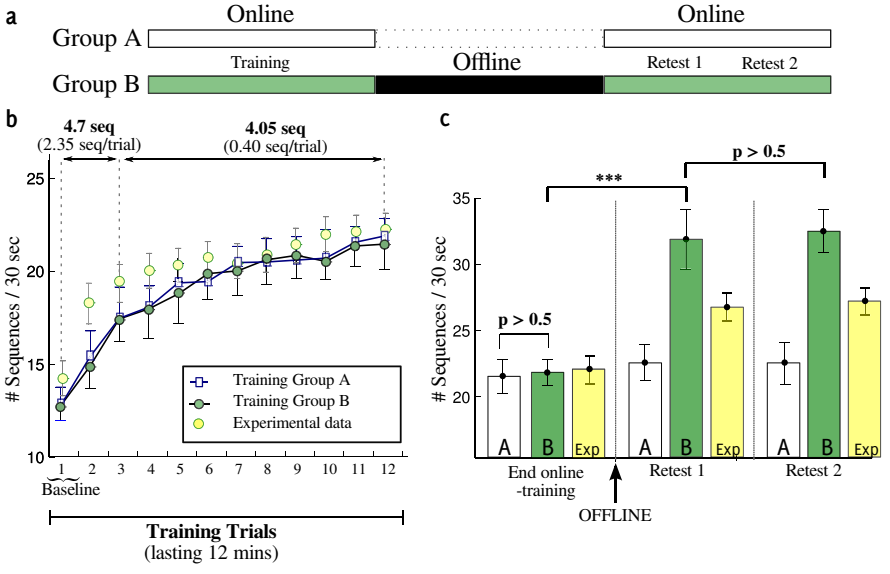


Fig. 6. *Sequential finger tapping task.* (a) Experimental protocol. (b,c) Simulation and experimental results (from Walker and and Stickfold 2004, [2]).

model, and the inverse model can adapt its dynamics only when this teaching information is available. Finally, both groups CM_{off} and CM_{CTRL} undertake a second probe test (trial 14) and their performances are compared.

Fig. 5c shows the results of this comparison both from our simulations and from experimental data obtained on human subjects [3]. A repeated measure analysis of variance and post-hoc tests show that the two groups have similar performances during the first probe test (i.e. when tested immediately after online training, trial 13). On the other hand, the second probe test (trial 14) shows that the mean directional error of CM_{off} subjects is significantly reduced compared to control subjects. Compared to the first probe test (trial 13), a performance enhancement of 12.7 ± 2.1 % is reached by CM_{off} subjects. By contrast, control subjects exhibit a lower performance improvement of 5.2 $\pm$ 1 %. The increase of performance of simulated CM_{off} subjects is consistent to that observed experimentally on human subjects after a night of sleep (yellow data, $+11 \pm 3$ % [3]). Since all parameters were controlled in our simulation, the improvement we report could only be explained by the offline consolidation process, and not by other factors such as circadian cycle. However, simulated control subjects appear to have better performances during the probe test (trial 14) compared to human subjects tested again after 8 hours of wakefulness, who do not show any significant improvement.

3.3 Cross-Task Validation: Results on a 'Sequential Finger Tapping Task'

In order to further validate the coupling scheme model, a second (totally different) procedural task was simulated: the sequential finger tapping task proposed by Walker and Stickfold (2004) [2].

This task requires subjects to press four numeric keys on a standard computer keyboard with the fingers of their non-dominant hand. A five elements sequence, 4-1-3-2-4, must be repetead as quickly and accurately as possible for a period of 30 s. Each 30 s trial is then scored according to the number of complete sequences achieved. The entire training consists of 12 trials (with 30 s rest periods between trials). The score from the first training trial is taken as a baseline, while the score from the final trial is taken as the post-training performance. 30 simulated subjects are allocated into 2 groups (A and B). Group A undergoes online adaptation only, whereas group B undergoes both online and offline learning.

The protocol and results are presented in Figs. 6a, and b,c, respectively. Subjects from groups A and B show similar performance improvement across the 12 training trials, with a non significant difference of 4.8% observed at the end of training (Figs. 6b). Overall performances improved by about 64% across the 12 training trials, with 40% occurring across the first three trials, and the remaining 24% occurring at a slower but relatively constant rate across the final ten trials. Figs. 6c illustrates the effect of offline learning and consolidation. It is shown that subjects from group B, after offline training, exhibit a significant improvement compared to control group A (probe test 1). They display no further significant improvement with additional online training (probe test 2).

Figs. 6b,c also show the correspondance between simulation results and experimental data obtained on human subjects [2].

4 Discussion

This work addresses the issue of coupling internal models (i.e. forward and inverse) in the cerebellum in order to enhance both online and offline learning capabilities. The proposed connectionist architecture takes inspiration from the cerebellar microcomplex circuit and it employs spiking neural populations to process information. Long-term synaptic plasticity (both LTP and LTD) is implemented to achieve adaptive motor control. It is shown that the system can acquire representations of closed-loop sensorimotor interactions, suitable to adapt the behavioural response to changing sensory contexts.

The coupling model reproduces the experimental findings on human procedural learning during the rotation adaptation task proposed by Huber *et al.* (2004) [3]. The sleep-dependent consolidation observed experimentally is mimicked here by an offline learning phase during which a replay of the contextual information elicited during online training occurs. This hypothesis is corroborated by several experimental studies: for example, it has been shown that patterns of activity recorded during online practice of a motor skill task reappear during episodes of REM sleep, while such activity is not seen in control subjects [14].

The same architecture is also validated on a second procedural task (i.e. the sequential finger tapping task proposed by Walker and and Stickfold 2004 [2]), which strengthens the idea that the proposed coupling scheme may offer a plausible model to *(i)* combine the advantages from fast online adaptation properties of forward models and accurate but slower convergence of inverse models, and *(ii)* achieve offline consolidation of procedural memories to enhance motor control capabilities.

In both cases, the model cerebellar microcomplex is used to adapt the dynamics of a fairly simple controller (e.g. two degrees of freedom arm). The model would probably need more neuronal resources to deal with more complex motor control tasks. One possible solution may be to use a modular approach as previously proposed by Wolpert and Kawato (1998) [10]. The coupling model would then be taken as a functional unit, and various behaviours could be generated by combining the output of several units. Because one unit could be used in different contexts, a large repertoire of behaviours could be generated, even with a limited number of modules.

Other questions related to the offline consolidation process can be further investigated using our model. As we observed, a sufficiently long offine consolidation leads to an improvement of overall performances. This observation raises a fundamental question concerning how the potential improvement varies as a function of the duration of the offline process. Finally, in our simulations to solve the rotation adaptation task, for instance, random sequences of entire trajectories were replayed when performing offline consolidation. It remains to be elucidated how the benefits of offline learning would vary if contextual information were only partially replayed. This question is currently under investigation using the presented coupling model.

To conclude, we voluntary omitted the role of the parietal lobe in this study, although this region is known to be implicated in target reaching tasks, and more generally in motor prediction processes (for a good review see [22]). The differential roles in prediction of the cerebellum and the parietal lobe are still under debate, and it is highly possible that these two structures work as a functional loop for predicting the sensory consequences of movement and making adequate corrections. It has been previously suggested that one of the distinctions may be that, contrary to the predictions made by the cerebellum, those of the parietal cortex would be made available to awareness [22]. However, to the best of our knowledge, this is still highly speculative, and has not been validated yet. A complementary view stresses the fact that the parietal cortex could be more involved in the comparison between sensory and motor information by maintaining the anticipated sensory consequence of the movement, this prediction being made by forward models located inside the cerebellum [23]. If this hypothesis were to prove true, then it would give insights on (1) how the teaching signal sent to drive learning of internal inverse models could be computed, and (2) why a strong correlation between the increase of slow wave activity in parietal cortex areas during sleep and the performance improvement after sleep are reported in procedural tasks [3]. An extended version of the coupling scheme could therefore integrate a simplified model of the parietal lobe, which could help to dissociate the role of both structures during online and offline motor adaptation.

References

[1] Ito, M.: The Cerebellum and Neural Control. Raven Press, New York (1984)
[2] Walker, M.P., Stickgold, R.: Sleep-dependent learning and memory consolidation. Neuron 44(1), 121–133
[3] Huber, R., Ghilardi, M.F., Massimini, M., Tononi, G.: Local sleep and learning. Nature 430(6995), 78–81

[4] Kawato, M., Furukawa, K., Suzuki, R.: A hierarchical neural-network model for control and learning of voluntary movement. Biol. Cybern. 57, 169–185 (1987)
[5] Kawato, M.: Internal models for motor control and trajectory planning. Curr. Opin. Neurobiol. 9, 718–727
[6] Lalazar, H., Vaadia, E.: Neural basis of sensorimotor learning: modifying internal models. Current Opinion in Neurobiology 18(6), 573–581
[7] Darlot, C., Zupan, L., Etard, O., Denise, P., Maruani, A.: Computation of inverse dynamics for the control of movements. Biological Cybernetics 75(2), 173–186
[8] Pasalar, S., Roitman, A.V., Durfee, W.K., Ebner, T.J.: Force field effects on cerebellar purkinje cell discharge with implications for internal models. Nature Neuroscience 9(11), 1404–1411
[9] Dean, P., Porrill, J., Ekerot, C., Jorntell, H.: The cerebellar microcircuit as an adaptive filter: experimental and computational evidence. Nat. Rev. Neurosci. 11(1), 30–43 (2010)
[10] Wolpert, D.M., Kawato, M.: Multiple paired forward and inverse models for motor control. Neural Networks 11(7-8), 1317–1329
[11] Kawato, M., Kuroda, T., Imamizu, H., Nakano, E., Miyauchi, S., Yoshioka, T.: Internal forward models in the cerebellum: fMRI study on grip force and load force coupling. Prog. Brain Res. 142, 171–188 (2003)
[12] Stickgold, R.: Sleep-dependent memory consolidation. Nature 437(7063), 1272–1278
[13] Ito, M.: Historical review of the significance of the cerebellum and the role of purkinje cells in motor learning. Ann. N. Y. Acad. Sci. 978, 273–288
[14] Maquet, P., Schwartz, S., Passingham, R., Frith, C.: Sleep-related consolidation of a visuomotor skill: brain mechanisms as assessed by functional magnetic resonance imaging. The Journal of Neuroscience 23(4), 1432–1440
[15] Ros, E., Carrillo, R., Ortigosa, E.M., Barbour, B., Agis, R.: Event-driven simulation scheme for spiking neural networks using lookup tables to characterize neuronal dynamics. Neural Computation 18(12), 2959–2993
[16] Ito, M.: Cerebellar circuitry as a neuronal machine. Prog. Neurobiol. 78, 272–303 (2006)
[17] Lev-Ram, V., Wong, S.T., Storm, D.R., Tsien, R.Y.: A new form of cerebellar long-term potentiation is postsynaptic and depends on nitric oxide but not cAMP. PNAS 99(12), 8389–8393
[18] Ito, M., Sakurai, M., Tongroach, P.: Climbing fibre induced depression of both mossy fibre responsiveness and glutamate sensitivity of cerebellar purkinje cells. The Journal of Physiology 324(1), 113–134 (1982)
[19] Carrillo, R.R., Ros, E., Boucheny, C., Coenen, O.J.D.: A real-time spiking cerebellum model for learning robot control. Bio. Systems 94(1-2), 18–27 (2008); PMID: 18616974
[20] Dayan, P., Abbott, L.F.: Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. The MIT Press, Cambridge
[21] Viviani, P., Flash, T.: Minimum-jerk, two-thirds power law, and isochrony: converging approaches to movement planning. J. Exp. Psychol. Hum. Percept. Perform. 21, 32–53
[22] Blakemore, S.J., Sirigu, A.: Action prediction in the cerebellum and in the parietal lobe. Exp. Brain Res. 153, 239–245
[23] MacDonald, P.A., Paus, T.: The role of parietal cortex in awareness of self-generated movements: a transcranial magnetic stimulation study. Cereb. Cortex 13, 962–967

eMOSAIC Model for Humanoid Robot Control

Norikazu Sugimoto[1,2], Jun Morimoto[2], Sang-Ho Hyon[2,4], and Mitsuo Kawato[3]

[1] National Institute of Communication Telecommunication
[2] Dept. of Brain Robot Interface, ATR Computational Neuroscience Labs
[3] ATR Computational Neuroscience Labs
[4] Department Robotics, Ritsumeikan University

Abstract. In this study, we propose a novel extension of the MOSAIC architecture to control real humanoid robots. The MOSAIC architecture was originally proposed by neuroscientists to clarify the human ability of adaptive control. The modular architecture of the MOSAIC model can be useful for solving nonlinear and nonstationary control problems. Both humans and humanoid robots have nonlinear body dynamics and many degrees of freedom. In addition, they can carry objects, and this makes the dynamics nonstationary. Therefore, the MOSAIC architecture can be considered a promising candidate as a motor-control model of humans and a control framework for humanoid robots. However, the application of the MOSAIC model has been limited to simple simulated dynamics. Since each module of the MOSAIC has a forward model, we can adopt this model to construct a state estimator. By using the state estimators, the extended MOSAIC model can deal with large observation noise and partially observable systems. Thanks to these advantages, the proposed control framework can be applied to real systems such as humanoid robots.

1 Introduction

Previous studies have suggested that the human central nervous system acquires and switches internal models of outside environments to adaptively perform motor control of the body in various environments [7,6]. The modular selection and identification for control (MOSAIC) architecture was originally proposed to explain the motor-control strategy of the human brain [12,4]. Humans and humanoid robots both have nonlinear body dynamics and many degrees of freedom. Moreover, they interact with objects in real environments, and this makes the dynamics nonstationary. Therefore, the MOSAIC architecture can be considered a promising candidate as a motor-control model of humans and a control framework for humanoid robots. However, the application of the MOSAIC model has been limited to simple simulated dynamics. This limitation partially comes from the following reasons: 1) the MOSAIC model does not explicitly consider the existence of noise input to sensory systems, and 2) the MOSAIC model assumes full observation and cannot deal with partially observable systems. In this study,

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 447–457, 2010.

we propose a novel extension of the MOSAIC architecture to cope with observation noise and partially observable systems. Since each module of the MOSAIC has a forward model, we can adopt this forward model to construct a state estimator. Moreover, using state estimators can provide a reasonable model of the sensorimotor function of the central nervous system as previously suggested [11]. In addition, a state estimation strategy using switching linear models is considered a useful approach to estimating hidden variables of complicated nonlinear dynamics [3]. The **e**xtended MOSAIC with state **e**stimators (eMOSAIC) can deal with large observation noise and partially observable systems. Thanks to these advantages, the proposed control framework can be applied to real systems such as humanoid robots.

In Section 2, we introduce our proposed eMOSAIC model. In this study, as proposed in an earlier work [2], we adopt an optimal control approach as the control method in the MOSAIC model.

We evaluate the control performance of the eMOSAIC model in environments with large observation noise and partially observable setups. We consider two tasks to show 1) the nonlinear control performance by a squatting task either using a two-link robot model or the humanoid robot, and 2) the nonstationary control performance by a carrying-object task using the humanoid robot with a weight. We show that the two-link robot model and the humanoid robot can maintain their balance using the eMOSAIC model when: 1) the two-link robot model or the humaniod robot squats periodically, and 2) the additional weight is suddenly placed on the humanoid robot model. Furthermore, we show that the eMOSAIC model significantly outperforms the original MOSAIC model on these tasks.

In Section 3, we consider the squatting task by using the two-link robot model in the simulated environment. In Section 4, we apply the eMOSAIC model to control our humanoid robot CB-i (Figure 1(a)) in order to show that the proposed model can be used in a real environment. We consider the squatting task and the carrying-object task. We show that a humanoid robot can maintain its balance while it is doing these tasks using the eMOSAIC model.

2 eMOSAIC Model

2.1 Target Problem

We consider an optimal control problem for nonlinear and nonstational dynamicas, where the dynamics is represented as:

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{n}(t), \tag{1}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), t) + \mathbf{v}(t), \tag{2}$$

where $\mathbf{x} \in \Re^N$, $\mathbf{u} \in \Re^D$ and $\mathbf{y} \in \Re^L$ are state, action and observation vector, respectively, and $\mathbf{n}(t) \sim \mathcal{N}(0, \Sigma_{\mathbf{x}})$ and $\mathbf{v}(t) \sim \mathcal{N}(0, \Sigma_{\mathbf{y}})$. $\mathcal{N}(0, \Sigma)$ denotes a Gaussian distribution with zero mean and covariance Σ. In the optimal control framework, the learning system is trying to find the optimal controller to

minimize the objective function: $J = E\left[\sum_{s=1}^{\infty} r(\mathbf{x}(s), \mathbf{u}(s))\right]$, where $r(\mathbf{x}(t), \mathbf{u}(t))$ is the cost function. To find the optimal controller to minimize the objective function, we estimate the value function:

$$V(\mathbf{x}(t)) = E\left[\sum_{s=t}^{\infty} r(\mathbf{x}(s), \mathbf{u}(s))\right]. \tag{3}$$

2.2 The eMOSAIC Model

The eMOSAIC model has the modular architecture. Each module is composed of a state estimator, a responsibility predictor, a value function estimator, and a controller. We approximate nonlinear and nonstational dynamics, and nonlinear cost function by switching linear models and quadratic models:

$$\mathbf{x}(t+1) = A_i\mathbf{x}(t) + B_i\mathbf{u}(t) + c_i + \mathbf{n}(t), \tag{4}$$

$$\mathbf{y}(t) = H_i\mathbf{x}(t) + \mathbf{v}(t), \tag{5}$$

$$r_i(\mathbf{x}(t), \mathbf{u}(t)) = -\frac{1}{2}\mathbf{x}(t)^{\mathrm{T}} Q_i \mathbf{x}(t) - \frac{1}{2}\mathbf{u}(t)^{\mathrm{T}} R_i \mathbf{u}(t), \tag{6}$$

where $A_i \in \Re^{N\times N}$, $B_i \in \Re^{N\times D}$, and $c_i \in \Re^N$ are parameters of i^{th} linear dynamics, $H_i \in \Re^{L\times N}$ is observation matrix. $Q_i \in \Re^{N\times N}$ and $R_i \in \Re^{D\times D}$ are parameters of i^{th} quadratic cost function. Therefore, each state estimator and each controller can be represented by a linear model, and the value function estimator can be represented by a quadratic model, respectively.

Figure 1(b) shows the schematic diagram of the eMOSAIC model. The responsibility predictor derives the responsibility of each module based on a state-prediction accuracy of the state estimator. Final output from the learning system is then derived as weighted sum of each module's output by the responsibility signal. Below, we explain the details of the state estimator, the responsibility predictor, the value function estimator, and the controller.

State estimator. The state estimator estimates states of the dynamics from the observation. We consider the state estimator that has the linear form:

$$\hat{\mathbf{x}}_i(t+1|t) = A_i\hat{\mathbf{x}}_i(t) + B_i\mathbf{u}(t) + c_i, \tag{7}$$

$$\hat{\mathbf{x}}_i(t+1) = \hat{\mathbf{x}}_i(t+1|t) + K_i(\mathbf{y}(t) - H_i\hat{\mathbf{x}}_i(t+1|t)) \tag{8}$$

where $\hat{\mathbf{x}}_i$ is the estimated state and K_i is the parameter for the state estimator. We derive the parameter K_i by solving the linear optimal estimation problem [8,9].

Responsibility predictor. The module selection is performed by probability distribution λ_i which we called "responsibility signal." The responsibility signal λ_i is given by following Bayes' rule:

$$\lambda_i(t) = \frac{P(i)p(\mathbf{x}(t) \mid \mathbf{y}(1:t), \mathbf{u}(1:t-1), i)}{\sum_{i'\in M} P(i')p(\mathbf{x}(t) \mid \mathbf{y}(1:t), \mathbf{u}(1:t-1), i')}. \tag{9}$$

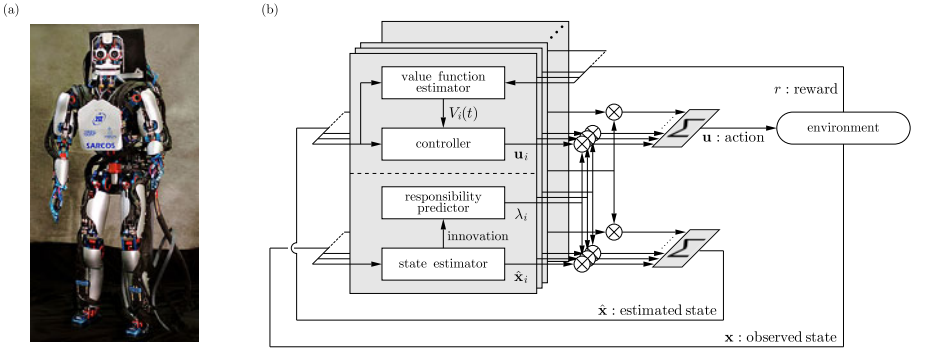


Fig. 1. (a) Humanoid robot CB-i, (b) **e**xtended MOSAIC with state **e**stimators (eMOSAIC)

Here, M is the set of module indices, $p(\mathbf{x}(t) \mid \mathbf{y}(1:t), \mathbf{u}(1:t-1), i)$ is the likelihood based on the accuracy of each state estimator, and $P(i)$ is the prior probability of module selection. The likelihood of state estimator $p(\mathbf{x}(t) \mid \mathbf{y}(1:t), \mathbf{u}(1:t-1), i)$ is given according to the estimation error. By assuming that the estimation error is Gaussian with variance σ^2, likelihood $p(\mathbf{x}(t) \mid \mathbf{y}(1:t), \mathbf{u}(1:t-1), i)$ is given by

$$p(\mathbf{x}(t)\,|\,\mathbf{y}(1:t), \mathbf{u}(1:t-1), i) \propto p(\mathbf{y}(t) \mid \mathbf{x}(t), \mathbf{y}(1:t-1), \mathbf{u}(1:t-1), i)\, p(\mathbf{x}(t) \mid \mathbf{y}(1:t-1), \mathbf{u}(1:t-1), i), \tag{10}$$

$$p(\mathbf{y}(t) \mid \mathbf{x}(t), \mathbf{y}(1:t-1), \mathbf{u}(1:t-1), i) = \frac{1}{\sqrt{(2\pi)^L |\Sigma_{\mathbf{y}}|}} \exp\left[-\frac{1}{2}\mathbf{e}_i(t)^{\mathrm{T}} \Sigma_{\mathbf{y}}^{-1} \mathbf{e}_i(t)\right], \tag{11}$$

$$\mathbf{e}_i(t) = \mathbf{y}(t) - H_i \hat{\mathbf{x}}_i(t), \tag{12}$$

$$p(\mathbf{x}(t) \mid \mathbf{y}(1:t-1), \mathbf{u}(1:t-1), i) = \frac{1}{\sqrt{(2\pi)^N |\Sigma_{\mathbf{x}}|}} \exp\left[-\frac{1}{2}\mathbf{d}_i(t)^{\mathrm{T}} \Sigma_{\mathbf{x}}^{-1} \mathbf{d}_i(t)\right], \tag{13}$$

$$\mathbf{d}_i(t) = \mathbf{x}_i(t) - \{A_i \hat{\mathbf{x}}_i(t-1) + B_i \mathbf{u}(t-1) + c_i\}, \tag{14}$$

$\hat{\mathbf{x}}_i(t) = E\left[P(\mathbf{x}(t) \mid \mathbf{y}(1:t-1), \mathbf{u}(1:t-1), i)\right]$ is the predicted state which is calculated by the state estimator of i^{th} module based on the observation vector $\mathbf{y}(1:t-1)$ and action output $\mathbf{u}(1:t-1)$, and $\mathbf{e}_i(t) = \mathbf{y}(t) - H_i \hat{\mathbf{x}}_i(t)$ is the so-called error of innovation.

If the hidden state exists, it is likely that the responsibility signal is underspecified based on instantaneous prediction error only. Therefore, we introduce the prior probability $P(i)$ based on "temporal continuity" can be given by

$$P(i) \propto \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2} E_i(t-1)\right]. \tag{15}$$

E_i represents a smoothed square error of innovation $\mathbf{e}_i(t)$ at time t,

$$E_i(t) = \sum_{s=0}^{t} \rho^{(t-s)\Delta t} \|\mathbf{e}_i(t)\|^2 \Delta t \tag{16}$$

where $0 < \rho < 1$ is a parameter that controls the strength of the temporal continuity and Δt is a time step of the observation. Eq. (16) can be expanded to follow a recursive form: $E_i(t) = \|\mathbf{e}_i(t)\|^2 \Delta t + \rho^{\Delta t} E_i(t-1)$.

Finally, the responsibility predictor weights the estimated state of each module,

$$\bar{\mathbf{x}}(t) = \sum_{i \in M} \lambda_i(t) \bar{\mathbf{x}}_i(t), \tag{17}$$

where $\bar{\mathbf{x}}_i(t)$ is the estimated state of i^{th} module at time t.

Value Function Estimator. We derive the controller by locally solving the linear-quadratic optimal control problem. Since we approximate the cost function by the quadratic functions as Eq. (6), we can locally estimate the value function by using a quadratic function:

$$V_i(\hat{\mathbf{x}}(t)) = v_i^0 - \frac{1}{2}(\hat{\mathbf{x}}(t) - \mathbf{x}_i^v)^{\mathrm{T}} P_i (\hat{\mathbf{x}}(t) - \mathbf{x}_i^v), \tag{18}$$

where the matrix P_i was given by solving the Riccati equation:

$$0 = P_i A_i + A_i^{\mathrm{T}} P_i - P_i B_i R_i^{-1} B_i^{\mathrm{T}} P_i + Q_i. \tag{19}$$

The center $\mathbf{x}_i^v$ and the bias v_i^0 of i^{th} value function were given by

$$\mathbf{x}_i^v = -(Q_i + P_i A_i)^{-1} P_i c_i, \tag{20}$$

$$\frac{1}{\tau} v_i^0 = r^0 - \frac{1}{2} \mathbf{x}_i^{v\,\mathrm{T}} Q \mathbf{x}_i^v. \tag{21}$$

Controller. Thus, from the linear optimal control theory [9], the controller can be derived as:

$$\mathbf{u}_i(t) = -R_i B_i^{\mathrm{T}} P_i (\hat{\mathbf{x}}(t) - \mathbf{x}_i^v). \tag{22}$$

The responsibility predictor weights the action of each module,

$$\mathbf{u}(t) = \sum_{i \in M} \lambda_i(t) \mathbf{u}_i(t). \tag{23}$$

Then, the controller outputs the weighted action $\mathbf{u}(t)$ to the environment.

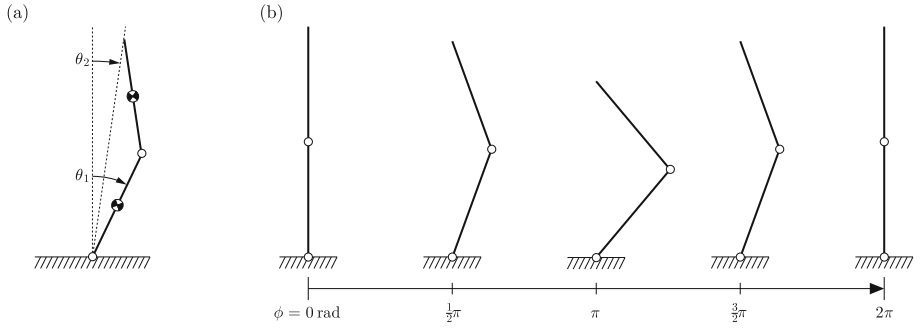


Fig. 2. (a) Two-link robot model. The mass and length of each link were 5kg and 0.5m, respectively, and the friction coefficient of each joint was 0.1. (b) Relationship between the phase of the periodic pattern generator ϕ and the posture of the two-link robot model.

3 Simulation

As an example of a nonlinear control problem, we consider a squatting task by using a two-link robot model (see Figure 2(a)). A basic squatting behavior is provided by a periodic pattern generator. Figure 2(b) shows the relationship between the phase of the periodic pattern generator ϕ and the posture of the two-link robot model. The frequency of the squatting movement was 0.5Hz.

Since a robot flexes and extends a leg periodically, a complex control law is required to prevent the falling of the robot. We apply eMOSAIC to control the two-link robot model to maintain its balance during the squatting movement. Note that the two-link robot model cannot maintain balance only using the output of the periodic pattern generator.

The angle of lower link and the tip of higher link are represented by θ_1 and θ_2 (see Figure 2(a)). The input state vector is $\mathbf{x} = [\theta_1\ \theta_2\ \dot{\theta}_1\ \dot{\theta}_2]^{\mathrm{T}}$. The output of eMOSAIC is a relative desired trajectory of the bottom joint ($\mathbf{u} = \theta_1^{\mathrm{des}}$). The sum of the relative desired trajectory (θ_1^{des}) and the output of the periodic pattern generator are used to derive torque at each joint based on a PD servo model. The cost is given in the quadratic form (see Eq.(6)) The parameters of quadratic cost were $Q_i = diag(0, 100, 0, 1)$ and $R_i = 0.5$. In this task, we use eMOSAIC with three modules ($i = 1, 2, 3$). One trial lasted 6 sec. Simulation and observation time-step Δt was 0.002 sec.

In this simulated environment, we focus on showing two advantages of eMOSAIC model: 1) eMOSAIC can be applied to a partially observable system, and 2) eMOSAIC can be applied to an environment with large observation noise.

First, we apply eMOSAIC both to the fully observable system and the partially observable system (only joint angles can be observed, i.e., $H_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ in Eq.(5)). The two-link robot model did not fall over even in the case of partial

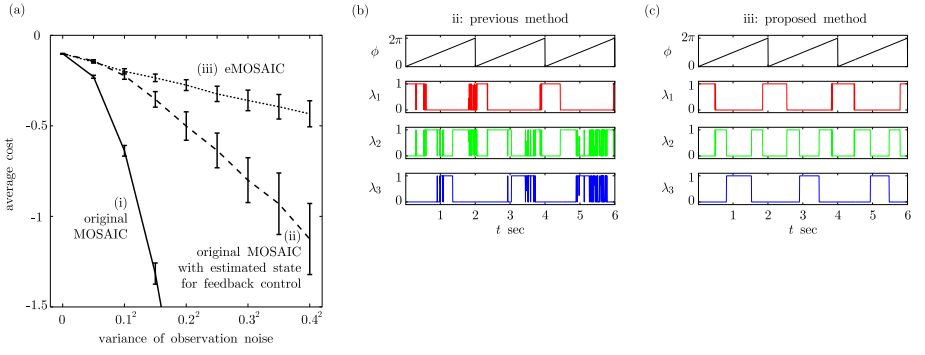


Fig. 3. Results of the squatting task. (a) Average cost with different observation noise. The means and standard deviations over 100 simulation runs are plotted. The responsibility signal of the (b) previous method (ii) and (c) proposed method (iii). The topmost panels indicate the phase of the periodic pattern generator, and the other panels indicate the responsibility signals of the three modules. (b) and (c) show the the results obtained when the responsibility signal is computed without and with the state estimators, respectively.

observation (see **Video 1** [1]). Furthermore, the mean costs acquired in the fully observable system and the partially observable system were -0.13 and -0.15, respectively. By using eMOSAIC, the performance of the controller in the partially observable system was close to the performance of the controller in the fully observable system.

Second, we evaluated the robustness of eMOSAIC for large observation noises. For this comparison, we considered three methods: (i) original MOSAIC, (ii) original MOSAIC with estimated state $\hat{\mathbf{x}}$ for feedback controller only (thus state estimation is not used for the responsibility signal's weighting), and (iii) proposed method. We tested the control performance of these three methods for observation noise ($\Sigma_{\mathbf{y}} = \sigma_{\text{obs}}^2 I$ with $\sigma_{\text{obs}}^2 = 0 \sim 0.4^2$). Figure 3(a) shows the relationship between the size of the observation noise and the average cost in the squatting task. The mean and standard deviation of the cost over 100 simulation runs are plotted.

Figures 3(b) and (c) show the responsibility signals estimated without and with the state estimator, respectively (ii, iii). The variances of the observation noises were $\Sigma_{\mathbf{y}} = 0.2^2 I$. The topmost panel shows the phase of the periodic pattern generator (ϕ), and the other panels show the responsibility signals of each module ($i = 1, 2, 3$). The responsibility signal estimated without the state estimator shows frequent chattering, which causes low control performance. In contrast, by using eMOSAIC, the responsibility signal shows smooth transition without chattering. Each module ($i = 1, 2, 3$) was activated at approximately

[1] The videos are uploaded on the web site:
http://www.cns.atr.jp/~xsugi/supplement/SAB2010/

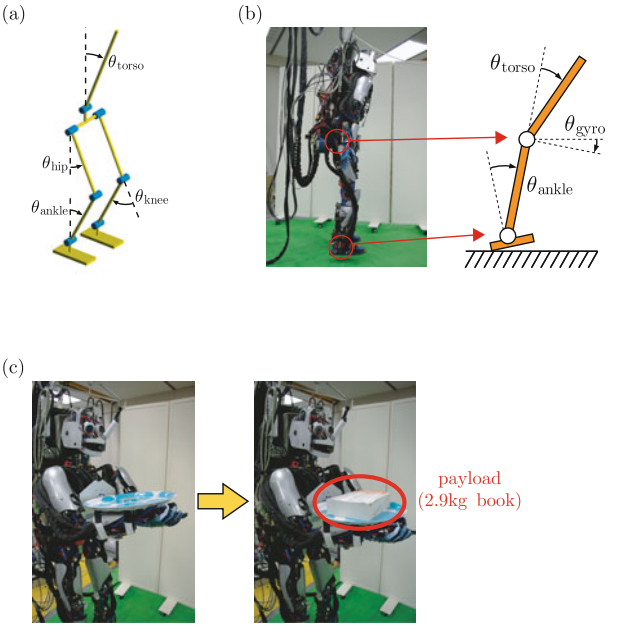


Fig. 4. (a) Pitch joint angles of the humanoid robot CB-i. (b) Two-link robot with a foot model of the CB-i. The two joints of the two-link model represent the torso and ankle joint of the CB-i. The gyro sensor is attached to hip of the CB-i. (c) A lifting task. The mass of payload is 2.9kg. Left: Without payload. Right: With payload.

$\phi = 0(2\pi)$, $\frac{1}{2}\pi(\frac{3}{2}\pi)$, and π. By using eMOSAIC, an appropriate module could be selected around appropriate phase despite the large observation noise.

4 Real Robot Experiment

In a real environment, we are often confronted with two difficulties, namely, nonlinearity and nonstationarity. In the real robot experiment using the humanoid robot CB-i (see Figure 1(a)), we consider the squatting task to evaluate the nonlinear control performance and the lifting task to evaluate the nonstationary control performance.

4.1 Squatting Task

We tried to maintain the balance of the CB-i in the squatting task. Figure 4(a) shows the pitch-joint coordination of CB-i. A periodic pattern generator outputs the desired trajectories to the hip, knee, and ankle joints. Each desired trajectory is given as follows: $\theta^{\text{d}}_{\text{hip}} = D\frac{1-\cos\phi(t)}{2} + \theta^{\text{res}}_{\text{hip}}$, $\theta^{\text{d}}_{\text{knee}} = 2D\frac{1-\cos\phi(t)}{2} + \theta^{\text{res}}_{\text{knee}}$ and $\theta^{\text{d}}_{\text{ankle}} = D\frac{1-\cos\phi(t)}{2} + \theta^{\text{res}}_{\text{ankle}}$. Here, $D = \frac{15}{180}\pi$ is the amplitude of a squatting

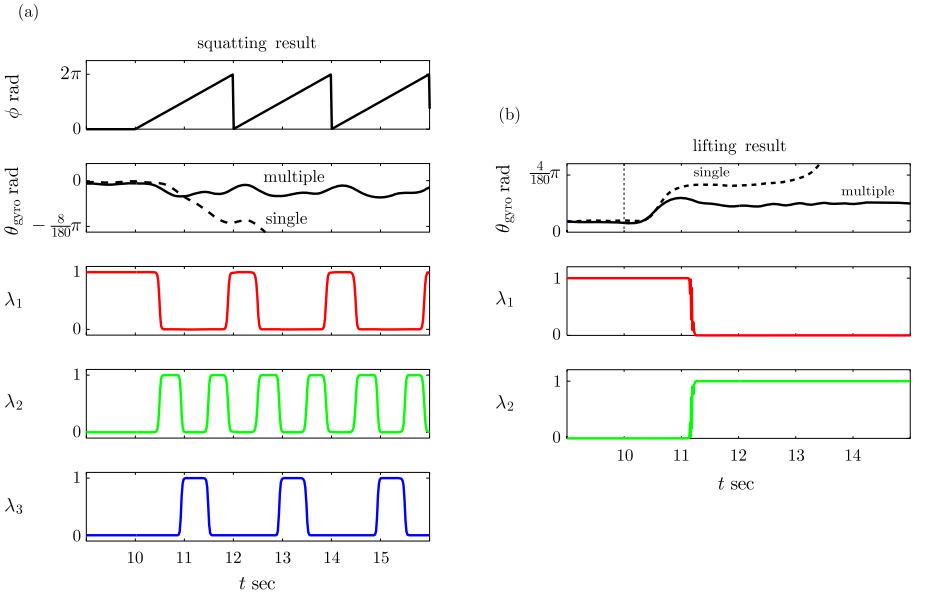


Fig. 5. Results of real-robot experiments. (a) Results of 0.5Hz squatting task. The top panel shows the phase of the periodic pattern generator (ϕ). The second panel shows the pitch angle of the gyro sensor. The solid line represents the result of the proposed method (**Video 2**), and the dashed line represents the result obtained by using only the 1st module (**Video 3**). The other panels show the responsibility signals of the proposed method. The results of 1.5Hz squatting task is uploaded as **Video 4**. (b) Results of lifting task. The top panel shows the pitch angle of the gyro sensor in the multiple-module and single module cases (**Video 5**, **Video 6**). The other panels show the responsibility signals of the proposed method.

movement. $\theta^{\text{res}}_{\text{hip}} = \frac{5}{180}\pi$, $\theta^{\text{res}}_{\text{knee}} = \frac{10}{180}\pi$, and $\theta^{\text{res}}_{\text{ankle}} = \frac{5}{180}\pi$ represent the rest posture of the robot. ϕ is the phase of the periodic pattern generator.

We approximated the dynamics of CB-i by a two-link robot with a foot model [5,1,10] (see Figure 4(b)). The input state was six-dimensional: $\mathbf{x} = \left[\theta_{\text{torso}}\ \theta_{\text{ankle}}\ \theta_{\text{gyro}}\ \dot{\theta}_{\text{torso}}\ \dot{\theta}_{\text{ankle}}\ \dot{\theta}_{\text{gyro}}\right]^{\text{T}}$. The output of eMOSAIC is a relative desired torso and hip joint angles: $\mathbf{u} = \left[\theta^{\text{add}}_{\text{torso}}\ \theta^{\text{add}}_{\text{hip}}\right]^{\text{T}}$. These relative desired joint angles are added to the desired joint angles generated by the periodic pattern generator. To follow the desired joint angles, the torque output at each joint is derived by a PD servo controller.

Here, the purpose of a linear optimal controller in each module is to keep the pitch angle of gyro sensor horizontal $\theta_{gyro} = 0$ (see Figure 4(b)). The cost function is defined as Eq.(6). The parameters of quadratic cost were $Q = diag(0, 0, 1, 0, 0, 1)$ and $R = diag(0.01, 0.01)$, respectively.

Figure 5(a) shows the results of 0.5Hz squatting task. The top panel shows the phase of the periodic pattern generator (ϕ). The second panel shows the trajectory of the pitch angle of the gyro sensor (θ_{gyro}). The solid line represents the result of eMOSAIC (**Video 2**). The pitch angle of the gyro sensor (θ_{gyro}) is maintained approximately in the range $\pm\frac{2}{180}\pi$rad for which the robot does not fall over. The dashed line represents the result obtained by using only the 1st module (**Video 3**). At approximately $t = 10.8$ sec, the robot in which only the single module was used fell backward; by using the multiple-module method, we successfully balanced the robot because the 2nd and 3rd modules were subsequently selected. The other panels show the responsibility signal of three modules. We have uploaded the video of the fastest squatting (1.5Hz) as **Video 4**.

4.2 Lifting Task

Here, we tried to maintain the balance of the CB-i in the lifting task (see Figure 4(c)). Again, we approximated the dynamics of our humanoid robot by using the above-mentioned two-link robot with a foot model (Figure 4(b)). We used two modules ($i = 1, 2$) in eMOSAIC. The state estimator of the 1st and 2nd estimate the state of the CB-i without the payload and with payload, respectively.

We used the same cost function as in the squatting task. Figure 5(b) shows the results of the lifting task. The top panel shows the pitch angle of the gyro sensor. We applied a payload to the tray at time $t = 10$ sec. The solid and dashed lines represent the proposed method (**Video 5**) and the single module method (**Video 6**), respectively. The robot with single module fell forward at approximately $t = 13$ sec; however, by the proposed method, the pitch angle of the gyro sensor θ_{gyro} was successfully maintained at less than $\frac{2}{180}\pi$ rad. The bottom panel shows the responsibility signals. First, the proposed method selected the 1st module, which controls the environmental dynamics without a payload. After applying the payload, the 2nd module, which controls the environmental dynamics when a payload is used, was automatically selected.

5 Discussion

We extended the MOSAIC architecture by using state estimators to cope with partially observable environment and large observation noise. We compared the proposed method with the previous method by performing simulations using a simple model. The results of the simulations indicated that our extension to the MOSAIC architecture improved its robustness and adaptiveness of control despite the partial observation and large observation noise. Then, we implemented our proposed method to the humanoid robot CB-i. We tried to stabilize CB-i in a squatting task and a lifting task. In these tasks, the CB-i could not be balanced by using the single module, but our proposed method was successfully used to stabilize the real humanoid robot.

In the future, we will consider a hierarchical module selection approach in which the higher layer has a longer time constant than the lower layer. Such a hierarchical architecture may be more adaptive and achieve better robust control than the present extended MOSAIC architecture.

Acknowledgments. The authors would like to thank the National Institute of Communication Telecommunication (NICT), Japan. The authors gratefully acknowledge N. Nakano for the experimental setups.

References

1. Atkeson, C., Stephens, B.: Multiple balance strategies from one optimization criterion. In: The IEEE-RAS 2007 International Conference on Humanoid Robots (2007)
2. Doya, K., Samejima, K., Katagiri, K., Kawato, M.: Multiple model-based reinforcement learning. Neural Computation 14, 1347–1369 (2002)
3. Ghahramani, Z., Hinton, G.E.: Variational learning for switching state-space models. Neural Computation 12(4), 831–864 (2000)
4. Haruno, M., Wolpert, D.M., Kawato, M.: Mosaic model for sensorimotor learning and control. Neural Computation 13 (2001)
5. Hyon, S., Osu, R., Otaka, Y.: Integration of multi-level postural balancing on humanoid robots. In: 2009 IEEE International Conference on Robotics and Automation (2009)
6. Imamizu, H., Kuroda, T., Yoshioka, T., Kawato, M.: Functional magnetic resonance imaging examination of two modular architectures for switching multiple internal models. Journal of Neuroscience 24(5), 1173–1181 (2004)
7. Imamizu, H., Sugimoto, N., Osu, R., Tsutsui, K., Sugiyama, K., Wada, Y., Kawato, M.: Explicit contextual information selectively contributes to predictive switching of internal models. Experimental Brain Research 181(3), 395–408 (2007)
8. Kalman, R.E., Bucy, R.S.: New Results in Linear Filtering and Prediction Theory. Trans., ASME, Series D, J. of Basic Engineering 83(1), 95–108 (1961)
9. Lewis, F.L.: Optimal Estimation. John Wiley & Sons, Chichester (1986)
10. Stephens, B.: Humanoid push recovery. In: The IEEE-RAS 2007 International Conference on Humanoid Robots (2009)
11. Wolpert, D.M., Ghahramani, Z., Jordan, M.I.: An internal model for sensorimotor integration. Science 269(5232), 1880–1882 (1995)
12. Wolpert, D.M., Kawato, M.: Multiple paired forward and inverse models for motor control. Neural Networks 11, 1317–1329 (1998)

Noisy-or Nodes for Conditioning Models

Jean Marc Salotti

Groupe Cognitique et Ingénierie Humaine, EA487
Ecole Nationale Supérieure de Cognitique,
146 Rue Léo Saignat, 33076 Bordeaux Cedex, France
jean-marc.salotti@ensc.fr

Abstract. In order to adapt the behavior of robots to varying environments, conditioning models provide interesting ideas. A prediction system is an important part of such models. The problem is to update it according to the sequence of stimuli perceived by the robot. Bayesian networks can be used to implement the prediction system. However, update rules are very complex and we need an incremental and fast learning process. We propose the use of noisy or nodes with appropriate learning rules. Numerous features of conditioning have been tested and promising results have been obtained.

Keywords: Pavlovian conditioning, operant conditioning, noisy or, Bayesian network.

1 Introduction

A predator learns through experience that a specific smell predicts the presence of a prey and a prey learns that the sudden flight of a group of birds predicts the presence of a predator. Such behaviors obey to the rules of classical conditioning, also called Pavlov conditioning [13]. Fundamentally, in classical conditioning, animals learn the correlation between specific events. These mechanisms are at the root of any complex interaction between animals and their environments. They are fully appropriate to the real world. For that reason, they are interesting sources of inspiration for the definition of the basic learning mechanisms of robots. The context of this type of learning is very specific. Learning is life long, every experience is taken into account, memory is never reset and environments are varying without warning. This is typically the domain of developmental robotics (see the survey from Lungarella [12]). In this paper, we focus on the prediction system of classical conditioning models with possible extensions to operant conditioning. It is based on the dynamic construction of Noisy Or structures. In part 2, we recall some important properties of classical conditioning and suggest important issues. In part 3, we present our prediction model based on the construction of Noisy Or nodes with appropriate update rules. Some results are then presented and discussed in part 4.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 458–467, 2010.

2 Classical Conditioning

2.1 Observed Behaviors

In the basic experience of classical conditioning, there is an unconditioned stimulus (US), a conditioned stimulus (CS), and a response R [13]. If the conditioning is successful, the specific response R should be observed whatever the presence or absence of the US. An important parameter is the time interval between the presentation of the CS and the presentation of the US (ISI=Inter Stimuli Interval). The conditioning is strong and fast for very small ISI and it becomes more and more difficult as the value of the ISI increases. Other interesting behaviors have been observed when subtle variations are introduced in conditioning experiments. Latent inhibition occurs when the CS is presented alone several times before the standard conditioning protocol. A "blocking" of a CS2-US association occurs when a CS1-US conditioning has already been performed and CS1 is always presented before CS2. Secondary conditioning occurs when a first CS (CS1) is used for classical conditioning and a second CS (CS2) is introduced before CS1. CS2 predicts CS1 and finally becomes a predictor for the US. The response is therefore observed when CS2 alone is presented.

2.2 Conditioning Models

Since the synthesis of experimental results presented by Pavlov, there has been substantial work and different models have been proposed. Most of them are based on the original model proposed by Rescorla and Wagner [4], [15]. Equation (1) gives the modification of the associative strength of a given stimulus X after a new trial. The increase is proportional to the salience of X (parameter α) and the efficiency of conditioning (parameter β). λ is the maximum strength and V_{Total} is the sum of all associative strength of the present stimuli.

$$V_X^{n+1} = V_X^n + \alpha_X \beta(\lambda - V_{Total}^n) \tag{1}$$

The associative strength of a given stimulus can be interpreted as the degree of prediction of the US. From a conceptual viewpoint, another important model has been proposed by Klopf with further considerations by Grossberg [4], [11]. For the first time, stimuli were associated to neurons and the associative strength was represented by synaptic weights. Conditioning was clearly seen as the ability to build a prediction system. Other authors followed the same principles [1], [2], [17]. Sutton and Barto established the basic principles of reinforcement learning [22]. They also proposed a temporal difference model of classical conditioning (TD model, 1987 and 1990 [20], [21]). Furthermore, Schmajuk proposed to take into account the novelty of the stimulus to explain latent inhibition [18]. Animals pay indeed more attention to new stimuli than to well known ones.

A comparison of several methods can be found in the paper from Balkenius [1]. All models fail in one or more simple experiments of blocking or second order conditioning with different timings and durations of CS and US (see figure 1). Moreover, both Schmajuk and TD models make the conditioning with a CS arriving after the US

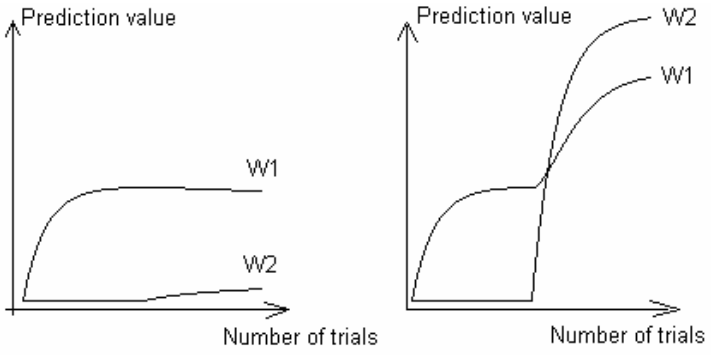


Fig. 1. Overriding and serial blocking experiment with the TD model. Left, we reproduced an experiment reported by the authors. In a first phase, during 70 trials, the first stimulus (CS1) is immediately followed by the US. The prediction value (W1) increases as expected. In a second phase, during 70 trials, the same CS1-US configuration is used but another stimulus (CS2) is added. When CS2 starts after CS1 but ends at the same time, it is called overriding blocking. W2 does not increase much. Right, the same experiment is performed, but in the second phase CS2 starts after the end of CS1 and finishes just before the start of the US. Theoretically, it is a serial blocking experiment (Kehoe 1981, [10]). However, this time, no blocking is observed. Surprisingly, prediction values W1 and W2 increase with a high slope. Variations of the TD model in which the representation and timing are differently taken into account may nevertheless provide the correct result [3] [8].

possible. There is still a controversial debate in the literature to explain some experimental results supporting backward conditioning [9], [19]. In this paper, we propose to focus on the prediction system of a conditioning model. If the CS occurs after the US, it cannot be a predictor of the US. In our paradigm, a backward conditioning is therefore not a desired property.

3 Bayesian Networks

3.1 Noisy or Nodes

The main feature of a conditioning model is a prediction system dedicated to the prediction of a reward or a punishment. Bayesian networks are well known tools to implement prediction systems [6], [7]. The problem is that the network is almost empty at the beginning of the agent's life. We therefore have to perform an automatic learning of the structure and the parameters of the network. In general, in terms of computational time, the determination of the structure is known to be a NP-complete problem [5]. However, the context of that learning differs in many aspects from usual learning contexts for Bayesian networks. First of all, learning is life long and there is no end for a specific learning phase. We therefore have to implement an incremental learning process. Second, the number of examples is quite short. Thus the convergence of the method should be fast. In comparable situations, Pearl proposed the "noisy or" [14]. We propose to implement his method, which can be summarized as follows. First of all, we define P(Y|X) as the conditional probability of observing

event Y during a limited period of time following the observation of event X. If $X_1..X_n$ are predictor events of Y, and if we assume that the causal links between every X_i and Y are independent and if there is no hidden variable, it is sufficient to have an estimate of all $P(Y|X_i)$ to compute $P(Y|X_1..X_n)$. Formally, if a list of predictor events $\{X_1..X_n\}$ are observed, the conditional probability is given by equation (2).

$$P(Y|X_1..X_n) = 1 - \prod_i (1 - P(Y|X_i)) \tag{2}$$

3.2 Reinforcement

An important problem is to take into account varying environments. For instance, in a conditioning trial, an agent has to learn the prediction of the reward after a given stimulus. Later on, in an extinction trial, the agent has to learn that its prediction is not valid any more. In the literature, it is often desired to obtain a network structure independent from the sequence of examples (see for instance the inference method from Cooper [6]). In our case, since we consider varying environments, the learning technique should be fully dependent on the exact sequence of examples. In fact, each new experience should be taken into account and should have an impact on the Bayesian network. In other words, we can not estimate a given conditional probability by a simple statistical computation based on the number of positive observations divided by the total number of observations. We should rather forget most of old observations and take into more consideration more recent ones. This is typically the case of all reinforcement equations proposed in previous models. The idea is to update the conditional probability by means of small increments or decrements of the previous value. If the probability that Y is caused by X has to be increased (X and Y have just been observed), we propose to update P(Y|X) with equation (3).

$$P_{t+1}(Y|X) = P_t(Y|X) + k_1 \alpha_{X,t} \alpha_{Y,t} \prod_{i\,/\,ti<ty \text{ and } (t-ti<\Delta t\max)} (1 - P(Y|X_i)) \tag{3}$$

where: t_i is the time of event X_i , t is the current time, Δt_{max} is a maximum delay to accept the causality between two events, k_1 is a constant and $\alpha_{x,t}$ and $\beta_{y,t}$ are attention parameters at time t.

The right term of the equation is justified. It is indeed proportional to the global error and it enables the blocking effect by reducing the increment if other predictor stimuli are simultaneously present. If the X-Y sequence is observed for the first time, (conditional probability equal to 0), a link is created between X and Y and P(Y|X) is stored. Let us consider the situation in which all conditions are fulfilled and a new link has been added in the network or simply reinforced if it already existed. Depending on time discretization, one step further the same configuration might be present. Do we have to update the prediction once again? In classical conditioning, it is well known that the efficiency is greater if the delay between the CS and the US is small. That interesting property can be taken into account. The configuration is the same as long as their trace in short term memory overlap ($t-t_x < \Delta t_{max}$ and $t-t_y<\Delta t_{max}$). In other words, the shorter the delay the longer the traces overlap and the greater number of times equation (3) applies.

3.3 Decrease of the Conditional Probability

What happens if the expected reward event never occurs? When t-tx>Δtmax the next event is not predicted any more. The conditional probability must be decreased. We propose a similar equation (see equation (5)).

$$P_{t+1}(Y|X) = P_t(Y|X) - k_2 \alpha_X \alpha_y \prod_{\substack{i \,/\, Xi=true \\ and\ t-ti<\Delta t\max}} P(Y|X_i)) \tag{4}$$

The proposed equation is justified as follows. The idea is to decrease the conditional probability such that high values are strongly penalized and if the experience is repeated it slowly converges to 0. However, if another stimulus X_2 has been detected and if $P(Y|X_2)$ is small, the decrement should be much smaller because X_2 is probably an inhibitor of event Y, so there is no reason to penalize P(Y|X). Though the proposed equation intuitively provides interesting results, it is not clearly linked to the global error. Other formulas have been tried, but they did not respect the convergence towards 0.

3.4 Attention and Novelty

In Schmajuk's model, the novelty of a stimulus plays an important role [18]. It is taken into account to explain latent inhibition and related phenomena. We propose a similar approach. In our model, we introduced a parameter called α. That parameter typically depends on the attention strength, which is high when the intensity of the stimulus is high, when it is new or when the animal or the agent is motivated. We propose a simple equation (5) to take into consideration those different aspects.

$$\alpha_{X,t} = M_t I_{X,t} N_{X,t} \tag{5}$$

M is the motivation, I is the intensity of the stimulus (consider for example the intensity of a noise) and N is the novelty score. While parameter I only depends on perceptual parameters at time t, the computation of N requires the memorization of past events. If a stimulus is new, its novelty is set to 1. Then equation (6) gives the evolution of the novelty associated to a stimulus X after a new observation.

$$N_{X,t+1} = N_{X,t} - k_3(N_{X,t} - N_{\min}) \tag{6}$$

with N_{min} minimum novelty value and k_3 constant.

4 Assessment

4.1 Acquisition and Extinction

Our model has been implemented in a simple simulator for an accurate assessment of multiple conditioning properties. In the following experiments, the same parameters have been used: Δt_{max} = 5 seconds; time discretization: 0.2 second; M_t=I_t=1.0; k_1 = 0.005; k_2=0.1; k_3=0.1; N_{min}=0.5. All values have been empirically chosen. However,

choosing other values is possible and lead to similar results. We tested k_1 in [0.001;0.01], k_2 in [0.01;0.5], k_3 in [0.01;0.9] and $N_{mi}n$ in [0.3;0.8]. Changes are rather minor and mainly impact the number of trials for acquisition or extinction of conditioning. Acquisition and extinction of classical conditioning have been tested. Results are presented in Figure 2. The acquisition and extinction curves are very similar to those obtained with other models. There is no surprise here since our equations are also inspired from Rescorla and Wagner's work. What is important is that the curves are exactly the same whatever the durations of the CS and the US. The timing nevertheless plays a role in the acquisition phase. In agreement with the literature, acquisition is faster when the interstimulus interval is short. Another important result is that the conditional probability P(US|CS) remains equal to zero if the US precedes the CS. Backward conditioning is not possible with our model.

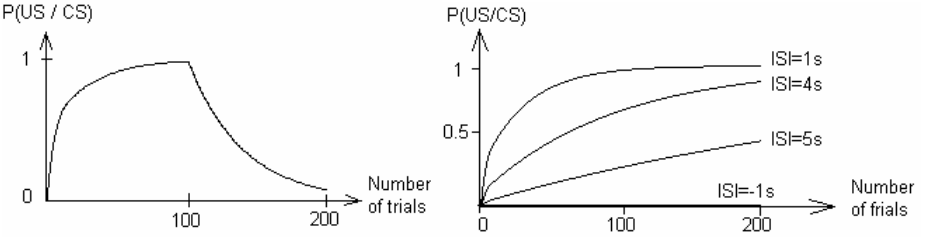


Fig. 2. Left: Acquisition followed by extinction. In a first phase, during 100 trials, a CS is presented at time t=1.0 and the US is presented at time t=2.0. Then, in a second phase, during 100 new trials, the CS is presented alone. Right: Different acquisition curves are obtained according to the interstimulus interval (ISI). ISI=-1s means that the US precedes the CS.

4.2 Blocking, Second Order Conditioning and Latent Inhibition

The blocking phenomenon has been tested. The results are presented in Figure 3. The curves are similar to those obtained with the TD model for overriding blocking (left part of Figure 1). However, in a serial blocking experiment, while the TD model would fail (right part of Figure 1), using our model the same results were obtained.

Concerning second order conditioning the results are presented in Figure 4. As it can be observed, while the US is never presented after CS1, it is nevertheless predicted thanks to the link between CS1 and CS2. It is important to note that since there is no direct link between CS1 and the US, P(US|CS1) is inferred using the conditional probabilities associated to the existing links (see equation (7)).

$$P(US|CS1) = P(US|CS2) * P(CS2|CS1) \tag{7}$$

In the first trials of the second phase, P(US|CS1) quickly increases. Its highest value is close to 0.5, which means that the US is expected but remains uncertain. Then it logically decreases towards 0. Such results are in agreement with the literature on second order conditioning.

A latent inhibition scenario has been implemented. The results are presented in Figure 5. The probability still converges towards 1 but the slope is weaker due to the decrease of the novelty (equations (5) and (6)).

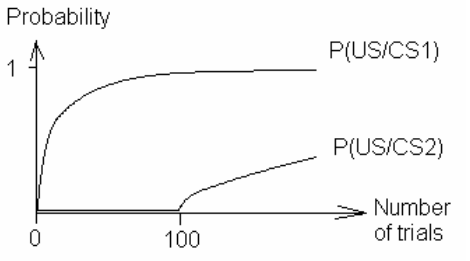


Fig. 3. Blocking experiment. In a first phase, during 100 trials, CS1 is presented at time t=1.0 and the US is presented at time t=3.0. Then, in a second phase, during 100 new trials, CS1 is presented at time t=1.0, CS2 at t=2.0 and the US at t=3.0. Since probability updates do not depend on the duration of the stimuli, the same results are obtained for overriding and serial blocking experiments.

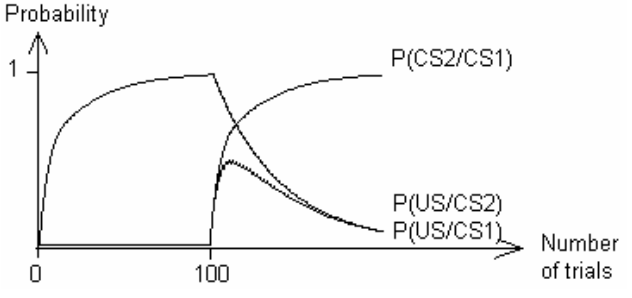


Fig. 4. Second order conditioning experiment. In a first phase, during 100 trials, CS2 is presented at time t=2.0 and the US is presented at time t=3.0. Then, in a second phase, during 100 new trials, CS1 is presented at time t=1.0 followed by CS2 at time t=2.0. The US is absent.

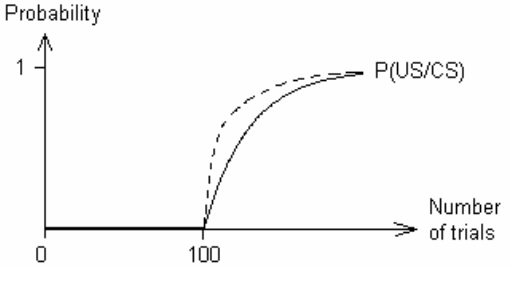


Fig. 5. Latent inhibition experiment. During 100 trials, a CS is presented at time t=1.0 and the US is absent. Then, during 100 new trials, the CS is presented at time t=1.0 followed by the US at time t=2.0. The theoretical curve that would be obtained in the case of a new CS is in dashed line.

4.3 Overexpectation

A specific compound conditioning experiment has been conducted to illustrate a major difference between our model and most others. The results are presented in Figure 6. If two CS are alternatively used for a strong acquisition of conditioning, all models predict a high associative strength. However, if later on the same stimuli are simultaneously used, all models based on Rescorla and Wagner's equation predict a decrease of the associative strength (see equation (1)). Such a decrease may be justified if the reward amount remains the same in all trials and if we consider that more rewards were expected during the compound trial. This experiment is called "overexpectation" and a decrease of the response is indeed observed with animals.

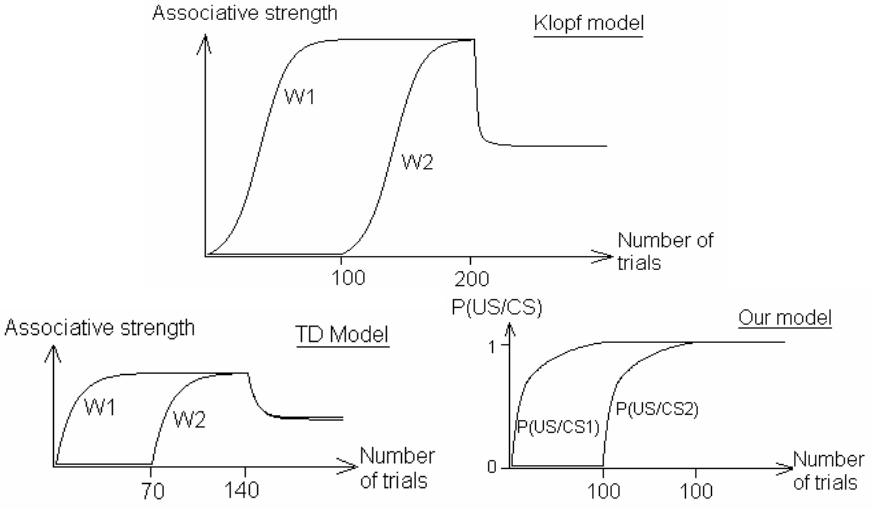


Fig. 6. Overexpectation experiment with models Klopf, TD and ours. In a first phase, CS1 is presented before the US. In a second phase, CS2 is presented before the US. And in the final phase, CS1 and CS2 are presented at the same time before the US.

However, we should make the difference between the prediction of the reward amount and the prediction of the reward event. In 2006, Rescorla recalled that the response decrease produced by overexpectation does not diminish the ability of stimuli to transfer control to instrumental responses with which they share a reinforcer [16]. This property can be explained by a decrease of the predicted reward amount while at the same time maintaining the conditional probability of the reward event very high. Since we focus in this paper on the prediction of events, the conditional probability associated to single stimuli should not be decreased. The result provided by our model is therefore correct and consistent with the results of the other models. Let us examine the difference between equations. In equation (1), the decrease is due to the last term, which is the negative sum of all associative strengths. If the sum exceeds λ, the associative strength decreases. In our equation, there is a product instead of a sum. If we develop the product in more simple terms, we obtain equation

(8). The same sum is therefore present but it is augmented by a sum of products. As long as that last sum is weak, the product is equal in first approximation to the sum of equation (1). But if it is not and this is the case in the current experiment since P(US|CS1) and P(US|CS2) are high, our model predicts no change in the prediction of the reward: Both probabilities remain close to 1.

$$\prod_i (1 - P_i) = 1 - \sum_i P_i + \sum_{i \neq j} P_i P_j - \varepsilon \tag{8}$$

5 Conclusion

A prediction system has been presented and can be used for classical conditioning. The key idea is to use Noisy Or nodes and appropriate rules to govern the creations and updates of the links. Our model differs from others in several aspects. First the conditioning is based on the onset (or offset) of the stimuli and not on their presence. Second, it is a prediction system that uses conditional probabilities with values between 0 and 1. It determines the probability of observing the reward event, while most other models provide a prediction of the reward amount. Interesting results have been obtained. We are currently still working on the model and the perspectives are numerous. There are issues that neither our model nor existing ones can currently solve, for example in the situation where stimuli are strongly dependent on each other. In the case of inhibitory conditioning the probability of observing the reward is weak if the inhibitor is present even if an excitatory stimulus is observed. A Noisy Or is clearly inappropriate in this case because it is based on the assumption that the presence of a single excitatory stimulus is sufficient to predict the reward. However, we are currently investigating a new structure with two concurrent Noisy Or nodes. The first one would be excitatory and the second inhibitory. We hope to be able to present interesting results in a near future. Another important perspective is to establish a clear link with operant conditioning. Since actions can also be decomposed into events, our model can easily be extended to the prediction of their consequences.

References

1. Balkenius, C., Morén, J.: Computational models of classical conditioning: a comparative study. In: Mayer, J.-A., Roitblat, H.L., Wilson, S.W., Blumberg, B. (eds.) From Animals to Animats, vol. 5. MIT Press, Cambridge (1998)
2. Balkenius, C.: Attention, habituation and conditioning: toward a computational model. Cognitive Science Quarterly 1(2), 171–214 (2000)
3. Bertin, M., Schweighofer, N., Doya, K.: Multiple model-based reinforcement learning explains dopamine neuronal activity. Neural Networks 20, 668–675 (2007)
4. Commons, M.L., Grossberg, S., Staddon, J.E.R. (eds.): Neural Network Models of Conditioning and Action. Lawrence Erlbaum Associates, Hillsdale (1991)
5. Cooper, G.: Computational complexity of probalistic inference using Bayesian belief networks. Artificial Intelligence 42(2), 393–405 (1990)
6. Cooper, G., Hersovits, E.: A bayesian method for the induction of probalistics networks from data. Machine Learning 9, 309–347 (1992)

7. Courville, A.C., Daw, N.D., Touretzky, D.S.: Bayesian theories of conditioning in a changing world. Trends in Cognitive Sciences 10(7), 294–300 (2006)
8. Daw, N.D., Courville, A.C., Touretzky, D.S.: Representation and Timing in Theories of the Dopamine System. Neural Computation 18, 1637–1677 (2006)
9. Hall, J.: Backward conditioning in Pavlovian type studies. Integrative Psychological and Behavioral Science (1984)
10. Kehoe, E.J., Schreurs, B.G., AModei, N.: Blocking Acquisition of the Rabitt's Nictating membrane Response to Serial Conditioned Stimuli. Learning and Motivation 12, 92–108 (1981)
11. Klopf, A.: A neuronal model of classical conditioning. Psychobiology 16(2), 85–125 (1988)
12. Lungarella, M., Metta, G., Pfeifer, R., Sandini, G.: Developmental robotics: a survey. Connection Science 15(4), 151–190 (2003)
13. Pavlov, I.P.: Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex (translated by G. V. Anrep). Oxford University Press, London (1927)
14. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc., San Mateo (1988)
15. Rescorla, R.A., Wagner, A.R.: A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In: Black, A.H., Prokasy, W.F. (eds.) Classical conditioning II: Current Research and Theory, pp. 64–99. Appleton-Century-Crofts, New York (1972)
16. Rescorla, R.A.: Spontaneous recovery from overexpectation. Learning and Behavior 34(1) (2006)
17. Salotti, J.M., Lepretre, F.: Classical and Operant Conditioning as Roots of Interaction for Robots. In: Proceedings of the Workshop From Motor to Interaction Learning in Robots, Conference on Intelligent Robotics Systems, Nice, September 22–26 (2008)
18. Schmajuk, N.A., Larrauri, J.A.: Experimental challenges to theories of classical conditioning: Application of an attentional model of storage and retrieval. Journal of Experimental Psychology: Animal Behavior Processes 32, 1–20 (2006)
19. Silva, F.J., Timberlake, W.: A Clarification of the Nature of Backward Excitatory Conditioning. Learning and Motivation 31, 67–80 (2000)
20. Sutton, R.S., Barto, A.G.: A temporal-difference model of classical conditioning. In: Proceedings of the 9th Annual Conference of the Cognitive Science Society, pp. 355–378 (1987)
21. Sutton, R.S., Barto, A.G.: Time-derivative models of Pavlovian reinforcement. In: Gabriel, M., Moore, J. (eds.) Learning and Computational Neuroscience: Foundations of Adaptive Networks, pp. 497–537. MIT Press, Cambridge (1990)
22. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)

Adaptation of Coupled Sensorimotor Mappings: An Investigation towards Developmental Learning of Humanoids

Martin Hülse and Mark Lee

Department of Computer Science, Aberystwyth University
Aberystwyth, SY23 3DB, UK
{msh,mhl}@aber.ac.uk

Abstract. The engineering of humanoid or similar robot systems requires frameworks and architectures that support the integration of a variety of sensorimotor modalities. Within our computational framework for visually guided reaching we ask how coupled sensorimotor mappings of different modalities can be learned autonomously from scratch. Based on a learning process that allows continuous adaptation of a single sensorimotor mapping, we introduce three strategies (parallel, sequential, and synchronous) for the learning of coupled mappings. These strategies are systematically tested in a simplified simulation. The experiments indicate that stages of development can emerge from synchronous adaptation of sensorimotor mappings of different characteristics. Thus, observed stages in development are not necessarily the result of explicitly defined and triggered learning tasks.

1 Introduction

The computational architecture which provides the context of this study enables an anthropomorphic robotic system or a humanoid to perform visually guided reaching. In particular it integrates active vision and reaching and is the result of a series of experiments on sensorimotor learning of eye-saccades [1], visual search [2] and hand-eye coordination [3]. Here we don't introduce this architecture in detail; the general concept is sufficient for the motivation of this work. The architecture operates in three different computational domains, which are the retinotopic reference frame, the gaze and the reach space (see Fig. 1 Left). The three different domains are linked together by two mappings: the sensorimotor mapping for eye-saccades and the mapping between gaze and reach space. The last core element of our architecture is the visual memory. It stores motor configurations of the active vision system which enables the system to perform a systematic visual scan of the environment. Furthermore, its content represents potential reach targets since the entries in the gaze space can directly be mapped into the reach space.

In summary, the two mappings provide the transformation of visual data into reach coordinates. Obviously, this is only possible via the gaze space and not

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 468–477, 2010.

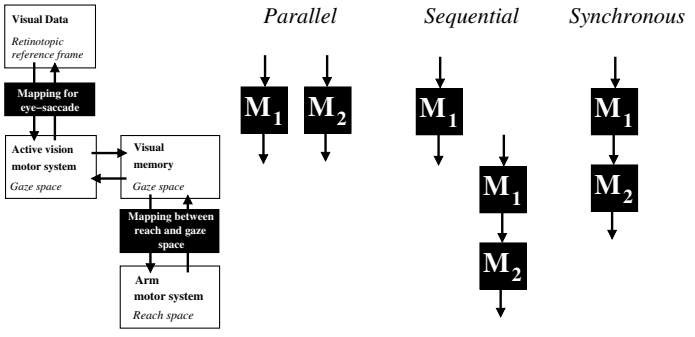


Fig. 1. Left: Schema of visually guided reaching for anthropomorphic robot systems. **Right:** Three strategies of sensorimotor learning, see text for details.

directly. Thus, the gaze space is the central element of this framework. More about the background and motivation of this framework can be found in [2,3].

The current implementation of this architecture validates our approach and provides an evaluation of the technical requirements and limitations. Our final goal, however, is an autonomously learning process of visually guided reaching without human intervention. Consequently, we need to ask: How shall these two mappings be learned from scratch? From the engineering point of view one can think of three principal strategies (Fig. 1 Right) which will be explained in the following.

1.1 Learning Strategies for Coupled Sensorimotor Mappings

Assume two mappings M_1 and M_2 which must be combined in order to solve the final task and both can only be learned through robot-environment interaction, as in our example of visually guided reaching. One way to learn them is separately or, as we call it, in *parallel*. After the learning of these mappings is completed, the system links them together providing new competences. Parallel learning assumes that both mappings serve independent sensorimotor systems. For a robotic system this is not always the case. In our example of visually guided reaching the learning of hand-eye coordination only makes sense if eye-saccades towards objects are performed robustly and in a reproducible manner. Hence, only after the mapping for eye-saccades is complete then the learning of the second mapping shall start. The second strategy, called *sequential*, represents this case. After M_1 is learned it will be chained or coupled with M_2 and the adaptation of M_2 starts. Thus the coupled mappings are learned in a specific sequence *and* most importantly, at any time only one mapping is subject of adaptation.

Parallel and *sequential* strategy need to implement additional mechanisms that detect when learning is completed in order to stop adaptation and link the mappings or to trigger the start of the next mapping learning. A way to avoid such explicit mechanisms is to chain the mappings right from the beginning and

run the adaptation processes synchronously or simultaneously. This strategy is called *synchronous sensorimotor learning*.

In the synchronous case, obviously, as long M_1 hasn't completed learning, M_2 adapts to the "faulty signals" of M_1. However, M_1 will show progress and this will gradually be reflected in M_2 too. Synchronous adaptation heavily relies on fast and continuously running adaptation processes which also have to provide re-learning. They have to be continuous because there is no start or ending of a mapping learning process anymore. As long as the robot is active all the mappings undergo an adaptation process. Furthermore, sensorimotor patterns in a learning robot system change significantly and other sensorimotor mappings of the system must be able to adjust accordingly. In other words, for synchronous sensorimotor learning the mappings must be able to forget.

1.2 Specific Aim of This Study

In this work we introduce a learning process that provides fast and continuous adaptation of sensorimotor mappings. The main objective of this paper is the analysis of essential properties of *synchronous* and *sequential* sensorimotor learning, as they are the most relevant strategies for robotics. Although this is done in a non-robot context we will see that the results will provide important insights for our framework towards autonomous developmental learning of visually guided reaching for humanoid robots.

2 Methods

2.1 Computational Substrate for Continuously Adapting Sensorimotor Mappings

The sensorimotor mappings in the following experiments are implemented as a case-based strategy. Assuming two spaces $X \subseteq R^n$ and $Y \subseteq R^m$ of arbitrary dimension, where

$$x = (x_1, x_2, \ldots, x_n) \in X, \qquad y = (y_1, y_2, \ldots, y_m) \in Y.$$

A mapping $\mathcal{M}$ stores the pairs (x, y) representing concrete examples that indicate how one point in one space X is related to space Y. Here, a pair is referred to as a *link*. The direct coding of these links allows bi-directionality, i.e. x refers to y and vice versa.

In addition, a metric is required for both spaces. This provides the generalisation of the mapping between X and Y because it is represented by concrete examples (links) which are unlikely to occur again during a robot-environment interaction in the real world. Thus, a definition of distance (i.e. a metric) between the points in space is required to allow a search for the 'closest neighbor' stored in the mapping (where 'closest neighbor' leads to the best estimation of the corresponding point the mapping can provide). For each space in a mapping a different metric can be applied. In this setup the Euclidean distance is used.

This and similar implementations have been applied to a variety of sensorimotor learning tasks for real robot systems, such as eye-saccades [1] and hand-eye coordination [3]. In a robot scenario learning sensorimotor mappings means the generation of links through robot-environment interaction. The quality of the mappings is obviously determined by the stored links, namely their number and distribution [3]. In addition to an insertion method of links, a mapping also has to provide mechanisms for the deletion of links in order to allow continuous adaptation to changing environmental conditions. In the following we explain the mechanisms which provide continuous adaptation in more detail.

Assume a mapping $\mathcal{M}$ learning the relation between the two spaces X and Y. We further need to introduce two parameters, $T > 0.0$ (tolerance) and $Q > 0$ (minimum age). Each link in a mapping has an "age" value. Whenever the mapping $\mathcal{M}$ is applied, or speaking precisely, whenever an estimation is made, which is formally written as:

$$y_e = \mathcal{M}(x),$$

then the age value of each link in the mapping is increased. A new link added has age value zero. Links having age values larger than Q can be deleted from the mapping.

Regarding the adaptation process, the robot-environment interaction generates sensor values x. This value is applied to the mapping producing an estimation for an appropriate actor signal y_e. Now, this signal or the corresponding action outcome is evaluated according to a given value system. For simplicity, assume a supervised system which has access to the correct motor/target signal y. Hence, an error value can be derived $d = |y - y_e|$, which we call the *estimation error*. If $d > T$, i.e. estimation error is larger than the given tolerance T of $\mathcal{M}$, then (x, y) is added to $\mathcal{M}$ as a new link. If the estimation was good enough, $z \leq T$, then no link is added, but the age value of the link which has provided the good estimation is set to zero. This ensures that for the time being this link remains in the mapping. The update of the age value for well performing links is important because the final step in this update process is the test for old links. If the age value of oldest link in $\mathcal{M}$ is larger than Q then this link will be removed from the mapping. Notice, only one link is deleted, even if there are more links of age values larger than Q.

Without going into detail, it shall be mentioned that T determines the generalisation capabilities of the mapping. The larger T the more the mapping generalises. The smaller the more it is likely to overfit. Q has an impact on the number of links in a mapping. The larger its value, the longer an individual link remains in the mapping. Thus, a higher number of links in a mapping can be expected if it has a larger Q-value. See [3] for more details.

2.2 A Simple Simulation for the Learning of Coupled Sensorimotor Mappings

In this study of sensorimotor learning we couple only two mappings, each links only 1-dimensional spaces, no robot system is involved. Our learning simulation

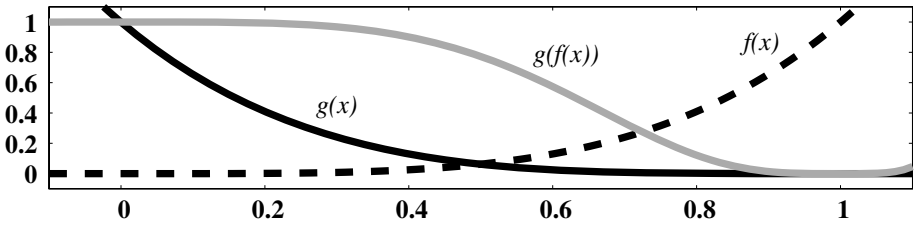


Fig. 2. Plot of the functions f(x),g(x) and g(f(x)) as defined and used in this work

is a simple function approximation task. The two functions $f(x)$ and $g(x)$ we use are defined as follows:

$$f(x) = x^4, \qquad g(x) = (x-1)^4, \qquad x \in [0,1].$$

Thus both operating in the closed interval $[0,1]$ only.

From the plots in Figure 2 one can see that $f(x)$ and $g(x)$ are symmetric with respect to the axis $x = 0.5$, and therefore inherently contain the same complexity. In other words, using the same approximation technique similar levels of accuracy can be expected.

In the learning simulation two mappings $\mathcal{M}_f$ and $\mathcal{M}_g$ are trained to approximate function $f(x)$ and $g(x)$, respectively. The overall target function that is to approximate is a combination of both, namely:

$$g(f(x)) = (x^4 - 1)^4, \qquad x \in [0,1].$$

This function is plotted in Fig. 2 too.

In the learning process we have two mappings $\mathcal{M}_f$ and $\mathcal{M}_g$ that individually adapt as described above. The only crucial issue is the way the mappings are coupled. This coupling determines the learning strategy, parallel, sequential or synchronous. In the following we describe the synchronous learning, see Box 1.

At the start of the simulation both mappings are initialised (for each $T = 0.01$, $Q = 200$) which also includes the insertion of 200, or Q-many, randomly selected links into each mapping. Finally the two functions $f(x)$ and $g(x)$ are defined analytically. In the next step a concrete example (x, y, z) of the combination $g(f(x))$ is derived by selecting a random x-value, uniform distributed in the interval $[0,1]$, and deriving y and z analytically. Noise is added to these signals in order to simulate the uncertainty of measurement present in every real sensorimotor system. The noise is Gaussian, where mean value is zero and standard deviation is T. It is worth mentioning that T is the tolerance value of both mappings. Choosing a tolerance value not smaller than the expected uncertainty makes an adaptation of the mapping to noise less likely. Step `3` begins by applying mapping $\mathcal{M}_f$ to the noisy signal x_o which leads to the estimation y_e. This result is tested versus the actual value y. Based on the resulting discrepancy or the error value d_f, mapping $\mathcal{M}_f$ will be updated accordingly (step `4`). In the case of a poor estimation (step `4.3`) a new y-value, y_o, is generated. This value y_o is assumed to be the

Box 1. Synchronous learning of $g(f(x))$, where $\mathcal{M}_f$ is applied first and approximates $f(x)$ while $\mathcal{M}_g$ approximates $g(x)$:

```
1    Initialise mapping M_f and M_g
1.1  set value of  T = 0.01 and Q = 200 for both mappings ;
1.2  add Q random links to both mappings,  M_f and M_g
1.3  define functions analytically as f(x) = x^4 and  g(x) = (x − 1)^4 ;
2    Simulate robot-environment interaction
2.1  create random example x = U(0, 1) ; -- uniform distributed in [0, 1]
2.2  derive y = f(x) and z = g(f(x)) ;
2.3  x_o = x + N(0, T) ; -- Gauss Dist.
3    Derive estimation error d_f of the first mapping M_f
3.1   y_e = M_f(x_o) ; y_o = y_e
3.2   d_f = |y − y_e| ;
4    Update first mapping M_f
4.1  if (d_f ≤ T)
4.2  then: set age to 0 for link which provided the good estimation;
4.3  else: create a new y_o where |y − y_o| ≤ d_f ;
           add new link  (x_o, y_o) to  M_f;
4.4  remove oldest link in M_f if age value is larger than Q;
5    Derive estimation error d_g of the combined mapping M_g(M_f(x))
5.1   z_e = M_g(y_o) ;
5.2   d_g = |z − z_e| ;
6    Update second mapping M_g
6.1  if (d_g ≤ T)
6.2  then: set age to 0 for link which provided the good estimation;
           z_o = z_e ;
6.3  else: create z_o where |z − z_o| ≤ d_g ;
           add new link  (y_o, z_o) to  M_g;
6.4  remove oldest link in M_g if age value is larger than Q;
7.0  go back to 2;
```

results of a targeted robot-environment interaction or exploration which leads to a better motor response. Notice, it is a better response but not necessarily the optimal or the best, which would be y. This condition can be formally written as: $0.0 \leq |y_o - y| \leq |y_e - y|$. Consequently, if x_o leads to a bad estimation then the new link (x_o, y_o) is added to $\mathcal{M}_f$, which is likely to improve the estimation performance of the mapping. In the case of a good estimation (step `4.2`) the age value of the link in the mapping which has delivered the good result is set to zero. Furthermore, y_o is set to y_e, the original estimation of $\mathcal{M}_f(x_o)$. The test of the second mapping $\mathcal{M}_g$ is based on the estimation resulting from the first mapping, namely y_o. The corresponding estimation error d_g is handled in the same way as described for the first mapping in this chain. However, after the update of the second mapping $\mathcal{M}_g$ is completed the whole process is repeated at step 2 starting with the generation of a new example. In summary, the process of continuous adaptation is established by the iteration of this loop, which we call an update or *learning cycle*.

This was the description of the synchronous leaning process for $g(f(x))$. However, this process can easily altered in oder to simulate sequential learning. Assuming that the first mapping was already learned, then the adaptation of the second mapping only needs to deactivate (or remove) step 4.

3 Experiments and Results

Simulations for all three strategies were conducted: parallel, sequential and synchronous. Thus, we have three different learning setups. For each setup we run 10 individual simulations. In each learning cycle and for each mapping involved as well as for the target function the average estimation error and its standard deviation was calculated. The validation set contained 500 randomly chosen x-values (uniform distributed). In addition the number of links was recorded too. The plots in Fig. 3 show the evolution of these values values over the learning cycles. For reasons of clarity only the plots of target function and the average number of links have error bars indicating the corresponding standard deviation.

The parallel learning (Fig. 3 A and B) provides the base lines in terms of accuracy and the needed number of learning cycles. One can see, on average the adaptation of both mappings is completed after $\approx$ 1200 cycles. At this point the average error values (individual function and target function) doesn't improve any more. Also the the number of links (Fig. 3 B) doesn't change distinctively anymore. This doesn't mean that no links are removed and added anymore. The adaptation process is still ongoing, since links get "older" and must be replaced with new links. But the total number reaches a kind of equilibrium.

The sequential learning (Fig. 3 C and D) was conducted over 4000 learning cycles. Since two mappings were learned in sequence and each mapping should have 2000 learning cycles as it was for the parallel learning. The switch of the learning process from the first mapping to the second is clearly indicated by the average number of links. After 2000 cycle the number of links of the first mapping remains constant, while it starts to increase for the second. After the switch to the second mapping, the average error value of the target function drops faster than it does for the second mapping. Hence, the progress of adaptation is much faster for the target function $g(f(x))$ than it is for the second mapping approximating $g(x)$.

On the other hand, as long as the switch to the second mapping isn't done (before cycle 2000), the improvements of the first mapping are not reflected in the target function at all. In other words, no matter how well the first mapping adapts the average error of target function remains at the high level of the beginning.

The adaptation of the second mapping is faster compared with the parallel case. At step 3000 (1000 steps after the switch) the learning is completed for the second mapping in both cases. Hence, assuming that the switch could already have been done at learning cycle 1200, when the first mapping is completed, then a total number of 2200 learning cycles can be estimated for sequential learning.

The data of synchronous adaptation (Fig. 3 E and F) show that the whole learning process is completed after $\approx$ 1300 learning cycles. As one would expect,

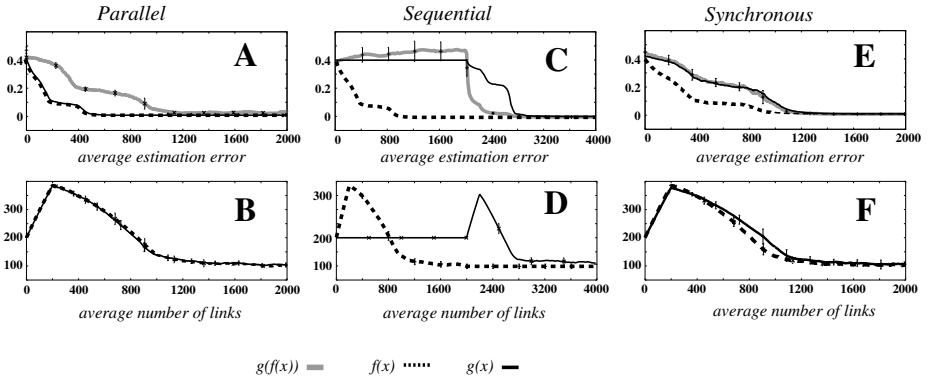


Fig. 3. Plots of the learning simulations, see text for details

with respect to the average estimation errors the progress of the second mapping is slower than for the first mapping. But surprisingly, the evolution of the estimation errors of the target function follows directly the progress of the second mapping.

Summarising all experiments by the final average estimations errors is done in Fig. 4. Sequential and synchronous learning always generate optimal error levels, since they are similar to the error levels achieved for the single function approximation in the parallel case. However, there is an noticeable higher error of the second mapping for sequential learning. Regarding the number of links we see in all runs the average number is quite similar.

4 Discussion

Sequential and synchronous learning outperform parallel learning. This can only be caused by the second mapping, since the adaptation process of the first mapping is identical to the parallel learning. But the sequential learning, on the other hand, shows that the second mapping generates much higher errors when tested against the individual function which it is supposed to approximate. This indicates that the second mapping is actually learning the errors of the first mapping in order to compensate them. Therefore, we have optimal error levels for the target function but lower estimation performance when testing the second mapping individually.

It is worth mentioning what make sequential adaptation different to parallel. In parallel adaptation both mappings are trained by random examples derived from a uniform distribution. In sequential learning the two mapping are coupled. Hence, the second mapping is trained by the signals generated by the first mapping. Since our mapping approximate non-linear functions the output values of the first mapping have a distribution very distinct from the uniform distribution. In other words, in sequential learning the second mapping is performed with a specific selection of training data. This specific set of data might be not best

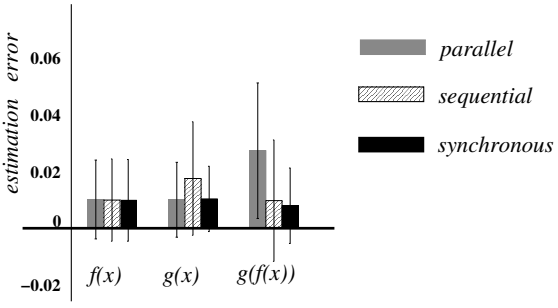


Fig. 4. Final average estimation errors for the complete set of experiments

suited for the general approximation of function $g(x)$. The individual test of the mappings was based on uniform distributed test values.

The effect of such compensatory adaptation is highly visible for sequential learning, but not in synchronous adaptation. The reason might be that at the beginning the faulty signals of the first mapping provide a "better" distribution of training examples for the second mapping. Consequently, the effect of compensatory adaptation might occur only latter since the adaptation process never stops as long the systems runs. Additional experiments need to provide more evidence for this hypothesis.

At this point we can summaries the experiments as follows. Synchronous learning is faster compares with sequential learning (1200 vs. 2200 learning cycles), it is even almost as fast as parallel learning (1200 learning cycles). With respect to the approximation of the target function, sequential and synchronous learning achieve distinctive lower error levels than parallel learning. Therefore, synchronous learning should be the preferred strategy for learning coupled mappings.

5 Conclusion

Synchronous learning outperforms parallel strategies with respect to average estimation errors. Compared to the amount of training data synchronous learning again has an advantage over sequential learning. For the implementation of autonomous development within our specific framework of visually guided reaching it follows that synchronous learning is efficient and should be the preferred strategy. Moreover, chained sensorimotor mappings can be learned without additional effort, namely additional mechanisms that explicitly measure the progress of learning in order to trigger the adaptation of the next sensorimotor mappings. All sensorimotor mappings involved in visually guided reaching can adapt at the same time and right from the beginning. Future research will demonstrate this on real robots.

A question might occur when considering research activities trying to model and to implement staged competence learning for humanoid robots which are inspired by child development [4]. In human infant development very distinct

stages of competences can be identified. Where are these stages for synchronous sensorimotor learning?

One might see staged development in infants as evidence that sequential learning strategies are in place and explicit trigger mechanisms therefore essential part of computational models for developmental learning. Nevertheless, we argue that stages of development could also emerge from synchronous adaptation processes if the sensorimotor mappings have different characteristics. In other words, observable stages of development could emerge because some sensorimotor mappings need longer to fully develop than others. In addition, factors, like physical condition and maturation, are obviously important constraints which need to be considered during the first months and years in child development [5]. In summary, we argue, that our experiment provide evidence that the application of synchronous learning can simplify the engineering of learning processes in advanced robot systems. However, it also seems worth taking synchronous learning into consideration for models of child development in general.

Acknowledgement

This work was supported by the EC-FP7 projects IM-CLeVeR and ROSSI, and through UK EPSRC grant EP/C516303/1.

References

1. Chao, F., Lee, M.H., Lee, J.J.: A developmental algorithm for ocular-motor coordination. Robotics and Autonomous Systems 58, 239–248 (2010)
2. Hülse, M., McBride, S., Lee, M.: Gaze modulated visual search for active vision. In: RSS 2010 (submitted 2010)
3. Hülse, M., McBride, S., Lee, M.: Fast learning mapping schemes for robotic hand-eye coordination. Cognitive Computation 2(1), 1–16 (2010)
4. Baldassare, G., et al.: The IM-CLeVeR project: Intrinsically motivated cumulative learning versatile robots. In: 9th International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, pp. 189–190 (2009)
5. Butterworth, G., Harris, M.: Principles of Developmental Psychology. Lawrence Erlbaum Associates, Hove (1994)

Learning Inverse Kinematics for Pose-Constraint Bi-manual Movements

Klaus Neumann, Matthias Rolf, Jochen J. Steil, and Michael Gienger

Research Institute for Cognition and Robotics - CoR-Lab
Bielefeld University

Abstract. We present a neural network approach to learn inverse kinematics of the humanoid robot ASIMO, where we focus on bi-manual tool use. The learning copes with both the highly redundant inverse kinematics of ASIMO and the additional arbitrary constraint imposed by the tool that couples both hands. We show that this complex kinematics can be learned from few ground-truth examples using an efficient recurrent reservoir framework, which has been introduced previously for kinematics learning and movement generation. We analyze and quantify the network's generalization for a given tool by means of reproducing the constraint in untrained target motions.

1 Introduction

The ability to use tools is one of the cornerstones of behavioral intelligence. Tool use is fundamental to human life: Humans use tools to extend their reach, to amplify their physical strength, and to perform many other tasks. However, to overcome limitations induced by the anatomy, tools are used by many organisms to increase their abilities. On the other hand, tools also play a very important role in classical industrial robotics. In this context, tools are used to tailor standard robot arms for specific tasks. The respective kinematics are typically hard coded by a human programmer or incorporated in the kinematic function by a simple offset. With the advent of autonomous and highly redundant humanoid robots such as ASIMO, machines begin to display an unprecedented dexterity and start to feature very flexible motor capabilities with a high precision. Because of their humanoid anatomy such robots are expected to handle tools in a way similar to humans in a large variety of tasks. A predefined parametrization of arbitrary constraints introduced by a tool is not feasible in this scenario. Learning of the skill will be more efficient than a situation dependent reprogramming. Many practical manipulation tasks, like those we consider in this paper, can be treated as imposing a certain pose constraint on the motion of the robot's hands. Examples are bi-manual use of a stick or moving a large box, where both hands become coupled with respect to both orientation and position.

In this paper, we focus on the example of bi-manual tools used by the humanoid robot ASIMO [1], although the methodology is by no means restricted to this particular robot. Tools are described by a function, which maps the position and orientation of a given tool to the end effector configuration of the

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 478–488, 2010.

robot. The geometry of a given tool, which defines the constraint, is therefore only implicitly available through the training examples and is never explicitly used. We will show that the learned solution will reproduce this constraint when generalizing to new targets. Note that the robot needs to coordinate its full body in order to use tools, because the arms are also coupled through the torso and its respective hip motion.

For learning, we employ a recurrent neural framework that is a variation of *reservoir computing* and has previously been used for learning inverse kinematics [2] and movement generation [3]. It uses efficient learning rules [4] that are biologically plausible and follow the general idea of reservoir computing: Inputs are fed into a dynamic reservoir of hidden neurons, by which they are transformed into a high dimensional space, the state of the reservoir network. This method is a very data efficient scheme, which can cope with the typical constraints in developmental learning. Data efficiency means to learn without excessive sampling of all possible tool configurations in space. The learner can generalize within convex hull of the demonstrated examples and can extrapolate to unseen samples [5].

Other machine learning techniques have been very successfully applied to specific inverse kinematics problems [6]. In order to increase flexibility in such systems, several approaches have been used. Under the notion of *extendable* or *adaptive body schemata*, several studies investigate how motor and control knowledge can be re-learned for the case of tool use [7,8,9]. The incorporation of arbitrary constraints has been investigated for the control of specific actions, for instance by Howard *et al.* [10]. However, learning the incorporation of arbitrary constraints into voluntary control is not well investigated. Therefore, our method expands the state of the art towards flexible tool use. In the remainder of the paper we describe the learning setting in Sect. 2, the neural network approach in Sect. 3, the evaluation and experiments in Sect. 4, and conclude in Sect. 5.

2 Tools as Kinematic Constraints for ASIMO

Given a robot, the forward kinematics function $F : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is uniquely defined. It converts a set of joint angles into the corresponding end effector configuration. On ASIMO, the end effector configuration contains two subsets: the left and the right hand. The hand center points $p^{L,R}$ are described in cartesian coordinates x, y and z with respect to the world coordinate system. The orientations of the hands are expressed as spatial orientations of the grasp axis $d^{L,R}$. The grasp axis are the z-axis in hand centered coordinates (see Figure 1). Thus the task vector is a twelve-dimensional input variable

$$e = e^{L,R} = (e^L, e^R) = ((p^L, d^L), (p^R, d^R)) \in \mathbb{R}^{12},\ L = \text{left},\ R = \text{right}\ . \quad (1)$$

In the following, task constraints given by tools will define and couple positions for both hands and the directions of both grasp axis.

The ASIMO full body motion controller [11], which will be used to generate the ground truth examples, operates on 15 degrees of freedom (m = 15). Each arm is moved by controlling three rotational degrees of freedom in the shoulder,

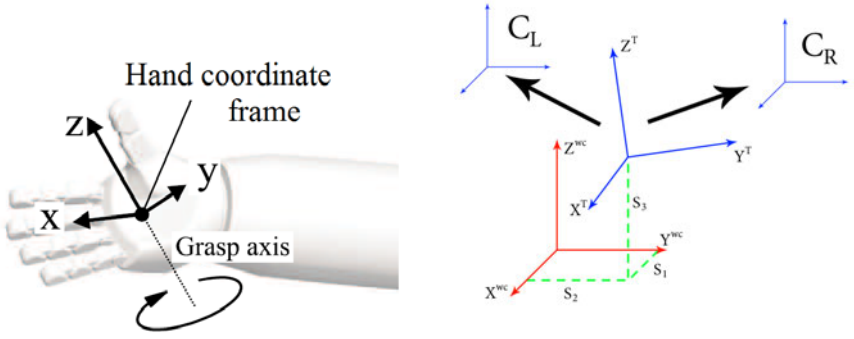


Fig. 1. Left: The grasp axis is identical with the z-axis in hand coordinates. Right: The translation is defined by $s_{1,2,3}$. C_L and C_R are the transformations to the end effectors. The Rotation from the world coordinates (WC) to the tool coordinates (T) is described by the angles $\theta_{1,2,3}$.

one in the elbow and one in the wrist. Additionally, four degrees of freedom are located in the hip: its height over ground and the rotation around all three spatial axes. The last degree of freedom is the heads pan orientation that is without effect on the task, but also controlled and learned.

An inverse kinematics function F^{-1} of a robot is defined by the forward kinematics in the following equation: $F(F^{-1}(e)) = e$. It maps a configuration of the end effector $e \in \mathbb{R}^n$ to the joint angels $q \in \mathbb{R}^m$ for the robot. There is no unique inverse kinematics function in the case of redundancy. ASIMO's kinematics is interesting for learning, because both arms are coupled by the upper body motion. The full body motion couples both arms by means of an augmented Jacobian [11] such that there is no separate kinematics for the arms. Targets that are out of reach for the hands can be approached, for instance, by leaning forward or backward.

2.1 Tool Kinematics as Constraints

We now describe the tool kinematics in order to be able to generate training examples. Once the training examples are known, the explicit tool geometry and kinematics are not further used directly, but all information is implicitly contained in the examples of tool positions and joint angles. We focus on bi-manual tool use. A tool is defined by a constraint, which couples both hands together. The constraint is described by a tool's position and orientation in Euler angles

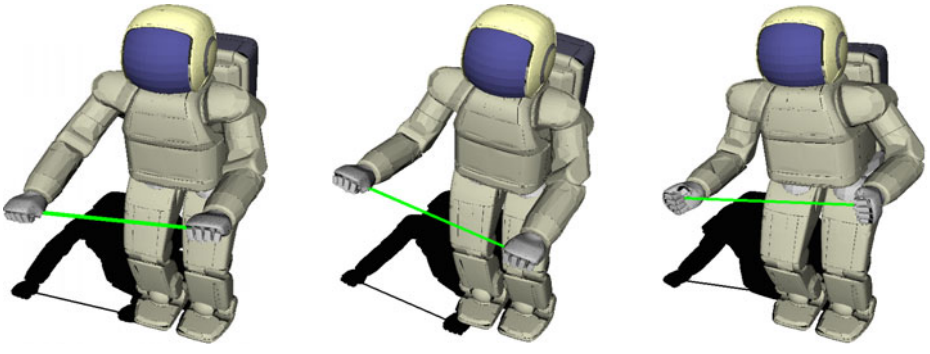
$$u(t) = (s_1, s_2, s_3, \theta_1, \theta_2, \theta_3) \in \mathbb{R}^6 \quad . \tag{2}$$

Given this input vector $u(t)$ the desired bi-manual end effector configuration $C(u(t)) = (C_L(u(t)), C_R(u(t))) = e(t) \in \mathbb{R}^{12}$ is uniquely defined. $C_L : \mathbb{R}^6 \rightarrow \mathbb{R}^6$ transforms the input vector to the world coordinates of the left hand. $C_R : \mathbb{R}^6 \rightarrow \mathbb{R}^6$ does the same for the right hand. For the control of the robot, a

function $T : \mathbb{R}^6 \rightarrow \mathbb{R}^{15}$, which couples the tool C and the inverse kinematics $F^{-1} : \mathbb{R}^{12} \rightarrow \mathbb{R}^{15}$ is required

$$T(u(t)) = F^{-1}(C(u(t))) = F^{-1}(e(t)) = q(t) \ . \tag{3}$$

Figure 2 shows some examples of ASIMO holding a 46 cm long stick. Given the position and orientation of the tool as input variable $u(t)$, the recurrent neural network (after learning as described below) computes joint angles $\hat{q}(t)$ to grasp the stick. In Fig. 2(a) and 2(b) both grasp axis are pointing towards each other, which is characteristic for a stick, while in 2(c) the grasp axis are vertical like for grasping a wheel with the stick as diameter.



(a) The stick axis is parallel with the y-axis in world coordinates.

(b) Stick rotated around the x-axis.

(c) ASIMO performs an end effector configuration where both grasp axis are in parallel.

Fig. 2. ASIMO controlled by a recurrent neural network

3 The Neural Network Learning Approach

The task is to learn the combined function consisting of the specified tool constraint and inverse kinematics with a recurrent neural network, such that T is approximated from a small number of examples. A new function $\hat{T} : \mathbb{R}^9 \rightarrow \mathbb{R}^{15}$ is defined by the neural network. The actual network outputs are denoted with

$$\hat{q}(t) = \hat{T}(u(t)) \ . \tag{4}$$

The goal is to minimize the error between (3) and (4) on the training set. For learning we use a recurrent neural network, which receives a time sequence of subsequent tool configurations $u(t)$ as input. The network is trained to compute joint angles $\hat{q}(t)$ such that the tool can be grasped $F(\hat{q}(t)) \approx C(u(t))$. Figure 3 shows the network setup. The respective network consists of 6 input-, 15 output- and 300 hidden reservoir- neurons. The output nodes receive input from both: input and reservoir neurons. The reservoir receives the input values and, in a

Table 1. Network parameters

Connection	Sparseness	Init. range
Input-Reservoir	0.2	0.1
Input-Output	1.0	0.1
Reservoir-Reservoir	0.02	0.02
Reservoir-Output	0.2	0.1
Output-Reservoir	0.2	0.1

Table 2. Learning parameters

BPDC-Learning		IP-Learning	
Rate-Start	0.15	Rate-Start	0.01
Rate-End	0.015	Rate-End	0.001
ϵ	0.002	μ	0.2

recurrent loop, the output values. The connectivity parameters are listed in Tab. 1. Formally, we consider the recurrent reservoir dynamics

$$x(k+1) = W_{net}y(k) + W_{in}u(k) \text{ where } y(k) = f_{a,b}(x(k)) \ .$$

k is the discrete time step and x_i , $i = 1...N$ are the neural activations. $y = f_{a,b}(x)$ is the vector of neuron activities obtained by applying parameterized Fermi functions $\frac{1}{1+exp(-a_i \cdot x - b_i)}$ component-wise to the vector x.

We assume that the neurons are enumerated such that the first O = 15 neuron activations $x_i, i = 1...O$ serve as output values. In our setting we can thus write

$$x(k) = (\hat{q}(k)^T, x_{O+1}(k), ..., x_N(k)) \ .$$

Our setup involves two learning rules that work in parallel. Connections to the output nodes are adapted with the supervised Backpropagation-Decorrelation rule (BPDC), which has been introduced in [4]. It can cope with feedback from output to the internal neurons [12] (Figure 3). Since only the output layer is adapted in a supervised manner, the approach is biologically rather plausible [13] compared to learning methods that require a deep backpropagation of errors. Such output layer adaption is also believed to occur in the cerebellum, which is heavily involved in human motor learning [14]. The initialization and

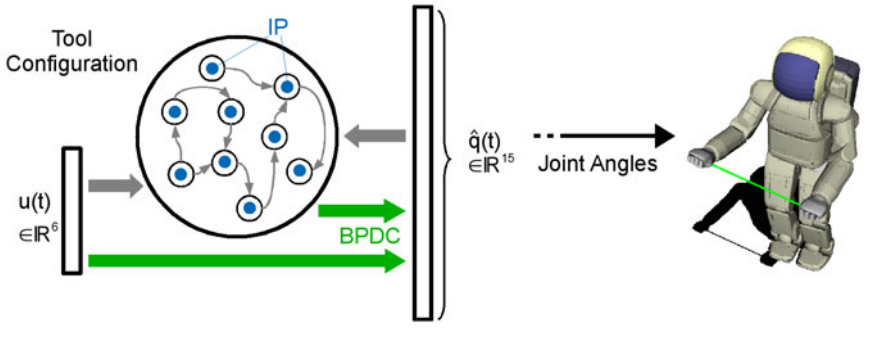


Fig. 3. The reservoir network. The network is trained towards a constraint inverse kinematics solution by BPDC adaption of output weights and an Intrinsic Plasticity (IP) rule within the reservoir. The estimated joint angles are applied on ASIMO.

handling of the other connections follow the reservoir computing paradigm and are therefore randomly chosen from a uniform distribution and stay fixed, see Tab. 1 and 2. An unsupervised Intrinsic Plasticity (IP) rule is applied inside the reservoir that accounts for an efficient neural coding as it can be found in different visual cortical areas [15]. The IP rule was first introduced by Triesch [16], inspired by soma-intrisic adaptions that are found in biological neurons [17], and was first used for reservoir optimization in [18]. Details can be found in these references.

4 Experiments and Results

In order to acquire ground truth training data, we use an analytic velocity-based feedback controller. This whole body motion (WBM) controller [11] uses all upper body degrees of freedom of ASIMO to perform a target motion of both hands. It selects one particular out of the infinite number of solutions based on additional criteria. It is important to note here that the goal of learning is not to replicate the velocity mapping. Rather, we learn a pure feedforward control, that solves the inverse kinematics directly. This is not the case for the velocity-based feedback controller. Since the demonstration and execution of targets to the controller is also temporal, in practice a target e is never exactly reached and this way to generate training examples actually introduces some noise.

Given a certain tool C the inverse kinematics function F^{-1} and a trajectory of the tool $u(t)$, trajectories of samples $q(t) = T(u(t))$ are created by executing the WBM controller and recording the respective joint angles. The analytic forward kinematics F is then used to additionally compute the corresponding end effector configurations $e(t) = F(q(t))$, which are later used to calculate different error measurements.

The training is organized in epochs and cycles. A cycle is one full temporal presentation of the training motion $u(t)$. In each epoch we first re-initialize the network-state randomly and present one cycle to the network without training to wash-out the randomly chosen initial state of the reservoir. Subsequently we show the complete pattern five times with enabled learning: after the presentation of each new target position $u(t)$, the output connections are adapted towards the target output $q(t)$ using the BPDC rule. An IP rule is used for reservoir optimization. A final cycle is used to estimate the error of the output joint angles $\hat{q}(t)$, while learning is disabled. We use three error measures:

– The mean relative euclidean distance between the desired and actual joint angles:

$$E_{jts} = \frac{1}{T} \sum_{t=1}^{T} \sqrt{\sum_{i=1}^{15} \left(\frac{\hat{q}_i(t) - q_i(t)}{ub_i - lb_i} \right)^2} \ .$$

ub_i is the upper bound and lb_i is the lower bound of the i-th joint.

– The mean distance between desired and actual hand positions (in meters) as interpretable and realistic error measure:

$$E_{pos} = \frac{1}{2T} \sum_{t=1}^{T} ||\hat{p}^L(t) - p^L(t)|| + ||\hat{p}^R(t) - p^R(t)||$$

– The mean distance between desired and actual hand orientations (in radiants):

$$E_{dir} = \frac{1}{2T} \sum_{t=1}^{T} |\measuredangle(\hat{d}^L(t), d^L(t))| + |\measuredangle(\hat{d}^R(t), d^R(t))| \ .$$

where $\measuredangle(a, b) = arccos\left(\frac{a \cdot b}{|a| \cdot |b|}\right)$ is the enclosed angle between a and b.

All series were learned over 1000 epochs with continuously decreasing learning rates of BPDC and IP. The rates follow an exponential function from a given start to a given end.

Evaluation and Generalization

Previous studies have shown that excellent generalization with the proposed network scheme [2,3,12] is possible, however for much simpler tasks not involving additional grasp constraints. The following Tabs. 3, 4 and 5 demonstrate that this holds true also for the more complex scenario considered in this paper. The errors are shown for nine different training sets. The training sets demonstrate learning to manipulate a 46 cm long stick, which is moved in a circle in front of ASIMO without changing the sticks orientation. For instance "XZ50" means that the stick center was moved in the X-Z-Plane in world coordinates in a circle with a radius of 50mm, divided equally in steps of one degree:

$$u(t) = (s_1 - 0.05m \cdot cos(t), s_2, s_3 + 0.05m \cdot sin(t), 0, 0, 0) \text{ with } s_1 = 0.45m,$$
$$s_2 = 0m,\ s_3 = 0.75m \text{ and } t = k \cdot \frac{2\pi}{360} \text{ and } k = 0..360 \ .$$

Figure 4(a) shows the visualization of the example time series.

The rows of the tables show the reservoir network errors on the nine training sets. The tables demonstrate that generalization to similar training sets is possible.

All series in Fig. 5 are produced by a movement of the stick in the x-z-plane, without changing the sticks orientation, such that the end points of the stick draw circles (Figure 4(a)). The lines in Figs. 5 and 6 are the outcome of the controller. In this case the end effector positions $\hat{p}^{L,R}(t)$ are plotted in Fig. 5(a) and the grasp axis coordinates $\hat{d}^{L,R}(t)$ in Fig. 5(b). The points represent the network output $\hat{q}$, trained with the inner time series, which was transformed by the analytic forward kinematics F into the end effector configuration $F(\hat{q})$ for visualization. The figure points out the networks' remarkable generalization ability.

Table 3. Relative joint errors E_{jts}

te./tr.	XY25	XY50	XY75
XY25	**0.027**	0.066	0.122
XY50	0.029	**0.037**	0.083
XY75	0.128	0.157	**0.218**

te./tr.	XZ25	XZ50	XZ75
XZ25	**0.048**	0.096	0.149
XZ50	0.040	**0.034**	0.088
XZ75	0.063	0.040	**0.037**

te./tr.	YZ25	YZ50	YZ75
YZ25	**0.156**	0.311	0.473
YZ50	0.029	**0.044**	0.089
YZ75	0.076	0.137	**0.207**

Table 4. Position errors E_{pos}

te./tr.	XY25	XY50	XY75
XY25	**0.005**	0.014	0.028
XY50	0.010	**0.011**	0.023
XY75	0.020	0.018	**0.025**

te./tr.	XZ25	XZ50	XZ75
XZ25	**0.008**	0.017	0.028
XZ50	0.005	**0.007**	0.016
XZ75	0.013	0.009	**0.008**

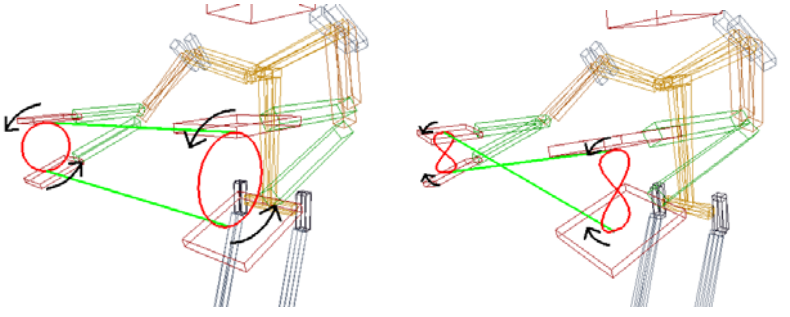
te./tr.	YZ25	YZ50	YZ75
YZ25	**0.017**	0.034	0.052
YZ50	0.006	**0.009**	0.015
YZ75	0.011	0.015	**0.021**

Table 5. Orientation errors E_{dir}

te./tr.	XY25	XY50	XY75
XY25	**0.030**	0.064	0.106
XY50	0.027	**0.016**	0.042
XY75	0.038	0.050	**0.070**

te./tr.	XZ25	XZ50	XZ75
XZ25	**0.012**	0.024	0.039
XZ50	0.008	**0.015**	0.029
XZ75	0.029	0.022	**0.023**

te./tr.	YZ25	YZ50	YZ75
YZ25	**0.016**	0.034	0.0560
YZ50	0.018	**0.019**	0.0357
YZ75	0.043	0.053	**0.0844**



(a) The time series XZ50 forms an circle with 5.0 cm radius at the endpoints of the stick.

(b) The time series RYZ100 forms an eight at the endpoints of the stick.

Fig. 4. A stick grasped by ASIMO in two different configurations

Figure 6 shows the movement of a stick by changing its orientation such that the end effector positions draw eights (Figure 4(b)). The grasp axis is shown in Fig. 6(b). The network was trained by the outer time series and interpolates the other time series with high accuracy.

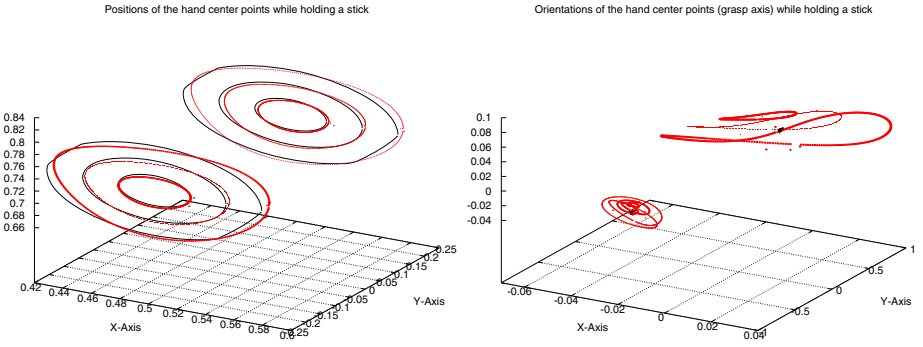


Fig. 5. Stick, which was moved in the x-z-plane without change of the orientation. Left: the position of the hands. Right: components of the grasp axis. The recurrent neural network was trained with the training set producing the inner circle.

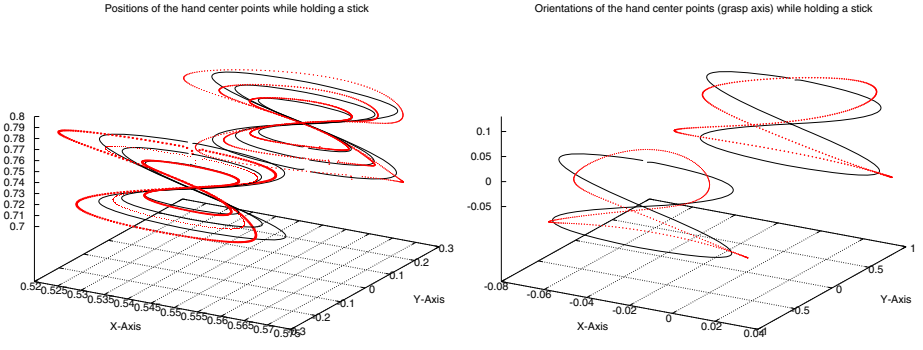


Fig. 6. Stick, which orientation was changed, such that eights at the ends of the stick were created. Left: the position of the hands. Right: components of the grasp axis. The recurrent neural network was trained with the training set producing the outer eight.

5 Discussion and Outlook

We present a neural learning task defined by a tool for the humanoid robot ASIMO. Necessary for learning control are data efficient learning mechanisms. The presented experiments show that our methodology allows such learning. The network can learn to coordinate the robot's upper body and the coupled kinematic chain defined by a tool. Our learning technique is able to deal with temporally correlated data and online learning, which are fundamental prerequisites to enable an incremental acquisition and also an ongoing refinement of motor skills. It is fast enough to be used in real time on the real robot system.

The networks allow remarkable generalization from very few, expert-generated examples, which makes the approach appealing for motor learning. One smooth sample motion, consisting out of 360 closely connected samples, is sufficient to

learn the whole body kinematics for a given tool. Future work will address a more systematic analysis of the generalization ability. The main focus will lie on the constraint, which couples the robots end effectors, defined by a tool. To analyze the generalization ability across constraints in more detail a measure which is able to quantify the network's capability to satisfy the constraint is needed.

Acknowledgement

Matthias Rolf acknowledges the financial support from Honda Research Institute Europe for the Project "Neural Learning of Flexible Full Body Motion".

References

1. Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., Fujimura, K.: The intelligent asimo: system overview and integration. In: Intelligent Robots and System, pp. 2478–2483 (2002)
2. Rolf, M., Steil, J.J., Gienger, M.: Efficient exploration and learning of full body kinematics. In: 8th IEEE ICDL (2009)
3. Reinhart, F.R., Steil, J.J.: Reaching movement generation with a recurrent neural network based on learning inverse kinematics. In: Int. Conf. on Humanoid Robots, Paris, pp. 323–330. IEEE, Los Alamitos (2009)
4. Steil, J.J.: Backpropagation-decorrelation: Recurrent learning with O(N) complexity. In: Proc. IJCNN, vol. 1, pp. 843–848 (2004)
5. Argalla, B.D., Chernovab, S., Velosob, M., Browninga, B.: A survey of robot learning from demonstration. Robotics and Autonomous Systems 57(5) (2009)
6. D'Souza, A., Vijayakumar, S., Schaal, S.: Learning inverse kinematics. In: Proc. IROS, pp. 298–303 (2001)
7. Stoytchev, A.: Computational model for an extendable robot body schema. Technical Report GIT-CC-03-44, Georgia Institute of Technology (2003)
8. Nabeshima, C., Kuniyoshi, Y., Lungarella, M.: Adaptive body schema for robotic tool-use. Advanced Robotics 20(10) (2006)
9. Hersch, M., Sauser, E., Billard, A.: Online learning of the body schema. International Journal of Humanoid Robotics 5(2) (2008)
10. Howard, M., Klanke, S., Gienger, M., Goerick, C., Vijayakumar, S.: A novel method for learning policies from variable constraint data. Autonomous Robots 27(2) (2009)
11. Gienger, M., Janssen, H., Goerick, C.: Task-oriented whole body motion for humanoid robots. In: Int. Conf. on Humanoid Robots. IEEE, Los Alamitos (2005)
12. Reinhart, F.R., Steil, J.J.: Attractor-based computation with reservoirs for online learning of inverse kinematics. In: Proc. ESANN, pp. 257–262 (2009)
13. Yamazaki, T., Tanaka, S.: The cerebellum as a liquid state machine. Neural Networks 20(3), 290–297 (2007)
14. Ramnani, N.: The primate cortico-cerebellar system: anatomy and function. Nature Reviews Neuroscience 7(7) (2006)

15. Baddeley, R., Abbott, L.F., Booth, M.C., Sengpiel, F., Freeman, T.: Responses of neurons in primary and inferior temporal visual cortices to natural scenes. Proc. R. Soc. London, Ser. B 264(1389), 1775–1783 (1998)
16. Triesch, J.: A gradient rule for the plasticity of a neuron's intrinsic excitability. In: Proc. ICANN, pp. 65–79 (2005)
17. Stemmler, M., Koch, C.: How voltage-dependent conductances can adapt to maximize the information encoded by neuronal ring rate. Nature Neuroscience 2(6), 521–527 (1999)
18. Steil, J.J.: Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning. Neural Networks, Special Issue on Echo State and Liquid State networks, 353–364 (2007)

TeXDYNA: Hierarchical Reinforcement Learning in Factored MDPs*

Olga Kozlova[1], Olivier Sigaud[1], and Christophe Meyer[2]

[1] Institut des Systèmes Intelligents et de Robotique
Université Pierre et Marie Curie - Paris 6, CNRS UMR 7222
4 place Jussieu, F75252 Paris Cedex 05
Olivier.Sigaud@upmc.fr, Olga.Kozlova@isir.upmc.fr
[2] Thales Security Solutions & Services, ThereSIS Research and Innovation Office
Route départementale 128, F91767 Palaiseau Cedex
Christophe.Meyer@thalesgroup.com

Abstract. Reinforcement learning is one of the main adaptive mechanisms that is both well documented in animal behaviour and giving rise to computational studies in animats and robots. In this paper, we present TeXDYNA, an algorithm designed to solve large reinforcement learning problems with unknown structure by integrating hierarchical abstraction techniques of Hierarchical Reinforcement Learning and factorization techniques of Factored Reinforcement Learning. We validate our approach on the LIGHT BOX problem.

1 Introduction

One central issue in animat research consists in facing the complexity of the real world with adaptive methods. Among such adaptive methods, reinforcement learning (RL) is one of the most studied. However, due to the "curse of dimensionality" inherent to the Markov Decision Process (MDP) framework, standard RL algorithms cannot address large scale real world problems mostly because they must enumerate all states [1]. In order to apply RL to robotics problems, two main lines of research can be followed. One consists in making profit of the structure of the sensory system of the robot using Factored MDPs (FMDPs), using a representation where a state is implicitly described by an assignment of values to some set of state variables. Then, one can exploit the dependencies between variables to get a compact representation [2]. The second line of research consists in making profit of the structure of the decision problem itself, using Semi-MDPs (SMDPs), an extension of MDPs where the number of time steps between one decision and the next is a random variable. The SMDP framework is the basis for Hierarchical Reinforcement Learning (HRL) algorithms as it decomposes the original task into subtasks that are easier to solve individually [3].

The methods to solve FMDPs and SMDPs usually assume that the structure of the problem is given, though that structure is rarely available in practice. As to hierarchical representations, HEXQ [4] and VISA [5] are two algorithms designed to solve this problem. Besides, for FMDPs, [6] have proposed SDYNA to solve Factored Reinforcement

* This work was founded by CIFRE convention - 1032/2006.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 489–500, 2010.

Learning (FRL) problems. Here, we propose TeXDYNA, an algorithm that combines the benefits of HRL and FRL, building an HRL-augmented version of SDYNA.

The paper is organized as follows. First, in Section 2, we give the minimal background necessary to understand what follows. In Section 3, we present TeXDYNA, introducing the method for discovering options from the internal structure of the problem and showing how to introduce options into FRL algorithms. In Section 4, we present an experimental study of our system based on the LIGHT BOX problem. In Section 5, we discuss the contributions and limitations of our work as well as related work.

2 A Quick Index to the Background

An FMDP is described by a set of state variables $S = \{X_1, \ldots, X_n\}$, where each X_i takes its value in a finite domain $Dom(X_i)$. Structured Dynamic Programming (SDP) algorithms such as SVI [7] make profit of this structure to compute a policy compactly. Structured-DYNA (SDYNA) [6] is a general framework that adapts indirect RL of the DYNA family [8] to the FMDP framework. SPITI is a particular instance of SDYNA based on a decision trees induction process to learn the structure of the problem and on SVI to obtain an efficient policy. While the FMDP representation technique reduces the size of the representation of state-action space of large problems by decomposing states into a set of random variables, the HMDP representation decomposes the overall state-action space into a set of smaller state-action spaces each of which can be factored. SMDPs are an extension to MDPs where the number of time steps between two decisions is a random variable. Among several other frameworks (see [3] for an overview), the options framework [9] is designed to solve SMDPs by building a hierarchy of options. An option is a tuple $\langle \mathcal{I}, \pi, \beta \rangle$, where $\mathcal{I} \subseteq S$ is an initiation set, that is a subset of states in which it is possible to execute o, $\pi : S \times O \rightarrow [0, 1]$ is a policy executed in o, and $\beta : S \rightarrow [0, 1]$ is a termination condition function, that is the probability of terminating the option in each state. Options are a generalization of primitive actions including temporally extended courses of actions. As a result, the algorithms deal only with options and do not have to distinguish options from primitive actions. Previous algorithms combining FMDP representation and HMDP structure learning are HEXQ [4] and VISA [10]. The work most closely related to ours, Incremental-VISA [11,12] is discussed in Section 5.

3 TeXDYNA

TeXDYNA hierarchically decomposes an FMDP by automatically splitting it into a set of options. Meanwhile, the local policy of each option is incrementally improved by a SDYNA-like approach. The central contribution of this work comes from the fact that the discovery of options and the construction of the model of the FMDP, as well as policy computation, are simultaneous. To achieve simultaneous SMDP structure learning, FMDP structure learning as well as local and global policy computation, TeXDYNA is built on top of SPITI. There are two main advantages to our approach. First, we make profit of the learning method used in SDYNA to learn only a local model for each option. As a result, the models are smaller and, therefore, easier to learn. Second, introducing options in the planning stage results in the possibility to plan over

smaller partitions of the state-action space. In order to decompose hierarchically the overall FMDP into sub-FMDPs represented by options, TeXDYNA builds a global transition function that represents the structure of the problem and uses this function to build a hierarchy of options. For each option, TeXDYNA computes a local transition function and a local policy. Therefore, TeXDYNA can be decomposed into two simultaneous processes: (1) Learning options: learning the transition function of the overall FMDP (updating the FMDP model with (s,a,$s\prime$,r)) and adding or updating options (Algorithm 1); (2) Planning with options: using a modified version of the SPITI algorithm for model learning, planning and acting with options, i.e. updating hierarchical policy $\pi = \langle \pi_{o_0}, \pi_{o_1}, ...\pi_{o_n}, \rangle$(Algorithm 2).

3.1 Defining Options

The purpose of our approach is to decompose the overall FMDP into smaller subtasks or mutually independent stand-alone policies. Our representation takes advantage of the FMDP structure using temporal abstraction techniques. We use a specific options representation inspired from the goal-oriented exit options of VISA [10] and HEXQ [4], where options are defined by their exit states that can be seen as subgoals of the task. Our options are noted o: $o = \langle \mathcal{I}, \pi, e \rangle$ where $\mathcal{I} \subseteq S$ is an initiation set, π is a policy executed in o and e is the related exit. We do not use the usual termination function β since it is defined by the exit and computed at each time step during the option execution.

An exit corresponds to changes of values of state variables linked to the reward function, as in HEXQ. However, unlike HEXQ, where exits are state-action pairs, we define exits as a tuple $\langle v, a, v_{ch}, c \rangle$, where v is the variable whose value is changed by this exit, a is the exit action that makes the value of v change at the next state, v_{ch} is a variable change, i.e. a pair of values $\langle x, x' \rangle$ where x is the value before a is executed and $x\prime$ the value after a is executed. In the stochastic case, the variable change is a probability distribution over v_{ch} and the highest probability is kept in the exit definition. Finally, $c = \{x_1, ..., x_n\}$ is the context, that is the set of constraints (i.e. assignment of values to a subset of state variables) that makes this exit available. In this representation, the primitive actions have an empty context. Thus, there is at most one exit per action and per variable change. Thus, each option corresponds to a unique exit.

The initiation set I_o of option o defines the state space where this option can be executed. On the one hand, it determines if the resources necessary for the successful execution of the option are available, and, on the other hand, it specifies the state space from where the exit of the option is reachable. In practice, it is the union of its own exit context and all the exit contexts of its sub-options. If a sub-option is a primitive action, its exit variable is added to I_o with all possible values. Otherwise, the exit variable of the sub-option is added with its value change. By convention, a sub-option with an empty initiation set is admissible everywhere. This is particularly true for primitive actions. Therefore, all the values of the corresponding exit variables of these options are accepted. Thus, I_o contains all the states from which the exit of the option is reachable. This property of direct reachability is ensured by the fact that the exit context copies the constraints of the corresponding branch in the transition tree. In other words, the nodes of the branch represent the variables of the context. When those variables can be

changed by a sub-option, the exit of the parent option becomes reachable from all the states where its sub-options are accessible.

The hierarchical structure of the options set is determined by the variables interdependencies expressed in the structure of the transition function. However, for the sake of sound planning, a rank is assigned to each option as the highest rank of its sub-options plus one. As a result, the planning algorithm chooses an option to execute at each abstraction level in the hierarchy by going down from the most abstract options to primitive actions. Transition model learning and options discovery are simultaneous. Thus the options discovered first represent the transitions learned first and consequently have less constraints as they represent the most accessible subgoals. If the problem has a hierarchical structure, this gives a bottom-up direction to the options discovery. First the options with the lowest abstraction level are discovered, then their execution gives access to more constrained options. Defined this way, each option represents a sub-FMDP $\mathcal{M}_o = \langle S_o, O_o, \Psi, T_o, R_o \rangle$ containing a reduced partition of the initial state-action space, where S_o is a set of context variables, O_o is a set of sub-options, Ψ is a set of admissible state-option pairs defined by the initiation set, T_o is the local transition function and R_o is the local reward function. In this respect, when an option is created, we initialize its local FMDP tree structure composed of the transition trees for its context variables, as well as a local planning algorithm. Thus, when learning the internal FMDP structure of the option, the states injected in the learning algorithm are reduced to contain only variables x_i such that $\forall i, x_i \in X_o$.

3.2 Learning: Adding and Updating Options

To discover options, the overall task transition function is learned in a decision tree form. The FMDP model provides the structure used for the options discovery process, as this structure represents the dependencies between variables and constraints under which those variables change their values.

Algorithm 1 describes the procedure for adding or updating options. An option is introduced each time there is an action that can change one variable value. The sub-options are the options available in the sub-FMDP represented by the corresponding parent option. Sub-options are added in the following way: for each variable in the context of the parent option, if the set of options $\mathcal{E}$ contains an option that modifies the value of this variable, this option is added to the set of sub-options. Note that, when computing the context of an exit, the exit variables are excluded from the context to avoid cross-dependencies between options. This procedure supposes that there is one option per variable change and that each option changes at most one variable value. For instance, if some options change more than one variable, then one variable can be changed by more than one option. As a consequence, these options would be sub-options of one another, creating loops in the hierarchy. Nevertheless, this constraint can be relaxed by reorganizing the hierarchy once the options have been added. The procedure is the following: if a cross-dependency is detected between two options, the hierarchical link between them is removed and both options are attributed the lowest rank of the two. This way, more than one option may have the same exit variable.

The options are defined over the model of the problem while this model is learned. Thus some are incomplete or erroneous, especially in the first stages of the learning

Algorithm 1. Add and update options

init : options set $\mathcal{E} = \emptyset$
input: FMDP $\mathcal{F} = [\forall x_i \in X : Tree(P(x'|x))]$

```
1  forall transition tree Tree(P(x'|x)) ∈ F do
2      forall leaf l of the Tree(P(x'|x)) do
3          if branch contains an action a & a modifies the value of the variable x in leaf l
           then
4              if E does not contain a definition of option o with the exit corresponding to
               variable x and action a then
5                  introduce new option o defined by exit e : ⟨v, a, v_ch, c⟩ with :
6                  • variable v ← x
7                  • action a ← current action a
8                  • variable change v_ch ← ⟨value in the branch, value in the leaf⟩
9                  • context c ← variables of the branch that leads to the leaf l
10             else if E contains a partial definition of o then
11                 update o with new information
```

process. To ensure the relevance of discovered options, they are updated every time the model of transitions changes. The algorithm checks if the set of options $\mathcal{E}$ already contains an option defined by the same action and variable, but with a different context definition. If so, it updates it (line 10 in Algorithm 1) using the following procedure:

1. Update the exit context c of the option;
2. Update the transitions trees (add missing ones and discard irrelevant ones);
3. Update the sub-options list (add missing ones and discard irrelevant ones);
4. Re-compute the rank k of the option;
5. Re-compute the Initiation set I_o.

As to incrementality, [6] proposes to re-initialize the value function each time the reward function changes, mainly because the structure of the value function results from the reward function. In the incremental options learning case, the value function tree is reset each time the reward function changes in order to take into account every modification that changes the structure of the policy. There are exactly as many options as possible variable value changes. Thus there is no need to remove incorrect options given that as soon as their context is correct, they become accurate. Meanwhile, however, inaccurate options can influence planning and exploration by building an erroneous policy (see the next section).

3.3 Planning: FRL over Options

The planning stage builds a hierarchical policy over options by incrementally improving and modifying it simultaneously with the learning process. The planning algorithm, built upon the ideas coming from HRL algorithms and FRL methods based on SPITI, is given in Algorithm 2.

Algorithm 2. SPITI with options

input : FMDP $\mathcal{F}$, hierarchy of options $\mathcal{E}$

for each *time step* t

1 **if** *no option is running* **then**
  option $o \leftarrow$ChooseOption$(s, Tree(\pi), \mathcal{E})$

2 **if** *terminal condition of* o *is satisfied* **then**

2.1 execute exit action a; observe next state $s\prime$ and immediate reward r

2.2.a update local FMDP $\mathcal{F}_o$ with $(s,a,s\prime,r)$

2.2.b update parent FMDP $\mathcal{F}_{parent(o)}$ with $(s,o,s\prime,r)$

2.3.a update local policy π_o with $\mathcal{F}_o$

2.3.b update parent policy $\pi_{parent(o)}$ with $\mathcal{F}_{parent(o)}$

3 **else**
  sub-option $i \leftarrow$ChooseOption$(s, Tree(\pi_o), \mathcal{E})$
  if *sub-option* i *is primitive action* **then**

3.1 execute i; observe $s\prime$ and r

3.2 update local FMDP $\mathcal{F}_o$ with $(s,i,s\prime,r)$

3.3 update local policy π_o with $\mathcal{F}_o$

3.4 **return**: i

4 **else**
  call SPITI over sub-option $i : i \rightarrow$SPITI $(\mathcal{F}_i, \mathcal{E})$

To operate with options and take into account the hierarchical structure of the policy, we use a modified instance of SPITI for model learning, planning and acting with options. The algorithm executes options recursively by going down the hierarchy of options up to primitive actions that can be executed by the agent in its environment, then performs the updates by going up in the hierarchy. This upward update guarantees that the model of transitions includes the changes that occur during the incremental learning process. Thus, local planning is performed for each option with respect to the current structure of its local model of the transition and reward functions. The inner loop of SPITI is decomposed into three stages:

- *Acting*: choose an action using ϵ-$greedy$ exploration;
- *Learning*: update the model of the transition and reward functions of the FMDP from $\langle X, a, X', R\rangle$ observations using ITI;
- *Planning*: update $Tree(V)$ and $Tree(\pi)$ using one sweep of SVI.

Then, in the learning stage, the ITI algorithm is modified to work with options by replacing primitive actions by options in trees and allowing to remove or add new transition trees while learning. Moreover, ITI uses reduced states representations containing only variables present in the context of the option, so that the local model is updated with observations $\langle S_o, a, {S_o}', R\rangle$ where S_o is a set of local context variables.

Finally, in the planning stage, the adaptation of SDP algorithms like SVI to the options framework is straightforward. Since the action space is restricted by the number of sub-options available on a given hierarchical level, the algorithm has an additional argument that specifies the list of options to iterate through.

The primitive actions available in the environment are all initialized as options with empty exit context and initiation set. TEXDYNA builds the hierarchical policy $\pi = \langle \pi_{o_0}, \pi_{o_1}, ...\pi_{o_n}, \rangle$ where a higher level policy is based on a set of lower level policies. Each option follows the policy of its FMDP, its sub-options follow their respective policies and so on. Moreover, to propagate the external rewards to the local policies of options, when a high level option is discovered, an additional reward, named "internal reward" r_o (by contrast with the external reward received from the environment) is assigned to its exit action. We set $r_o = \frac{R_o}{2}$, where R_o is the internal reward of the parent option. For the options on the top level of the hierarchy, R_o is the maximal immediate external reward that the agent can get. This heuristics is inspired by the "salient event" heuristics introduced in [13].

As mentioned above, in the first iterations, the hierarchy of options and often options themselves are inaccurate or irrelevant. This is why the procedure to choose options (Algorithm 3) must take inaccuracies into account and use a strategy that favors exploration in the first steps and chooses the options at the right level of the hierarchy.

Algorithm 3. Choose Option

input: current state s, policy $Tree(\pi)$, hierarchy of options $\mathcal{E}$, level k

Get option o from to the current policy, $o = max_o[leaf(Tree(\pi)|s)]$
if *o is null* **then**
  choose option o from $\mathcal{E}$ accessible in the current state s with current rank k
while *o is null* **do**
  choose option o from $\mathcal{E}$ accessible in the current state s with rank $k-1$
  (if $k = 0$ choose random primitive action)
return o

The root node of the option hierarchy represents the overall FMDP. The first option o is chosen according to the root policy while its initiation set contains the current state s. The sub-options are selected according to the internal policy π_o of the option o augmented with an ϵ-*greedy* exploration policy. If the policy is incomplete and does not contain any information about options available in a current state, then an option is chosen randomly from the options set $\mathcal{E}$ on the current rank level with respect to its initiation set. If no option is available with a given rank, the algorithm chooses from options with lower rank and so forth. Thus, in the first stages of learning, only the options with rank 0 are available, that is primitive actions chosen randomly and in the end the preference always goes to the options with the higher level of abstraction. This provides a high level of exploration at the beginning and then for unexplored parts of the state space, therefore this speeds up the overall learning process.

In order to avoid an option from being stuck within an erroneous policy that fails to achieve its exit, we introduce the notion of *selection penalty* that forbids one option for a given number of time steps. In practical terms, when an option reaches its exit state but fails to change the corresponding variable value, that means that its structure is inaccurate. In this case we forbid this option selection for $\frac{(max\ time\ steps\ per\ episode)}{10}$ time steps in order to let the model of transitions of its parent option be improved.

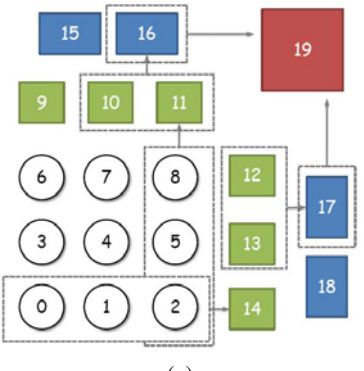
4 Experimental Study

Here, we have chosen to evaluate TEXDYNA on the LIGHT BOX domain [11], presented in Fig. 1. It consists of a set of twenty "lights", each of which is a binary variable corresponding to ON and OFF, named "0", "1", "2", etc. Each light has a corresponding action that toggles the light ON or OFF. Thus there are twenty actions, $2^{20} \approx 1$ million states, thus approximately 20 million state-action pairs. The 9 white lights are simple toggle lights that can be turned ON or OFF by executing the 9 corresponding actions. The green lights are toggled similarly, but only if certain configurations of white lights are ON, with each green light having a different set of dependencies. Similarly, the blue lights depend on certain configurations of green lights being ON, and the red light depends on configurations of the blue lights. The goal is to turn the red light on, in which case the agent receives a reward of 20.

The results presented below are averaged over 20 runs of 150 episodes where each episode is limited to 300 steps. The curves are smoothed by computing the moving average weighted over ten neighboring values. The algorithms use the following parameters: $N = 300$ in DYNA-Q and $\epsilon = 0.1$ in ϵ-$greedy$. The ITI algorithm uses $\chi^2 = 30$ in stochastic problems and $\chi^2 = 0$ in deterministic ones. The algorithms are coded in C# and run on Intel Core2Duo 1.80GHz processor with 2Go RAM.

An example of hierarchy of options obtained on the LIGHT BOX problem is given in Fig. 2. The complete state representation in the LIGHT BOX problem is given by 20 variables corresponding to 20 lights, but the states used to update the internal structure of the FMDP corresponding to the option $toggle16$ contains only 2 variables "10" and "11". As a consequence, the sub-FMDP has 2 variables $\times$ 2 sub-options instead of 20 variables $\times$ 20 actions.

Fig. 3 shows the performance in number of time steps required to complete one episode of the LIGHT BOX problem within three experimental contexts: random policy, TEXDYNA and DYNA-Q applied to both the deterministic version of the LIGHT BOX



(a)

Light	Depends on
9	0 3 6
10	1 4 7
11	2 5 8
12	6 7 8
13	3 4 5
14	0 1 2
15	9 10
16	10 11
17	12 13
18	13 14
19	16 17

(b)

Fig. 1. (a) The LIGHT BOX problem: number and color of "lights" with their dependencies. (b) Internal dependencies of the LIGHT BOX problem.

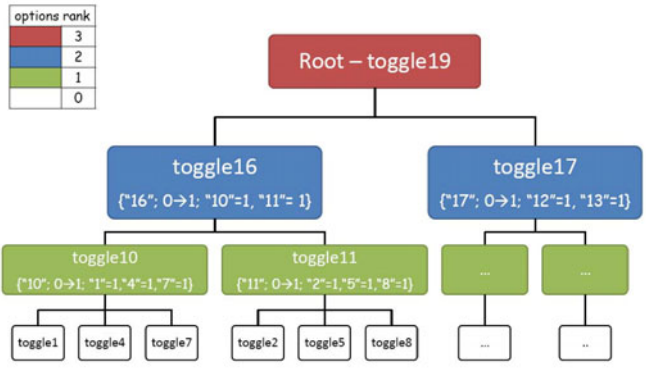


Fig. 2. Example of options discovered in the LIGHT BOX problem

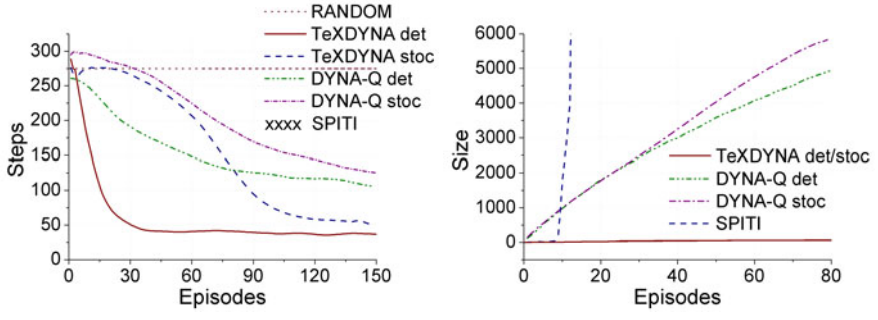


Fig. 3. (a) Convergence over episodes on the LIGHT BOX problem. (b) Policy size on the LIGHT BOX problem.

problem and a stochastic version where the agent performs random actions with a 10% probability. We could not obtain the convergence curve for SPITI because SPITI attempts to build the complete value tree representing all possible combinations that is 1 million leaves. This assertion is explained by the results presented in Table 1 that recaps the average time in seconds per step and the size of the functions in the stochastic LIGHT BOX problem. TEXDYNA requires less time to perform one step since it works on a smaller representation. While SPITI is struggling to check if the red light is dependent on the white light, the policy built by TEXDYNA goes straight to the goal state by achieving subgoals one by one at each level.

The algorithm finds the same structure in the stochastic case as in the deterministic one. One can notice that it takes more time to the algorithm to learn a stochastic transition function, but in the end it discovers the same hierarchy of options and the same policy and value functions. Similarly, stochastic DYNA-Q needs more episodes than its deterministic counterpart to converge. Furthermore, as shown in Table 1, TEXDYNA builds much smaller representations than its competitors in the stochastic case.

In order to explain the lesser performance of SPITI compared to TEXDYNA, we record the policy function size for SPITI and TEXDYNA, and the number of state-action pairs in DYNA, at the end of each episode. Fig. 3(b) shows the evolution of the corresponding

Table 1. Performance on the stochastic LIGHT BOX problem (Policy and value function size in total number of nodes in decision trees)

	TEXDYNA	SPITI	DYNA-Q
Transition function size	780 ± 14	790 ± 25	–
Value function size	240 ± 20	> 15000	> 10000
Policy function size	180 ± 8	> 15000	> 10000
Time/step(sec)	0.04	> 100	> 2

functions size over episodes. TEXDYNA quickly reaches a plateau on its optimal policy size, while the policy discovered by DYNA continues to grow up until representing all possible states. Therefore, DYNA results given in Fig. 3(a) are biased by the fact that even if it discovered a kind of sub-optimal policy in a reasonable time, the algorithm is unable to perform the complete policy computation. As to SPITI, the system fails to achieve convergence because of a too strong memory requirement. Indeed, as soon as one tree size exceeds 15000 nodes, the system runs out of memory. Further code optimization might reduce this limitation, but the main point is that exponential task structure simplification allows TEXDYNA to scales much better. Indeed, local models combined with state reduction to the set of context variables ignoring anything else provides a significant state-action space reduction within the structure of each option.

5 Discussion

TEXDYNA ideas are first inspired by Sutton's DYNA architecture [8], enriched and adapted to FMDPs by [6]. Second, as to the exit oriented options representation, some ideas come from the HEXQ [4] and VISA [10,5] frameworks, where the exit definition proposed in HEXQ is extended to include variable change and context in order to address more complex structures. TEXDYNA builds a hierarchy of options online and directly from the transition trees taking advantage of their structure, whereas VISA builds a variable influence graph from the given DBNs and then builds transition graphs and reachability trees to determine the initiation sets of the options. Furthermore, it only discovers options linked to the variables directly connected to reward, while TEXDYNA backpropagates the reward among subgoals.

Finally, Incremental-VISA [11,12] is adapting VISA to the case where the model of the problem is not known in advance. Like TEXDYNA, it attempts to simultaneously learn the hierarchical (options discovery) and factored (DBN learning) structure of the MDP. The approach learns incrementally and autonomously both the causal structure of the environment and useful skills that exploit this structure. It uses DBN structure learning techniques to learn the environment structure and SDP algorithms like SVI to build hierarchical policies online. The authors propose an active learning scheme to improve the efficiency with which this structure is acquired that bootstraps on existing structural and procedural knowledge. As new structure is discovered, more complex skills are learned, which in turn allow the agent to discover more structure, and so on. The major differences between TEXDYNA and Incremental-VISA is, for the moment, that

the latter is limited to the deterministic case, whereas TeXDYNA is adapted to stochastic problems. The second difference lies in the way of introducing options into planning: Incremental-VISA waits the option to be "mature enough" before introducing it in the hierarchy by using a measure of entropy on the transition functions whereas TeXDYNA inserts options directly in the hierarchy in order to accelerate its completion. In the same way as VISA, Incremental-VISA uses the DBNs to discover dependencies between state variables. Therefore, it needs to build intermediate graphs and trees to catch the internal hierarchical structure, while TeXDYNA operates on decision trees and discovers the structural links directly. Unfortunately, we were not able to compare the performance mainly because of the absence of a common metrics. Indeed, the experimental results published in [12,11] are based on the number of value changes and the time to compute the policy without giving the corresponding metrics criteria.

At this point in time, the algorithm presents some limitations. Firstly, instead of using "internal reward" to propagate the external reward to local policies so that all options have fixed interest, the options discovery algorithm could be combined with task-specific knowledge to identify useful, salient or challenging subroutines. Secondly, we consider that there is at most one option per variable value change. This assumption simplifies computations within the algorithms, but can be relaxed. More importantly, as to the problem representation, our option-specific state abstraction is strongly goal-oriented, that is reaching a unique exit context. This can result in the creation of excessive number of options in problems where an action can change more than one variable at the same time. Finally, the hierarchy of options is strictly ordered, that means that we cannot address problems where the FMDP structure includes synchronic arcs or post-action variable dependencies, because it would introduce cross-dependencies between options and cycles in the hierarchy of options.

6 Conclusion

We have presented TeXDYNA, a powerful framework that combines factored and hierarchical reinforcement learning. This framework is built on three main ideas:

- The use of the transition function structure represented as decision trees to discover options results in efficient learning and planning capabilities that are integrated into the (factored) model-based RL framework.
- The localization of the models results in an exponential reduction of the state-action space of each option. Instead of taking transition trees from the global structure, the models of transitions are learned locally for each option. This accelerates the solution process.
- The immediate use of the just discovered options in the planning process speeds up the learning of its internal structure and of its parent option structure.

We have evaluated on the LIGHT BOX problem the capability of TeXDYNA to efficiently generate hierarchical policies and shown that it performs better than its non factored and non hierarchical ancestors. The main issue for future work consists in replacing the basic ϵ-$greedy$ exploration strategy by a more sophisticated policy, either along the "optimism in the face of uncertainty" line [14] or based on adding internal motivations

such as an artificial curiosity process [15, 13] into the framework. Then the application of TEXDYNA to a robotics problem will be the matter of a more experimental work.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
2. Boutilier, C., Dearden, R., Goldszmidt, M.: Exploiting structure in policy construction. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, pp. 1104–1111 (1995)
3. Barto, A., Mahadevan, S.: Recent advances in hierarchical reinforcement learning. Discrete Event Systems Journal 13, 41–77 (2003)
4. Hengst, B.: Discovering hierarchy in reinforcement learning with HEXQ. In: Proceedings of the 19th International Conference on Machine Learning, pp. 243–250 (2002)
5. Jonsson, A., Barto, A.: Causal graph based decomposition of factored MDPs. Journal of Machine Learning Research 7, 2259–2301 (2006)
6. Degris, T., Sigaud, O., Wuillemin, P.H.: Learning the structure of factored markov decision processes in reinforcement learning problems. In: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, Pennsylvania, pp. 257–264. ACM, New York (2006)
7. Boutilier, C., Dearden, R., Goldszmidt, M.: Stochastic dynamic programming with factored representations. Artificial Intelligence 121(1-2), 49–107 (2000)
8. Sutton, R.S.: DYNA, an integrated architecture for learning, planning and reacting. In: Working Notes of the AAAI Spring Symposium on Integrated Intelligent Architectures (1991)
9. Sutton, R., Precup, D., Singh, S.: Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence 112, 181–211 (1999)
10. Jonsson, A.: A causal approach to hierarchical decomposition in reinforcement learning. PhD thesis, University of Massachusetts Amherst (2006)
11. Vigorito, C.M., Barto, A.G.: Autonomous Hierarchical Skill Acquisition in Factored MDPs. In: Yale Workshop on Adaptive and Learning Systems, New Haven, Connecticut (2008)
12. Vigorito, C., Barto, A.: Hierarchical Representations of Behavior for Efficient Creative Search. In: AAAI Spring Symposium on Creative Intelligent Systems, Palo Alto, CA (2008)
13. Singh, S., Barto, A., Chentanez, N.: Intrinsically motivated reinforcement learning. Advances in Neural Information Processing Systems 18, 1281–1288 (2004)
14. Szita, I., Lörincz, A.: The many faces of optimism: a unifying approach. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1048–1055. ACM, New York (2008)
15. Oudeyer, P.Y., Kaplan, F., Hafner, V.: Intrinsic motivation systems for autonomous mental development. IEEE Transactions on Evolutionary Computation 11 (2007)

Learning Robot-Environment Interaction Using Echo State Networks

Mohamed Oubbati, Bahram Kord, and Günther Palm

Institute of Neural Information Processing, University of Ulm
89069 Ulm, Germany
{mohamed.oubbati,bahram.kord,guenther.palm}@uni-ulm.de
http://www.uni-ulm.de/in/neuroinformatik.html

Abstract. Learning robot-environment interaction with echo state networks (ESNs) is presented in this paper. ESNs are asked to bootstrap a robot's control policy from human teacher's demonstrations on the robot learner, and to generalize beyond the demonstration dataset. Benefits and problems involved in some navigation tasks are discussed, supported by real-world experiments with a small mobile robot.

Keywords: robot-environment interaction, echo-state network, learning from demonstration.

1 Introduction

Any cognitive activity arises from interaction between bodies, brains and environments [1]. This means that a behaviour of a robot operating in and interacting with an environment cannot be analysed in isolation, since it is a result of properties of the robot itself (embodiment), the environment (situatedness), and the control program (task) the robot is executing. Thus, analysing the triangle (robot-task-environment) as one dynamical system could provide better understanding of phenomena governing the robots behaviours over time. Recurrent neural networks (RNNs) are powerful tools to learn such complex dynamical systems, for two main reasons. First, they are universal approximators of dynamical systems [2]. Second, they can exhibit continuous dynamics; a suitable property to model robot-environment interaction. The second reason is motivated by the fact that dynamics of nervous systems and the physical world are continuous in nature [3]. Under this perspective, many efforts have been done to analyze robot-environment interaction using RNNs. Beer and colleagues carried out rigorous analysis on understanding of a humanoid robot-environment interaction using continuous-time recurrent neural network [4]. Pasemann et. al. have shown that attractors formed in a recurrent neural network (RNN) can be used to characterise robot-environment interaction [5,6]. Their investigation shows that a RNN controller has four relevant attractors, which can be directly mapped to some environmental states like free space, obstacles, or deadlock situation. Tani et. al. explored higher cognitive abilities of robots through a multiple timescales RNN to generate reusable behaviors [7].

In this work, we assume that the robot learns by demonstration. A teacher demonstrates a task using the body of the robot *learner*, then a RNN learns from the robot-environment interaction to derive a control policy for the demonstrated behavior. This

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 501–510, 2010.

procedure is particularly important when considering real robots. Due to the fact that demonstration dataset are exactly those that the robot would observe/execute, the learning system will be less sensitive to noisy sensors. Also in term of system integration this approach provides fast and efficient way of obtaining the control code, i.e. a nonlinear dynamical model that maps environmental situations to actions. Our learning system is an *Echo State Network* (ESN), which has two principal properties: (1) a large RNN is used as a "reservoir" of excitable complex dynamics; this network will be not trained; (2) only the weights of output connections are to be adjusted [8]. Many dynamical systems, which were difficult to learn with the existing methods, have been easily learnt by ESN [9,10]. Recently, we used ESNs to develop a dynamic controller for mobile robots [11]. The advantage is that no knowledge about the robot model is required; a useful property in practical situations, where the exact knowledge about the physical parameters of the robot is almost unattainable. We also explored the notion that a well trained ESN needs to change only its internal state to change its behavior policy [12,13]. The ESN, in this work, is asked to bootstrap a robot's control policy from teacher's demonstration, and to generalize beyond the demonstration dataset. Naturally, we expect adequate performances only for system conditions that are close to those seen during training, such that valid solutions could be acquired for similar states that may not have been seen during demonstration.

The rest of this paper is organized as follows. Section 2 explains the learning design, and section 3 presents some implementation results. A discussion and conclusion are drawn in section 4.

2 Learning Design

The world (robot-task-environment) consists of states S and actions A, with the mapping between states from which actions being defined by a probabilistic transition function $T(\acute{s}|s,a) : S \times A \times S \rightarrow [0,1]$. We define the demonstration set D as k paires of observation and actions $D = \{(s_i, a_i)\}$, $s_i \in S$, $a_i \in A$, $i = 1, \ldots, k$. The ESN is then provided with the set D, and asked to aquire the task dependent control policy $\pi : S \rightarrow A$ to select desired actions based on current states.

2.1 Echo State Networks

An ESN (Fig. 1) is formed by a so-called "Dynamic Reservoir"(DR), which contains a large number of sparsely interconnected neurons with non-trainable weights. The activation of internal neurons is updated according to

$$X(n+1) = f(W^{in}U(n+1) + WX(n) + W^{back}Y(n+1)) \tag{1}$$

and the outputs are calculated as

$$Y(n+1) = f^{out}(W^{out}(U(n+1), X(n+1), Y(n))) \tag{2}$$

An essential condition for successful using of ESN is the "echo state" property. It is a condition prior to training, where the actual network state is required to be an "echo" of its history. This means that the state of each internal neuron x_i can be mapped by input/output (u/d) histories through a function e_i, i.e. $x_i(n) = e_i(d(n-1), d(n-2), \ldots, u(n), u(n-1), \ldots)$ [9]. If this condition is met, only weights connections from internal neurons to the output (W^{out}) are to be trained. This could be done by any suitable training method (least squares method, etc.) in a one-shot fashion [8].

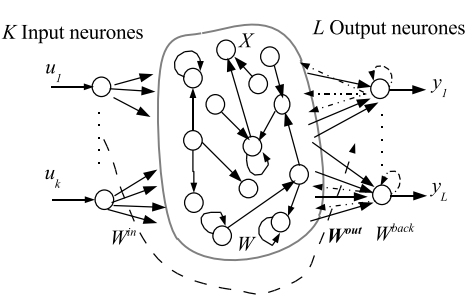
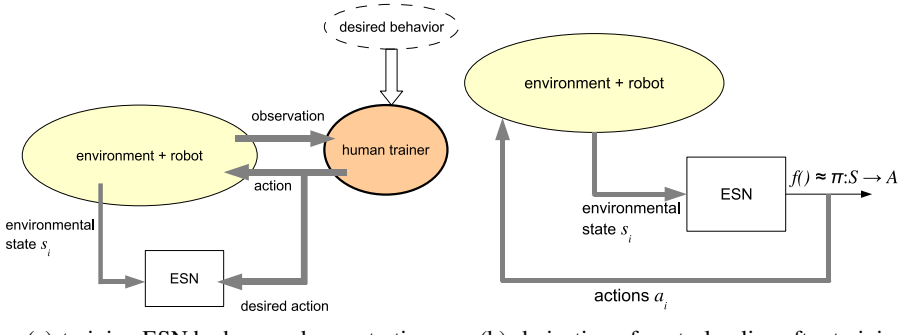


Fig. 1. Basic architecture of ESN. Dotted arrows indicate connections that are possible but not required.

2.2 Deriving Policy with ESNs

During demonstration, a teacher operates the robot while recording its sensor/actuator data. Once the demonstration set $D = \{(s_i, a_i)\}$ have been gathered, the ESN performs a batch learning to derive the control policy $f \approx \pi : S \rightarrow A$ (Fig. 2).

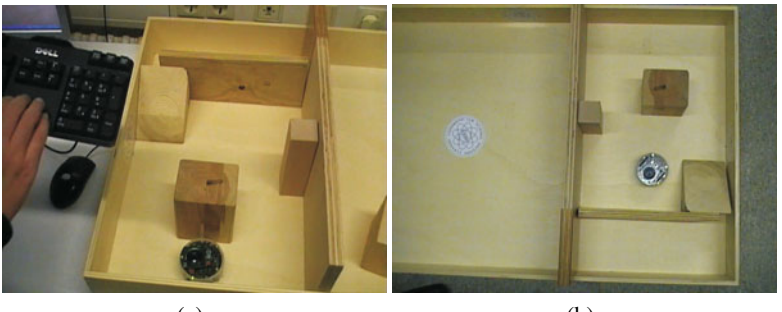


(a) training ESN by human demonstration (b) derivation of control policy after training

Fig. 2. Learning design

3 Implementation

We implemented the learning design on a small mobile robot, called e-puck [14], equiped with 8 infrared (IR) proximity sensors and two stepper motors. We drove the robot manually several times for the task to be learned. The linear velocity of the robot (v) was kept constant while the angular velocity (w) was controlled by a human operator using a keyboard (Fig. 3). During the movement we collected sensor readings and angular velocities for training. Training was performed using ESNs with 8 inputs (8 IRs), N internal neurons, and one output that represents the desired angular velocity transformed to two wheel speeds. No back-connection from the output to the DR, and no synaptic weight connections from the input directly to the output. The input and the internal synaptic connections weights were randomly initialized from a uniform distribution over $[-1, +1]$. The internal weight matrix W has a sparse connectivity of 20% and scaled such that its maximum eingenvalue $|\lambda_{max}| = \alpha$ (also called spectral radius). After training, the ESN provides the desired angular velocity based on the actual sensor-readings (Fig. 4). We performed several experimental tests; three of them are reported here: *door-passing*, *wall-following*, and *route-learning*. We will show the effect of the internal neurons number N and the spectral radius α on the ESN performance.



(a) (b)

Fig. 3. Implementation. (a) demonstration made on the robot learner. (b) after training, the ESN controls the robot.

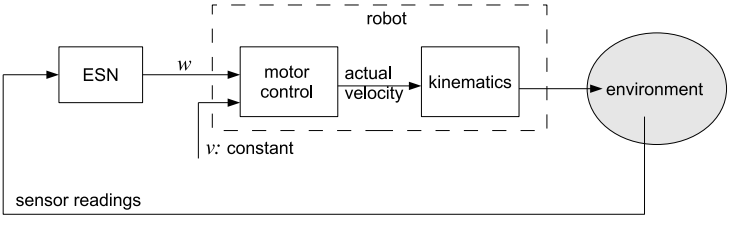


Fig. 4. Robot control with ESN

3.1 Door-Passing

Here, the robot learns how to move through a door. Demonstration data were prepared by driving the robot several times in a door-passing scenario, while collecting sensor readings and correspondent angular velocities (Fig. 5). After training, we put the robot in a scenario 1 where the robot should pass through two openings (Fig. 6, a). The spectral radius $\alpha = 0.8$, and the number of neurons in the reservoir N recieves the values $\{6, 8, 12\}$. During this test we were surprised to see that even with 6 internal neurons the ESN could generalize successfully and bring the robot through the two doors. In a more complicated environment (scenario 2), we expect that the robot moves through the openings D_2 and D_4 (Fig. 6, b). The number of neurons in the reservoir is kept to 12, and α is varied from 0.4 to 0.8 in a step of 0.2. We can see that the ESN with $\alpha = 0.8$ performed the best, regarding the smoothness of the robot path obtained with this spectral radius.

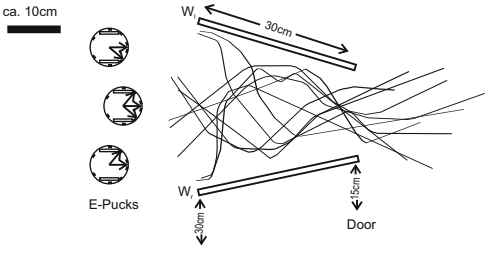
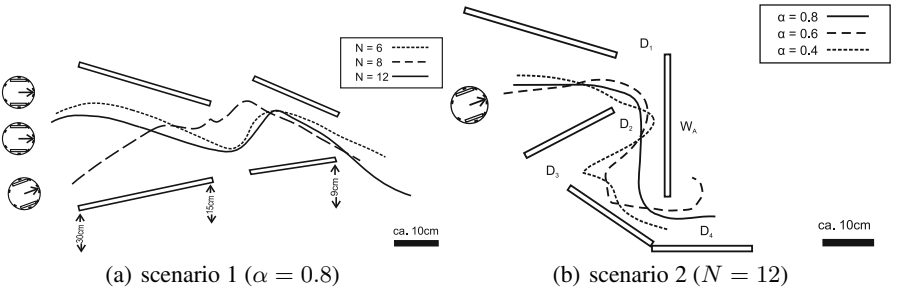


Fig. 5. Collection of training data for door-passing: The teacher moves the robot from different start positions in the environment in order to demonstrate the most possible situations. Demonstrations were in a form of 10 movements producing 750 data pairs sampled with a period of 0.1 second.



(a) scenario 1 ($\alpha = 0.8$) (b) scenario 2 ($N = 12$)

Fig. 6. Door-passing

3.2 Wall-Following

In this task two main situations need to be learned: (1) move parallel to a wall, and (2) turn in a concave corner. We have first collected training data by driving the robot several times in these two situations as illustrated in Fig. 7. Due to the smooth movement of the robot during previous experiment we kept the spectral radius to $\alpha = 0.8$, and we tested the ESN performance with different number of internal neurons $N = \{6, 8, 12\}$ (Fig. 8). To show the degree of similarity between the actual and the demonstrated robot behaviors we did a test in a form of occurence of the robot distance to the wall in training and test data (Fig. 9). The distance was measured using one infrared sensor (IR_3) on the robot. By merely looking at these results, we can subjectively say that ESNs could reproduce successfully the demonstrated behavior, since the IR_3 delivered values lie often between $[0, 1000]$. Fig. 10 shows a recovery testing situation, in which we put an object (A) at the corner and we made a small opening (B) (2 cm) in the wall. Moving near (A) or (B) means that the ESN has to deal with a convex corner at (A) (not seen before), and maintain its stability in presence of sensory inputs totally beyond the intervall of traning data at (B). An ESN with small dimension (6 and 8 internal neurons) showed a good performance to recover those perturbations, whereas using large dimension (more than 12 internal neurons) the control loop lost stability at many times. We can see that with 12 neurons the system began to lose its stability, when reaching (B).

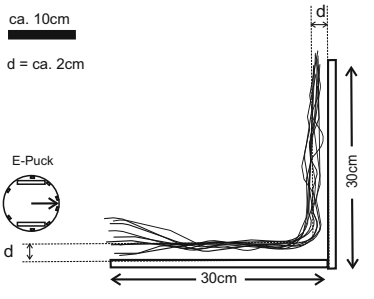
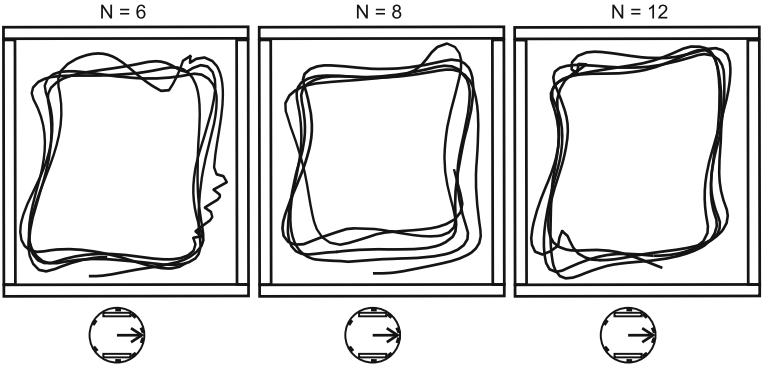


Fig. 7. Demonstration data for the task wall-following. IR sensors and wheel speeds were collected by moving the robot several times to show how to follow a wall and how to deal with a concave corner. 10 movements were collected in a form of 693 pairs sampled with a period of 0.1 second.

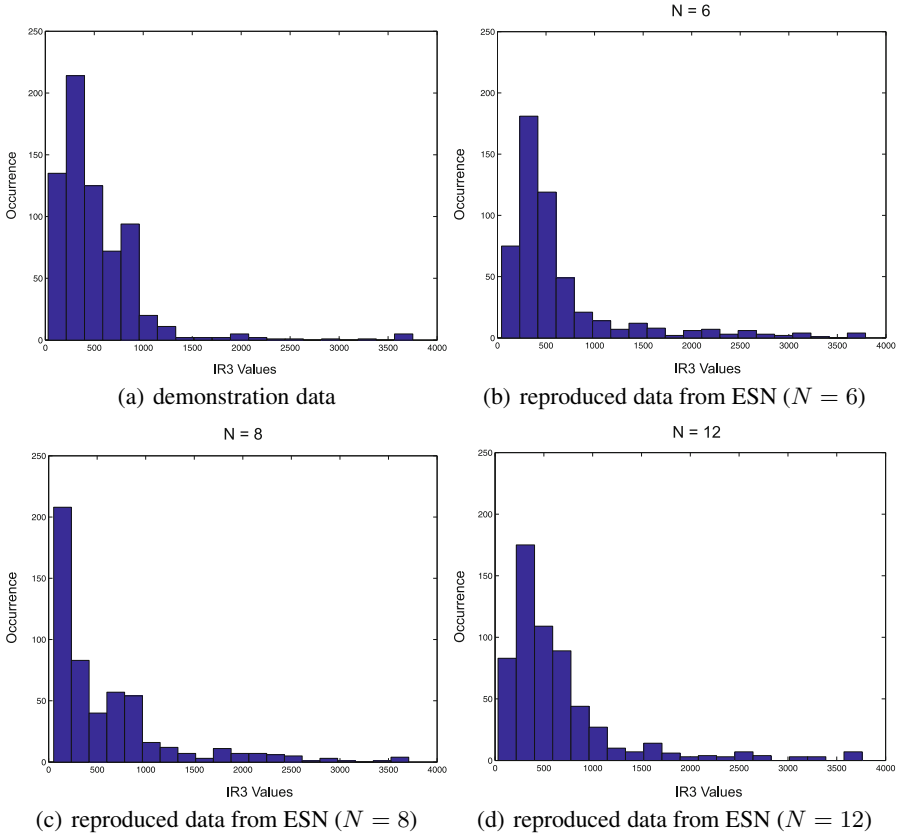
3.3 Route-Learning

Here, we demonstrate the ability of an ESN to learn a more challenging task; *learning a route*. The robot was led many times along a desired route to collect rich information about the environment. Fig. 11. (a) shows 20 rounds collected to train the ESN controller, where the two red dashed lines represent the boarders of route to be learned. After training process is completed, the acquired perceptions are associated with motor actions, enabling the robot to follow the route autonomously. Fig.11. (b) shows that the trained robot follows the desired trajectory well, with few deviations. This result shows also how the ESN is reliable and copes with noise and new environmental perceptions.



(a) reproduced data from ESN ($N = 6$) (b) reproduced data from ESN ($N = 8$) (c) reproduced data from ESN ($N = 12$)

Fig. 8. Wall-following: The results show that the trained ESNs could accomplish this task successfully, but the robot behavior is found to be almost identical for the different neurons N.



(a) demonstration data (b) reproduced data from ESN ($N = 6$)

(c) reproduced data from ESN ($N = 8$) (d) reproduced data from ESN ($N = 12$)

Fig. 9. Occurences of the robot distance to the wall from demonstrated and reproduced data

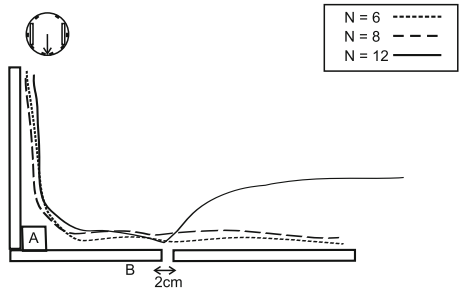


Fig. 10. Wall-following with recovery testing: We put an object (A) at the corner, which means that the robot has to deal also with a convex corner, and we made a small opening (B) (2 cm) in the wall, such that sensory data in those situations are totally beyond the intervall of traning data. ESNs with 6 and 8 neurons could recover those perturbationslearner, whereas with 12 neurons the control loop began to lose stability.

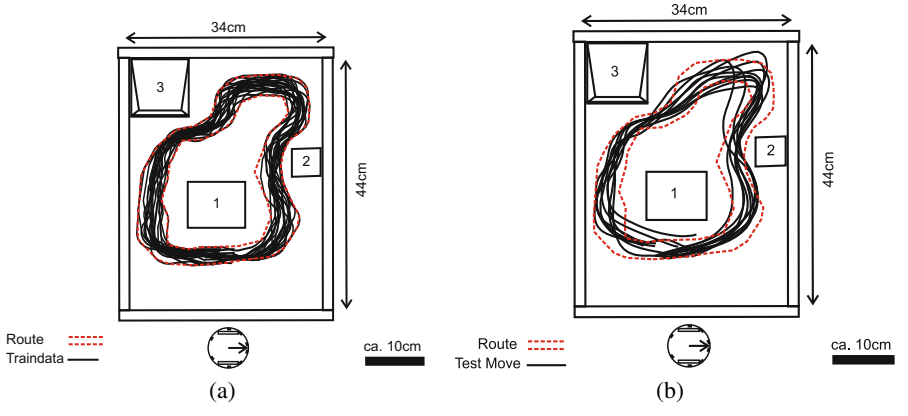


Fig. 11. Route learning with ESN. (a) human teacher led the robot between obstacles (1,2,3) to collect training data in a form of 2390 samples. (b) ESN with 20 internal neurons and $\alpha = 0.7$ controls the robot to reconstruct the desired route.

For example when the robot deviate from the trained route, due to its kinematic constraints, the ESN could generalize, and generate appropriate control signals in order to bring the robot again on its desired path.

4 Discussion and Conclusion

In this paper we have implemented the ESN as a learning system to derive a control policy from robot-environment interaction. A similar work has been done in [15], where the task consists in finding and then getting closer to a possibly moving target. The ESN has shown great performance to solve that task; a result that pushed us to test the ESN in more complicated tasks. Furthermore, we wanted, in this paper, to see the effect of

the internal neurons number N and the spectral radius α on the ESN performance in performing the tasks of *door-passing*, *wall-following*, and *route-learning*. Once these behaviors have been demonstrated using the body of the robot learner, three ESNs performed batch learning on demonstration datasets to derive a control policy for those tasks. The results have shown that an ESN even with 6 internal neurons could reproduce and generalize successfully the door-passing task. Experiments have also shown that an ESN with $\alpha = 0.8$ performed the best, regarding the smoothness of the robot path obtained with this spectral radius. In the wall-following task the robot has learned how to maintain a distance parallel to the wall, and how to deal with a concave corner. The robot reproduced successfully the demonstrated behavior, and also showed a great robustness against new situations, i.e. in presence of convex corner, and an openning in the wall. The task of route-learning was the more challenging to be learned, since each IR-sensor plays an important role in the learning process. The result shows that the trained robot follows the desired trajectory well, with few deviations. These deviations means that the ESN recieves completely new environmental perception; a hard generalization test. In those new states the ESN could generalize and generate appropriate control signals in order to bring the robot again on its desired route.

We note that the learning system has also shown poor performances in many situations. We identified three main causes of that. The first cause is due to the dataset sparsity; undemonstrated areas in the state space. Using a keyboard, the human trainer was not able to demonstrate behaviors in all areas of the state space. This has raised the question of how the robot should act when it encounters a state without a demonstration. When novel states (very different from previously demonstrated states) are encountered by the robot, the ESN could not generalize optimally. In those situations, it was necessary to acquire additional demonstrations, and re-make training. The second cause was the quality of the dataset. The poor quality of the employed sensors as well as the kinematic precision of the utilized robot has generated dataset ambiguity in some situations. For example, during multiple execution of a desired task, different actions has been mapped to almost identical states. The third cause was the learning system itself. During preparation of the ESN, it was not easy to find its optimum parameters. Using a "relatively" large dimension (more than 20 internal neurons) the network lost stability at many times. This is possibly due to the high dimensionality of the system, or to the relatively small amount of training data. We have made a similar observation in another experiment, when we trained an ESN as an adaptive velocity controller for an omnidrive mobile robot [13].

There are many research questions to be addressed in the future. We summarize three main points:

- What is the behavioral capacity of ESNs? In this work, we trained separate ESNs for each task. We did not discuss the question whether a single ESN could learn and reproduce all demonstrated behaviors. In a previous work [12] we have shown that an ESN needs to change only its internal state to change its behavior policy. Does learning with fixed-weights perform well in these tasks?

- How to improve generalization? When a completely novel state is encountered by the robot, the ESN is unable to produce an adequate action. In this case, the ESN should be able to recognize the lack of knowldge and solicit help from the teacher.

– How to deal with symetrie in a route learning? In a route learning it is possible to encounter similar situations which need different actions. The short term memory in ESNs might provide a solution.

References

1. Pfeifer, R., Bongard, J.C.: How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books). The MIT Press, Cambridge (2006)
2. Funahashi, K.-i., Nakamura, Y.: Approximation of dynamical systems by continuous time recurrent neural networks. Neural Network 6(6), 801–806 (1993)
3. Beer, R.D.: A dynamical systems perspective on agent-environment interaction. Artif. Intell. 72(1-2), 173–215 (1995)
4. Beer, R.D.: The dynamics of active categorical perception in an evolved model agent. Adaptive Behavior 11(4), 209–243 (2003)
5. Pasemann, F.: Dynamics of a single model neuron. International Journal of Bifurcation and Chaos 3, 271–278 (1993)
6. Hülse, M., Zahedi, K., Pasemann, F.: Representing robot-environment interactions by dynamical features of neuro-controllers. In: Butz, M.V., Sigaud, O., Gérard, P. (eds.) Anticipatory Behavior in Adaptive Learning Systems. LNCS (LNAI), vol. 2684, pp. 222–242. Springer, Heidelberg (2003)
7. Tani, J., Yamamoto, J.: On the dynamics of robot exploration learning. Cognitive Systems Research 3(3), 459–470 (2002)
8. Jaeger, H.: Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach. Technical Report 159, AIS Fraunhofer, St. Augustin (2002)
9. Jaeger, H.: The 'echo state' approach to analysing and training recurrent neural networks. Technical Report 148, AIS Fraunhofer, St. Augustin, Germany (2001)
10. Jaeger, H., Haas, H.: Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. Science (April 2004)
11. Oubbati, M., Schanz, M., Buchheim, T., Levi, P.: Velocity control of an omnidirectional robocup player with recurrent neural networks. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 691–701. Springer, Heidelberg (2006)
12. Oubbati, M., Schanz, M., Levi, P.: Meta-learning for Adaptive Identification of Non-linear Dynamical Systems. In: Proc. Joint 20th IEEE International Symposium on Intelligent Control and 13th Mediterranean Conference on Control and Automation, Limassol, Cyprus. IEEE, Los Alamitos (June 2005)
13. Oubbati, M., Palm, G.: A neural framework for adaptive robot control. Journal of Neural Computing and Applications 19(1), 103–114 (2010)
14. Mondada, et al.: The e-puck, a robot designed for education in engineering. In: Proc. of the 9th Conf. on Autonomous Robot Systems and Competitions, vol. 1, pp. 59–65 (2009)
15. Hartland, C., Bredèche, N.: Using Echo State Networks for Robot Navigation Behavior Acquisition. In: ROBIO 2007, Sanya Chine (2007)

A Novel Information Measure for Predictive Learning in a Social System Setting

Paolo Di Prodi[1,⋆], Bernd Porr[2], and Florentin Wörgötter[3]

[1] University of Glasgow
epokh@elec.gla.ac.uk
[2] University of Glasgow
b.porr@elec.gla.ac.uk
[3] BCCN Göttingen, Germany
worgott@bccn-goettingen.de

Abstract. We introduce a new theoretical framework, based on Shannon's communication theory and on Ashby's law of requisite variety, suitable for artificial agents using predictive learning. The framework quantifies the performance constraints of a predictive adaptive controller as a function of its learning stage. In addition, we formulate a practical measure, based on information flow, that can be applied to adaptive controllers which use hebbian learning, input correlation learning (ICO/ISO) and temporal difference learning. The framework is also useful in quantifying the social division of tasks in a social group of honest, cooperative food foraging, communicating agents.

Simulations are in accordance with Luhmann, who suggested that adaptive agents self-organise by reducing the amount of sensory information or, equivalently, reducing the complexity of the perceived environment from the agents perspective.

1 Introduction

Information measures are usually defined for input/output systems where they determine the quality of the transmission. Behaving agents, however, act as closed loop systems in which there is no clearly defined difference between input and output. What matters most for the organism is to compensate for disturbances introduced by the environment in the perception action loop. If there is no disturbance, the organism cannot differentiate between themselves and the environment. Consequently, the concept of information in these systems needs to be revised [5].

A method for defining closed loop information has been proposed by Ashby - the so called *requisite variety* [1]. The measure is based on the premise that closed loop systems aim to maintain a desired state. The goal of a feedback loop is then to minimise the deviation from the desired state i.e. the number of bits required to successfully compensate a disturbance acting on the forward

⋆ Webpage: http://isg.elec.gla.ac.uk.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 511–522, 2010.

loop. In this way, the method quantifies the variety, or bits, originating from the disturbance. For example, if the disturbance has a variety of 10 bits and survival requires a desired state of 2 bits, then the reaction to that disturbance must provide a variety of 8 bits. Ashby then proved that error controlled closed loop systems (like PID controllers [21]) cannot achieve perfect regulation. More recently, Touchette et al. [24] in Theorem 10 proved that the entropy reduction achieved by a closed loop system is bounded by the entropy reduction achieved by the open loop control plus the mutual information gathered by the estimation of the state. However the advent of predictive controllers, such as Q-learning [22], that predict future states requires an extension of the information theory for predictive learning.

In this paper we present an extension to the law of requisite variety, called *the predictive requisite variety*, that quantifies the theoretical limits of control (as well as providing a performance index) for predictive adaptive controllers. We argue that a predictive adaptive controller acts as a reactive system before learning and as an open loop forward system after learning. A reactive system comprises an error controlled closed loop and is non optimal because it only reacts after a deviation from its desired state has happens. The environment contains usually predictive signals which can help the agent to react before the error is presented [16]. Thus, bio inspired controllers can be provided with a predictive signal (like vision) and a reflexive signal (like touch). Learning then has the task of avoiding the trigger of the reflexive reaction - thus creating an open loop forward controller which discards the information of the reflexive signal.

Learning is then quantified by the increase in the information flow of the predictive loop and by a corresponding decrease in the information flow of the closed loop. Information flow, or transfer entropy, is not a new idea (see for example [3,23]) but it has never been applied to predictive agents in order to assess their learning performance. The analysis of a predictive agent with a single behaviour, say for example obstacle avoidance, can be done calculating the information flow of the sensory-motor loop.

Analysis becomes more complicated when an agent is provided with a set of competitive behaviours in a social scenario where agents use predictive learning-see, for example, ISO [4] or ICO [17,4] - and are therefore learning from each other. The task of the social system in this analysis is cooperative food foraging in which every agent has 3 adaptive behaviours which are: avoidance for obstacles, attraction to food disks and attraction to others with food. Agents communicate honestly, always signalling to others when they find food. When the social system is adapting, it self-organises into 2 sub-systems each described by a dominant behaviour: seekers have a dominant attraction for food disks, parasites have a dominant attraction to others with food. The information flow explains how the social system divides itself into sub-systems by looking at the information processing of every agent. Luhmann [13] proposed that differentiation of social systems is caused by a decrease in information processing of each subsystem and this is consistent with our information flow measurements.

The paper is divided in sections covering the following topics: regulation and entropy (as defined originally by Ashby), a new information measure for predictive learning, a simulation model with social adaptive agents, results, and a discussion.

2 Ashby's Law of Requisite Variety

First, we review Ashby Law of Requisite Variety for the forward (see Fig.1(B)) and closed loop controller (see Fig.1(A)). Fig.1 uses the same notation introduced by Ashby:

- D= finite state machine whose states are the disturbances from the environment
- E= finite state machine whose states are the essential variables partitioned in $E = \eta \cup \overline{\eta}$, where η is a partition of desired states or goals of the organism and its complementary partition $\overline{\eta}$ represents the non-desired states.
- R= finite state machine whose states are the available regulations/actions that the organism can perform
- T= finite state machine whose states are the set of possible states of the environment

In this work we consider deterministic finite state machines but the analysis can also be extended to Markov processes [6]. It is very important for our analysis to understand that only the forward controller can achieve perfect regulation whereas the closed loop controller cannot because the reflex always comes too late. Gatsby [1] stated that a good controller R blocks the flow of variety[1] from disturbances D to essential variables E: if R is a regulator, the insertion of R between D and E decreases the variety that is transmitted from D to E. An organism can be described by a body R with goals to be achieved η and an environment T which forms a closed loop between actions and sensors. As an analogy, the organism is a perfect regulator if is able to keep the essential variables E within a desired sub-set η in spite of the disturbances D -thus having a null entropy for E, $H(E) = 0$.

If no regulator R is provided (see Fig.1(C)), the disturbance D tends to drive E_0 outside a set of desired states η by means of the environment T, .Thus, in the worse case, the disturbance completely controls the status of the organism:

$$H(D) = H(E_0) \tag{1}$$

The regulator R can be connected in a feed-forward configuration as in Fig.1(B) or in a closed loop configuration as in Fig.1(A). The performance of the forward regulator is measured by the maximum entropy reduction $\Delta H^{max}_{forward}$ which is the difference between the entropy of the essential variable $H(E_0)$ before regulation and after regulation $H(E)$.

$$\Delta H^{max}_{forward} = H(E_0) - minH(E) \tag{2}$$

[1] Ashby defines variety precisely as the number of different states a variable can take and is equivalent to the Shannon's entropy H measured in bits.

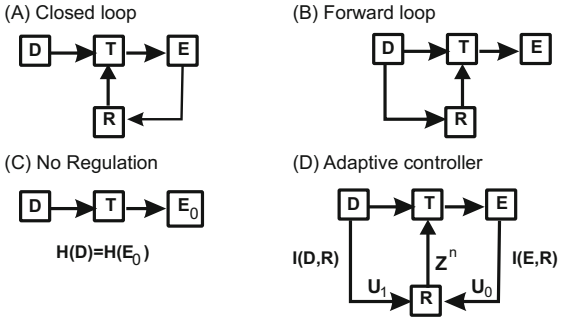


Fig. 1. (A) The organism with a closed loop controller. (B) The same organism with an forward controller. (C) The organism before regulation. (D) An adaptive controller is a mix of forward and closed loop control. Every block is a finite state machine whose inputs are indicated by incoming arrows and outputs are indicated by outgoing arrows.

The maximum entropy reduction in the forward condition $\Delta H^{max}_{forward}$ can be calculated by using the Law of Requisite Variety:

$$H(E) \geq H(D) + H(R|D) - H(R) \tag{3}$$

where $H(R|D)$ is the regulator noise[2]. Thus:

$$\Delta H^{max}_{forward} = H(R) - H(R|D) \tag{4}$$

because combining Eq.2 and Eq.3 gives:

$$\Delta H^{max}_{forward} = H(E_0) - H(D) - H(R|D) + H(R) \tag{5}$$

Considering the initial condition in Eq.1 we obtain Eq.4:

$$\Delta H^{max}_{forward} = H(D) - H(D) - H(R|D) + H(R) = H(R) - H(R|D) \tag{6}$$

The quantity $\Delta H^{max}_{forward}$ in Eq.4 tells us that better performance can be achieved by either increasing the regulation entropy $H(R)$ or by decreasing the controller noise $H(R|D)$.

We will now show that a closed loop controller cannot achieve perfect regulation ($H(E) = 0$) as it requires a deviation from the desired state η to work $H(E) > 0$. Thus, the disturbance transmits all its entropy to the essential variable $H(D) = H(E)$ and no entropy reduction can be achieved:

$$\Delta H^{max}_{close} = 0 \tag{7}$$

If for $H(E) = 0$ then R blocks the information flow in the channel $D \rightarrow E$ and thus no information is transmitted to R for the regulation task: the regulator R is asserting a perfect control on E without knowing the status. In the next section we extend the law of requisite variety for adaptive controllers.

[2] If the controller is not noisy $H(R|D) = 0$.

3 Law of Adaptive Requisite Variety

An adaptive controller (see Fig.1(D)) is a mix of a forward [8] and closed loop controllers [21] because R has now 2 inputs: D and E. We can think of D as a predictor of the deviation of E, because D transfers its entropy to E by means of the environment T.

In order to explain the new law, we introduce the mutual information $I(E,R)$ for the closed loop channel $E \rightarrow R$ with the corresponding channel capacity $C_{E,R}$:

$$I(E,R) = H(E) + H(R) - H(E,R) \tag{8}$$

$$C_{E,R} = \max_{p(E)} I(E,R) \tag{9}$$

the mutual information $I(D,R)$ for the forward channel $D \rightarrow R$ with the corresponding channel capacity $C_{D,R}$:

$$I(D,R) = H(D) + H(R) - H(D,R) \tag{10}$$

$$C_{D,R} = \max_{p(D)} I(D,R) \tag{11}$$

The channel capacity of the regulator channel $D \rightarrow T$ is then $C_{R,T}$.

The adaptive controller (denoted ada) begins as a closed loop controller with $\Delta H_{ada}^{max}(before) = H_{close}^{max}$ (see Eq.7) as it mainly uses the $E \rightarrow R$ reflex channel and blocks the $D \rightarrow R$ predictor channel whose mutual information is very low. In summary:

$$0 < I(E,R) \leq C_{E,R} \tag{12}$$

$$I(D,R) \simeq 0 \tag{13}$$

$$\Delta H_{ada}^{max}(before) = 0 \tag{14}$$

The adaptive controller achieves perfect regulation (see Eq.4) when

$$\Delta H_{ada}^{max}(after) = H_{forward}^{max} \tag{15}$$

because it blocks the $E \rightarrow R$ reflex channel and opens the $D \rightarrow R$ predictor channel. To summarise:

$$0 < I(D,R) \leq C_{D,R} \tag{16}$$

$$I(E,R) \simeq 0 \tag{17}$$

$$\Delta H_{ada}^{max}(after) = H(R) - H(R|D) \tag{18}$$

If we assume realistically that the regulator has a common channel capacity $C_{E,R} = C_{D,R} = C_{R,T}$, the constraint for learning becomes:

$$I(E,R) + I(D,R) \leq C_{R,T} \tag{19}$$

thus an adaptive controller can achieve optimal regulation $\Delta H_{ada}^{max}(after)$ when is able to compensate the mutual information of the closed loop $I(E,R)$ with

the mutual information of the forward controller $I(D,R)$. An imperfect regulator will likely work in the sub-optimal regime $I(D,R) < I(E,R)$. So to quantify the performance of an adaptive predictive controller we have to compute the mutual informations $I(D,R)$ and $I(E,R)$. This is however not always possible because it is hard to identify the reflex channel and the predictor channel. Therefore in the next section we use an approximation of these 2 quantities using the concept of information flow.

4 Information Flow for Adaptive Predictive Controllers

Looking at Fig.1(D), we can estimate $I(E,R)$ by computing the information flow of the reflex-output channel $Z^n \rightarrow U_0$ and $I(D,R)$ by computing the information flow of the predictive-output channel $Z^n \rightarrow U_1$. We denoted them as:

$$MI^n_{U0} = I(Z^n, U_0) \leftrightarrow I(E,R) \tag{20}$$

$$MI^n_{U1} = I(Z^n, U_1) \leftrightarrow I(D,R) \tag{21}$$

where $U0$ is the reflex input, $U1$ is the predictor input and Z^n the extended output:

$$Z^n = [z(k)z(k+1)\dots z(k+n-1)] \tag{22}$$

which contains n outputs of the agent and U the random variable describing the temporal signal $u(k+n)$ which is the input of the agent resulting from previous actions(for more details see [11,14]). Fig.2(A) shows an organism composed of 3 ICO [17] controllers and the corresponding information flow measures for every controller. Each ICO controller takes 2 continuous inputs $U0, U1$ and one continuous output Z_n. ICO correlates the predictive signal u_1[3] with the derivative of the reflexive signal u_0 according to the formula:

$$\frac{d\omega_1}{dt} = \mu \cdot u_1 \cdot \frac{du_0}{dt} \tag{23}$$

where ω_1 is the gain of the predictive signal u_1 and μ is the learning speed (see Fig.2(C)). Since the ICO controller works in continuous mode, the input and output signals must be discretized in order to compute the information flow and channel capacity (see Simulation Details). The two measures MI^n_{U0}, MI^n_{U1} are used to compute the channel capacities $C_{E,R}$ and $C_{D,R}$:

$$\zeta^n(Z^n \rightarrow U0) = \max_{p(Z^n)} MI^n_{U0} \leftrightarrow C_{E,R} \tag{24}$$

$$\zeta^n(Z^n \rightarrow U1) = \max_{p(Z^n)} MI^n_{U1} \leftrightarrow C_{D,R} \tag{25}$$

In the simulations in the next section, we will estimate the mentioned quantities for individual agents of a social group.

[3] u_1 and u_0 indicates temporal signals $u_1(t)$ and $u_0(t)$.

5 Methods

The previous measures are applied to a social system where all agents learn continuously from each other and from the environment. This scenario is very interesting because the social system is able to self-organise by forming 2 subsystems with task division. The social system described in [15] is composed of N identical agents and M food disks randomly placed in a square world for every simulation. Food disks contain a certain amount of food that is depleted when an agent finds it. The task is cooperative food foraging. The simulated agent is shown in Fig.2(B) and has also been used by [12]: it is a Braitenberg [7] vehicle with 2 lateral wheels and 2 antennas. By default the agent drives straight forward, with speed $v = 1$ units per time step. It has 2 sensor-pairs, near contact antennas and far contact antennas. Every agent has a MISO (multiple inputs single output) controller and a variable of 1 bit for the food status. The agent has competitive 3 tasks: avoid obstacles (empty food disks and other agents without food), find food from the disks, find foods from other agents with food. The MISO is composed of 3 parallel ICO controllers (see Fig.2(A)) which are provided with a reflex input error u_0, a predictive signal error u_1, a learnt weight ω_1 and an output z. The outputs of the 3 ICO controllers are summed to $z = z_{Av} + z_{Fo} + z_{Af}$[4] which gives the steering angle: $z = 0$ the robot goes straight forward at speed v, $z > 0$ the robot rotates clockwise, $z < 0$ the robot rotates anti-clockwise. Every

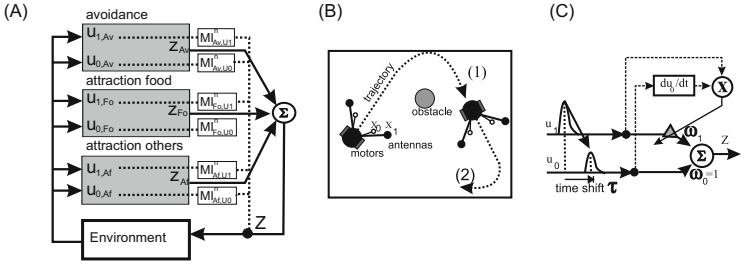


Fig. 2. (A) MISO controller composed of 3 stacked ICO controllers for avoidance, food attraction and attraction to others. The output of every controller is summed to z. For every controller/behaviour the pair of mutual information is computed between the output and the input MI^n_{U0}, MI^n_{U1}. (B) Agent with short antennas (reflexive inputs, x_0) and long antennas (predictive inputs, x_1). The agent is learning to avoid obstacles. The motor reaction will reduce the intensity of the painful reflex x_0 as well as delay its occurrence. (C) Schematic diagram of the input correlation learning rule and the signal structure [17]. The u_0 and u_1 are, respectively, the difference between the filtered values of the left and right antennas of the agent. During learning the u_0 peak will be shifted in time and reduced in amplitude as the agent learn successfully by increasing the predictor gain ω_1.

[4] Av stands for obstacle avoidance, Fo for food attraction and Af for attraction to others with food.

simulation is run for $0 \leq k \leq 6 \cdot 10^5$ time steps and is divided in 3 stages. At every stage, each agent produces 6 input time series and 1 output time series $z(k)$ which means that we can calculate the information flow for every pair of reflex-output and predictor-output: MI^n_{U0}, MI^n_{U1}. For a single simulation:

1. for $0 \leq k_1 \leq 2 \cdot 10^5$ all agents are reactive ($\mu = 0$). For each agent $i = 1, ..., N$ we have 3 pairs of information flow:
   (a) avoidance: $MI^n_{Av,U1}, MI^n_{Av,U0}$
   (b) food attraction: $MI^n_{Fo,U1}, MI^n_{Fo,U0}$
   (c) others attraction: $MI^n_{Af,U1}, MI^n_{Af,U0}$
2. for $2 \cdot 10^5 < k \leq 4 \cdot 10^5$: every agent is learning $\mu = 1.0$ and the weight for every ICO controller $\omega_{1,Av}$,$\omega_{1,Fo}, \omega_{1,Af}$ is increasing.
3. for $4 \cdot 10^5 < k_3 \leq 6 \cdot 10^5$: every agent stop learning $\mu = 0.0$ and is using the last weight set at $k = 4 \cdot 10^5$. For each agent we compute again the 3 pairs of the MI^n.

The channel capacities for every agent are computed by providing each isolated output $z = z_{Av}$,$z = z_{Fo}$,$z = z_{Af}$ with a source of independent randomness during a simulation of $2 \cdot 10^5$ time steps for every case. Then we apply the Blahut-Arimoto algorithm [20,18] with a bound error of $\varepsilon = 10^{-11}$ and 5000 maximum iterations to estimate the channel capacity for every agent in the reflex-output loop $\zeta^n(Z^n_k \rightarrow U0)$. There is no difference between $\zeta^n(Z^n_k \rightarrow U0)$ of every agent so we define ζ^n_{all}. To compute the capacity for the predictor-output loop $\zeta^n(Z^n_k \rightarrow U1)$ we use the same approach but preset the weights of every agent to an arbitrary high value to simulate perfect learning:

$$\omega_{1,Av} = 10.0, \omega_{1,Fo} = 10.0, \omega_{1,Af} = 10.0 \tag{26}$$

and we obtain the same results

$$\zeta^n(Z^n_k \rightarrow U1) = \zeta^n(Z^n_k \rightarrow U0) = 2.0 \tag{27}$$

for $n \geq 2$ as anticipated in Eq.24,25.

6 Results

The results of this sections are based on a simulation with $N = 10$ agents and $M = 5$ food disks. All agents start with the same weights for every ICO controller $\omega_{1,Av} = 0.1$,$\omega_{1,Fo} = 0.1, \omega_{1,Af} = 0.1$. In stage 3 there are 5 agents with $\omega_{1,Af} < \omega_{1,Fo}$ and 5 agents with $\omega_{1,Af} > \omega_{1,Fo}$. The first group is identified by a strong attractive behaviour for the food disks (seekers), whereas the second group is identified by a strong attractive behaviour for others agent with food (parasites).

We estimate the MI^4 in stage 1 and stage 3 for every agent by using the corrected standard deviation formula [19]. Before learning (Fig.3 (A),(D)) the

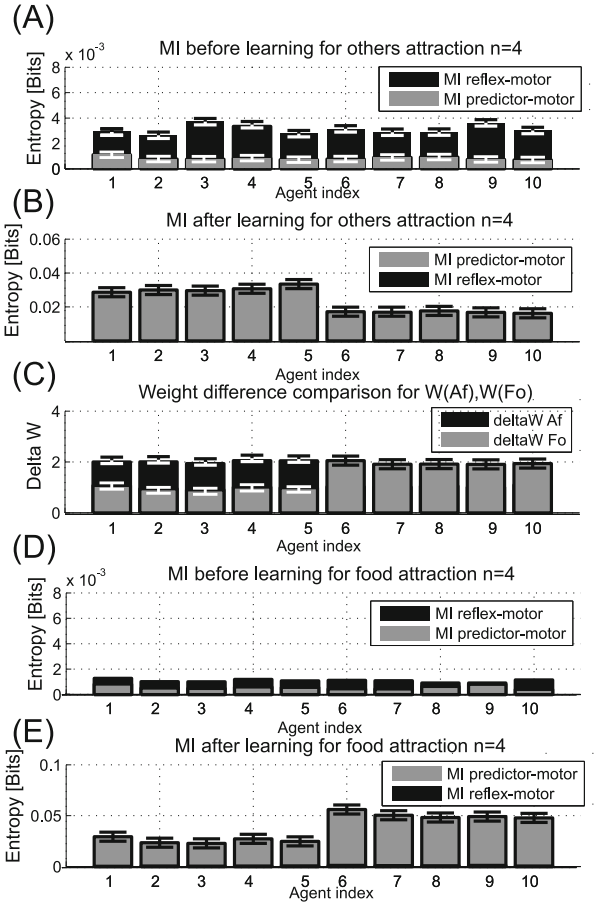


Fig. 3. (A) Information flow before learning for attraction to others $MI^4_{Af,U1}$ (grey bars), $MI^4_{Af,U0}$ (black bars) expressed in bits. (B) Information flow after learning for attraction to others in bits. (C) Weight difference for every agent: $\Delta W_{Af} = \omega_{1,Af} - 0.1$, $\Delta W_{Fo} = \omega_{1,Fo} - 0.1$ (D) Information flow before learning for attraction for food $MI^4_{Fo,U1}$ (grey bars), $MI^4_{Fo,U0}$ (black bars) in bits. (E) Information flow after learning for attraction for food in bits. Error bars are centered on the average for 100 simulations. The error width is equal to the maximum-minimum interval of the computed measures over 100 simulations.

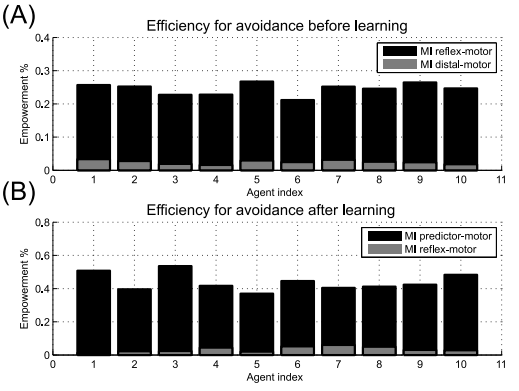


Fig. 4. (A) Efficiency for every agent of the reflex-output and predictive-output loop in terms of capacity before learning (stage 1): $MI^4_{Av,U0}/\zeta^4_{all}\%$ (dark bars), $MI^4_{Av,U1}/\zeta^4_{all}\%$ (grey bars). (B) Efficiency after learning (stage 3).

reflex-output loop predominates over the predictor-output loop for both the food attraction behaviour and the others attraction behaviour:

$$MI^4_{Af,U1} < MI^4_{Af,U0} \simeq 0.0025 \tag{28}$$

$$MI^4_{Fo,U1} < MI^4_{Fo,U0} \simeq 0.001. \tag{29}$$

After learning (stage 3). The configuration is reverted and the predictor-output loop dominates the reflex-output loop for both behaviours as in Fig.3(B),(E):

$$MI^4_{Af,U0} \ll MI^4_{Af,U1} \tag{30}$$

$$MI^4_{Fo,U0} \ll MI^4_{Fo,U1} \tag{31}$$

This result matches our expectations in terms of the increase of $I(D,R)$ and decrease of $I(E,R)$. If we compare the $MI^4_{Af,U1}$ in Fig.3(B) to $MI^4_{Fo,U1}$ in Fig.3(E) we can see that the agents with indices 1,2,3,4,5 (parasites) have a larger weight $\Delta W_{Af} \simeq 2.0$ (see Fig.3(C)) for the attraction to others and, therefore, a larger information flow $MI^4_{Af,U1} > MI^4_{Fo,U1}$, whereas agents with indices 6,7,8,9,10 (seekers) have a larger weight change $\Delta W_{Fo} \simeq 2.0$ for the food attraction and so a bigger $MI^4_{Fo,U1} > MI^4_{Af,U1}$.

Thus, the information measure is directly correlated with the weight change and can be used to quantify the learning performance of a single agent before and after learning. However, it can also be used to quantify the dominant behaviour and, consequently, the self-organising properties of social systems.

In Fig.4 we measure the efficiency of the reflex-output and predictive-output loop $MI^4_{Av,U1}, MI^4_{Av,U0}$ for the avoidance behaviour in relation to the capacity for the agents $\zeta^4_{all} = 2.0$. Fig.4(A) shows that before learning $MI^4_{Av,U0}$ is using 0.25% of the full channel capacity and Fig.4(B) shows that after learning $MI^4_{Av,U1}$ is using about 0.45% of the channel capacity. The MI of order

$n = 1, 2, 3$ does not provide enough discrimination for the previous analysis because the output history of the agent is too short to be correlated with the inputs. The capacity ζ^n_{all} takes its maximum of 2 bits when $n \geq 2$.

7 Discussion

In summary, we have introduced an extension to Ashby's requisite variety theory called the law of adaptive requisite variety, computed the information flow to measure the learning performance for agents with competitive behaviours and found the relation between the efficiency of the information flow MI and the weight change of the adaptive controller $\Delta\omega_1$. We also linked our information approach to the Luhmann theory that sub-systems are formed to reduce the perceived complexity of the environment. In our simulations, after the learning experience 5 agents have a dominant attraction behaviour for food disks (seekers) and 5 have a dominant attraction behaviour for others (parasites). The seekers mainly use the predictive information of the food disks while the parasites mainly use the predictive information of the others who posses food. Thus, we conclude that predictive learning in a social context leads to the formation of subsystems. This can be demonstrated with the help of our approach. While Polani [11,9] and Lungarella [16,10] used the empowerment measure as a general cost function to optimise the agent's behaviour or evolution, we use it as the upper bound of the MI to measure the efficiency of the sensory-motor loop use. Ay in his work [2] uses an adaptive controller which maximises the excess entropy (the mutual information between past and present) at the input side to achieve a working regime exploratory and sensitive to the environment. We can calculate the MI for this case by considering the reflex as the present input and the predictor as the past history. Our approach is not restricted to MISO controllers. Kulvicius et al. [12] measures the temporal input development, the output and path entropy of the adaptive agents to study the optimality of the antenna ratio for an avoidance task, thus completing the tools required to evaluate a single task controller. Current work is focusing on using a model checking approach to verify the properties of the system in terms of information flow.

8 Simulation Details

The world is a toroidal square of 300x300 units (Um), the agent has a diameter of 10 Um, the reflex antennas have a range of 40 Um, the predictor antennas have a range of 60 Um, every food disk has a diameter of 20 Um, the agent consumes food after 30 time steps. Every food disk starts with 100 food units and, if depleted, is reset after 5 time steps. To compute the entropy, the input space is discretized into 4 equally spaced bins and normalised in the range [-1,1] both for the predictor U_1 and the reflex U_0 signal, the output signal Z is discretized in 8 directions.

References

1. Ashby, W.: An Introduction to Cybernetics. Chapman & Hall, Boca Raton (1956)
2. Ay, N., Bertschinger, N., Der, R., Güttler, F., Olbrich, E.: Predictive information and explorative behavior of autonomous robots. The European Physical Journal B 63(3), 329–339 (2008)
3. Polani, D., Ay, N.: Information ows in causal networks. Adv. Compl. Syst. (2007)
4. Wörgötter, F., Porr, B.: Isotropic sequence order learning in a closed loop behavioural system. In: Roy. Soc. Phil. Trans. Mathematical, pp. 2225–2244
5. Wörgötter, F., Porr, B.: Inside embodiment what means embodiment to radical constructivists? Kybernetes, 105–117 (2005)
6. Booth, T.L.: Sequential Machines and Automata Theory, 1st edn. (1967)
7. Braitenberg, V.: Vehicles: Experiments in synthetic psychology. MIT Press, Cambridge (1984)
8. Zhang, J., Bi, D., Wang, G.L.: Novel learning feed-forward controller for accurate robot trajectory tracking. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3611, pp. 266–269. Springer, Heidelberg (2005)
9. Nehaniv, C., Klyubin, A.S., Polani, D.: Empowerment: A universal agent-centric measure of control. Proceedings of the IEEE Congress on Evolutionary Computation 1, 128–135 (2005)
10. Nehaniv, C.L., Klyubin, A.S., Polani, D.: Keep your options open: An information-based driving principle for the sensorimotor systems. In: PLoSOne, vol. 3 (2008)
11. Polani, D., Klyubin, A.S., Nehaniv, C.L.: Organization of the information flow in the perception-action loop of evolved agents, pp. 177–180 (June 2004)
12. Kolodziejski, C., Kulvicius, T.: On the analysis of differential hebbian learning in closed-loop behavioral systems. In: Frontiers in Computational Neuroscience. Conference Abstract: Bernstein Conference on Computational Neuroscience (2009)
13. Luhmann, N.: Social Systems (1996)
14. Bullwinkle, D., Lungarella, M., Pegors, T.: Methods for quantifying the information structure of sensory and motor data. Neuroinformatics 3, 243–262 (2005)
15. Di Prodi, P., Porr, B., Wörgötter, F.: Adaptive communication promotes subsystem formation in a multi agent system with limited resources. In: LAB-RS 2008: Proceedings of the 2008 ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems, pp. 89–96 (2008)
16. Sporns, O., Pfeifer, R., Lungarella, M., Kuniyoshi, Y.: On the information theoretic implications of embodiment - principles and methods. In: Proc. of the 50th Anniversary Summit of Artificial Intelligence, pp. 76–86 (2008)
17. Porr, B., Wörgötter, F.: Strongly improved stability and faster convergence of temporal sequence learning by utilising input correlations only. Neural Computation 18(6), 1380–1412 (2006)
18. Blahut, R.E.: Computation of channel capacity and rate distorsion functions. IEEE Trans. on Inform. Theory 18(4), 460–473 (1972)
19. Rouslton, M.S.: Estimating the errors on measured entropy and mutual information. Physica D, 285–294 (1999)
20. Arimoto, S.: An algorithm for computing the capacity of arbitrary memoryless channels. IEEE Transactions on Information Theory 18(1), 14–20 (1972)
21. Stuart, B.: Nicholas minorsky and the automatic steering of ships. IEEE Control Systems Magazine 4(4) (1984)
22. Sutton, A.G., Barto, R.S.: Reinforcement learning: An introduction. MIT Press, Cambridge (1998)
23. Schreiber, T.: Measuring information transfer. Phys. Rev. Lett. 85, 461–464 (2000)
24. Touchette, H., Lloyd, S.: Information-theoretic limits of control. Phys. Rev. Lett. 84(6), 1156–1159 (2000)

Co-development of Linguistic and Behavioural Skills: Compositional Semantics and Behaviour Generalisation

Elio Tuci*, Tomassino Ferrauto, Gianluca Massera, and Stefano Nolfi

ISTC-CNR, Via San Martino della Battaglia n. 44, 00185 Rome, Italy
{elio.tuci,tomassino.ferrauto,gianluca.massera,stefano.nolfi}@istc.cnr.it
http://laral.istc.cnr.it/

Abstract. Population of simulated agents controlled by dynamical neural networks are trained by artificial evolution to access linguistic instructions and to execute them by indicating, touching or moving specific target objects. During training the agent experiences only a subset of all object/action pairs. During post-evaluation, some of the successful agents proved to be able to access and execute also linguistic instructions not experienced during training. This is owe to the development of a semantic space, grounded on the sensory motor capability of the agent and organised in a systematised way in order to facilitate linguistic compositionality and behavioural generalisation.

Keywords: Grounding, CTRNNs, Artificial Evolution.

1 Introduction

During the last few years, several researchers have been building robotic and simulated systems in which communication and linguistic skills are grounded in perception and action [1,2,3,4]. One reason that explains the interest in these works is constituted by the fact that they represent a suitable methodology to investigate with precise operational models important aspects of cognition and action [5,6,7,8]. This work is motivated by an intention to contribute to deepen our understanding of the relation between action and language in order to verify the nature of their strict interdependence. Indeed, as we will see, the results of this type of research can help us to answer important questions such as: how agents linguistic abilities are dependent on, and grounded in, other behaviours and skills; how action-language interaction supports the bootstrapping of the agents cognitive system, e.g. through the transfer of properties of action knowledge to that of linguistic representations (and vice versa).

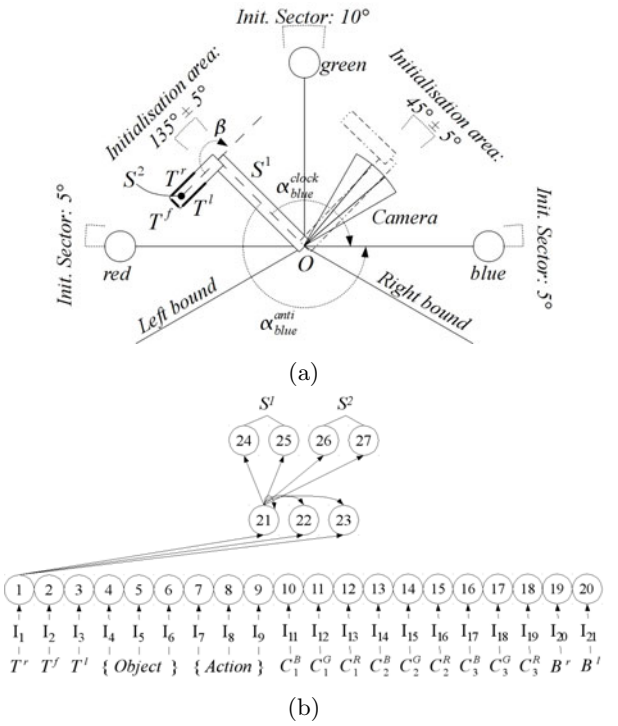
In this paper, we describe a model in which a simulated agent interacts with coloured objects located in its peripersonal space by exhibiting three behaviours (indicating, touching, and pushing) during a series of trials. In each trial, the

* This research work was supported by the *ITALK* project (EU, ICT, Cognitive Systems and Robotics Integrating Project, grant n° 214668).

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 523–532, 2010.

agent receives as input a linguistic instruction (constituted by two units, one that defines an object and another that defines an action) and is rewarded for the ability to exhibit the corresponding behaviour (i.e., executing the action on the target object). During training, an agent experiences only a subset of all possible object/action instructions. The goal of this work is to design neural mechanisms that allow the agent to access and execute both the experienced and the non experienced linguistic instructions, through the development of a compositional semantics that underpins linguistic and behavioural skills required by the task. This study has been strongly inspired by the work illustrated in [9] in which the authors trained a wheeled robot to interact with three coloured objects (located on the left, frontal, and right side of the agent) through three actions (indicating, hitting, and pushing). Also the idea of studying semantic combinatoriality through the co-development of linguistic and behavioural skills has been strongly inspired by the above seminal work in which the authors demonstrated how the linguistic and behavioural skills developed by the agents can be bounded together in order to allow the agent to react to a new linguistic instructions not experienced during training. Yet, we look at the problem with different methodological tools to provide further alternatives to those issues that we perceive as current limitations of the work described in [9]. In particular, in [9], the agent is controlled by two separated modules (one dedicated to perception and action, the other to linguistic comprehension) trained through a learning by demonstration process in which the sequence of sensory-motor states experienced while the experimenter drives the agent actuators during a demonstration session are used as teaching input for a supervised learning algorithm. Moreover, in [9] the sensory-motor module is trained to execute all the possible behaviours, even those associated to the linguistic instructions used to test the agent's generalisation capabilities. Contrary to [9], we propose to study the emergence of situated semantics in single non modularised artificial neural networks trained through a trial and error process (based on an evolutionary algorithm) in which the agents are rewarded on the basis of their ability to execute the linguistic instructions being free to determine how to execute such instructions. In our model, behavioural and linguistic competences co-evolve in a single neural structure in which the semantics is fully grounded on the sensory-motor capabilities of the agents and fully integrated with the neural mechanisms that underpin the agent's behavioural repertoire. Moreover, the agents are evolved to execute only the behaviours corresponding to the linguistics instructions experienced during training. Therefore, the capability of the agents to generalise concerns both the capability to access not experienced linguistic instructions as well as the capability to generate not experienced behaviours.

At the end of the training process successful agents display an ability to translate the linguistic instructions experienced during training into the corresponding situated behaviours. By analysing how successful agents react to specific combination of object/action instructions not experienced during training, we observed that some of the agents display an ability to spontaneously produce the appropriate behaviours, despite these behaviours have never been produced or rewarded



	MOVE Act_o^M					
	Object			Action		
	I_4	I_5	I_6	I_7	I_8	I_9
Blue	0	0	1	1	0	0
Green	0	1	0	1	0	0
Red	1	0	0	1	0	0

	TOUCH Act_o^T					
	Object			Action		
	I_4	I_5	I_6	I_7	I_8	I_9
Blue	0	0	1	0	1	0
Green	0	1	0	0	1	0
Red	1	0	0	0	1	0

	INDICATE Act_o^I					
	Object			Action		
	I_4	I_5	I_6	I_7	I_8	I_9
Blue	0	0	1	0	0	1
Green	0	1	0	0	0	1
Red	1	0	0	0	0	1

(c)

Fig. 1. (a) The agent structure and its world. The vision system of the agent is drawn only with respect to the arm initialised on the right initialisation area. (b) The structure of neural network. Continuous line arrows indicate the efferent connections for the first neuron of each layer. Underneath the input layer, it is shown the correspondences between sensors/linguistic instructions, the notation used in equation 1a to refer to them, and the sensory neurons. (c) The linguistic instructions. In grey the instructions not experienced during training. INDICATE is considered only in Exp. A.

before during training. Post-evaluation analyses on the behaviour of successful agents suggest that their capability to access unlearnt instructions and to generate the corresponding unlearnt behaviour partially results from the emergence of temporal (rather than topological as in [9]) structures of the semantic space. Finally, we observed that the development of systematised knowledge underpinned by a compositional semantic system is facilitated by evolutionary circumstances in which the agents are explicitly required to display elementary behavioural skills that can be recruited for the generation of more complex behaviours.

2 Methods

The task and the agent

Each agent lives in a two-dimensional world and is comprised of an arm with two segments referred to as S^1 (100 cm) and S^2 (50 cm), and two degrees of freedom

(DOF). Each DOF is comprised of a rotational joint which acts as the fulcrum and an actuator. One actuator causes S^1 to rotate clockwise or anticlockwise around point O, with the movement restricted within the right ($-30°$) and the left ($210°$) bound. The other actuator causes S^2 to rotate within the range [$-90°$, $90°$] with respect to S^1. Friction and momentum are not considered (see Fig. 1a). In the environment there are three rounded objects of different colours (i.e., a blue, a green, and a red object). The objects are placed at 150 cm from point O with their centre placed anywhere on the cord delimiting their corresponding Init. sector (see Fig. 1a). The objects do not move unless pushed by the arm. The agent is equipped with a linear camera with a receptive field of $30°$, divided in three sectors, each of which has three binary sensors (C_i^B for blue, C_i^G for green, and C_i^R for red, with $i \in [1, 2, 3]$ sectors). Each sensor returns 1 if the blue/green/red object falls with the corresponding sector. The camera and S^1 move together. The experimental set up is built in a way that at each time step there can be only one object in the camera view. If no coloured object is detected, the readings of the sensors are set to 0. The agent is also equipped with right and left bound binary sensors which activate (i.e., their reading is set to 1) whenever S^1 reaches the right or the left bound, respectively. Finally, three binary touch sensors (i.e., T^r, T^f, T^l) are placed on the right, front, and left side of S^2. Collisions between the agent and an object are handled by a simple model in which whenever S^2 pushes the object the relative contact points remain fixed.

In a first series of simulations (referred to as Exp. A), agents are trained to execute the following three actions: TOUCH (Act_o^T), MOVE (Act_o^M), and INDICATE (Act_o^I), where o is the object on which the action is executed, and can be either the *blue*, the *green* or the *red* object (see Fig. 1c). TOUCH requires an agent to remain in contact with the target object with the right side of S^2 (that is, by activating the touch sensor T^r) for an uninterrupted interval of 100 time steps. During this interval, S^1 does not have to rotate. MOVE requires an agent to rotate S^1 more than $35°$ while S^2 is touching the object with its right side. The rotation of S^1 while S^2 is touching the object determines the movement of the object. INDICATE requires an agent to rotate S^1 until the angular distance between S^1 and the object is less than $30°$. INDICATE is correctly executed only if S^1 remains at less than $30°$ from the target object for more than 100 time steps. During the execution of INDICATE, an agent must not collide with any object. During the execution of TOUCH and MOVE, an agent must not collide with the non target objects (i.e., the objects not mentioned in the current linguistic instruction). In a second series of simulations (referred to as Exp. B), agents are trained to execute only the action TOUCH (Act_o^T), and MOVE (Act_o^M).

The agent controller and the evolutionary algorithm

The agent controller is composed of a continuous time recurrent neural network (CTRNN) of 20 sensor neurons, 3 inter-neurons and 4 motor neurons [10]. At each time step sensor neurons are activated using an input vector I_i with $i \in [1, .., 20]$ corresponding to the sensors readings (see Fig. 1b).

The inter-neuron network is fully connected. Additionally, each inter-neuron receives one incoming synapse from each sensory neuron. Each motor neuron receives one incoming synapse from each inter-neuron. There are no direct connections between sensory and motor neurons. The states of the motor neurons are used to control the movement of S^1 and S^2 as explained later. The states of the neurons are updated using the following equations:

$$\frac{\Delta y}{\Delta T} = \begin{cases} \Big(- y_i + gI_i \Big) \dfrac{1}{\Delta T}; & \text{for } i \in \{1, .., 20\}; \quad (1a) \\ \Big(- y_i + \displaystyle\sum_{j=1}^{23} \omega_{ji} \sigma(y_j + \beta_j) \Big) \dfrac{1}{\tau_i}; & \text{for } i \in \{21, 22, 23\}; \quad (1b) \\ \Big(- y_i + \displaystyle\sum_{j=21}^{23} \omega_{ji} \sigma(y_j + \beta_j) \Big) \dfrac{1}{\Delta T}; & \text{for } i \in \{24, .., 27\}; \quad (1c) \end{cases}$$

with $\sigma(x) = (1+e^{-x})^{-1}$. In these equations, using terms derived from an analogy with real neurons, y_i represents the cell potential, τ_i the decay constant, g is a gain factor, I_i the intensity of the perturbation on sensory neuron i, ω_{ji} the strength of the synaptic connection from neuron j to neuron i, β_j the bias term, $\sigma(y_j + \beta_j)$ the firing rate (hereafter, f_i). All sensory neurons share the same bias (β^I), and the same holds for all motor neurons (β^O). τ_i and β_i with $i \in \{21, 22, 23\}$, β^I, β^O, all the network connection weights ω_{ij}, and g are genetically specified networks' parameters. At each time step the angular movement of S^1 is $2.9H(f_{24} - 0.5)sgn(0.5 - f_{25})$ degrees and of S^2 is $2.9H(f_{26} - 0.5)sgn(0.5 - f_{27})$ degrees, where H is the Heaviside step function and sgn is the sign function.

A generational genetic algorithm is employed to set the parameters of the networks [11]. The population contains 100 genotypes. Generations following the first one are produced by a combination of selection with elitism, recombination and mutation. For each new generation, the five highest scoring individuals from the previous generation are retained unchanged. The remainder of the new population is generated by fitness-proportional selection from the 70 best individuals of the old population. Each genotype is a vector comprising 90 real values. Each gene is chosen uniformly random from the range $[0, 1]$. Cell potentials are set to 0 when the network is initialised or reset, and circuits are integrated using the forward Euler method with an integration time step $\Delta T = 0.05$.

The fitness function

During evolution, each genotype is translated into an arm controller and evaluated more than once for different object-action pairs and different starting positions. In Exp. A (i.e., with INDICATE), agents are evaluated 14 times initialised in the left and 14 times in the right initialisation area, for a total of 28 trials. For each initialisation area, an agent experiences 2 times all the linguistic instructions with the exception of Act^M_{blue} and Act^T_{green}. These two instructions are never experienced during the training phase. In Exp. B (i.e., without INDICATE), agents are evaluated 8 times initialised in the left and 8 times in the right initialisation area, for a total of 16 trials. 4 out of 6 linguistic instructions

are experienced during the evolution process, while 2 are not (as before, the instructions which are not experienced are Act^{M}_{blue} and Act^{T}_{green}). In both Exp. A and Exp. B, at the beginning of each trial, the agent is randomly initialised in one of the two initialisation area, and the state of the neural controller is reset. A trial lasts 12 simulated seconds ($T = 250$ time steps). A trial is terminated earlier in case the arm collides with a non target object.

In each trial k, an agent is rewarded by an evaluation function which seeks to assess its ability to execute the desired action on the target object. The final fitness FF attributed to an agent is the sum of two fitness components F^1_k and F^2_k. F^1_k rewards the agent for reducing the angular distance between S^1 and the target object. F^2_k rewards the agent for performing the required action on the target object. F^1_k and F^2_k are computed as follows:

$$F^1_k = \max\left(0, \frac{d^i - d^f}{d^i} \cdot P^1_k, \mathbb{1}_{d^f<4.6^\circ}\right) \tag{2}$$

where d^i and d^f are respectively the initial (i.e., at $t = 0$) and final (i.e., at the end of the trail k) angular distances between S^1 and the target object and $\mathbb{1}_{d^f<4.6^\circ}$ is 1 if $d^f < 4.6^\circ$, 0 otherwise. P^1_k is the penalty factor, which is set to 0.6 if the agent collides with a non target object, to 1.0 otherwise. The angle between S^1 and the target object o can be measured *clockwise* (α^{clock}_o) or *anticlockwise* (α^{anti}_o). In equation 2, d^i and d^f are the minimum between the clockwise and anticlockwise distance, that is $d = \min\left(\alpha^{clock}_T, \alpha^{anti}_T\right)$.

$$F^2_k = \begin{cases} \dfrac{\textit{steps-on-target}}{\textit{max-steps-on-target}} \cdot P^2_k & \text{for TOUCH or INDICATE} \quad (3a) \\ \dfrac{\Delta\theta}{\textit{max-angular-offset}} \cdot P^2_k & \text{for MOVE} \quad (3b) \end{cases}$$

where *max-steps-on-target* $= 100$, $P^2_k = 0$ if $F^1_k < 1$ otherwise $P^2_k = 1$, and *max-angular-offset* $= 34.4^\circ$. For the action INDICATE, *steps-on-target* refers to the number of time steps during which $F^1_k = 1$, and S^2 does not touch the target object. For the action TOUCH, *steps-on-target* refers to the number of time steps during which $F^1_k = 1$, S^2 touches the target object by activating the touch sensor T^r, and S^1 does not change its angular position. $\Delta\theta$ is the angular displacement of the orientation of S^1 recorded while $F^1_k = 1$, and S^2 is touching the target object by activating the touch sensor T^r. A trial is terminated earlier if *steps-on-target* $=$ *max-steps-on-target* during the execution of INDICATE or TOUCH and when $\Delta\theta =$ *max-angular-offset* during the execution of MOVE.

3 Results

For both Exp. A and Exp. B, we run for 10000 generations ten evolutionary simulations, each using a different random initialisation. Recall that our objective is to generate agents that are capable of successfully performing all the possible

Table 1. Result of post-evaluation test performed on the best agents of each generation for each run and for Exp. A and Exp. B. The table shows the number of successful agents on linguistic instructions experienced during evolution, and the percentage of successful agents on linguistic instructions not experienced during evolution indicated by the corresponding row (see text for details).

run		n. 1	n. 2	n. 3	n. 4	n. 5	n. 6	n. 7	n. 8	n. 9	n. 10
Exp. A											
Num. Suc. Agents		8634	0	7182	0	5491	3466	8812	8312	4627	8632
(%)	Act^M_{blue}	30.87	0.00	17.96	0.00	0.00	57.73	29.43	27.96	12.19	3.56
	Act^T_{green}	17.88	0.00	0.56	0.00	2.77	1.13	16.00	21.19	3.41	1.00
	Act^M_{blue} and Act^T_{green}	9.07	0.00	0.61	0.00	0.00	1.59	6.97	15.56	0.35	0.00
Exp. B											
Num. Suc. Agents		6044	6011	8689	8893	0	8385	9060	7620	9151	8304
(%)	Act^M_{blue}	20.43	14.59	11.67	19.98	0.00	0.01	1.10	16.18	3.05	7.70
	Act^T_{green}	0.00	0.32	1.63	2.11	0.00	10.10	1.62	0.59	1.22	0.87
	Act^M_{blue} and Act^T_{green}	0.00	0.00	0.44	0.16	0.00	0.00	0.00	0.21	0.00	0.00

behaviours corresponding to the execution of all the possible linguistic instructions by undertaking a training focused only on a subset of them. We run two different series of simulations (i.e., Exp. A and Exp. B) to see whether the training on a more elementary action (i.e., INDICATE) bears upon the development of functionally compositional neural structures.

The best agents of each generation in both experimental conditions have been post-evaluated by running sets of 80 trials for each linguistic instruction. Agents of Exp. B are not tested on linguistic instructions that require action INDICATE. In half of the trials the agents are randomly initialised in the right and half of the trials in the left initialisation area (see Fig 1a). We considered successful at the post-evaluation tests the agents that managed to obtain a success rate higher that 80% in performing the behaviours corresponding to the execution of the linguistic instructions experienced during evolution. Successful agents have been further classified in i) *non compositional* agents, referring to those successful agents that proved to be less than 80% successful at performing the behaviour corresponding to the execution of both the not experienced instructions, Act^M_{blue} and Act^T_{green}; ii) *partially compositional* agents referring to those successful agents that proved to be more than 80% successful at performing the behaviour corresponding to the execution of only one of the two non experienced instructions, Act^M_{blue} or Act^T_{green}; iii) *fully compositional* agents referring to those successful agents that proved to be more than 80% successful at performing the behaviour corresponding to the execution of both the not experienced instructions, Act^M_{blue} and Act^T_{green}. Results of post-evaluation tests are shown in Table 1.

All the runs, with the exception of run n. 2 and n. 4 in Exp. A, and run n. 5 in Exp. B, generated plenty of successful agents. For what concerns

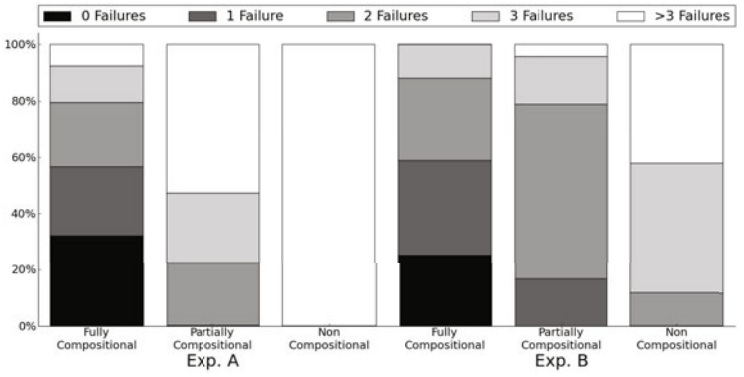


Fig. 2. Percentage of *fully compositional*, *partially compositional* and *non compositional* agents in the two experimental conditions grouped by the number of failure

compositionality, the results can be summarised in few relevant points. First, *fully compositional* agents are a very small percentage of the successful agents, in Exp A, and they are almost absent in Exp. B. Moreover, in those run that generated them, *fully compositional* agents keep on appearing and disappearing during evolution, while successful agents once generated they are almost never lost (data not shown). These data suggest that compositionality is not automatically associated with, and is not a prerequisite for developing the capability of successfully performing the evolutionary task. Second, in both Exp. A and Exp. B,*partially compositional* agents are slightly more frequent than fully compositional agents. Moreover, *partially compositional* agents capable of performing Act^{M}_{blue} are more frequent than *partially compositional* agents capable of performing Act^{T}_{green}. Third, although successful agents are slightly less likely to be generated in Exp. A than Exp. B, *fully compositional* or *partially compositional* agents are definitely more frequent in Exp. A than in Exp B. This suggests that the training on the more elementary action INDICATE seems to facilitate the development of behavioural and linguistic compositionality.

Having ascertained that some of the successful agents are also *partially* or *fully compositional*, we try to understand more about the mechanisms underpinning compositionality. Looking at the behaviour of all types of *compositional* agents, we noticed that they first move S^1 keeping S^2 bent in order to point to the target object (as required for the INDICATE instruction). After that, if TOUCH or MOVE is required, they rotate S^2 and eventually S^1 again depending on the current linguistic instruction. If INDICATE is required, they keep S^1 pointing to the object, and S^2 fully bent as at start. A very parsimonious hypothesis on how *compositional* agents generate these behavioural patterns is based on the capability to "parse" the linguistic instruction and to "pay attention" to its parts in a sequential order. According to our temporal sequencing hypothesis, compositionality may result from the fact that at the beginning of a trial, when the agents have to approach the target object, only the part of the instruction referring to the object bears on its behaviour. When an agent is ready to execute

the action on the target object, then only the part of the instruction referring to the action bears on the agent behaviour. In other words, compositionality may be underpinned by a systematised knowledge of the task obtained by paying attention to different parts of the linguistic instruction at different times of a trial. Linguistic instructions, including those not experienced during training, would be "decomposed" in already experienced elementary units which trigger known (i.e., already experienced) elementary behaviours in a specific temporal sequence (i.e., first the movement on the target object, then the execution of the desired action).

To test the temporal sequencing hypothesis, we run a further series of post-evaluation tests on successful agents of both Exp. A and Exp. B. In these tests, the linguistic command referring to the action is changed during the agents' life time as soon as the agents have completed the movement toward the target object (i.e., when $d^f < 0.08$, see Sec. 2). According to the temporal sequencing hypothesis, *compositional* agents should pay attention to the part of the linguistic instruction referring to the action only after having reached the target object. Therefore, they should correctly execute the second-given action, while ignoring the first-given one. The performance of *non compositional* agents should result severely disrupted by this type of unexpected manipulation of the linguistic instruction. The agents undergo sets of 80 trials for each possible transition from a first-given action to a second-given action different from the first one, and for each object. In half of the trials the agents are randomly initialised in the right and half of the trials in the left initialisation area. There are 18 possible transitions in Exp. A and 6 in Exp. B. The performance of an agent on each specific transition is considered a failure if the agent fails to execute the second-given action in more than 64 out of 80 trials. The results shown in Fig. 2 indicate that only some of the *fully compositional* agents are able to perform all transitions without any failure. These agents appear to have acquired a systematised knowledge of the task in accordance with what suggested by the temporal sequencing hypothesis. The higher the number of failure, the less structured the knowledge of the task with a higher number of linguistic instructions learnt by rote and represented as "atomic" operations in a semantics space progressively less compositional. Note that it is possible to be a *compositional* agent and having few linguistic instructions learnt by rote. This is probably the case of *fully compositional* agents that make several failure on specific transitions (remember that we do not enforce by any means compositionality). Note also that Exp. A and Exp. B generate similar results. This may imply that *fully compositional* agents exploit the same mechanisms to achieve compositionality in spite of the fact that in Exp. B the evolutionary conditions seem not to facilitate their evolution.

4 Conclusions

The results of this study shows that dynamical neural networks designed by artificial evolution can provide the required mechanism to develop a compositional semantic neural structures which allow autonomous agents to access linguistic

instructions not experienced during training and to execute the corresponding behaviours also non experienced during training. Although we haven't carried out yet any analysis on the neural mechanisms, we run some behavioural tests which showed that evolved compositional semantic systems seem to be underpinned by temporal structures. That is, *fully compositional* agents possess the required mechanisms to "parse" different part of the instruction and to execute different sub-behaviours at different time of their life span. Evolutionary conditions in which the agents are explicitly required to execute more elementary behaviour than those on which their compositional skills are evaluated seem to facilitate the emergence of *fully compositional* agents. Leaving the agents free to determine how to achieve the goals associated to each linguistic instruction allowed the agents to organise their behavioural skills in ways that facilitate the development of compositionality thus enabling the possibility to display a generalisation ability at the level of behaviours (i.e., the ability to spontaneously produce new behaviours that have not been displayed or rewarded before). In future research we plan to investigate the characteristics that favour the emergence of compositional solutions (that ensure behavioural generalisation) and/or that reduce the chance to converge on non-compositional solutions and the possibility to scale the model with respect to the number and the complexity of the linguistic/behavioural repertoire.

References

1. Cangelosi, A., Parisi, D. (eds.): Simulating the evolution of language. Springer, New York (2002)
2. Roy, D.: Grounding words in perception and action: computational insights. Trends in Cognitive Sciences 9(8), 389–396 (2005)
3. Steels, L.: Experiments on the emergence of human communication. Trends in Cognitive Sciences 10(8), 347–349 (2008)
4. Nolfi, S., Mirolli, M. (eds.): Evolution of Communication and Language in Embodied Agents. Springer, Berlin (2010)
5. Glenberg, A., Kaschak, M.: Grounding language in action. Psychonomic Bulletin & Review 9(2), 558–565 (2000)
6. Rizzolatti, G., Arbib, M.: Language within our grasp. Trends in Neuroscience 21, 188–194 (1998)
7. Pulvermuller, F.: The Neuroscience of Language. In: On Brain Circuits of Words and Serial Order. Cambridge University Press, Cambridge (2003)
8. Cappa, S., Perani, D.: The neural correlates of noun and verb processing. Journal of Neurolinguistics 16(2-3), 183–189 (2003)
9. Sugita, Y., Tani, J.: Learning semantic combinatoriality from the intercation between linguistic and behvioral processes. Adaptive Behavior 13(1), 33–52 (2005)
10. Beer, R.D., Gallagher, J.C.: Evolving dynamic neural networks for adaptive behavior. Adaptive Behavior 1(1), 91–122 (1992)
11. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)

Indirectly Encoding Neural Plasticity as a Pattern of Local Rules

Sebastian Risi and Kenneth O. Stanley

School of Electrical Engineering and Computer Science
University of Central Florida, Orlando, FL 32816
{risi,kstanley}@eecs.ucf.edu

Abstract. Biological brains can adapt and learn from past experience. In neuroevolution, i.e. evolving artificial neural networks (ANNs), one way that agents controlled by ANNs can evolve the ability to adapt is by encoding local learning rules. However, a significant problem with most such approaches is that local learning rules for every connection in the network must be discovered separately. This paper aims to show that learning rules can be effectively indirectly encoded by extending the Hypercube-based NeuroEvolution of Augmenting Topologies (HyperNEAT) method. *Adaptive HyperNEAT* is introduced to allow not only patterns of weights across the connectivity of an ANN to be generated by a function of its geometry, but also patterns of arbitrary *learning rules*. Several such adaptive models with different levels of generality are explored and compared. The long-term promise of the new approach is to evolve large-scale adaptive ANNs, which is a major goal for neuroevolution.

Keywords: Adaptation, Learning, HyperNEAT, Neuroevolution.

1 Introduction

Research in neuroevolution, i.e. evolving artificial neural networks (ANNs) through evolutionary algorithms, often focuses on static ANNs (i.e. weights do not change during the network's lifetime). However, in many control and decision-making problems, the environment may change too quickly to allow phylogenetic adaptation; thus the controller needs to adapt *online* to maintain performance. For example, a robot may need to remember a location that changes over time.

One way that agents controlled by ANNs can evolve the ability to adapt over their lifetime is by encoding *local learning rules* in the genome that determine how their synaptic connection strengths should change in response to changing activation levels in the neurons they connect [1, 2, 3]. This approach resembles the way organisms in nature, which possess plastic nervous systems, cope with changing and unpredictable environments.

Although demonstrations of this approach have suggested the promise of evolving adaptive ANNs, a significant problem is that local learning rules for every connection in the network must be discovered separately. That is, although

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 533–543, 2010.

interest has grown in recent years in *indirectly encoding* the weights of ANNs so that they can be discovered as patterns [4,5,6,7,8,9], the power of indirect encoding is rarely applied to encoding *learning rules*. Yet the distribution of rules across a network likely conforms to discoverable regularities just as weights.

Additionally, as noted by Yao [10], the right learning rule depends on the ANN architecture, which makes it difficult to design an optimal such rule a priori. Yao further points out that designing learning rules by hand, which is common in this area [1,10], requires making assumptions that might not hold in practice.

This paper aims to show that learning rules can be effectively indirectly encoded by extending the Hypercube-based NeuroEvolution of Augmenting Topologies (HyperNEAT) method [11,6,12], which currently indirectly encodes large geometric patterns of fixed weights for high-dimensional problems [13,11,6,12]. The new method introduced here, called *adaptive HyperNEAT*, allows not only patterns of weights across the connectivity of an ANN to be generated by a function of its geometry, but also *patterns of learning rules*. The idea that learning rules can be distributed in a geometric pattern is new to neuroevolution but reflects the intuition that synaptic plasticity in biological brains is not encoded in DNA separately for every synapse in the brain. Thus the main idea in this paper is a step towards more biologically plausible adaptive systems.

An important contribution of this work is to show that there is a tradeoff between the *generality* of an indirect encoding of plasticity and its computational cost. Yet, as experiments in a variant of the T-Maze learning domain [14,3] will show, in special cases, e.g. when the reward signature is nonlinear and the ANN topology is restricted, a most general encoding may be necessary. Thus, rather than offering a single approach to all problems, this paper reveals the existence of a continuum of adaptive encodings that trade off generality with computational expense. From this perspective the practitioner can make the most informed choice on the ingredients that may be necessary for a particular domain.

Building on the ability of HyperNEAT to evolve large-scale connectivity patterns, the long-term promise of the new approach is to evolve *large-scale adaptive ANNs*, which is a major goal for neuroevolution.

2 Background

The HyperNEAT method that enables learning from geometry in this paper is an extension of the original NeuroEvolution of Augmenting Topologies (NEAT) algorithm that evolves ANNs through a *direct* encoding [15,16]. NEAT starts with a population of small, simple neural networks and then *complexifies* them over generations by adding new nodes and connections through mutation. By evolving networks in this way, the topology of the network does not need to be known a priori. The important feature of NEAT for the purpose of this paper is that it evolves *both* the topology and weights of a network.

However, in direct encodings like NEAT, each part of the representation maps to a single piece of structure in the solution [17]. The significant disadvantage of this approach is that even when different parts of the solution are similar,

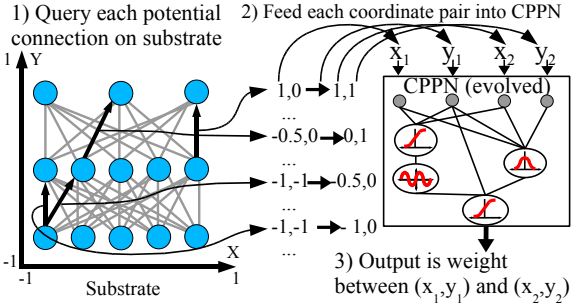


Fig. 1. How an ANN is Encoded by a CPPN. A collection of ANN nodes, called the *substrate*, is assigned coordinates that range from −1 to 1 in all dimensions. (1) Every potential connection in the substrate is queried to determine its presence and weight; the dark directed lines in the substrate depicted in the figure represent a sample of connections that are queried. (2) Internally, the CPPN (which is evolved) is a graph that determines which activation functions are connected. As in an ANN, the connections are weighted such that the output of a function is multiplied by the weight of its outgoing connection. For each query, the CPPN takes as input the positions of the two endpoints and (3) outputs the weight of the connection between them. Thus, CPPNs can produce regular patterns of connections in space.

they must be encoded and therefore discovered separately. Thus HyperNEAT employs an *indirect* encoding instead, which means that the description of the solution is compressed such that information can be reused, allowing the final solution to contain more components than the description itself [4,5,6,7,8,9].

In HyperNEAT, NEAT is altered to evolve an indirect encoding called *compositional pattern producing networks* (CPPNs [8]) *instead* of ANNs [13,11,6,12]. The main idea in HyperNEAT is that the CPPN, which is itself a network composed of a variety of activation functions, acts as a pattern generator that outputs a *pattern of connection weights* situated within the geometry of the ANN. The activation functions within the CPPN, such as sine and Gaussian, allow it to express regularities across the geometry of the ANN [11,6,12].

Formally, CPPNs are *functions* of geometry (i.e. locations in space) that output connectivity patterns whose nodes are situated in n dimensions, where n is the number of dimensions in a Cartesian space. Consider a CPPN that takes four inputs labeled x_1, y_1, x_2, and y_2; this point in four-dimensional space *also* denotes the connection between the two-dimensional points (x_1, y_1) and (x_2, y_2), and the output of the CPPN for that input thereby represents the weight of that connection (Fig. 1). By querying every possible connection among a pre-chosen set of points in this manner, a CPPN can produce an ANN, wherein each queried point is a neuron position. Because the connections are produced by a function of their endpoints, the final structure is produced with *knowledge* of its geometry. In effect, the CPPN is painting a pattern on the inside of a four-dimensional hypercube that is interpreted as the isomorphic connectivity pattern, which explains the origin of the name *hypercube-based NEAT* (HyperNEAT). Connectivity patterns produced by a CPPN in this way are called *substrates* so that they can be

verbally distinguished from the CPPN itself, which has its own internal topology. As a rule of thumb, nodes are placed on the substrate to reflect the geometry of the task [13,11,12]. That way, the connectivity of the substrate is a function of the task structure and while the task may be complex, the domain geometry is often intuitive.

For example, the sensors of an autonomous robot can be placed from left to right on the substrate in the same order that they exist on the robot. Outputs for moving left or right can also be placed in the same order, allowing HyperNEAT to understand from the outset the correlation of sensors to effectors. In this way, knowledge about the problem geometry can be injected into the search and HyperNEAT can exploit the regularities (e.g. adjacency, or symmetry) of a problem that are invisible to traditional encodings.

For a complete overview of HyperNEAT, see Gauci and Stanley [6] and Stanley et al. [12]. The next section extends this approach to evolve adaptive ANNs.

3 APPROACH: Adaptive HyperNEAT

The main idea in adaptive HyperNEAT is that CPPNs can not only encode connectivity patterns but also *patterns of plasticity rules*. As in the brain, different *regions* of the ANN should be more or less plastic and employ different learning rules, which HyperNEAT allows because it sees the ANN geometry. In general, a learning rule changes the weight of a connection based on presynaptic activity o_i, postsynaptic activity o_j, and the current connection weight w_{ij}:

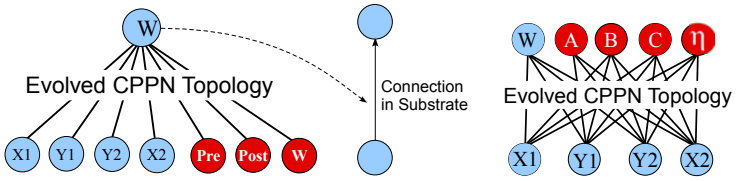
$$\Delta w_{ij} = \Phi(o_i, o_j, w_{ij}) \ . \tag{1}$$

In this paper three different adaptive HyperNEAT models are compared that are able to encode different levels of learning rule generality. The goal of this comparison is to elucidate the advantages and disadvantages of different levels of generality to modeling dynamic learning processes. All three models allow learning rules to be distributed as patterns across the connectivity of an ANN.

The most general **iterated model** (Fig. 2a) augments the four-dimensional CPPN that normally encodes connectivity patterns with three additional inputs: presynaptic activity o_i, postsynaptic activity o_j, and the current connection weight w_{ij}. That way, the synaptic plasticity of a connection between two two-dimensional points (x_1, y_1) and (x_2, y_2) can be described by

$$\Delta w_{ij} = \text{CPPN}(x_1, y_1, x_2, y_2, o_i, o_j, w_{ij}) \ . \tag{2}$$

The update of the synaptic weights can thereby be iteratively performed by the same CPPN that normally encodes network connectivity, which allows evolving increasingly complex learning rules. In effect, the CPPN encodes an entire dynamical system, including how changes depend on both location *and* activity. The CPPN is *requeried* on every tick of the clock to update the ANN weights. The initial weight configuration is determined by querying the CPPN as in the original HyperNEAT approach (Sec. 2) with the presynaptic activity, postsynaptic activity, and weight inputs all set to zero.



(a) Iterated Model (b) ABC Model

Fig. 2. Adaptive HyperNEAT. CPPNs for the iterated (a) and ABC models (b) are shown. The CPPN in (a) is continually requeried during the lifetime of the agent to determine the weight change given the location of the connection, activation of the presynaptic and postsynaptic neuron, and the current weight as input. In contrast, the CPPN in (b) is only activated once to determine the three parameters A–C and the learning rate η, which control synaptic plasticity during the lifetime of the agent, in addition to the initial weight w.

The less general **Hebbian ABC model** augments the CPPN instead with four additional *outputs* (Fig. 2b): learning rate η, correlation term A, presynaptic term B, and postsynaptic term C. When the CPPN is initially queried, these parameters are permanently stored, which allows the synaptic weight to be modified during the lifetime of the agent by the following plasticity rule:

$$\Delta w_{ij} = \eta \cdot [A o_i o_j + B o_i + C o_j] \ . \tag{3}$$

Traditional approaches to evolving adaptive ANNs with direct encodings also evolve the coefficients of Equation (3) but because of the limitations of direct encodings often only employ *one* such evolved rule throughout all ANN connections [18,3]. The difference here is that A, B, C, and η are indirectly encoded by HyperNEAT in a geometric pattern across the connectivity of the whole network. Therefore each connection could potentially employ a different rule if necessary. However, unlike the more general iterated model, this CPPN only produces variants of the ABC Hebbian rule. Thus the space of possible rules is more restricted.

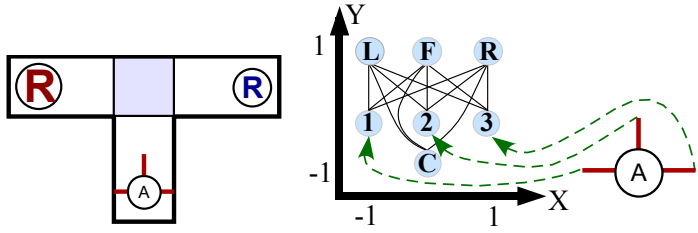
Finally, the simplest model is **plain Hebbian**. The CPPN has only one additional output that encodes the learning rate η:

$$\Delta w_{ij} = \eta \cdot o_i o_j \ . \tag{4}$$

This variant tests for the minimal sufficient dynamics to solve the T-Maze domain given in this paper, which is explained in the next section.

4 T-Maze Domain

T-Mazes are often studied in the context of operant conditioning of animals; they are also studied to assess the ability of plastic ANNs [14,3]. The discrete T-Maze in this paper (Fig. 3a) consists of two arms that either contain a high or low reward. The agent begins at the bottom of the maze and its goal is to



(a) T-Maze (b) Substrate

Fig. 3. T-Maze and Substrate Configuration. (a) In this depiction, high reward is located on the left and low reward is on the right side, but these positions can change over a set of trials. The challenge for the agent is to remember the location of the high reward from one trial to the next. (b) The autonomous agent A is equipped with three distance sensors and a reward color sensor that is set to zero during navigation.

navigate to the reward position. This procedure is repeated many times during the agent's lifetime. One such attempted trip to a reward location is called a *trial*. A *deployment* consists of a set of trials. When the position of the reward sometimes changes, the agent should alter its strategy accordingly to explore the other arm of the maze in the next trial and *remember* the new position in the future (requiring adaptation). The goal of the agent is to maximize the amount of reward collected over all deployments.

5 Experiments

To generate a controller for the T-Maze domain, the evolved CPPNs query the substrate shown in Fig. 3b. The locations of inputs and outputs are designed to geometrically correlate (e.g. seeing something on the left correlates to turning left). Thus the CPPN can exploit the geometry of the agent. The agent is equipped with three rangefinder sensors that detect walls to the left, front, and right of the robot. The *Color* input (explained shortly) is set to the color of the collected reward at the maze end, which determines the amount of reward given to the agent. The three output neurons are *Left*, *Forward*, and *Right*. At each simulated time step, the agent continues to move straight at a constant pace if the *Forward* output has the highest activation level. Otherwise the agent turns 90 degrees in the direction of the highest activated neuron (Left or Right).

An agent crashes if it does not (1) maintain a forward direction in corridors or (2) turn either right or left when it encounters the junction. If the agent crashes then the current trial is terminated.

In this paper, *two* T-Maze scenarios are studied to elucidate the advantages and disadvantages of encoding different levels of plasticity rule generality. **Scenario 1** resembles the traditional T-Maze domain described in the previous section. Each agent is evaluated on four deployments with 100 trials each. The starting position of the high reward alternates between deployments and switches

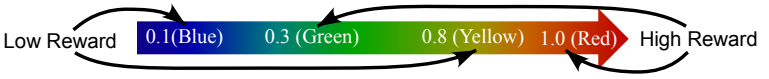


Fig. 4. Nonlinear Reward Color Encoding. The agent receives a high reward for green or red and a low reward for blue or yellow. The ANN color encoding together with the given ANN topology requires the agent to include a nonlinear learning rule.

positions after 50 trials on average. Color input values of 1.0 and 0.1 encode the high (red) and low (blue) reward, respectively.

In **scenario 2**, the agent is exposed to a total of four different colored rewards. The first deployment resembles scenario 1 with reward signatures of 0.1 and 1.0. However, in the *second* deployment, color input values of 0.3 and 0.8 are introduced to encode new high yellow and low green rewards, respectively (Fig. 4). Adding these intermediate reward colors yields a reward signature that is *not linearly separable*. Because the ANN controlling the agent does not have any hidden neurons, the learning rule must *itself* be nonlinear. Scenario 2 therefore makes a good domain for this study because it requires evolving a specific learning rule that depends on the ANN topology.

The fitness function, which is the same for all compared approaches and identical to Soltoggio et al. [3], is calculated as follows: Collecting a high reward has a value of 1.0 and a low reward is worth 0.2. A penalty of 0.4 is subtracted if the agent does not maintain forward motion in corridors or does not turn left or right at a junction. The total fitness of an individual is determined by summing the fitness values for each of the 100 trials over all deployments.

Note that although Risi et al. [19] showed that novelty search [20], which abandons objective-based fitness and instead simply searches only for novel behavior, significantly outperforms fitness-based search in the traditional T-Maze domain, a standard fitness function is employed in this paper to keep the experiment focused on the issue of adaptation.

5.1 Experimental Parameters

All experiments were run with a modified version of the public domain SharpNEAT package [21] called HyperSharpNEAT. Runs consisted of 500 generations with a population size of 500 and 10% elitism. Sexual offspring (50%) did not undergo mutation. Asexual offspring (50%) had 0.94 probability of link weight mutation, 0.03 chance of link addition, and 0.02 chance of node addition. The available CPPN activation functions were sigmoid, Gaussian, absolute value, and sine, all with equal probability of being added. A connection is not ever expressed if the magnitude of its initial weight is below a minimal threshold of 0.4. Parameter settings are based on standard SharpNEAT defaults and prior reported settings for NEAT [15,16]. For all adaptive HyperNEAT models synaptic strength is bound within the range $[-1.0, 1.0]$.

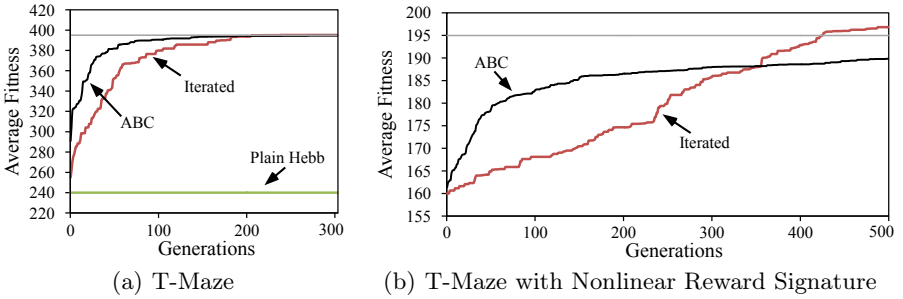
6 Results

The standard T-Maze (scenario 1) is solved when the agent reaches a fitness of 395. A minimum amount of exploration (i.e. collecting the low reward) is required at the beginning of each deployment and when the reward positions switch. The T-Maze with nonlinear reward signature (scenario 2), consisting of two deployments with different reward signatures, is solved with a fitness of 195. All reported results are averaged over 20 runs.

Figure 5a shows the average training performance over generations for the standard T-Maze (scenario 1). It took the ABC model 141 generations (σ=141) on average to find a solution. The iterated model took 89 generations (σ=61) on average. While the fitness for the iterated model initially increases more slowly than for ABC, it finds a solution slightly (though not significantly) faster on average. The plain Hebbian model cannot solve the task. Although both the more general iterated model and the ABC model can solve the task, the iterated model is computationally more expensive because the CPPN must be continually requeried for every ANN connection.

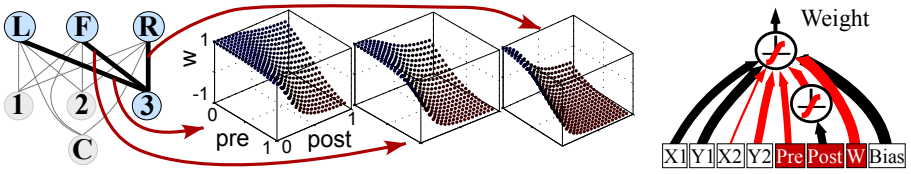
The average training performance over generations for scenario 2 is shown in Fig. 5b. The plain Hebbian rule is not tested in this variant because it is not able to solve the standard T-Maze. Whereas the iterated model solves the domain in 19 out of 20 runs, in 367 generations (σ=101) on average, ABC is not able to solve the task with the given ANN topology, which suggest the need for a nonlinear learning rule in this scenario (or potentially an ANN with hidden nodes). The more general iterated model is able to evolve such a rule.

Figure 6a shows CPPN-encoded learning rules of an ANN solution discovered by the iterated model. The function embodied by the CPPN (Fig. 6b) encodes a geometric pattern of nonlinear learning rules. Interestingly, the evolved rules



(a) T-Maze (b) T-Maze with Nonlinear Reward Signature

Fig. 5. Training Performance. The change in performance over evaluations for both scenarios is shown in this figure. All results are averaged over 20 runs. The horizontal line (top) indicates at what fitness the domain is solved. The iterated and ABC model are both able to solve the standard T-Maze domain (a) in about the same number of generations whereas the plain Hebbian approach does not show the necessary dynamics. The T-Maze domain with a nonlinear reward signature (b) requires a nonlinear learning rule, which only the iterated model discovers.



(a) Evolved Learning Rules (b) CPPN

Fig. 6. Discovered Learning Rules of an ANN Solution Created by the Iterated Model and the Underlying CPPN. The nonlinear learning rules shown in (a) are encoded by the evolved CPPN shown in (b). A geometric pattern of learning rules can been seen that varies with the target node's x location. The amount of synaptic change is a function of the pre- and postsynaptic activity and the corresponding positions of the pre- and postsynaptic neurons in the substrate (weight input w on the CPPN is set to zero in this depiction).

resemble postsynaptic-based learning rules that have been shown essential in the T-Maze domain [18].

7 Discussion and Future Work

The indirect HyperNEAT encoding is able to generate ANNs with millions of connections based on underlying geometric motifs [12]. This paper introduced an extension called adaptive HyperNEAT that generates not only patterns of weights across the connectivity of an ANN, but also patterns of learning rules (Fig. 6). The long-term promise of the new approach is therefore to evolve large-scale adaptive ANNs, which is a major goal for neuroevolution that may bring it closer to evolving brain-like structures.

While the ABC model together with an adequate ANN topology should be sufficient for most domains, the nonlinear variant of the T-Maze learning domain reveals that sometimes a more general encoding may be necessary. Although the ANN topology could potentially have been extended to allow a less general model to solve the nonlinear T-Maze domain, this experiment confirms the risk of making a priori assumptions about the type of necessary learning rules [10].

However, the generality of the indirect encoding of plasticity trades off with its computational cost. The most general iterated model is computationally expensive because the CPPN must be continually requeried for every ANN connection. The computational complexity for every time step is $O(n)+nO(m)$, where $O(n)$ and $O(m)$ are the costs of simulating an ANN with n connections and an underlying CPPN with m connections, respectively. Thus the most general model in its current form might be too computationally expensive for practical purposes that require large CPPNs and ANNs. However, it gives us a reference point from which to derive more specialized models such as the ABC model.

In the current iterated model the synaptic weights are updated at every time step. Characterizing how often a weight update is necessary is an important future research direction that may allow cutting down the computational cost of

even the most general model. Additionally, synaptic plasticity could be controlled by neuromodulation [19, 3, 2], which means that some neurons can enhance or dampen the neural plasticity of their target nodes. Such modulation could allow precise timing of CPPN weight queries. Finally, another potentially promising approach is combining the iterated and ABC models.

8 Conclusion

A new method called adaptive HyperNEAT was presented, which allows not only patterns of weights across the connectivity of an ANN to be indirectly encoded as a function of its geometry, but also patterns of arbitrary learning rules. Importantly, this paper shows that there is a tradeoff between the generality of an indirect encoding of plasticity and its computational cost. Yet, as a variation of the T-Maze domain demonstrates, the most general encoding may be necessary in some cases. The main conclusion is that the indirect HyperNEAT encoding may enable evolving large-scale adaptive ANNs.

Acknowledgments

This research was supported by DARPA under grant HR0011-09-1-0045 (Computer Science Study Group Phase 2).

References

1. Floreano, D., Urzelai, J.: Evolutionary robots with online self-organization and behavioral fitness. Neural Networks 13, 431–443 (2000)
2. Niv, Y., Joel, D., Meilijson, I., Ruppin, E.: Evolution of reinforcement learning in uncertain environments: A simple explanation for complex foraging behaviors. Adaptive Behavior 10(1), 5–24 (2002)
3. Soltoggio, A., Bullinaria, J.A., Mattiussi, C., Dürr, P., Floreano, D.: Evolutionary Advantages of Neuromodulated Plasticity in Dynamic, Reward-based Scenarios. In: Artificial Life XI, pp. 569–576. MIT Press, Cambridge (2008)
4. Bentley, P.J., Kumar, S.: Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999), pp. 35–43 (1999)
5. Bongard, J.C.: Evolving modular genetic regulatory networks. In: Proceedings of the 2002 Congress on Evolutionary Computation (2002)
6. Gauci, J., Stanley, K.O.: Autonomous evolution of topographic regularities in artificial neural networks. Neural Computation (to appear 2010)
7. Hornby, G.S., Pollack, J.B.: Creating high-level components with a generative representation for body-brain evolution. Artificial Life 8(3) (2002)
8. Stanley, K.O.: Compositional pattern producing networks: A novel abstraction of development. Genetic Programming and Evolvable Machines Special Issue on Developmental Systems 8(2), 131–162 (2007)
9. Stanley, K.O., Miikkulainen, R.: A taxonomy for artificial embryogeny. Artificial Life 9(2), 93–130 (2003)

10. Yao, X.: Evolving artificial neural networks. Proceedings of the IEEE 87(9), 1423–1447 (1999)
11. Gauci, J., Stanley, K.O.: A case study on the critical role of geometric regularity in machine learning. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008). AAAI Press, Menlo Park (2008)
12. Stanley, K.O., D'Ambrosio, D.B., Gauci, J.: A hypercube-based indirect encoding for evolving large-scale neural networks. Artificial Life 15(2), 185–212 (2009)
13. Clune, J., Beckmann, B.E., Ofria, C., Pennock, R.T.: Evolving coordinated quadruped gaits with the hyperneat generative encoding. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2009) Special Section on Evolutionary Robotics. IEEE Press, Piscataway (2009)
14. Blynel, J., Floreano, D.: Exploring the T-Maze: Evolving Learning-Like Robot Behaviors using CTRNNs. In: Raidl, G.R., Cagnoni, S., Cardalda, J.J.R., Corne, D.W., Gottlieb, J., Guillot, A., Hart, E., Johnson, C.G., Marchiori, E., Meyer, J.-A., Middendorf, M. (eds.) EvoIASP 2003, EvoWorkshops 2003, EvoSTIM 2003, EvoROB/EvoRobot 2003, EvoCOP 2003, EvoBIO 2003, and EvoMUSART 2003. LNCS, vol. 2611, pp. 593–604. Springer, Heidelberg (2003)
15. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary Computation 10, 99–127 (2002)
16. Stanley, K.O., Miikkulainen, R.: Competitive coevolution through evolutionary complexification 21, 63–100 (2004)
17. Floreano, D., Dürr, P., Mattiussi, C.: Neuroevolution: from architectures to learning. Evolutionary Intelligence 1(1), 47–62 (2008)
18. Soltoggio, A.: Neural Plasticity and Minimal Topologies for Reward-Based Learning. In: Proceedings of the 2008 8th International Conference on Hybrid Intelligent Systems, pp. 637–642. IEEE Computer Society, Los Alamitos (2008)
19. Risi, S., Vanderbleek, S.D., Hughes, C.E., Stanley, K.O.: How novelty search escapes the deceptive trap of learning to learn. In: GECCO 2009: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 153–160. ACM, New York (2009)
20. Lehman, J., Stanley, K.O.: Exploiting open-endedness to solve problems through the search for novelty. In: Bullock, S., Noble, J., Watson, R., Bedau, M. (eds.) Proceedings of the Eleventh International Conference on Artificial Life (Alife XI). MIT Press, Cambridge (2008)
21. Green, C.: SharpNEAT homepage (2003–2006), http://sharpneat.sourceforge.net/

Fractal Gene Regulatory Networks for Robust Locomotion Control of Modular Robots

Payam Zahadat[1,2], David Johan Christensen[1], Ulrik Pagh Schultz[1], Serajeddin Katebi[2], and Kasper Stoy[1]

[1] The Maersk Mc-Kinney Moller Institute,
University of Southern Denmark
{paza,david,ups,kaspers}@mmmi.sdu.dk
[2] Department of Computer Science and Engineering,
School of Engineering, Shiraz University, Shiraz, Iran
{zahadat,katebi}@shirazu.ac.ir

Abstract. Designing controllers for modular robots is difficult due to the distributed and dynamic nature of the robots. In this paper fractal gene regulatory networks are evolved to control modular robots in a distributed way. Experiments with different morphologies of modular robot are performed and the results show good performance compared to previous results achieved using learning methods. Furthermore, some experiments are performed to investigate evolvability of the achieved solutions in the case of module failure and it is shown that the system is capable of come up with new effective solutions.

Keywords: Fractal Gene Regulatory Networks, Modular Robots, Robot Control, Evolutionary Computation.

1 Introduction

The purpose of this paper is to investigate the capability of Fractal Gene Regulatory Networks (FGRNs) to control modular robots. FGRN [1] is a special type of computational Gene Regulatory Networks (GRNs) which utilizes fractal proteins to interact with a genotype. Modular robots are robots built from a number of mechanically coupled modules which can connect in different ways and each module is controlled by its own local controller. They have the potential to be versatile and robust, but due to their distributed, dynamic nature they are difficult to control.

Complex successful living phenotypes can be found everywhere in nature. Many of them consist of several cells each performing its own function related to position and role in the phenotype. Nature employs a complicated process of indirect mapping to develop a complete multi-cellular phenotype from a genotypic code. Instead of direct phenotype-genotype mapping normally used in conventional Evolutionary Computation (EC), the lifelong process of natural development is controlled by an ongoing interaction between genotype and intermediate substrates called proteins which are encoded by the genotype. This interaction is considered a network of genes which is called Gene Regulatory Network.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 544–554, 2010.

AN FGRN cell contains a genotype –called genome- that encodes fractal proteins, and a compound substrate -called cytoplasm- that maintains the proteins inside the cell. Developmental process of a cell is controlled by interaction between cytoplasm and genome. The process can be affected by information provided by the environment which is also represented by fractal proteins. During the lifetime of the FGRN cell, complex output patterns can be produced and used for different purposes such as controlling robots [2, 3, 4].

FGRN systems can be implemented distributedly. In a distributed system, all cells use the same genotype, but they run in parallel to each other. By providing proper environmental information for each cell, different cells in a system might follow different developmental trends and make appropriate output patterns. This distributed nature of the system potentially makes it suitable for controlling modular robots since each FGRN cell can be used to control one module of the robot. But the question is if this works in practice and what properties the resulting system has.

Modular robots are resource-constrained. They usually have little processing power and low inter-module communication abilities. In addition, based on the dynamic nature of a modular robot, failures might happen in modules, they can break, or the user may take apart the robot or detach some modules for different reasons while the robot is still supposed to work. In designing controllers for modular robots, it is desirable to have an acceptable level of robustness encountering these properties along with the characteristics of scalability, usability in different morphologies, and biological plausibility.

Distributed control of ATRON modular robots [5] which are supposed to perform a locomotion task is investigated here. Three different morphological configurations of the robot are used as experimental case studies. Previous works [6] have shown good performances for learning methods in these cases. The results achieved here, demonstrates evolvability of FGRN systems as distributed controllers of the robots which is the first step towards implementing FGRN systems to cope with more complex challenges in modular robots.

In an additional experiment, one of the three robots is selected and the behavior of the evolutionary system after a module failure is investigated. The results demonstrate that the FGRN system is evolvable to find new solutions for the new situations.

2 Related Works

In the field of computation systems different approaches have been used to create evolutionary systems with a developmental process for genotype to phenotype mapping. In some works, models of GRNs are evolved for making mathematical output functions such as sinusoid, exponential and sigmoid [7]. Some researchers have designed GRN systems for developing neural networks for controlling robots [8, 9, 10] or specifying the morphology of 3D organisms [11]. Also, GRN models have been used to develop the morphology of robots as well as their neural network controllers [12]. Other models of GRNs have been proposed in [13, 14, 15, 16].

In a model of GRN called FGRN, Bentley [1] introduces fractal proteins as an intermediate substrate that resembles the role of proteins in the cell. The recursive and self-similar nature of fractal proteins make the fractal genetic space evolvable,

complex, and redundant [2, 3, 17]. FGRNs are evolved to produce desired patterns [18], controlling conventional robot and motion planning [2, 4]. On the other hand, in the field of robotics, different approaches have been investigated by researchers to control modular robots. Co-evolving morphology and control of simulated modular robots [19, 20], learning strategies [6, 21] and applying central pattern generators to control modular robots [22, 23] are some of the reported researches in the field.

3 Gene Regulatory Networks

3.1 Biological Inspiration

Development of phenotypes can be thought of as a product of interaction between genes and proteins in their environment. Almost everything inside a cell is carried out by proteins. Proteins drive development and functioning of a cell and are used for communication between cell and its environment that might include other cells.

A cell contains a genome and a cytoplasm which are surrounded by a membrane (Fig. 1). The membrane separates the interior of a cell from the outside environment. Receptor proteins are embedded in the membrane and control the movement of environmental proteins into the cell. The cytoplasm contains a compound of proteins inside the cell. The genome consists of a set of genes. Every gene contains a sequence that encodes a protein (coding region) and a sequence that determines the conditions for activation or suppression of that gene (promoter region) (Fig. 1).

Fig. 1. An example cell (left) and a gene (right)

An active gene expresses and produces its appropriate protein as encoded in its coding region. For a gene to be activated, a proper amount of appropriate protein compounds in cytoplasm must match the promoter region of the gene.

The cytoplasm content is altered by proteins produced by genes inside the cell or the environmental proteins which have entered the cell passing through receptors.

During the development of a cell, the protein content of the cytoplasm might match against the promoter of some genes and get them to suppress or express proteins.

Every produced protein will merge to the cytoplasm and would alter its content. The new content, in turn, affects the expression of genes in the next step. It might cause new proteins to be produced or the amount of the current proteins in the cytoplasm to be changed. In this way, every gene which makes protein inside a cell might

influence the expression of other genes (including itself) directly or indirectly. In the same way, the proteins which enter the cell from the environment can influence expression of genes and participate in development of the cell. On the other hand, the functional behavior of a cell is determined by special proteins in the cell. These proteins may change the shape, structure, or other properties of the cell, or might be used as signals to the outside environment. Production of these proteins is determined by the corresponding genes and the protein content of the cytoplasm. Therefore, variations in cytoplasm content might lead to variations in the behavior of the cell to the outside world.

The ongoing interaction between proteins and genes continues for whole lifetime of a cell and is considered a network of genes which regulate the expression of each other and is called a Gene Regulatory Network (GRN).

3.2 Implementation

In a series of works reported by Bentley [1, 2, 3, 18] a protein model called fractal protein is developed as the protein substance of gene regulatory networks in an evolutionary system.

Each fractal protein is a square window on the Mandelbrot fractal set with a prespecified resolution (Fig. 2). Fractal proteins are represented by a square matrix of integer values but can be encoded by only three values (x, y, z). (x, y) determine the center of the window on the fractal set. z specifies the length of the sides and can be inversely considered as the amount of magnification in the fractal image. Iterating Mandelbrot formula along with the three values specifies every entry in the matrix of a fractal protein and determines the image. Fig. 2 shows an example fractal protein.

In addition to a square matrix of integer values, a single integer value relates to each fractal protein as its concentration level. The concentration level represents the current amount of the protein. The value increases when more of the protein is produced and decreases slowly over time to resemble normal degradation that happens in real cells.

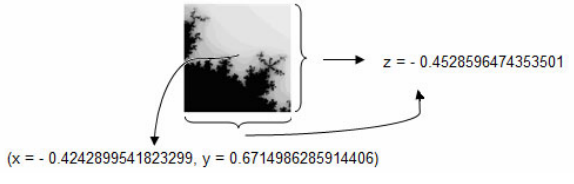


Fig. 2. An example fractal protein and the three values which specify it

Fractal proteins can merge together and make protein compounds. A fractal protein compound is represented by a square matrix of integer values in the same way as fractal proteins. In order to merge a protein into a protein compound, for every entry in the corresponding matrices, the winner is the paler pixel in the fractal image. See Fig. 3 for an example.

The cytoplasm of an FGRN cell is a compound of all the proteins inside the cell. Every protein that is produced in the cell or enters the cell from outside will be merged into the content of the cytoplasm.

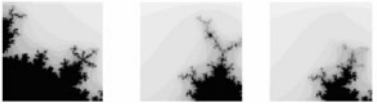


Fig. 3. Two proteins (left and middle) are merged (right)

Fig. 4. The cytoplasm protein compound (left) matches against the promoter of a gene (middle) and the absolute difference is calculated from the result (right)

Fig. 5. Environmental protein (left) passes through the receptor protein (middle) and some portion of it (right) is allowed to enter the cytoplasm

A genome in an FGRN cell consists of a set of genes. Genes consist of a sequence of values representing promoter region, coding region, threshold parameters, and type of the gene.

The coding region contains the three real values which encode a fractal protein. In the same way as the coding region, the promoter region consists of three real values that encode a square matrix of fractal values as well. This matrix works as a window that will be put on the cytoplasm protein compound matrix and is used to calculate the matching degree between the promoter of the gene and cytoplasm content (See Fig. 4 for an example). The matching degree along with the total concentration of matched proteins on promoter region, determine the degree of activation (or suppression) of the gene and might specify its protein production rate. Threshold parameters are used to calculate the matching degree and protein production rate of each gene.

To assimilate different types of genes in a cell, each gene contains an integer value representing its type. Every gene belongs to one of the following types:

- Regulatory gene, which comprises both promoter and coding region. Its encoded protein will be produced and merged into cytoplasm and participate in regulation of expression (or repression) of genes.

- Environmental gene, determines the proteins which might be present in the environment of the cell.
- Cell receptor gene, contains a coding region and produces a receptor protein. Receptor proteins merge together and act as a mask to permit variable portions of environmental proteins to the cytoplasm (See Fig. 5).
- Behavioral gene, which comprises a promoter region and a coding region. The values in the coding region can directly participate to determine the outputs of the cell.

Lifetime of an FGRN cell consists of a number of developmental cycles which can be summarized as the following steps:

- Produce receptor and environmental proteins.
- Pass the environmental proteins through receptors and merge them into the cytoplasm content.
- For every behavioral and regulatory gene,
  - If the content of cytoplasm matches the promoter,
    - If the gene is behavioral: utilize the coding region to specify the cell's outputs
    - If the gene is regulatory: express the coding region and merge the produced protein into the cytoplasm
- Update concentration level of proteins in the cytoplasm.

For more detailed descriptions of FGRN systems and the corresponding formulas see [1, 2, 4].

4 Evolving FGRNs to Control Modular Robots

Every module of a robot is considered a cell in a multi-cellular creature. Each module contains an FGRN cell which includes its genome and cytoplasm. All the FGRN cells run in parallel and independent of each other and make their own sequence of output commands for the modules containing them.

All the cells are genetically identical which means they contain an identical copy of a genome. Environmental information about the number of connections and the initial orientation of the module which contains the cell is provided for each cell in the form of environmental proteins. Therefore, two cells which are contained in two modules with different environmental situations initially contain different proteins in their cytoplasm. Different cytoplasm content might activate different genes of the genome of each cell and leads to different internal interactions and developmental trends. Consequently, while the cells are genetically identical, different phenotypic characteristics might be formed and different output commands might be generated by the cells during their lifelong development.

In this work proper genomes are evolved such that when they are copied in all the cells of one modular robot, each cell can generate a right sequence of commands for its module using the appropriate environmental information and make the robot perform its locomotion task.

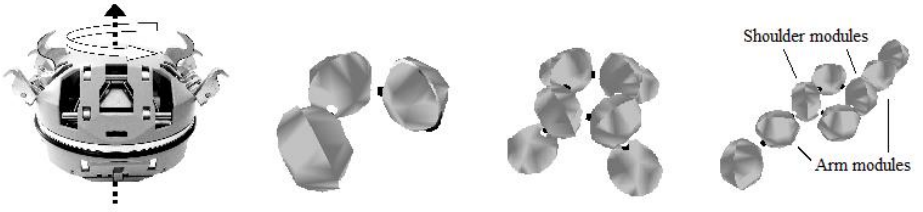


Fig. 6. From left to right: An ATRON module, Two-wheeler, Quadrupedal, and Crawler robots

4.1 Experimental Setup

Robot Simulator. Simulation experiments are performed in an open-source simulator named Unified Simulator for Self-Reconfigurable Robots (USSR) [24]. The simulator provides physics-based simulations of different modular robots including the ATRON robot. The ATRON robot is a homogenous, lattice-based self-reconfigurable modular robot. An ATRON module weighs 0.850kg and has a diameter of 110mm. A module consists of two hemispheres which can rotate infinite relative to each other with a speed of 360 degrees in six seconds. Each hemisphere contains two passive (bars) and two active connectors (hooks), see Fig. 6.

Table 1. Genetic parameters

#population size	#generations	crossover rate	mutation rate
20	50	40 %	1 %

# regulatory genes	#receptor genes	#environmental genes	# behavioral genes
4	5	9 / 10 (for crawler)	1

Genetic and developmental configurations. A population of 20 FGRN genomes is evolved for 50 generations using a variant of steady-state genetic algorithm with life-span limits [1]. Each genome is initialized with randomly generated regulatory, receptor, environmental, and behavioral genes. The initial number of each type of gene and the genetic parameters are shown in Table 1. Evolution is allowed to regulate the number of each type of genes (See [1, 4] for more details).

To evaluate a genome, identical versions of a genome are copied to all the modules' FGRN cells. Each cell receives some environmental proteins describing the number of connections and the initial orientation of the module in which it is situated. Also an additional environmental protein common between all the cells is initially provided.

In order to make an action for each module in every step, modules independently run their own FGRN cell for one developmental cycle and receive an output from the cell. The cell output is calculated on the basis of activation of behavioral genes inside the cell. The output is mapped to one of the following three commands that will be performed by the module in that step:

- rotateRight – rotates clockwise 90 degrees
- rotateLeft – rotates counterclockwise 90 degrees
- stop – rotate zero degrees

After a specified time span (50 sec.), fitness is evaluated as the distance between the initial position and the end position of center of mass of the robot.

The run-time procedure of a robot can be summarized as follows:

- Create genome
- For every Module of the robot:
  - Make an empty FGRN cell and put a copy of the genome into it.
  - During the run-time of the robot:
    - Receive information about the module's environment and activate the relevant environmental proteins.
    - Develop the cell for one cycle according to the developmental steps in section 3.2 and receive cell output.
    - Translate cell output to the module command and
    - Execute the command.

Case studies. We have evolved multi-cellular FGRN controllers for three robots with different morphologies and the same genetic configurations. Fig. 6 shows the three morphologies which are used. In order to keep things as simple as possible we didn't use any communication between modules. As it might be expected, for the two-wheeler robot, evolution leads to controllers which rotate the two opposite modules in the opposite directions to move the robot like a car. For the quadrupedal robot, a swimming-like behavior evolved. For the crawler robot, different crawling gaits evolved. In order to evaluate the robots, the velocity of the locomotion is calculated for each robot. The best and population-average velocities are shown in Fig. 7. The figure shows the results averaged over 10 independent runs. The results are compared with the results achieved by a learning strategy reported in [6]. Table 2 shows the higher velocities achieved by the FGRN controllers and the learning controllers. The learning strategy is reinforcement learning accelerated by a heuristic which detects and repeats potentially underestimated actions to accelerate the estimation accuracy and presumably accelerates the learning.

In another experiment, the evolvability of the FGRN system is investigated after a module failure. The crawler robot is selected for this experiment. We considered the solutions found in the last experiment. Different gaits were recognizable between the solutions evolved in the 10 runs. Based on the position of the modules which had more effect in the locomotion, the solutions can be categorized in two main groups-solutions which mainly use the shoulder modules and solutions which mainly use the arm modules (See Fig. 6). The second group which has the velocity of higher than average-velocity is selected. In order to resemble a situation of failure, one of the modules of high importance (one of the arms) is disabled while the robot uses the previously evolved FGRN controller. Since the controller is not suitable for this new situation, the fitness falls considerably. Afterwards, the controllers are allowed to

evolve for 30 generations and the velocities of the new solutions are evaluated. As it is shown in Fig. 7, the velocity of robots falls after failure, and then rises when evolution continues. The performance of the new evolved controllers is investigated for robots both with the broken module and intact module (after repairing the broken module). Table 3 shows the velocities in different situations and represents a good performance for the new evolved controllers in both cases of intact and broken modules. Furthermore, the experiment repeated with the broken module to evolve controllers from scratch (See Fig. 7). The velocities are averaged over 10 runs of evolution (Table 3).

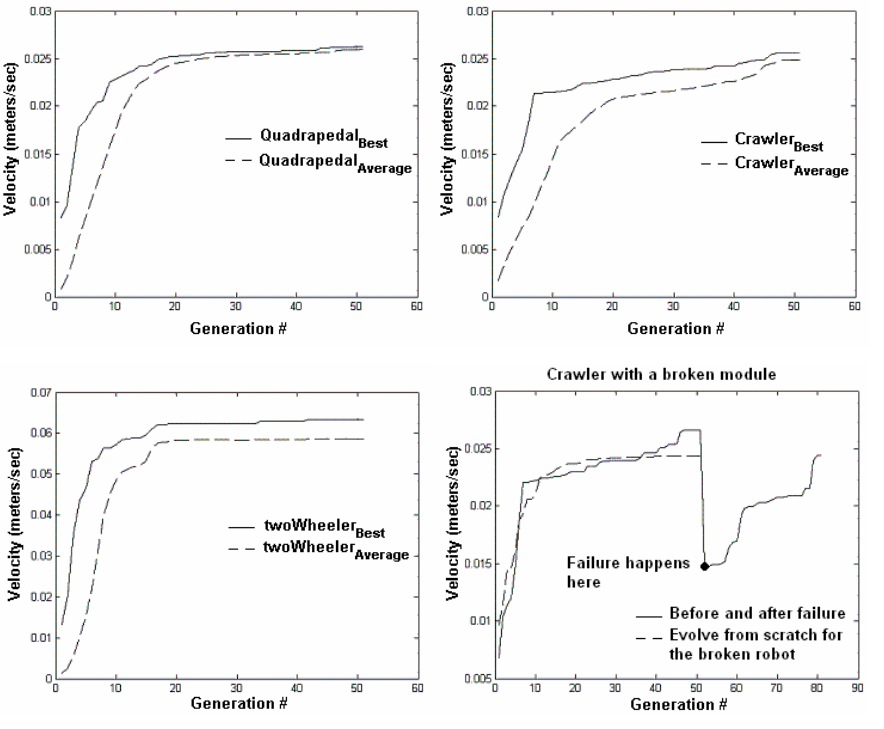


Fig. 7. Velocities for the three robots and velocities of the crawler with a broken module

Table 2. Comparison of the best velocities achieved by FGRN and [6] learning algorithm

Robot Configuration	Learning [6]	FGRN (Population average)	
	Mean	Mean	Standard deviation
Quadrupedal	0.0208	0.0260	0.0011
Crawler	0.0210	0.0248	0.0038
Two-wheeler	0.0383	0.0586	0.0007

Table 3. Averaged velocities of the Crawler robot

Velocity (all runs)	Before failure (selected runs)	After failure (selected runs)
0.0248	0.0262	0.0145

After more evolution (selected runs) – robot with failed module	After more evolution (selected runs) – module repaired	Velocity for broken robots evolved from scratch (all runs)
0. 0245	0.0235	0.0244

5 Conclusion

In this paper, we explored application of FGRN systems to control of modular robots. FGRN systems are inspired by natural cells and due to their internal interactions are able to generate complex output patterns which might be used as control commands. Implementing the FGRNs in multi-cellular way provides us a distributed controller for ATRON modular robots. The local controllers for all modules are encoded identically and run independently. In order to keep the system as simple as possible, there is no communication between modules in the current implementation. Communication between modules and different sensory information might be included in the future works.

We carried out experiments with different morphologies of the ATRON in a locomotion task and reached good performances. Results are compared to the previously reported results of robots employing a reinforcement learning strategy. Furthermore, we investigated the capability of the FGRN system to evolve more in case of a failure and the achieved controllers are evaluated for both intact and broken robots. The results show that the FGRN system is still evolvable to find new solutions for new situations of the robot.

Acknowledgments. The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 - Future Emerging Technologies, Embodied Intelligence, under grant agreement no. 231688.

References

1. Bentley, P.J.: Fractal proteins. J. Genetic Programming and Evolvable Machines 5(1), 71–101 (2004)
2. Bentley, P.J.: Adaptive Fractal Gene Regulatory Networks for Robot Control. In: Genetic and Evolutionary Computation Conference, Seattle, USA (2004)
3. Bentley, P.J.: Evolving Fractal Gene Regulatory Networks for Graceful Degradation of Software. In: Babaoğlu, Ö., Jelasity, M., Montresor, A., Fetzer, C., Leonardi, S., van Moorsel, A., van Steen, M. (eds.) SELF-STAR 2004. LNCS, vol. 3460, pp. 21–35. Springer, Heidelberg (2005)
4. Zahadat, P., Katebi, S.D.: Tartarus and Gene Regulatory Networks with Input. J. Adv. Complex Sys. 11(6), 803–829 (2008)
5. Ostergaard, E.H., Kassow, K., Beck, R., Lund, H.H.: Design of the ATRON Lattice-Based Self-Reconfigurable Robot. J. Auton. Robots 21(2), 165–183 (2006)

6. Christensen, D.J., Bordignon, M., Schultz, U.P., Shaikh, D., Stoy, K.: Morphology Independent Learning in Modular Robots. In: International Symposium on Distributed Autonomous Robotic Systems, pp. 379–391 (2008)
7. Kuo, P.D., Leier, A., Banzhaf, W.: Evolving Dynamics in an Artificial Regulatory Network Model. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 571–580. Springer, Heidelberg (2004)
8. Jakobi, N.: Harnessing Morphogenesis. In: Paton, R. (ed.) International Conference on Information Processing in Cells and Tissues, Liverpool, UK, pp. 29–41 (1995)
9. Dellaert, F., Beer, R.: A Developmental Model for the Evolution of Complete Autonomous Agents. In: Fourth International Conference on Simulation of Adaptive Behavior, pp. 393–401. MIT Press, Cambridge (1996)
10. Federici, D.: Evolving a Neurocontroller through a Process of Embryogeny. In: Schaal, S., et al. (eds.) Eighth International Conference of Simulation and Adaptive Behavior, pp. 373–384. MIT Press, Cambridge (2004)
11. Eggenberger, P.: Evolving Morphologies of Simulated 3D Organisms Based on Differential Gene Expression. In: Husbands, P., Harvey, I. (eds.) Proc. 4th European Conference on Artificial Life (ECAL), pp. 205–213. MIT Press, Cambridge (1997)
12. Bongard, J.C., Pfeifer, R.: Evolving Complete Agents Using Artificial Ontogeny. In: Hara, F., Pfeifer, R. (eds.) Morpho-functional Machines: The New Species (Designing Embodied Intelligence), pp. 237–258. Springer, Heidelberg (2003)
13. Federici, D., Downing, K.: Evolution and Development of a Multi-Cellular Organism: Scalability, Resilience and Neutral Complexification. J. Artificial Life 12(3), 381–409 (2006)
14. Banzhaf, W.: On evolutionary design, embodiment and artificial regulatory networks. In: Iida, F., Pfeifer, R., Steels, L., Kuniyoshi, Y. (eds.) Embodied Artificial Intelligence. LNCS (LNAI), vol. 3139, pp. 284–292. Springer, Heidelberg (2004)
15. Hornby, G.S., Pollak, B.: The Advantages of Generative Grammatical Encodings for Physical Design. In: Congress on Evolutionary Computation, pp. 600–607. IEEE Press, Los Alamitos (2001)
16. Kennedy, P.J., Osborn, T.R.: A Model of Gene Expression and Regulation in an Artificial Cellular Organism. J. Complex Systems 13(1), 1–28 (2001)
17. Bentley, P.J.: Methods for Improving Simulations of Biological Systems: Systemic Computation and Fractal Proteins. J. R. Soc. Interface (2009)
18. Bentley, P.J.: Evolving fractal proteins. In: Tyrrell, A.M., Haddow, P.C., Torresen, J. (eds.) ICES 2003. LNCS, vol. 2606, pp. 81–92. Springer, Heidelberg (2003)
19. Sims, K.: Evolving 3d morphology and behavior by competition. In: Brooks, R., Maes, P. (eds.) Proc. Artificial Life IV, pp. 28–39. MIT Press, Cambridge (1994)
20. Marbach, D., Ijspeert, A.J.: Co-evolution of Configuration and Control for Homogenous Modular Robots. In: Proc. 8th International Conference on Intelligent Autonomous Systems, Amsterdam, Holland, pp. 712–719 (2004)
21. Maes, P., Brooks, R.A.: Learning to Coordinate Behaviors. In: National Conference on Artificial Intelligence, pp. 796–802 (1990)
22. Kamimura, A., Kurokawa, H., Yoshida, E., Murata, S., Tomita, K., Kokaji, S.: Automatic Locomotion Design and Experiments for a Modular Robotic System. J. IEEE/ASME Transactions on Mechatronics 10(3), 314–325 (2005)
23. Sproewitz, A., Moeckel, R., Maye, J., Ijspeert, A.: Learning to Move in Modular Robots using Central Pattern Generators and Online Optimization. J. Rob. Res. 27(3-4), 423–443 (2008)
24. Christensen, D.J., Schultz, U.P., Brandt, D., Stoy, K.: A Unified Simulator for Self-reconfigurable Robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2008)

The Dependence of Braking Strategies on Optical Variables in an Evolved Model of Visually-Guided Braking

Didem Kadihasanoglu[1], Randall D. Beer[1,2], and Geoffrey P. Bingham[1,3]

[1] Cognitive Science Program
[2] School of Informatics and Computing
[3] Department of Psychological and Brain Sciences
Indiana University, Bloomington, IN 47406 USA
{dikadiha,rdbeer,gbingham}@indiana.edu

Abstract. This paper presents results from two sets of experiments which investigate how strategies used by embodied dynamical agents in a simple braking task are affected by the perceptual information that the agents receive. Agents are evolved in a simple 2D environment containing one stationary object. The task of the agents is to stop as close as possible to the object without hitting it. The results of these experiments demonstrate that most of the evolved agents use an impulsive braking strategy, in which deceleration is not controlled continuously. Potential causes of this impulsive braking strategy and possible future directions are discussed.

Keywords: Evolutionary robotics, visually-guided braking, image size, image expansion rate, tau, tau-dot.

1 Introduction

There is a growing interest in applying the evolutionary approach to model experimental paradigms from psychology. For example, inspired by the psychological experiments such as double-TV-monitor experiments and perceptual crossing, Iizuka and Di Paolo [1] investigated how embodied agents establish live interactions and discriminate this type of interaction from the identical recorded motions. Rohde and Di Paolo [2] implemented an evolutionary robotics simulation to guide the analysis of empirical data on adaptation to sensory delays. In another work, Wood and Di Paolo [3] applied evolutionary robotics techniques to model the famous "A-not-B" error paradigm. Considering the sensory and motor capabilities of evolved model agents, ecological psychology in general, and control of locomotion in particular, provides an excellent research area in which evolutionary robotics techniques can be used.

Visual control of locomotion, which requires coordination between perception and action, is essential for any mobile agent, whether it is a human, animal or a robot, to move around, explore and interact with the world. One approach to the control of locomotion is based on internal representations such as world models and plans [4]. An alternative approach, which is developed by Gibson [5,6,7] is based on the idea that adaptive behavior is controlled by the perceptual information that is available to the observer. When an observer moves in an

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 555–564, 2010.

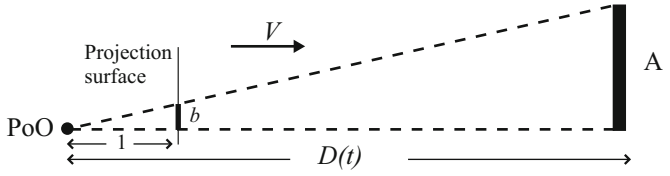


Fig. 1. A schematic view of an observer approaching an object of size A. PoO denotes the point of observation, D is the distance from the object, V is the observer's velocity, and b is the image size at time t. The retina of the observer (i.e. the projection surface) is approximated as being 1 unit distance away from the PoO.

environment, a pattern of motion is produced at the eye of the observer called *the optic flow* [5,7]. Optic flow provides information about the 3D layout of the environment, objects in the environment, and observer's self-motion through the environment, and can be used to control locomotion.

In this paper, we explore how perceptual information received by an agent affects the behavior of the agent in the context of a simple braking task based on the experimental paradigm used in Yilmaz and Warren [8]. During a direct approach to an object, the image of the object on the observer's retina expands (see, Fig. 1). Gibson [6] argued that the rate of optical expansion could be used to control braking. There are a number of ways in which optical expansion rate could be used. The first one is to keep expansion rate at a constant positive value, the magnitude of which depends on the point when braking is initiated [8]. Another strategy is based on the tau (τ) variable. Lee [9] demonstrated that the optical variable tau, which is the ratio of object's image size (b) to the image expansion rate ($\dot{b}$), specifies the time-to-contact (TTC) with the object as long as the current velocity is held constant:

$$\tau = \frac{b}{\dot{b}} = \frac{D}{V} = TTC \; . \tag{1}$$

Lee [9] also showed that the time derivative of τ ($\dot{\tau}$, or tau-dot) could be used to control deceleration during braking. If $\dot{\tau} < -0.5$, the current deceleration is too low and if it is maintained, it will result in a crash. If $\dot{\tau} > -0.5$, the current deceleration is too high and one will stop away from the object. If $\dot{\tau} = -0.5$, then the current deceleration will bring the observer to a stop right at the object. One can control braking by adjusting deceleration so as to keep $\dot{\tau}$ around -0.5, which is known as the "constant $\dot{\tau}$" strategy. A third hypothesis claims that braking could be controlled by computing the required deceleration from spatial variables such as distance to the object, the velocity of the observer, object size together with the optical variables [8]. Yilmaz and Warren [8] list two other strategies, in which deceleration is not continuously controlled but the brake is used in an impulsive fashion. The first strategy is the "slam on the brake" strategy in which an observer approaches the object with a constant velocity and then applies maximum deceleration. The second strategy is to apply a large deceleration at the beginning of the approach and then slowly drift to the object, using one or more deceleration spikes later to stop. Yilmaz and Warren call this strategy the "bang-bang" strategy.

The work presented in this paper explores how braking strategies of the evolved model agents are influenced by the perceptual information that is available to the agents. The simulations are based on the experiment carried out by Yilmaz and Warren [8], in which participants viewed computer displays simulating an approach to a stationary road sign. The task of the participants was to stop as close as possible to the road sign. The deceleration was regulated via a spring-loaded mouse. The only difference between our simulations and original experiment is that in our simulations the initial TTCs are longer. The reason to keep the TTC values longer is to investigate when braking is initiated.

2 Methods

In a series of experiments, model agents that are placed in a simple 2D environment with one stationary line object are evolved (see, Fig. 2(a)). The length of the object was 60. The agent has a circular body with a diameter of 30, and four sensors. The first sensor receives an input proportional to the image size (b) of the object, which is calculated using the geometry illustrated in Fig. 1. The second sensor receives an input proportional to the image expansion rate ($\dot{b}$). In the experiments reported here, we simulated translation through a rigid environment with no rotations, such as eye and head rotations. In this case, $\dot{b}$ can be calculated using Formula 1. The third and the fourth sensors detect the optical variable τ and $\dot{\tau}$, respectively. The task of the agent is to stop as close as possible to the object without hitting it. The agent can only move forward, i.e., its heading is fixed and it can only decelerate. So, it is a second-order, Newtonian system. The braking force is controlled by the motor neuron.

The behavior of each agent is controlled by a continuous-time recurrent neural network (CTRNN) with the following state equation:

$$T_i \dot{s}_i = -s_i + \sum_{j=1}^{N} w_{ji}\sigma(g_j(s_j + \theta_j)) + I_i \qquad i = 1, ..., N \; . \tag{2}$$

where N is the number of the CTRNN nodes, s is the state of each neuron, T_i is the time constant, w_{ji} is the strength of the connection from the j^{th} neuron to the i^{th} neuron, g is a gain, θ is a bias term, $\sigma(x) = 1/(1 + e^{-x})$ is the standard logistic activation function and I is the external input. The output of a neuron is $O_i = \sigma(s_i + \theta_i)$. All neurons, except for the sensory neurons, had a gain of 1.0. The agent's four sensors are fully connected to four fully interconnected interneurons which are in turn fully connected to one motor neuron controlling the vertical motion of the agent (see, Fig. 2(b)). The agent's deceleration is calculated using the following formula:

$$-\dot{V} = k \times O_m \; . \tag{3}$$

where O_m is the output of the motor neuron and k is a scaling constant which is set to be 3.0 in all of the experiments that will be reported in the next section.

The connection weights ($w_{ji} \in [-16, 16]$), biases ($\theta \in [-16, 16]$), time constants ($T \in [1, 10]$) and the gains ($g \in [1, 5]$) were evolved using a real-valued hill climbing algorithm with fitness-proportionate selection. New generations were created by applying random Gaussian mutations to the selected parents. The

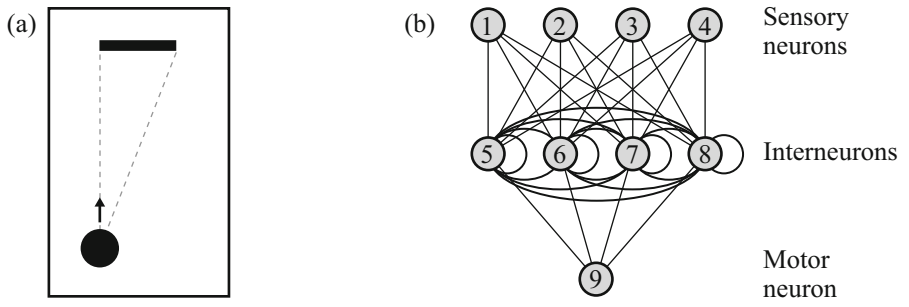


Fig. 2. (a) Basic experimental set-up (b) The CTRNN architecture

mutation variance was 0.45. The fitness scaling multiple was 1.03. Simulations were integrated using the Euler method with an integration step size of 0.1.

An agent's performance is determined based on its behavior in a number of evaluation trials. The object's position in the environment is fixed across all trials. The horizontal position of the agent is also the same across trials but the vertical distance between the agent and the object varied. The agent has 7 different initial distances from the object (120, 135, 150, 165, 180, 205 and 210) and 6 initial velocities (10.0, 11.0, 12.0, 13.0, 14.0 and 15.0). As a result, initial TTC with the object varies between 8.0 and 21.0. Each possible combination of the agent's initial distances and velocities was presented as a trial, resulting in $7 \times 6 = 42$ evaluation trials. At the beginning of each trial, the agent's neural states are initialized to zero. Then, the agent is placed in one of the 7 locations and its velocity is initialized to one of the 6 velocities. A trial ends when the velocity of the agent is 0.0 or when the agent touches the object, i.e., when the vertical distance between the center of the agent and the object is less than or equal to the radius of the agent. The overall fitness of the agent was determined by averaging the fitness of the agent over 42 evaluation trials.

Two different fitness functions were used in two different sets of experiments. The first fitness function was based on the velocity of the agent and the vertical Euclidean distance between the agent and the object at the end of a trial. It minimizes the agent's velocity and the distance between the agent and the object. Then, the performance measure to be maximized was:

$$\frac{\sum_{i=1}^{NumTrials} (1 - d_i/dMax_i)(1 - v_i/vMax_i)}{NumTrials}. \tag{4}$$

where $NumTrials$ is the total number of trials, d_i is the vertical distance between the agent and the object at the end of i^{th} trial, $dMax_i$ is the initial vertical distance of the agent from the object, v_i is the agent's velocity at the end of the i^{th} trial and $vMax_i$ is the agent's initial velocity. The second fitness function also minimizes the trial duration in addition to the velocity and the distance. In this case, the performance measure to be maximized was:

$$\frac{\sum_{i=1}^{NumTrials} ((1 - t_i/t_{max}) + (1 - d_i/dMax_i)(1 - v_i/vMax_i)/2)}{NumTrials} . \quad (5)$$

where t_i is the duration of the i^{th} trial and t_{max} is the maximum trial duration. Since it always takes some time for agents to end a trial, it is not possible to evolve agents with perfect fitness values using this second fitness measure.

3 Results

We conducted two sets of experiments. The aim of the first set of experiments is to investigate the effect of the perceptual information on the evolved braking strategies. The second set of experiments investigates the effect of the fitness measure on the evolved braking strategies. In the first set of experiments, we manipulated the perceptual information available to the agents. The fitness values of the agents were calculated using the fitness measure given in Formula 4. There were four groups of agents, each receiving a different type of information. The first group only received image size as the information. The input to the remaining three sensors was set to be zero. Similarly, the second, third and fourth groups received only the image expansion rate, tau and tau-dot as the information, respectively. From now on, agents in different groups will be referred by the information they receive such as image size agent or tau agent.

Preliminary results indicated that if the trial duration was not limited, most of the evolved agents exhibited the "bang-bang" strategy regardless of the visual information they were receiving. In other words, the agents decreased their velocities to near zero values right at the beginning of the trials and then slowly drifted to the object, giving rise to very long trial durations. In order to prevent agents from using the "bang-bang" strategy, the maximum trial duration was set to be 500 time steps in all of the experiments. For each group, 10 evolutionary runs were performed with different random seeds. For all evolutionary runs, the population size was 150 and the maximum generation number was set to 5000. Agents that can successfully solve the task were evolved in all four groups.

For the image size group, 9 out of 10 evolutionary runs produced agents that had a fitness value over 90% on the 42 evaluation trials. The best evolved image size agent across 10 runs attained a fitness value of 99.34%. It is important to note that the fitness measure does not explicitly punish the agents for non-zero velocities. This means that it is possible for the agents to have very small but non-zero velocities at the end of the trials. The velocity profile of the best evolved image size agent across 42 evaluation trials is given in Fig. 3(a). As can be seen from the figure, at the beginning of each trial the agent applies maximum deceleration and then moves with a constant velocity, the magnitude of which depends only on the agent's initial velocity. After that, it initiates its final braking. Although there are slight variations with the decreasing fitness values, these velocity profiles were essentially the same across agents. The velocity of the best agent is zero at the end of each evaluation trial and it stops very close to the target. The average final distance between the agent and the object across 42 evaluation trials was 0.94. The performance of the agent was also tested on 4641 generalization trials in which the agent's initial distance from the object was varied between 120 and 210, with an increment of 1 and the agent's initial

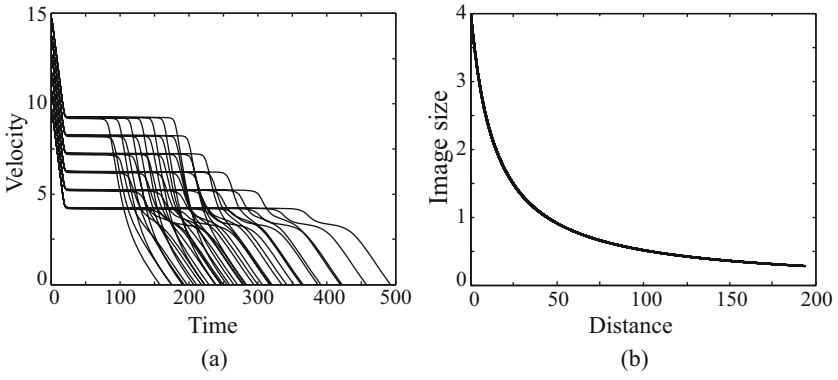


Fig. 3. (a) Velocity profiles of the best evolved image size agent across 42 evaluation trials. (b) Change in the image size as a function of distance across 42 evaluation trials.

velocity was changed from 10.0 to 15.0, with an increment of 0.1. The average performance of the best evolved image size agent was 98.76%. In 819 of the generalization trials, the agent touched the object with an average velocity of 0.42. All of these trials correspond to the trials in which the agent's initial velocity varied between 11.1 and 11.9. The average final distance of the agent from the object across generalization trials was 0.87.

Fig. 3(b) shows how the image size changes as the distance between the agent and the object changes across 42 evaluation trials. As can be seen from the figure, the shape of the curve is the same regardless of the initial conditions. Even though the agent did not explicitly receive image expansion rate or tau as the information, we also examined how these variables changed as the agent approaches to the object. The examination revealed that the image expansion rate was never held constant and the agent initiated its final braking when tau reached a certain value, the magnitude of which changes with the agent's initial velocity only. As the agent's initial velocity increased from 10.0 to 15.0, the tau value at which the braking was initiated decreased from 9.1 to 4.3.

For the image expansion rate group, all of the 10 evolutionary runs produced agents that had a fitness value over 96% on the 42 evaluation trials. The fitness value of the best evolved image expansion rate agent was 97.16%. The agent's velocity was always zero at the end of the evaluation trials but it stopped farther from the object compared to the image size agent. The average final distance was 4.09. The velocity profiles of the best image expansion rate agent are given in Fig. 4(a). The agent uses "slam on the brake" strategy and adjusts its braking in an impulsive fashion. Similar to the image size agent, the image expansion rate agent, too, applies maximum deceleration right at the beginning of the trials and then continues to move with a constant velocity. However, unlike the image size agent, it also applies maximum deceleration at the end of the trials. The image expansion rate was never held constant and the final braking was initiated when tau reaches a certain value, which varied between 6.6 and 3.2. The behavior of the rest of the agents was also very similar. The average performance of the best image expansion rate agent on the generalization trials was 97.01%. At the end of each generalization trial, the agent's velocity was zero. The average final

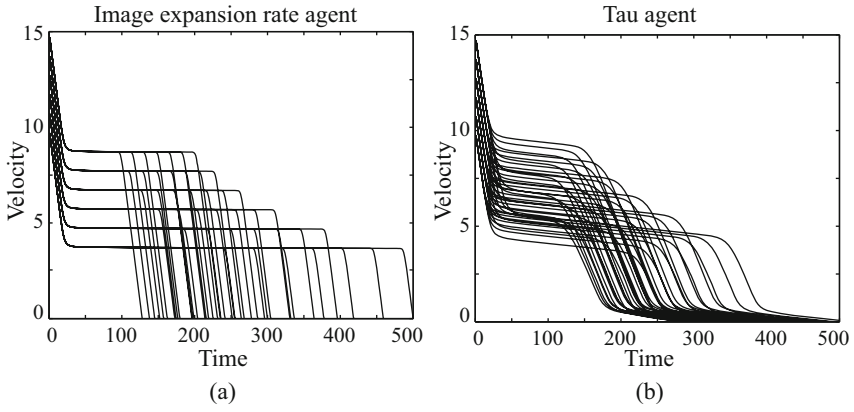


Fig. 4. Velocity profiles of the best evolved image expansion rate agent (a) and tau agent (b) across 42 evaluation trials

distance between the agent and the object was 4.31, which is greater than the average final distance for the image size agent.

For the tau group, the best evolved agent in each evolutionary run had a fitness value of at least 97.7%. The best evolved agent across 10 runs achieved a fitness value of 99.33%. This agent touched the object in 5 of the evaluation trials with an average velocity of 0.11. The average final distance between the agent and the object across all evaluation trials was 0.20. Similar to the image size agent, the tau agent also stopped very close to the object and risked touching it in some of the trials. However, its velocity profiles, which are illustrated in Fig. 4(b), are different. The agent seems to use a mixture of the "bang-bang" and the "slam on the brake" strategies. The tau agent also applies maximum deceleration at the beginning of the trials. But this time, the magnitude of the reduced velocity is not only dependent on the agent's initial velocity but also on the agent's initial distance. It varies between approximately 5.0 and 10.0. Then, the tau agent continues to decrease its velocity. After initiating its final braking, it slowly drifts to the object. The final braking was initiated when tau reached a certain value, which varied between 7.5 and 5.6. The agent's average performance on the generalization trials was 99.82%. The agent touched the object in 482 of the generalization trials with an average velocity of 0.05. The agent's average distance from the object at the end of generalization trials was 0.22. The examination of the velocity profiles of the rest of the agents revealed two more behaviors. The first behavior is similar to the behavior of the image expansion rate agent. The second behavior is more like the "slam on the brake" strategy, in which the agent moves with its initial velocity for a period of time and then rapidly increases its deceleration to the maximum value.

Finally, for the tau-dot group, in 9 of the evolutionary runs, the best evolved agent achieved a fitness value 90% or higher. The best evolved tau-dot agent across 10 evolutionary runs had a fitness value of 97.06%. The agent touched the object in 4 of the evaluation trials with an average velocity of 0.12. The average final distance between the agent and the object was 4.2 but the variation between the final distances among trials was greater. Fig. 5(a) shows the velocity profiles

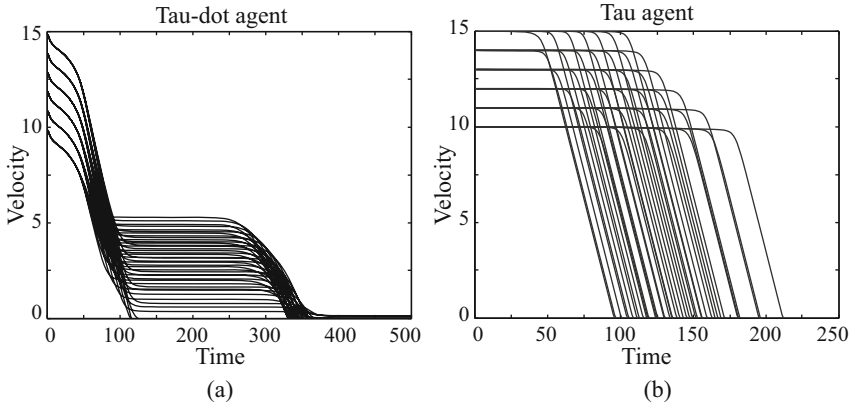


Fig. 5. (a) Velocity profile of the best evolved tau-dot agent across 42 evaluation trial. (b) Velocity profile of the best tau-agent that is evolved when the trial duration is explicitly included in the fitness measure.

of the best evolved tau-dot agent, which were very similar across agents. At the beginning of each trial, the agent decelerates at a decreasing rate. Regardless of the initial velocity or the distance, the agent applies the same deceleration. Then, the agent increases the deceleration to a maximum value, which varies with the agent's initial velocity and the distance. In three of the evaluation trials, this initial braking brings the agent to a stop very close to the object. In the remaining trials, the agent uses a kind of "bang-bang" strategy. It moves with a constant velocity, then initiates the final braking and then slowly approaches to the object. The tau value at which the final braking was initiated now varies with the agent's initial distance and the velocity but the agent never let the tau values go below 7.3. The agent's average performance on the generalization trials was 97.65%. It touched the object in 476 of the generalization trials with an average velocity of 0.09. Its final distance from the object varied between 0.01 and 18.16, with a mean of 3.36.

One common feature of the evolved agents is that all of them applied maximum deceleration at the beginning of each trial regardless of the information that they received. It is possible that at the beginning of an evolutionary search, it is easier for agents to increase their fitness by decreasing their velocity. As a result, rather than being governed by the visual information, the initial braking behavior might be due to the fitness measure. In order to test this prediction, we ran a second series of experiments using the fitness measure given in Formula 5. However, since this fitness measure puts an explicit time pressure on agents, it might encourage agents to adopt "slam on the brake" strategy. 10 evolutionary runs were performed with different random seeds, each having a population size of 150 and the maximum generation number of 5000. The agents received tau as the information. In all of the runs agents whose fitness values are at least 90% were evolved. Maximum trial duration was set to be 1000 time steps. The best evolved agent had a fitness value of 92.07% across evaluation trials and 92.09% across generalization trials. At the end of each evaluation trial, the agent's velocity was zero and the agent touched the object in 9 of the generalization trials

with an average velocity of 0.46. The average final distance between the agent and the object was 2.30 across all generalization trials. The velocity profile of this agent can be seen in Fig. 5(b). As predicted, putting an explicit time pressure on the agents eliminated the maximum deceleration at the beginning of the trials. However, it also encouraged the agent to adopt the "slam on the brake" strategy. The tau values that the agent initiated the braking varied between 5.6 and 7.5. This behavior was the same across agents.

4 Discussion

In this paper, we presented results from a series of experiments in which successful simulated agents are evolved to solve a simple braking task. In the first set of experiments, there were four different groups of agents, each receiving different perceptual information: image size, image expansion rate, tau and tau-dot. In each group, the agents that can successfully solve the task were evolved. All of the best evolved agents used an impulsive braking strategy in which the deceleration was not controlled continuously. One common feature of the velocity profiles of all of the best agents was the maximum deceleration that was applied at the beginning of each trial. This is probably due to the fitness measure since it makes it easier for the agents to increase their fitness by reducing their velocity. This behavior is eliminated in the second set of experiments, when the trial duration is explicitly included in the fitness measure. However, putting a time pressure on agents encourage them to use a pure "slam on the brake" strategy, in which they approached the object by keeping their initial velocities constant and then applied maximum deceleration at the end of the trials. Yilmaz and Warren [8] indicate that the "slam on the brake" strategy cannot be an efficient strategy to control braking in actual driving because of its inertial consequences. However, in our simulations the agents do not suffer from the side effects of applying a large amount of deceleration in a short time period. One way to prevent agents from adopting this strategy could be punishing the agents for high jerk. It is also possible that the use of a sigmoid activation function for the motor neuron caused the agents to apply either full braking or no braking at all, therefore, preventing them from continuously adjusting their velocity. One possible solution could be evolving the gain of the motor neuron which was set to be 1.0 in the experiments reported in this paper.

Another common feature of the image size, tau and tau-dot agents is that they all stopped very close to the object and sometimes risked touching the object with a velocity close to zero. The image expansion rate agent was safer. Its velocity was always zero at the end of the trials. However, it stopped farther away from the object compared to the other agents. The fitness measures that we used in the experiments give equal weights to the distance and velocity components. Stopping very close to the object and touching it with a low velocity result in similar fitness values. As a result, rather than being an indication of the inefficiency of the perceptual information, the non-zero velocities at the end of the trials might simply be an artifact of the fitness measures. They might be prevented by increasing the weight of the velocity component in the fitness measure or by punishing the agent for non-zero velocities.

Our main goal for evolving these model agents is to use them as a tool for studying human control of locomotion. Yilmaz and Warren [8] provided evidence

in favor of the constant tau-dot strategy. For now, none of the evolved agents seem to apply the constant tau-dot strategy or the constant image expansion rate strategy. Currently, we are investigating under what conditions those strategies evolve. We are also investigating the braking strategies used by humans in ongoing experiments involving human subjects. Another point is that although the velocity profiles of the agents change with the changing perceptual information, the perceptual information does not significantly alter the strategies adopted by the agents. However, it is possible that the effect of the changing perceptual information might be suppressed by issues related to the fitness measures that are mentioned above. Once those issues are solved we might be able to see the effect of changing perceptual information. Then, the next step is to allow agents to receive various combinations of visual information and to investigate under what conditions one type of information is preferred as opposed to others.

Acknowledgments. This work was supported in part by NSF grant IIS-0916409 to RDB.

References

1. Iizuka, H., Di Paolo, E.A.: Minimal agency detection of embodied agents. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 485–494. Springer, Heidelberg (2007)
2. Rohde, M., Di Paolo, E.: Adaptation to sensory delays: An evolutionary robotics model of an empirical study. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 193–202. Springer, Heidelberg (2007)
3. Wood, R., Di Paolo, E.: New models for old questions: Evolutionary robotics and the 'A Not B' error. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 1141–1150. Springer, Heidelberg (2007)
4. Loomis, J.M., Beall, A.C.: Model-based control of perception/action. In: Vaina, L.M., Beardsley, S.A., Rushton, S.K. (eds.) Optic Flow and Beyond, pp. 421–441. Kluwer, Dordrecht (2004)
5. Gibson, J.J.: The Perception of the Visual World. Houghton Mifflin, Boston (1950)
6. Gibson, J.J.: Visually controlled locomotion and visual orientation in animals. British Journal of Psychology 49(3), 182–194 (1958)
7. Gibson, J.J.: The ecological approach to visual perception. Houghton Mifflin, Boston (1979)
8. Yilmaz, E.H., Warren Jr., W.H.: Visual control of braking: A test of the hypothesis. Journal of Experimental Psychology: Human Perception and Performance 21, 996–1014 (1995)
9. Lee, D.N.: A theory of visual control of braking based on information about time-to-collision. Perception 5, 437–459 (1976)

Self-organizing Robot Teams Using Asynchronous Situated Co-evolution

Abraham Prieto, Francisco Bellas, Jose A. Becerra, Becerra Priego, and Richard J. Duro

Integrated Group for Engineering Research
Universidade da Coruña, Spain
{abprieto,fran,ronin,blanca.priego,richard}@udc.es

Abstract. Self-organizing without a central controller in order to achieve collaboration towards an objective is one the main challenges in the design and operation of multi-robot systems. It is of great interest in the field to explore different approaches in order to achieve this end. Here we consider a distributed open-ended evolutionary approach called Asynchronous Situated Co-evolution (ASiCO) and introduce a series of biologically inspired concepts in order to address the solution of complex multi-robot problems with several objectives and which require the coordination of robots within distinct groups carrying out heterogeneous tasks. Different elements are explored in this paper, including how to efficiently implement a co-evolutionary approach that can operate in real time using only local information perceived by the real robots as they act on the environment and how these experiments can be tweaked in order to produce the desired behaviors from the teams and individual robots.

Keywords: Coevolution, Adaptation, Multi-robot Systems, Coordination.

1 Introduction

Coordination in general and within robot teams, in particular, may be formulated in terms of cooperation or collaboration. However, there seems to be a lack of consensus in the literature on the precise definition of these terms, probably resulting from the fact that there are different possible points of view for the division of labor in social learning. To make things clear, here we adhere to the definition provided by [1] in which cooperative work "... is accomplished by the division of labor among participants, as an activity where each person is responsible for a portion of the problem solving...", whereas collaboration involves the "... mutual engagement of participants in a coordinated effort to solve the problem together." As a consequence, and following [2], both cooperation and collaboration imply the distribution of the task among the individuals of the society, but they differ in how it is divided. In the case of cooperation, the task is divided into independent subtasks whereas in the case of collaboration the subtasks are interdependent and it is in the cognitive processes where the division lies. Thus, when cooperating, coordination is needed only when constructing the final result from the partial results provided by the different individuals who have usually learnt their task according to their particular utility functions, whereas collaboration is "... a coordinated, synchronous activity that is the result of a continued

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 565–574, 2010.

attempt to construct and maintain a shared conception of a problem" [1] and the only valid utility is that of the society as a whole.

Based on the previous definitions, the problem addressed here is that of collaboration, that is, obtaining teams of individuals whose interaction leads to some emergent behavior of the society. In fact, this paper describes the application of a real time co-evolutionary strategy to the production of collaborative behaviors within sets of real robots in real environments. The approach is based on Watson et al's [3] Embodied Evolution (EE) and extends the concept to provide a means for groups of robots to self-organize and perform tasks efficiently. In the original implementation of EE, the authors sought to establish a completely distributed evolutionary algorithm embodied in physical robots. Their approach was based on the hypothesis that a large number of robots could be used for the evaluation stage of an evolutionary process devoted to obtaining a controller for a particular task. Their objective was different to the one sought here as they just wanted to obtain a single controller, but their efforts did lead to a set of interesting ideas and design requirements that had to be taken into account in order to use evolution within real robots operating in real environments. For instance, the evolutionary process had to be decentralized and thus the evaluations required for the determination of the fitness of an individual should take place directly within the individual, the physical robot itself, in an embodied and localized manner.

This way of addressing the problem differs radically from other strategies found in the Evolutionary Robotics (ER) literature [4][5] where centralized evolutionary algorithms perform the process of obtaining robot controllers using information from all the robots in a simulation (or even in some cases in real robots), and usually off-line. Much effort has been devoted to the study of the generation of coordinated behaviors for groups of robots in the last few years [6][7]. Different authors such as [8] with the swarm project or [9] who present a universal architecture for the decentralized control of groups of robots have addressed the general problem. Others have concentrated on implementation issues [10][11]. However, much of the work found on this topic is particular to a task (i.e. foraging or flocking) or environment. The authors often consider homogeneous sets of robots and/or controllers and quite often only contemplate the problem of cooperation. In general, they do not provide a general framework for obtaining collaborative behaviors. Recently some work such as [12] is being carried out to characterize some of the problems in this realm. The authors study the production of different collective solutions in order to determine the most appropriate for each case. However, again, collaboration, in which what needs to be determined is not really the task decomposition and its distribution, but, rather, the interactions among the individuals in order to jointly fulfill the objectives are not usually contemplated as the problem is extremely coupled and the controllers for all the individuals must be jointly obtained.

This paper presents the application of Asynchronous Situated Co-evolution (ASiCo) as a valid embodied, distributed and adaptive strategy for groups of real robots to collaborate in an efficient manner. ASiCo is inspired by some Artificial Life based distributed evolutionary approaches but includes some ideas from the multi-agent systems literature to provide a way to implement the objectives of the collective system through the creation of energy and interaction based utility distribution schemes.

The rest of the paper is structured as follows. First a brief description of the algorithm will be provided introducing some of the basic operators that allow it to be operational and efficient in real robot within real environment situations. This is the case of embryonic based reproduction, required for the adaptation of the algorithm to real time operation on real distributed multirobot systems. Section 3 presents some results from the application of ASICO with the objective of having a set of e-puck robots self-organize and collaborate in order to perform a collective cleaning task under different circumstances, some that will lead to homogeneous (in terms of controllers) robot teams and others that will generate heterogeneous teams. Finally, some conclusions are presented.

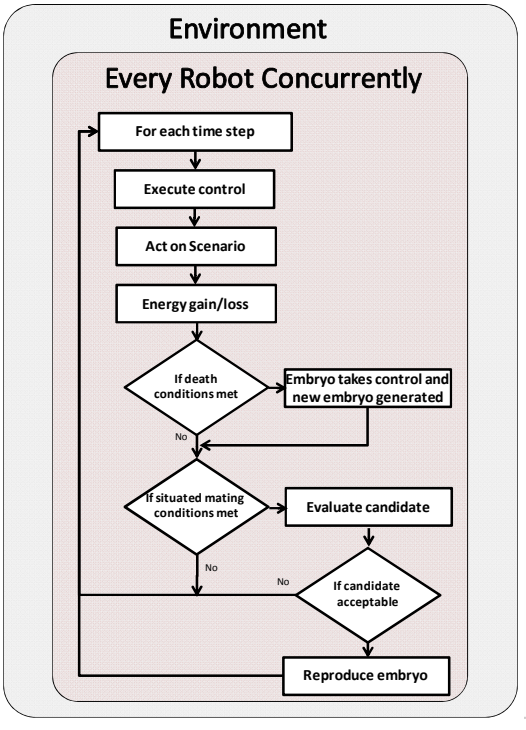
2 Overview of Asynchronous Situated Co-evolution

The ASiCo algorithm is based on a decentralized and asynchronous open-ended evolutionary approach. It is different from other evolutionary algorithms, such as genetic algorithms in which centralized selection and evaluation of the individuals based on an objective function at regular processing intervals is performed. In ASiCo, evolution is situated, meaning that all of the interactions in the population are local and depend on the spatial and temporal coincidence of the individuals. Consequently, reproduction, the creation of new individuals or their elimination is driven by events that may occur in the environment in a decentralized way.

This type of evolution has usually been employed for analysis purposes in the realm of Artificial Life, this is, to study how a system evolves in an open-ended manner, and not really with an engineering objective in mind. Therefore, no clear procedure exists to relate the global objective to be achieved with the local objectives of the agents that participate. To this end, the studies of utility functions and their distribution among individuals provide some clues in order to structure the energy dynamics of the environment to guide evolution towards the desired objectives sought. Specifically, we have considered the principled evaluation function selection procedure for evolving coordinated multirobot systems developed by Agogino and Tumer [13]. With this procedure, ASiCo open-ended evolution becomes an evolutionary optimization algorithm that provides a distributed solution through the collaboration of the whole population. For a deeper study about the validity of asynchronous situated coevolution as an alternative to classical evolution, see [14].

ASiCo is an interaction driven algorithm. Everything occurs in an environment and the environment can be simulated or real. Here as we are considering real robots operating in real environments there is no need for simulation. The basic operation of the algorithm has been schematically represented in Fig. 1 (left) together with its pseudocode (right). It works as follows:

Each robot starts with a random neural controller that covers as inputs its perceptions (either direct data from sensors or some type of processed perceptual information in the form of virtual sensor data) and provides as outputs the commands for its actuators. This is the main action controller. The robot also holds a copy of the genotype corresponding to its controller and a few other lines of code that regulate its reproductive interaction with the rest of the robots as well as its energy level. These are sort of instinctive or preset behaviors and, even though they could easily be




```
INITIALISATION() :- {
FOR EACH watcher
        Random generation of the controller_genetic_code;
        Random generation of the embryo_genetic_code;
        prequality = 0; quality = 10000; T_last_mating = 0;
}
ACT() :- {
FOR EACH watcher
        Move and cover cells in range;
        Refresh quality;
        Sense obstacles (obs) and watchers (wtc);
        Refresh trajectory memory (trm);
        Speed_difference = ArtNeuralNetwork(obs,agn,trm);
        Wheel_speed_1 = Speed_sum/2 + Speed_difference/2;
        Wheel_speed_2 = Speed_sum/2 - Speed_difference/2;
        age++;
        IF (age < maturity || quality < 1) THEN quality = 1;
        IF (closest_watcher_distance < range_meeting)
                THEN MEETING (closest_watcher);
        IF (quality < 0 || age > 1000) THEN REPLACE();
}
MEETING(watcher2) :- {
IF ((AVERAGE(quality,watcher2.quality) > prequality)
                && (T_last_mating < ActualTime - 10))
    THEN embryo_genetic_code =
        XOVER_MUTATION(genetic_code,watcher2.genetic_code);
        prequality = AVERAGE(quality,watcher2.quality);
}
REPLACE() :- {
controller_genetic_code = embryo_genetic_code;
embryo_genetic_code = MUTATION(embryo_genetic_code);
quality = prequality / 2.0; prequality /= 2.0;
}
```

Fig. 1. Schematic representation of ASiCo structure (left) and pseudocode (right)

included in the evolutionary process, for the sake of clarity here we are only going to consider the evolution of the action controller.

The survival of the controller is regulated by an energy level associated by the experiment designer with some interaction with the environment. For instance, in a cleaning scenario it can be the accumulated dirt level of the positions the robot cleans. In a social interaction scenario it may be the time the robot is capable of keeping another robot or human interested and interacting with it. This energy level increases through these interactions and decreases due to the expenditure of energy by the robot to do things.

The reproduction behavior regulates when and how the robots exchange genetic information. The basic idea is that some interaction event between robots (proximity, collision, a signal…) triggers a behavior whereby the robots evaluate their mating "desire". This desire is regulated by the energy levels of the two participants and, sometimes, by their affinity, that is, their distance in genotypic space.

It is important to avoid the instabilities caused by constant changes in the robot controllers due to possibly frequent reproduction during operation, which is a problem that arises in other strategies that have been proposed such as PGTA [3] due to a lack of evaluation time in the environment. It is also important to take into account that

this mechanism needs to be adapted to the objectives sought, and in this paper we are interested in groups of real robots with a fixed number of individuals. Consequently, as we cannot make robots "appear" in the environment and do not want real robots to "die" (actually, this case is considered in terms of robot failures), the reproduction mechanism has to contemplate the fact that the maximum number of robots is fixed and still provide a way for the population to evolve in a distributed manner. The approach followed here, is to synchronize death with birth. In fact, as the robots are preset, we can only work with their controllers and thus, a death-birth process within a robot is just a change of its controller. To allow for evolutionary pressure within this process, we have designed a reproduction mechanism for fixed size populations called Embryo Based Reproduction (EBR) [14].

The idea behind EBR is that each agent, carries, in addition to its own parameters, another set of parameters corresponding to its embryo and an associated pre-utility value for the embryo that estimates the utility of the agent generated from it. Thus, when a new agent is created, its embryo is generated as a mutation of the parent genotype with half of its energy. During the life of an agent, the embryo is modified whenever the agent meets another one and evaluates it positively; meaning that the average of the utility of the two parents is higher than the pre-utility of the current embryo and the affinity criteria is met. Finally, when the parent dies because it ran out of energy or time or for whatever other reason, the embryo substitutes the parent, that is, the control of the robotic unit is assumed by the embryo and a new embryo is generated within the robot. This way, we ensure that the size of the population remains constant and that the process takes place in an asynchronous and decentralized manner.

Summarizing, all of the robots are constantly acting in parallel following the strategies given by their action controllers. During operation they gain or lose energy according to some functions related to its interaction with the environment and its activity. When some specific interaction condition arises, usually related with spatial and temporal coincidence, a reproduction behavior is triggered and if the right conditions are present the robots exchange genetic material for their embryos. Finally, when a robot controller "dies", the embryo takes over and the process continues. It is through the regulation of the energy input/output and the reproduction conditions that a designer is able to lead this process towards the desired global objective for the robot population.

3 Experiments and Results

For the purpose of illustrating the use of this approach, this section presents a set of experiments carried out using a population of e-puck robots in a real environment. The idea of the experiments was to demonstrate the appropriateness of ASiCo as an algorithm that can operate in real time on real robots with very low computational capabilities and obtain self organized collaborative behaviors that would lead to achieving a predefined task. In this case, the task will be to mow a given grassy area or gather the leaves present there. However, depending on the scenario, the task will require different robot capabilities. In addition to demonstrating how the algorithm works, it is also an objective of this experimental section to show that the robot team

using ASiCo is capable of adapting to changing situations in the environment and that it is robust to the loss of members of the team.

There is an experimental area, which is basically a white board on which the robots move. In order to simplify the task of changing or controlling environments, we have decided to project the environment onto this white board. This is achieved through an overhead projector connected to the environmental control computer. There is also an overhead camera in order to provide feedback to the environment controller. The purpose of this setup is to be able to control the response of the environment to the actions of the different robots and thus be able to generate different scenarios in a simple way. For instance, the environment may consist in a grassy lawn and we can assume that the robots are mowing it. In this case, the environment control system just changes the color of an area from green to brownish whenever a robot goes over it and as time passes without any robot going over a particular area, the area becomes greener (more grass). It is important to note that this environment control system was generated just to make experimenting a lot simpler (it takes a long time for grass to grow back on a real lawn), but it does not affect the operation of the robots at all, as the only information they make use of is that which they can sense by themselves in the environment. The environment has no clue of which robot is which or what they are doing; it just reacts through a set of fixed rules to the actions of the robots, like any real environment.

Every robot has its local ASiCo algorithm implemented within it. That is, it carries its genotype, which has been converted into an RBF artificial network phenotype (but any other type of network could have been used) connected to its sensors and actuators and which acts as the controller, as well as the genotype of its embryo and a couple of memory elements, one for storing its current energy level and one for storing its relative position n instants ago (n is part of the genotype and it is the module and angle of the vector relating the two positions that are estimated using the odometry of the robots). It also contains the definition of how the energy level is changed. Here this is a simple algorithm whereby the robot, through its camera, perceives the color of the environment in front of it and depending on this color, adds an amount of energy (in terms of grass, the more grass it mows the larger the value it adds). As the robot moves, the energy level is depleted at a constant rate. Each robot implements its reproduction algorithm. Basically, when it detects another robot close by, it communicates with it to determine its energy level and genotype and compares it to its own. If the selection criteria, as described above, are met it takes a copy of its genotype and performs crossover and mutation with its own embryo. Finally, when the energy level of a robot goes to zero, it replaces its controller with the one that results from the embryo at that time and creates a new embryo as indicated in the previous section.

In terms of sensing, each robot can sense, through its camera, the color on the ground right in front of it, and the presence of other robots or obstacles around it using its infrared sensors. It is important to note here that the color on the ground will only be used by the robot to update its energy level and not as an input in the controller, because we are interested in obtaining collaborative or social behaviors, that is, we are interested in the society of robots obtaining a set of societal interaction rules which, through their instantiation, lead to an emergent behavior of the whole society that achieves the objective assigned to them. By knowing what is on the ground, an implicit division of the task could be achieved by following the paths of greatest

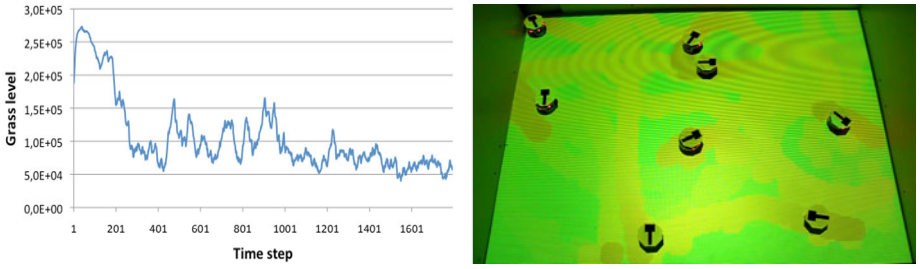


Fig. 2. Evolution of the global grass level (left) and a snapshot of the environment while the e-pucks are performing the task (right). Green intensity indicates grass level.

profit (more grass). With the current setup, the societal interaction rules are not related to those of the task. Consequently, the robot must infer the grass level from the number of robots that are placed in a given area.

The first experiment simply consists in the collaborative mowing of a grass area by a set of e-pucks. Here the grass grows fast and thus keeping the area mowed is a constant task. The global utility is the sum of the grass level in all the cells, and must be minimized. The individual utility is the sum of grass levels in the cells the robot has cleaned. This utility has been defined following the principles described in [13]. We must point out that the collective objective is a continuous mowing, that is, the robots must attend to the occupation of the environment in order to synchronize their behavior. As a consequence, this task cannot be solved through a cooperative approach requiring collaboration as defined in the introduction.

Fig. 2 left shows the evolution of the global grass level in an environment that is being patrolled by a set of 8 robots as described above. Initially it increases while the robots self-organize to perform the task. After a few interactions it decreases reaching around 0,1 of the maximum possible (around 5E+5), which is close to the minimum possible, in 1000 steps. Obviously, as there are very few robots, there are small oscillations in the performance of the behavior. In order to provide a visual indication of experimental setup, the right image of Fig. 2 shows a snapshot of the experiment. In this image intensity level represents the grass level. The fact that the whole area is mowed efficiently, as shown in Fig. 2 left, is a consequence of the emergence of a collaboration strategy forced by the global utility requirements. Another interesting result that can be extracted from the genotypes of the robots is that they tend towards a homogeneous genotype in this task. This seems reasonable given the fact that the environment is relatively homogeneous in terms of requirements for the agents.

To make things a little more complex, we now construct a second experiment with an environment where, in order to achieve the global objective, two different tasks need to be carried out. For instance, we can think that part of the area has to be mowed and part requires gathering leaves. To simulate this situation an environment was designed with two separate areas and we assume that all the robots can perform both tasks and the environment indicates one or the other through different colors. The neural controllers are modified including a new sensorial input that perceives the ground color transition, that is, if the robot crosses from one part to the other or

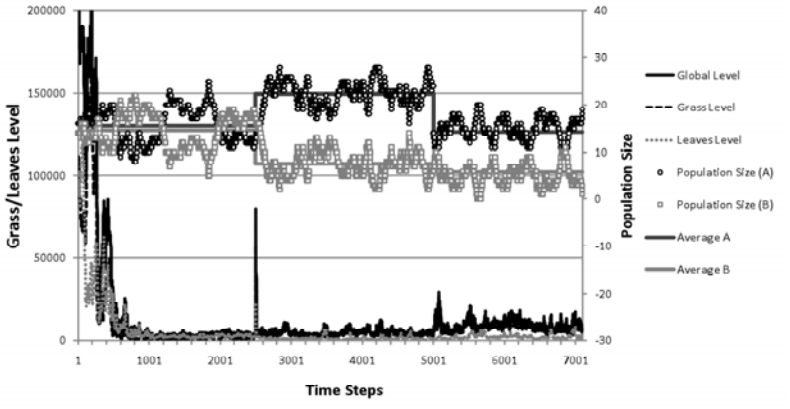


Fig. 3. Evolution of the grass/leaves levels as well as the populations in each main species for the second experiment

viceversa. Again, the ground color is perceived only to update the robot's energy level and it is not an input to the controller.

This experiment is really a concatenation of experiments, as the procedure has been to generate an environment with the two areas, each occupying 20% of the environment and when the populations had stabilized, the mowing area was increased to a 40% of the environment. After this situation was stabilized, 33% of the robots were turned off. The experiment involved 30 robots starting with random genotypes.

Fig. 3 displays the evolution curves. This figure reflects what the levels of grass and leaves are in the environment as well as their sum (the global level). It also shows the population size for each of the resulting genotypes as well as their averages. The behavior shown is very interesting and really supports the claim of ASiCo being able to provide an asynchronous localized approach for robot teams to self-organize in real time and adapt to changes in the environment and resources. Following the evolution of the graph, at the beginning the genotypes of all the robots are random and the phenotypes do not really do anything useful, they are basically moving randomly or not at all. As time passes there are some fluctuations in the population towards robots that are capable of solving either task, and consequently, there is a sharp decrease in the level in one zone or the other, but a high global level. As the robots interact, their grass mowing and leave collecting strategies as a team improve (much in the same way as in the first experiment), and the population clearly segregates into mostly two genotypes, one for one of the tasks and another for the other. After about 2000 time steps, around 15 robots are devoted to each task (the areas have the same size) and after about 3000, the grass and leave levels become very low. There are still slight changes in the population distribution due to the fact that it is a dynamic system that is always trying alternatives, but the task is performed correctly. The level obtained is an average of 2500 grass-leave units, which is quite low considering the initial level of 200000 and that the grass is continuously growing. Fig. 4 (left) displays a snapshot of the situation of the environment in terms of genotypes and area coverage in time step 1500. This figure shows two clearly differentiated species (different colors of the

Fig. 4. Final multirobot system obtained for two different cases. The left image corresponds to the result obtained when the areas have the same size and the central one when the size of the area to be mowed is increased. The colors of the robots on the computer view represent genotype and the color intensity in each area the level of grass/leaves. The right figure shows a view of the experimental setup with 8 e-puck robots.

robots) with a few individuals hopping between them. The main difference in the genotypes is a consequence of the new input that detects the ground color transition, which requires that one species attends to a transition and the other to the contrary.

In order to test the adaptability of ASiCo to changing environmental situations, in step 2500 the size of the grassy area is doubled and now it produces more grass per unit time (75% of the total grass/leave production per instant of time). As shown in Fig. 3, when the change in size of the area occurs, there is a sharp but short lived increase in the grass level, that is, the robots assigned to the now increased area are not enough to mow the area as well as before. However, in a very short time, the robot team is reorganized, with around 22 robots for this area and around 7 in the leaves area and the grass levels drop to close to the previous ones. The leaves level increases a little due to the fact that less robots are assigned to this area, and the global grass/leaves level ends at 5000. Fig. 4 right shows a snapshot of this situation.

Finally, after time step 5000, 33% of the robots are disabled, and thus, we are now working with only 20 robots. Again, initially we see a slight increase in the unmowed grass or uncollected leaves, but the population self-organizes to adapt to the new situation and 14 robots are assigned to mowing and 5 to leave collection. Obviously, the performance of the teams in this situation degrades, as there are less robots to perform the tasks, however, this decrease is not catastrophic and the levels are still acceptable at around 12000. This is especially so in the leaves area, which is a lot smaller and, thus, much easier to patrol.

4 Conclusions

This paper describes the application of Asynchronous Situated Co-evolution (ASiCo) as a general procedure for allowing real robot teams to self-organize and produce adaptive collaborative behaviors in an embodied fashion. To this end an embryonic like delayed reproduction mechanism (EBR) was proposed which leads to a very simple and low computational cost implementation for the individual robots and results in a quite effective and adaptable mechanism for the whole population. The robot controllers can evolve asynchronously in real time through the spatial and temporal coincidence of the robots generating heterogeneous populations of controllers depending on the task the whole group needs to perform as well as the number of

individuals present. These controller populations arise quite fast allowing the robot team to jointly achieve their objective adapting promptly to the new situations and objectives when the objective or environment changes and it degrades gracefully when some of the robots breakdown. The approach was implemented in an e-puck based robot team and the results obtained were quite successful.

Acknowledgments. This work was partially funded by the Xunta de Galicia through project (09DPI012166PR) and European Regional Development Funds.

References

1. Roschelle, J., Teasley, S.D.: The construction of shared knowledge in collaborative problem solving. In: O'Malley, C.E. (ed.) Computer-Supported Collaborative Learning, pp. 69–197. Springer, Berlin (1995)
2. Dillenbourg, P., Baker, M., Blaye, A., O'Malley, C.: The evolution of research on collaborative learning. In: Spada, E., Reiman, P. (eds.) Learning in Humans and Machine: Towards an interdisciplinary learning science, pp. 189–211. Elsevier, Oxford (1996)
3. Watson, R.A., Ficici, S.G., Pollack, J.B.: Embodied Evolution: Distributing an Evolutionary Algorithm in a Population of Robots. Robotics and Autonomous Systems 39(1), 1–18 (2002)
4. Harvey, I.: Artificial Evolution and Real Robots. In: Sugisaka, M. (ed.) Proceedings of the International Symposium on Artificial Life and Robotics (AROB), Beppu, Japan, pp. 138–141 (1996)
5. Mataric, M.J., Cliff, D.: Challenges in Evolving Controllers for Physical Robots. Journal of Robotics and Autonomous Systems 19(1), 67–83 (1996)
6. Szu, H., Chanyagorn, P., Hwang, W., Paulin, M., Yamakawa, T.: Collective and distributive swarm intelligence: evolutional biological survey. International Congress Series, vol. 1269, pp. 46–49. Elsevier, Amsterdam (2004)
7. Jolly, K.G., Ravindran, K.P., Vijayakumar, R., Sreerama Kumar, R.: Intelligent decision making in multi-agent robot soccer system through compounded artificial neural networks. Robotics and Autonomous Systems 55, 589–596 (2007)
8. Dorigo, M., Trianni, V., Sahin, E., Gross, R., Labella, T.H., Baldassarre, G., Nolfi, S., Deneubourg, J.-L., Mondada, F., Floreano, D., Gambardella, L.M.: Evolving self-organizing behaviors for a swarm-bot. Autonomous Robots 17(2-3), 223–245 (2004)
9. Peleg, D.: Distributed Coordination Algorithms for mobile robot swarms: new directions and challenges. In: Pal, A., Kshemkalyani, A.D., Kumar, R., Gupta, A. (eds.) IWDC 2005. LNCS, vol. 3741, pp. 1–12. Springer, Heidelberg (2005)
10. Burgard, W., Moors, M., Stachniss, C., Schneider, F.E.: Coordinated multi-robot exploration. IEEE Transactions on Robotics 21(3), 376–386 (2005)
11. Folgado, E., Rincón, M., Álvarez, J.R., Mira, J.: A Multi-robot Surveillance System Simulated in Gazebo. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2007. LNCS, vol. 4528, pp. 202–211. Springer, Heidelberg (2007)
12. Waibel, M., Keller, L., Floreano, D.: Genetic Team Composition and Level of Selection in the Evolution of Multi- Agent Systems. IEEE Transactions on Evolutionary Computation 13(3), 648–660 (2009)
13. Agogino, A., Tumer, K.: Efficient evaluation functions for evolving coordination. Evolutionary Computation 16(2), 257–288 (2008)
14. Schut, M.C., Haasdijk, E., Prieto, A.: Is situated evolution an alternative for classical evolution? In: Proceedings CEC 2009, pp. 2971–2976 (2009)

Emergence of an Internal Model in Evolving Robots Subjected to Sensory Deprivation

Onofrio Gigliotta[1,2], Giovanni Pezzulo[3], and Stefano Nolfi[2]

[1] Department of Relational Sciences, University of Naples Federico II
Via Porta di Massa, 1 80113 Naples, Italy
[2] Istituto di Scienze e Tecnologie della Cognizione - CNR
Via S.Martino della Battaglia, 44 - 00185 Rome, Italy
[3] Istituto di Linguistica Computazionale "Antonio Zampolli" - CNR
Via Giuseppe Moruzzi, 1 - 56124 Pisa, Italy
{Onofrio.Gigliotta,Giovanni.Pezzulo,Stefano.Nolfi}@istc.cnr.it

Abstract. In this study we show how simulated robots evolved to display a navigation skills can spontaneously develop an *internal model* and rely on it to complete their task when sensory stimulation is temporarily unavailable. The analysis of some of the best evolved agents indicates that their internal model operates by anticipating functional properties of the next sensory state rather than the exact state that sensors would have assumed. The characteristics of the states that are anticipated and of the sensory-motor rules that determine how the agents react to the experienced states, however, ensure that the agents produce very similar behaviour during normal and blind phases in which sensory stimulation is available or is self-generated by the agent itself, respectively. The characteristics of the agents' internal models also ensure an effective transition during the phases in which agents' internal dynamics is decoupled and re-coupled with the sensory-motor flow.

1 Introduction

The idea that cognitive agents act on the basis of internal models of their tasks rather than purely on the basis of the stimuli they receive from the external environment can be considered fundamental in cognitive science [1],[2],[3]. The structure and functioning of internal models is however much more debated.

Traditional theories in cognitive science describe internal models as symbolic mental structures that support higher level cognition and whose representational content is conceptual and is not tied to any sensorimotor modality. The de-emphasization of symbolic representations in cognitive science, and particularly cognitive robotics, has resulted in decreased attention to internal models in favor of a non-representational view [4]. Recently, however, the idea of internal modeling is gaining consensus anew, as numerous researchers in cognitive psychology, neuroscience, and robotics are increasingly reusing ideas originating from the domain of motor control [5,6] into more cognitive domains, therefore reintegrating the idea of internal modeling and representation in an "embodied", "motor" view of cognition [7,8,9,10,11,12,13].

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 575–586, 2010.

Internal models come in (at least) two varieties: *inverse models* and *forward models*. The former compute the necessary motor commands to achieve a certain goal given a starting condition, and the latter predict the sensory consequences of those motor commands.

Fig. 1 highlights the differences between (a) a stimulus-response system, and (b) one endowed with (multiple pairs of) internal, forward and inverse models, which is inspired by the architecture for motor control described in [14]. In the latter, the internal models (inverse and forward) realize an *inner loop*, which parallels actual sensorimotor interaction and mimics its input-output properties. Such loops can function on-line with action (b), or off-line (c), that is, *detached* from the current sensorimotor context. When this last condition holds, sensory inputs are substituted by predicted inputs, and motor outputs are inhibited. This last mode of functioning permits chaining multiple predictions (in principle, for an arbitrarily long number of steps) so to realize long-term lookahead predictions, or "mental simulations" [10,11].

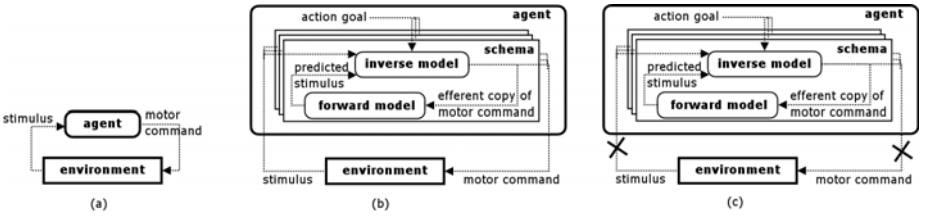


Fig. 1. Comparison between purely stimulus-response systems (b) and those endowed with anticipatory capabilities, which run an "internal loop" on-line with action (b), or off-line (c)

In other words, since internal models support the anticipation of action consequences, they can be used as "inner worlds" to try out actions, such as walking or reaching, internally rather than in the external environment. This novel view of internal modeling, which incorporates control-theoretic ideas and an embodied view of cognition, is clearly synthesized in the *emulation theory of representation* proposed by [9, p. 1]:

> in addition to simply engaging with the body and environment, the brain constructs neural circuits that act as models of the body and environment. During overt sensorimotor engagement, these models are driven by efference copies in parallel with the body and environment, in order to provide expectations of the sensory feedback, and to enhance and process sensory information. These models can also be run off-line in order to produce imagery, estimate outcomes of different actions, and evaluate and develop motor plans.

In this paper we investigate whether internal modeling could spontaneously arise in living organisms for the sake of effective motor control. More specifically, in

this paper we investigate whether artificial embodied agents, that are trained for the ability to exhibit a given behavioral skill, develop and use an internal model that allows them to anticipate forthcoming stimuli to overcome the problems caused by the fact that sensory stimulation is temporarily missing.

From a computational point of view, artificial organisms faced with a navigation problem in the presence of sensory stimuli will most likely develop a reactive, stimuli-based behavioral strategy (which can involve dynamical aspects, see later). The rationale behind our task design is that, when the environmental conditions change so that stimuli become temporarily unavailable, the artificial organisms have two options: either developing brand new behavioral strategies for dealing with the absence of stimuli, or learning to self-generate stimuli, so to reuse their already established behavioral strategy. Our study then aims to verify if this second option actually happens during neural evolution (and with which frequency) and if the ability to self-generate stimuli could create the adaptive conditions for the development of an internal model in an embodied and situated agent even in absence of any explicit reward for prediction. Note indeed that the simple self-generation of stimuli is not a guarantee that an internal model has been developed. Indeed, we are interested in differentiating the case of (self-) triggering of stored motor routines from the case of self-sustaining behavior through an on-line prediction of action effects—only the latter being, in our definition, an instance of internal modeling.

The fact that biological organisms can overcome the problem caused by the temporal lack of sensory information has been demonstrated, for example, in the experiment carried out in [15]. In this work a group of blindfolded human subjects were asked to perform a series of task (e.g. walking to a given marked location, avoiding obstacles, and throwing objects toward different location of the room) after having been asked to observe the room in which they were located and to direct their attention toward specific objects and markers. The fact that the subjects were able to accomplish these tasks rather well and almost as accurately with respect to a control situation in which they were not blindfolded clearly indicates that they are able to compensate the lack of visual information through some form of internal process, for example through an internal model that allows them to generate the required information by internally anticipating the consequence of their actions.

In a series of studies, Ziemke and collaborators have attempted to verify whether an artificial agent trained for the ability to accomplish a given task in normal and blind conditions could manage to overcome the problems caused by the lack of sensory information [16,17]. In the first work, the authors evolved a population of simulated wheeled robots for the ability to move along a square corridor in normal and blind conditions. The robots' sensors included only a linear camera able to detect four visual landmarks located at the four corresponding edges of the corridor itself. The analysis of the results obtained in this study demonstrates that, in some cases, the evolved agents display an ability to keep navigating within the environment also during blind phases. The analysis of one of the best individual indicates that the behaviour produced by the agents during blind phases always converges

on a sequence of actions that is very similar to those that are produced during normal phase (in a control conditions in which collision with walls are disabled). During the initial phase in which sensory stimulation is no longer available, however, the behaviour produced by the robot might differ significantly from the behaviour that is exhibited in normal conditions in the same circumstances and might thus lead to collisions between the robot and the walls. In other words, the best evolved individuals often fail to appropriately handle the transition between normal and blind phases. Moreover, contrary to the expectations of the authors, the evolved agents did not rely on an internal model or an ability to internally generate the simulated experience of the stimuli that are temporarily missing. The lack of sensory information in fact was not compensated by an ability to internally generate states that are identical or similar to those that would have been experienced in normal conditions but rather through the development of two different strategies that are executed depending on whether sensory information is available. Indeed, during normal phases the robot accomplished the task by moving forward while turning slightly toward right when the robot visually detecting a landmark and by turning right otherwise. The former elementary behaviour allows the robot to lose visual contact with the landmark toward the end of each corridor and then trigger the latter behaviour (as soon as the landmark is no longer in sight). The latter elementary behaviour allows the robot to negotiate the corner and then trigger the former behaviour (as soon as the robot visually detects the next landmark). During blind phases, instead, the robot solved the problem by executing the same two elementary behaviours described above for a certain time duration (approximately 30 and 5 time steps, respectively) by keeping track of the time spent executing the current behaviour in their internal neurons and by switching behaviour as soon as the appropriate time duration was reached. This study is particularly interesting since it indicates that stimulus prediction and internal modeling strategies do not spontaneously evolve by just forcing the system to act blindfold.

Therefore, another goal of paper is to identify the conditions (i.e. the characteristics of the task/environment and the agent control system) that represent a pre-requisite for the emergence of such an internal model. So far the idea of internal modeling has been mainly explored in a control-theoretic perspective, and numerous a-priori assumptions have been made such as the fixed time span of prediction, the specific arrangement of mechanisms (for instance, a comparison mechanism that "matches" real and predicted feedback so to calculate prediction error), and the fact that sensory predictions should be extremely close to "real" sensory input. On the contrary, our study employs a much simpler neural architecture where minimal design constraints were introduced, with the aim to analyze the specific solutions found by the evolutionary algorithm to answer basic questions such as what exactly is predicted in the internal forward models, what is the time scale of prediction, how accurate the predictions should be to be advantageous for an agent, to what extent the internal model can compensate the lack of sensory stimulation, etc. (see [18,19,20,21,22,23] for related studies using various computational modeling techniques).

Finally, from a technological perspective, our study aims to develop a methodology that can be used to synthesize artificial embodied agents (robots) able to operate effectively in uncertain conditions.

2 Methods and Scenario

To achieve the objectives described in the previous section, we set up an experimental scenario in which an embodied and situated agent should develop an ability to display a simple behaviour and keep producing it also when the sensory information is temporarily missing.

The agent consists of a simulated eye provided with a single photoreceptor located in front of a 500x500 pixel image generated by the combination of a blue and red gradient ranging continuosly from 0 to 255 along each axis (see fig. 2, left). Each time step, the photoreceptor detects the intensity of the blue and red in the pixel corresponding to the current position of the eye. The agent is also provided with two motors that allow it to move left-right and/or top-down, with respect to its current position, up to a maximum of ± 5 pixels along each axis.

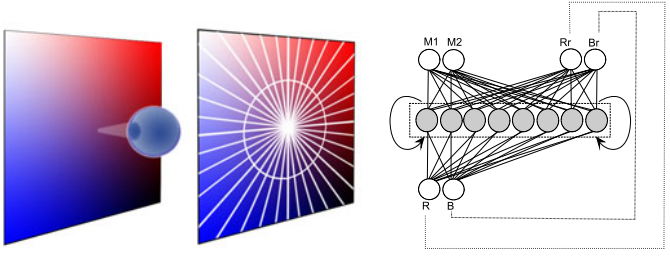


Fig. 2. Left: The environment consists of a screen displaying an image composed by the combination of a blue and red gradient distributed along the left-right and bottom-up axis. Center: The image has been divided into 36 sectors. Right: The architecture of agents' neural controller.

The task of the agent is to navigate on the image by turning around the center of the image at a distance of at least 130 pixels. For the purpose of measuring the agent's ability to exhibit such behaviour, the image has been ideally divided into 36 sectors located around its center (see fig. 2, center).

The agent's controller consists of an artificial neural network (see fig. 2, right) with two sensory neurons, eight internal neurons, two motor neurons, and two additional internal neurons (Rr and Br) that are used to replace the state of the sensory neurons when visual information is missing. The internal neurons receive connections from the sensory neurons and from themselves. The motor neurons receive connection from the internal neurons. The two sensory neurons (B and R) encode the intensity of blue and red colour currently perceived by the photoreceptor of the eye. The two motor neurons (M1, M2) determine the

amplitude of the eye movement along the left-right and top-down dimension within a range of [-5,5] pixels.

Internal neurons are leaky integrators (i.e. neurons that hold a certain amount of the previous activation)[24]. The internal and motor neurons are updated on the basis of a standard logistic function.

The architecture of the neural network is fixed. The connection weights and biases and the time constant of the internal neurons are encoded in free parameters and evolved [25]. The initial population consists of 100 randomly generated genotypes which encode the free parameters of 100 corresponding individuals. Each parameter is coded with 8 bits and is normalized in the interval [-5.0, +5.0] for the biases and the synaptic weights, and in the interval [0.0, 1.0] for the time constants. Each subsequent population is obtained by selecting the best 20 individuals of the previous population. Each selected individual is allowed to produce 5 offspring that are generated by duplicating the genotype of the reproducing individuals and by applying mutations (with 2% probability of flipping a bit).

Each individual is tested for 20 trials. At the beginning of each trial the eye is placed randomly in one of ten possible positions around the center of the image. The agent is then allowed to interact with the environment up to 4000 time steps. For each time step, the state of the agent's sensory neurons is updated on the basis of the current position of the eye, the state of the internal and motor neurons is updated, and the agent's eye is moved on the basis of the current state of the motor neurons. The agent experiences a succession of phases in which sensory information is available (normal phases), and phases in which it is missing (blind phases), according to the following rules. During the first half of each trial (i.e. during the first 2000 time steps) the agent always has access to the sensory stimulation coming from the environment (normal phase). During the second half of the trial, instead, the agent experiences a phase in which the sensory information is available (normal phase) followed by a phase in which sensory information is unavailable (blind phase), and vice versa. The length of both phases varies linearly during the 20 trials so to expose the robot to a progressively larger amount of sensory deprivation. On average, the percentage of sensory deprivation during the second half of the trial is 16%. During all the normal phases, the state of the two sensory neurons is set on the basis of the colour of the current portion of the image perceived by the agent, otherwise is replaced using Rr and Br outputs.

The performance (fitness) of the individual has been evaluated by computing the number of subsequent sectors of the image visited by the eye (see fig. 2, left). In particular, for each new visited sector(i.e. when current sector is different from the previous) the fitness F of the individuals is updated by adding dF:

$$dF = \begin{cases} \frac{1-\frac{|D_t-130|}{100}}{36} & \text{if } 30 \leq D_t \leq 230 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where D_t represents the distance between the point of the image observed by the agent at time t from the center of the image and 36 corresponds to the number of

sectors. Trials are terminated before the limit of 4000 time steps when the agent move in the wrong direction so to visit a sector already visited recently. The total performance of an individual is obtained by averaging the performance obtained during the 20 trials. The evolutionary process is continued for 1600 generations and replicated 40 times with randomly generated initial conditions.

3 Results

By analysing the behaviour of evolved individuals we observed that in 17 out of 40 replications of the experiment, the best individual succeeded in circling around the center of the image both in normal and blind phases. These individuals manage to compensate the lack of sensory information by self-sustaining their internal dynamics in two substantially different ways.

Agents belonging to the first "family" (13 out of 17) solve the problem by developing two qualitatively different strategies for normal and blind phases, and trigger the first or the second strategies during the two corresponding phases. Interestingly, although almost all these agents anticipate incoming stimuli during the normal phases with their neurons Rr and Br, their dynamics are different during the blind phases (like in [17], see below).

Agents belonging to the second "family" (4 out of 17), instead, keep reacting to the experienced sensory states in similar ways during normal and blind phases and compensate the lack of sensory information with the self-generation of equivalent information and by anticipating how the state of the sensors would vary as a result of the execution of the planned action. That is, the agents use a predictive strategy based on internal modeling.

Now we will discuss in more detail the nature of the solution evolved by the second family of succesful individuals. By observing the behaviour displayed by the best individual belonging to the second (internal modelling) family during

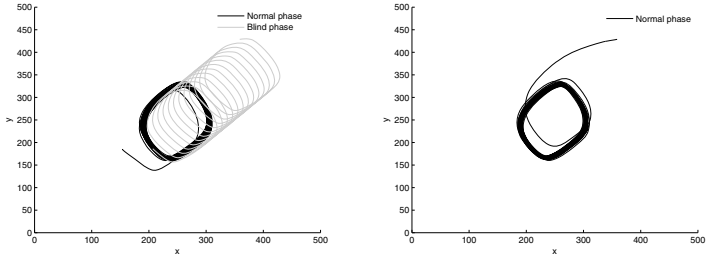


Fig. 3. Behaviour displayed by the best individual of the second family during a test in which the blind phase last 1000 time steps. Left, trajectory produced during the initial normal phase lasting 2000 time steps and during the succeeding blind phase lasting 1000 time steps (dark and light lines, respectively). Right: trajectory produced during the last 1000 time step in which the agent has access again to stimulation coming from the external environment.

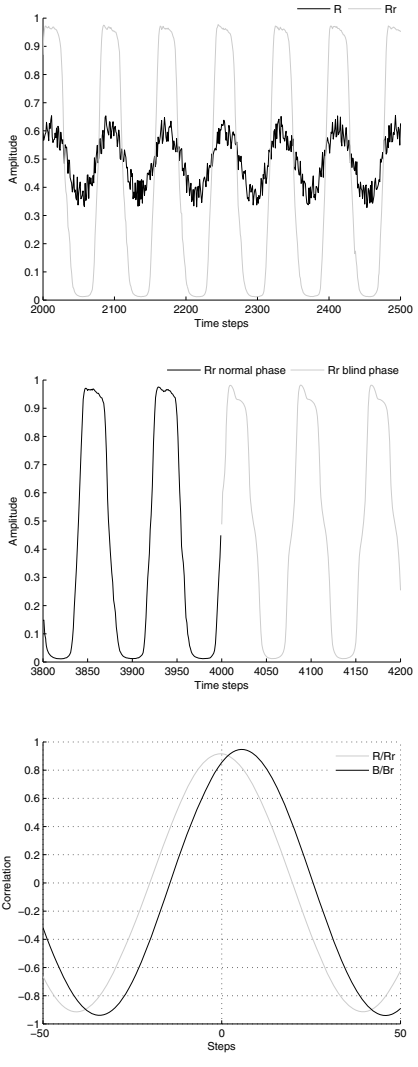


Fig. 4. Top: State of the sensors and self-generated outputs during a normal phase. The dark and left lines indicate the state of neurons R and Rr, respectively. Similar results were obtained for neurons B and Br (not shown). Center: State of outputs neurons over time. The dashed and dotted lines indicate the state of the neuron Rr during a normal and blind phase, respectively. Similar results were obtained for neuron Br (not show). Bottom: Cross-correlation over time between the states of the two output neurons and the states of the two corresponding sensory neurons. Analysis performed on the data collected for 1000 time steps during normal phase. The position of the peek along the x-axis indicates the extent of the anticipation (for value below 0) or of the delay (for value above 0) of the state variation of the output neurons at time t with respect to corresponding sensory neuron at time t+1. The peak is at -1, indicating anticipation of 1 time step.

a test in which the agent experiences a blind phase lasting 1000 time step after the first half of the trial (fig. 3, left), we can see how, during the blind phase, the agent keeps producing the same quasi-circular trajectory while slowly drifting toward the top-right part of the image (this process continues until sensors R and B are recovered). Moreover, by observing the trajectory produced by the agent during the successive normal phase (fig. 3, right), we can see how the agent manage to quickly recover from the drift as soon as the sensory stimulation become available again. These results demonstrate that the agent succeed in mastering also blind phases in which sensory information is temporarily missing. Moreover, the obtained results suggests that the lack of sensory information can be tolerated only for a limited amount of time since small differences between the behaviour produced in normal and blind conditions tend to cumulate over time during blind phases. Finally, this analysis shows that the agent is able to handle the transitions between normal and blind phases and vice versa by continuing to produce the desired behaviour.

The fact that the state of sensory neurons (R and B) at time t+1 and the state of the additional internal neurons (Rr and Br) at time t differ significantly during blind phases (fig. 4, top) indicates that the agent does not operate by predicting the exact state that the former neurons would assume at time t+1 in a normal condition. However, the cross-correlation analysis of the state of the two sets of neurons indicates that the variations of the output neuron at time t are in phase with the variations of sensory neurons at time t+1, during blind phases (fig. 4, bottom). In other words, it anticipates a property of how sensory states vary over time. It is worth noting that this is sufficient for behaving adaptively. Indeed, that despite the differences in the input signal profiles, the output signal profiles are very close in the two phases. The comparison of the dynamic of variation of the output neurons (Rr and Br) during a normal and a blind phase (fig. 4, center) indicates that the state of the neurons vary over time in a rather similar way independently from the fact that the sensory neurons are fed with actual data collected from the environment or with self-generated data, despite the two data differ significantly.

4 Discussion

Our study shows that a simulated robot trained with a genetic algorithm in a navigation task can develop an *internal model of the robot/environmental interaction*, and rely on it to fulfill the same task adaptively even when the robot is temporarily "blindfolded". The robot's internal model has several key characteristics: (1) It is *autonomously developed* depending on the demands of the agent-environment interactions rather than externally designed (the robot is not rewarded for anticipating the next sensory states or the way in which sensory states will vary). (2) It is primarily driven by its own dynamic properties, and can be triggered by both external and internal, self-generated inputs. That is, it is *self-sustained*, in the sense that the agent can endogenously (re)generate it by using self-produced rather than external stimuli, and "detachable" from the

sensorimotor loop. (3) It is of an *anticipatory nature*, since it correlates with future stimuli more than its past or present stimuli, and can be self-sustained while "real" sensory stimuli are missing.

5 Conclusions

Theoretical studies suggest that internal models could have originated in living organisms for the sake of adaptive behavior, not for cognition, and were therefore exapted for advanced cognitive and social operations [9,12]. Unfortunately, little effort has been devoted to the verification of this hypothesis–something that is admittedly very complicated to do empirically, but is more feasible by using the methodology of *evolutionary robotics*, which we adopted in this study.

The central hypothesis that motivated our design methodology is that a (temporary) deprivation of external stimuli can make it favorable, from an evolutionary perspective, the development of a robot's internal model even in the absence of any explicit reward for prediction. Indeed, once the robot has learned a reliable behavioral strategy and an associated dynamical representation of its task, it could be favorable to maintain the same strategy, and at the same time learn to self-maintain the same dynamics via self-generated inputs, rather than evolving two separate strategies to deal with the presence or absence of external stimuli.

Our study is part of a more general initiative in cognitive science that aims to draw a naturalized, embodied view of cognition by tracing it back to sensorimotor learning and motor control, some of which maintaining a representational perspective [9,12], and some others not [26].

Most studies mentioning internal modeling tend to frame the problem in control-theoretic terms; for example, [9] describes internal modeling loops in terms of Kalman filters. On the contrary, in our experiment we make fewer a-priori assumptions, for example about prediction and its time-scale, or the similarity between external and self-generated stimuli. By analyzing the best architectures selected by neural evolution, we observe that certain characteristics of the evolved internal models are actually close to abstract control-theoretic models; for example, one-step predictions emerge under appropriate environmental conditions. At the same time, the evolved internal models have certain characteristics that can hardly be studied from an a-priori perspective; for example, as illustrated in fig. 4, self-generated stimuli are different in amplitude and more regular than external stimuli. Overall, our experiments can be considered a further step in the clarification of this novel and multifaceted view of embodied cognition.

Acknowledgements

The research leading to these results has received funding from the Europeans Community 7^{th} Framework Programme under grant agreements ITALK (ICT-214668) and HUMANOBS (ICT-231453).

References

1. Craik, K.: The Nature of Explanation. Cambridge University Press, Cambridge (1943)
2. Tolman, E.C.: Cognitive maps in rats and men. Psychological Review 55, 189–208 (1948)
3. Johnson-Laird, P.: Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness. Cambridge University Press/Harvard University Press, Cambridge (1983)
4. Brooks, R.A.: Intelligence without representation. Artificial Intelligence 47(47), 139–159 (1991)
5. Kawato, M.: Internal models for motor control and trajectory planning. Current Opinion in Neurobiology 9, 718–727 (1999)
6. Wolpert, D.M., Gharamani, Z., Jordan, M.: An internal model for sensorimotor integration. Science 269, 1179–1182 (1995)
7. Clark, A., Grush, R.: Towards a cognitive robotics. Adaptive Behavior 7(1), 5–16 (1999)
8. Frith, C.: Making up the Mind. In: How the Brain Creates our Mental World. Blackwell, Malden (2007)
9. Grush, R.: The emulation theory of representation: motor control, imagery, and perception. Behavioral and Brain Sciences 27(3), 377–396 (2004)
10. Jeannerod, M.: Neural simulation of action: A unifying mechanism for motor cognition. NeuroImage 14, S103–S109 (2001)
11. Jeannerod, M.: Motor Cognition. Oxford University Press, Oxford (2006)
12. Pezzulo, G., Castelfranchi, C.: The symbol detachment problem. Cognitive Processing 8(2), 115–131 (2007)
13. Wolpert, D.M., Doya, K., Kawato, M.: A unifying computational framework for motor control and social interaction. Philos. Trans. R. Soc. Lond. B. Biol. Sci. 358(1431), 593–602 (2003)
14. Wolpert, D., Miall, C., Kawato, M.: Internal models in the cerebellum. Trends Cogn. Sci. 2, 338–347 (1998)
15. Lee, D.N., Thompson, J.A.: Vision in action: the control of locomotion. In: Ingle, D.J., Goodale, M.A., Manfield, R.J.W. (eds.) Analysis of Visual Behavior, pp. 411–433. MIT, Cambridge (1982)
16. Ziemke, T., Jirenhed, D.A., Hesslow, G.: Blind adaptive behavior based on internal simulation of perception. Technical Report HS-IDA-TR-02-001, Department of Computer Science (School of Humanities & Informatics), University of Skovde, Sweden (2002)
17. Ziemke, T., Jirenhed, D.A., Hesslow, G.: Internal simulation of perception: a minimal neuro-robotic model. Neurocomputing 68, 85–104 (2005)
18. Alnajjar, F., Hafiz, A.R., Zin, I.B.M., Murase, K.: Vision-motor abstraction toward robot cognition. In: Leung, C., Chan, J. (eds.) ICONIP 2009, Part II. LNCS, vol. 5864, pp. 65–74. Springer, Heidelberg (2009)
19. Johnsson, M., Balkenius, C., Hesslow, G.: Neural network architecture for cross-modal activation and perceptual sequences. Papers from the AAAI Fall Symposium (Biologically Inspired Cognitive Architectures), Arlington, Virginia, USA, pp. 85–86 (2009)
20. Linker, F., Niklasson, L.: Extraction and inversion of abstract sensory flow representations. In: Proceedings of the Sixth international Conference on Simulation of Adaptive Behavior, From Animals to Animates, vol. 6, pp. 199–208. MIT Press, Cambridge (2000)

21. Manoonpong, P., Wrgtter, F.: Efference copies in neural control of dynamic biped walking. Robotics and Autonomous Systems 57, 1140–1153 (2009)
22. Pezzulo, G.: A study of off-line uses of anticipation. In: Asada, M., Tani, J., Hallam, J., Meyer, J.A. (eds.) SAB 2008. LNCS (LNAI), vol. 5040, pp. 372–382. Springer, Heidelberg (2008)
23. Svensson, H., Morse, A., Ziemke, T.: Representation as internal simulation: A minimalistic robotic model. In: COGSCI (2009)
24. Gigliotta, O., Nolfi, S.: On the coupling between agent internal and agent/ environmental dynamics: Development of spatial representations in evolving autonomous robots. Adaptive Behavior 16(2-3), 148–165 (2008)
25. Nolfi, S., Floreano, D.: Evolutionary Robotics. MIT Press, Cambridge (2000)
26. Hesslow, G.: Conscious thought as simulation of behaviour and perception. Trends in Cognitive Sciences 6, 242–247 (2002)

Emergent Distribution of Computational Workload in the Evolution of an Undulatory Animat

Ben Jones[1], Yaochu Jin[2], Bernhard Sendhoff[2], and Xin Yao[1]

[1] School of Computer Science, The University of Birmingham, UK
[2] Honda Research Institute Europe GmbH, DE

Abstract. The coupling between an agent's body and its nervous system ensures that optimal behaviour generation can be undertaken in a specific niche. Depending on this coupling, nervous system or body plan architecture can partake in more or less of the behaviour. We will refer to this as the automatic distribution of computational workload. It is automatic since the coupling is evolved and not pre-specified. In order to investigate this further, we attempt to identify how, in models of undulatory fish, the coupling between body plan morphology and nervous system architecture should emerge in several constrained experimental setups. It is found that neural circuitry emerges minimalistically in all cases and that when certain body segmentation features are not coevolved, the agents exhibit higher levels of neural activity. On account of this, it is suggested that an unconstrained body plan morphology permits greater flexibility in the agent's ability to generate behaviour, whilst, if the body plan *is* constrained, flexibility is reduced with the result that the nervous system has to compensate.

1 Introduction

Animal behaviour is largely shaped by the coupling existing between nervous system and body plan morphology. If the coupling is optimal, the animal's body can interact appropriately with its environment and the animal survives; evolution becomes favourable. The process towards optimal coupling might go as follows. A progressive change in the environment causes a progressive change in the animal's body morphology and given the coupled nervous system, results in a commensurately progressive alteration of the nervous system architecture. Specifically, the coupling becomes driven by a need to optimise the distribution of computational workload existing between the body plan morphology and the nervous system architecture. In this paper, we hypothesise that in artificial fish-like agents situated *in silico*, neural circuitry will become more important when flexibility in the body plan morphology is constrained. We base this on the idea of *morphological computation* [13,3], that, due to the passive dynamics of the body plan morphology, the neural circuitry has to play more or less of a role in the computational process. This is highly applicable to aquatic agents given how the size and shape of the fish body are known to have a significant impact on behaviour generation [11,6]. We are in good stead to model fish-like agents given the abundance of research before us. We are inspired by Ekeberg [4,5] and Ijspeert [7,8] both of whom modelled undulatory fish such as lamprey, with results comparable to biophysical data with the latter also incorporating evolutionary processes; also, the work of Karl Sims [16] which was seminal in the study of embodied cognition.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 587–596, 2010.

This paper outlines a framework that can be used to explore the processes by which body and nervous system become coupled during a coevolutionary process. With a novel neural control system that is geometrically constrained, behaviour of the agent (that of swimming) is ultimately determined by how the nervous system architecture and body morphology become coupled together. The remainder of this paper is set out as follows. Section 2 introduces the model of an undulatory fish-like agent used throughout. The experimental setup is outlined in Section 3. Results are presented in Section 4. Discussion and conclusions are then provided in Section 5.

2 Simulation System

The agent is modeled on a three dimensional spring mesh, see Fig. 1 and has a geometrically constrained nervous system consisting of the following types of neuron.

Motor neuron. Within a given body segment, each of 4 motor neurons actuates two of the face's outer springs, see Figs. 1 and 2. The level of force applied during this compression process is proportional to the motor neuron's membrane potential and for reasons of stability, normalised to within the range [0, 120]. With sufficiently coordinated motor dynamics, the agent is endowed with the ability to locomote. Architecturally, a motor is centrally fixed to one of the four outer cuboid faces making up the body segment (i.e. it lies 'flat' on a given face).

Sensory neuron. Four sensory neurons serve as additional 'input' neurons. They are fixed to the top-middle of each of the head segment's four cuboidal faces. Computationally, the input current for a given sensory neuron is proportional to the angle of the target from the sensor, along the plane that runs perpendicular to the edge of the segment face that the sensor belongs to.

Interneuron. Additional neuron units residing within each body segment. The Euclidean locations of these interneurons are evolved through a process of simulated evolution and this is the crucial way in which we evolve the architecture of the nervous system. A given neuron can never move beyond a segment's architectural bounds.

2.1 Agent Body Plan Morphology

Inherent segmentation. The animat agent is inherently segmented, see Fig. 1. This is both for implementational convenience and for the fact that in all but the simplest organisms, segmentation is argued to be generic in metazoan organisation [12]. In the model, the number of segments and the length of each are optimized during a process of simulated evolution (thus the genotype is variably lengthed). The length of a body segment is a fraction of the total length of the agent, but this is then tuned by an evolved 'length factor' parameter, F_i. Note that this factor then also affects the neural distribution by modifying the y coordinate value of each segmental interneuron by proceeding with $n_y \leftarrow n_y \cdot F_i$.

Body symmetry. An abstraction of body symmetry is also incorporated into the model. Each body segment has 4 motor neurons, residing on the ventral, dorsal, left and right sides of the agent. Since they are fixed in place, they are considered part of the body morphology. The ability for a given motor neuron to take part in movement generation is established by an evolvable boolean parameter. If both left and right motors evolve to take part in movement generation, then the motor symmetry is considered bilaterally symmetric. Note that the emergence of symmetry is not considered in this paper. Please see [10,9] for an investigation.

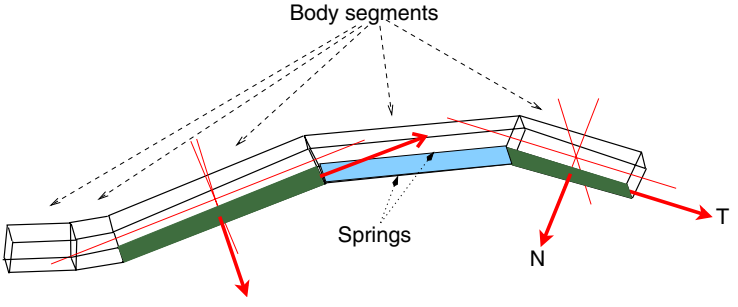


Fig. 1. Agent geometry showing construction out of of body segments. A 'compass system' is used for deriving the directional components of opposing water forces; example vector components are highlighted with red arrows. 'T' represents the tangent component and 'N' represents the normal component. A contracting segment face together with constituent springs has been further highlighted.

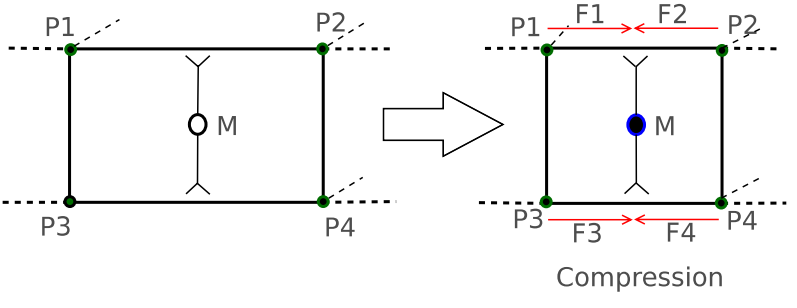


Fig. 2. Motor force system: the effect of a motor M in the centre of one of the segment faces contracting spring P1⟺P2 by applying equal but opposite forces F1 and F2 to point masses P1 and P2 and contracting spring P3⟺P4 by applying equal but opposite forces F3 and F4 to point masses P3 and P4.

2.2 Agent Nervous System

An analogue model. We use a continuous time recurrent neural network (CTRNN) to model the agent's nervous system. In comparison to a spiking neural network, a CTRNN is able to approximate the smooth dynamics required for adequate swimming behaviour. For a given neuron, the membrane potential u_j is modelled as follows, [2]:

$$\frac{du_j}{dt} = \frac{1}{\tau_j}\left(-u_j + \sum_{i=1}^{C} w_{ji}a_i + I_j\right) \tag{1}$$

where τ_j is a time constant, w_{ji} is a vector of presynaptic connection weights and I_j is an external input current. The value a_i is a presynaptic neuron's membrane activity computed as $\tanh(u_i - \beta_i)$ where β_j is a bias value. If a neuron is inhibitory then all of its outgoing connection weights are made to be negative.

Neuron positioning and synaptic efficacy. In our model, the distance between a pair of neurons (*i* and *j*) determines the connection strength of the connection between them, if such a connection exists. The formula $w_{ij} = \xi/d_{ij}$ is used to derive this value where d_{ij} is the Euclidean distance between neurons *i* and *j* and ξ has been empirically set to 20.0 to ensure that a suitable range of weights can be generated. The maximum bounds of d_{ij} are given by the geometrical properties of the body segment. Given the preceding formula, neurons have a higher connection weight if they are closer together. This is in consideration of synaptic delay which is not explicitly accounted for in the model but is known to have an effect on synaptic efficacy.

Neuron positioning and connectivity. A second aspect of our nervous system model is that interconnectivity also comes about as a function of neuronal Euclidean distance, using the following sigmoidal distribution,

$$\sigma(\lambda, s, d_{ij}) = \frac{2}{2 + \exp((\lambda/s) * d_{ij})} \tag{2}$$

where λ is an evolved parameter, s is a scaling parameter, which is set to the total length of the agent, and d_{ij} is the euclidean distance between neurons *i* and *j*. If the function generates a value >0.5 then a connection is established; also, the higher a given λ parameter, the closer a pair of neurons have to be to become connected. A set of four λ values is employed each of which caters for one type of connection: λ_{II}, λ_{IE}, λ_{SE}, λ_{AA} where I=interneuron, E=effector neuron, S=sensory neuron; AA indicates connections between interneurons in adjacent segments.

2.3 Simulation Environment

Virtual water. The environment model implements the effect of water drag. External 'water force' pushes against each face of an animat segment, the magnitude of which depends on the opposing force of the face *pushing back*. It is sufficient to use the face's velocity as an approximate measure of this 'pushing-back' force. The face velocity is taken to be the average over all four constituent point mass velocities (one at each corner of the face). The approach is simple, reliable and efficient in its implementation and has been pursued by others (e.g. [15]). Implementation wise, the face's velocity vector is initially split into two velocity components, the tangent component and the normal component ($t = \hat{\mathbf{t}} \cdot \mathbf{v}$;$n = \hat{\mathbf{n}} \cdot \mathbf{v}$). These are represented by 'T' and 'N' in Fig. 1. The actual water force, $\mathbf{w}$, to be applied to each of the four point masses making up the face is then simply calculated as $\mathbf{w} = \mathbf{f}cdA$ where $\mathbf{f}$ is a force vector derived from the

normal and tangent velocity components, c is a viscosity coefficient, d is drag, and A is the area of the segment face. The γ parameters are set to 1.0 and 0.9 for the normal and tangent components respectively; c and d are further set to 1.

2.4 Evolutionary Process

An evolutionary algorithm that harnesses both the power of adaptive mutation and local selection is used to evolve all or part the genotype illustrated in Fig. 3. Note that many of the gene groupings (e.g. the number of segment length factor genes) depend on the number of body segments, i.e. the genotype is variably lengthed; to account for this, when a segment is added / lost, affected gene groupings are added (with random initialisation) / removed. A local selection process of the type described in [14] is employed, since, (a) preliminary investigation showed it to have far better convergence; (b) within the field of Neuroevolution, a similar strategy has on prior occasion been employed to great success, see for example [17]; (c) the fact that local selection results in gradual phenotypic change is a practical advantage since it allows for evolutionary process to be more easily tracked. Discrete recombination and adaptive mutation which relies on strategy parameters $\tau_0 = 1.0/\sqrt{2.0 * D}$ and $\tau_1 = 1.0/\sqrt{2.0 * \sqrt{D}}$ [1] are further applied to the selected chromosome pool with preset probabilities 0.2 and 0.02 respectively.

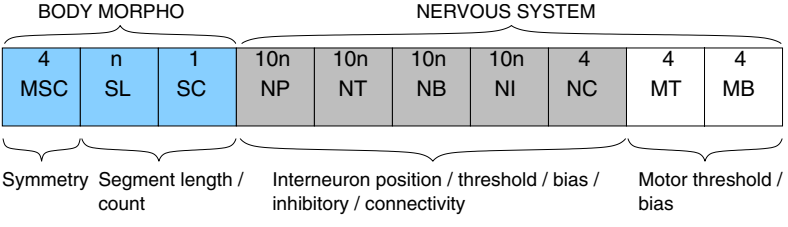


Fig. 3. The model genotype used within the simulation system. Each abbreviation refers to the type of gene and each number refers to the number of a type of gene. The variable 'n' refers to a variable number of body segments e.g. if there are 5 body segments, then there are 50 interneuron positional genes and 5 segment length factor genes. The 'NC' genes equate to the λ parameters in Eq. 2.

3 Experimental Setup

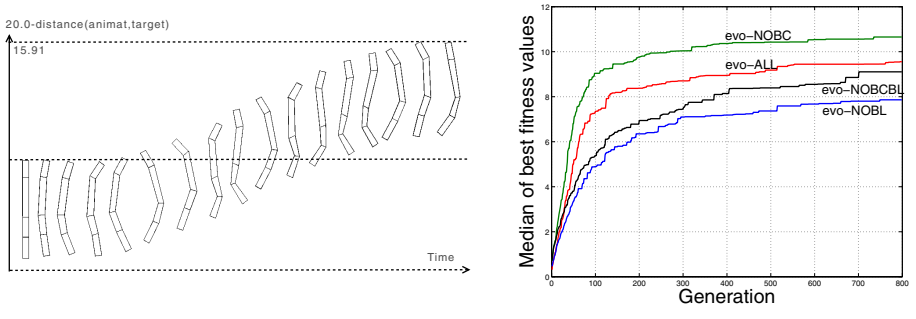
Four sets of 50 differently randomly initialised experiments were conducted; each placed a different level of constraint on the evolutionary process, as described:

1. **evo-ALL.** The full genotype as described above was evolved.
2. **evo-NOBCBL.** All parameters except for the number of body segments and the lengths of each segment, were evolved.
3. **evo-NOBC.** All parameters except for the number of body segments were evolved.
4. **evo-NOBL.** All parameters except for the lengths of each segment were evolved.

The aim of the first experiment was to essentially discover how 'everything' should become optimally tuned during the evolutionary process. That of the second was to find out how the agent should evolve at the other extreme i.e. when only its nervous system parameters (and body symmetry) are coevolved. The aim of the latter two experiments was to characterise how changes in segmentation properties (which we may equate to body morphology), affect fitness. In all experiments, the agent's task was to swim towards a fixed environmental target placed 20 units away from the head of the agent. Thus fitness was given by the distance of the agent's head from this target as measured at the end of the behavioural sequence; specifically, $20.0 - d(animat, target)$.

4 Results

A visualisation of the behaviour of the best evo-ALL agent is provided in Fig. 4a and plots of fitness are given in Fig. 4b. Statistical p-values are given in Table 1.



(a) (b)

Fig. 4. (a) Animat behavioural sequences for the best animat from evo-ALL showing undulatory type locomotion in which a wave of propulsion passes down the length of the agent's body. Snapshots taken every 25 behavioural iterations. (b) Medians over best individuals for each of the 4 sets of experiments (median readings are taken to prevent outlier artifacts).

Table 1. Statistical p-values generated from comparisons in best agent fitness (ranksum test) at the end of evolution

Comparison	evo-ALL	evo-NOBC	evo-NOBL	evo-NOBCBL
evo-ALL	-	0.0689	0.0393	0.4862
evo-NOBC	-	-	0.0003	0.025
evo-NOBL	-	-	-	0.1394

Effect on neural architecture. In Fig. 5, the neural architectures for the best individuals to have emerged are visualised. There are several observations to be made. Firstly, there are no connections between the sub-network architectures (one sub-network per body segment). This indicates a preference for fully decentralised, computationally independent sub-networks. Secondly, in all cases, few connections were established from

the sensory neurons. In instances where connections did emerge, it seems doubtful that the sensory neurons were actually employed, since a simple test in which the target object was removed from the environment subsequent to the evolutionary process found that agents could still successfully locomote. Thus, most agents (if not all) evolved without a true sensory system. We can also observe differences in wire length, see Fig. 5, with evo-ALL agents evolving to have fewer connections than agents from the other simulations. It would seem that generally, connectivity has a tendency to emerge minimally in all cases and more so in evo-ALL. Statistical differences in this property are given in Table 2.

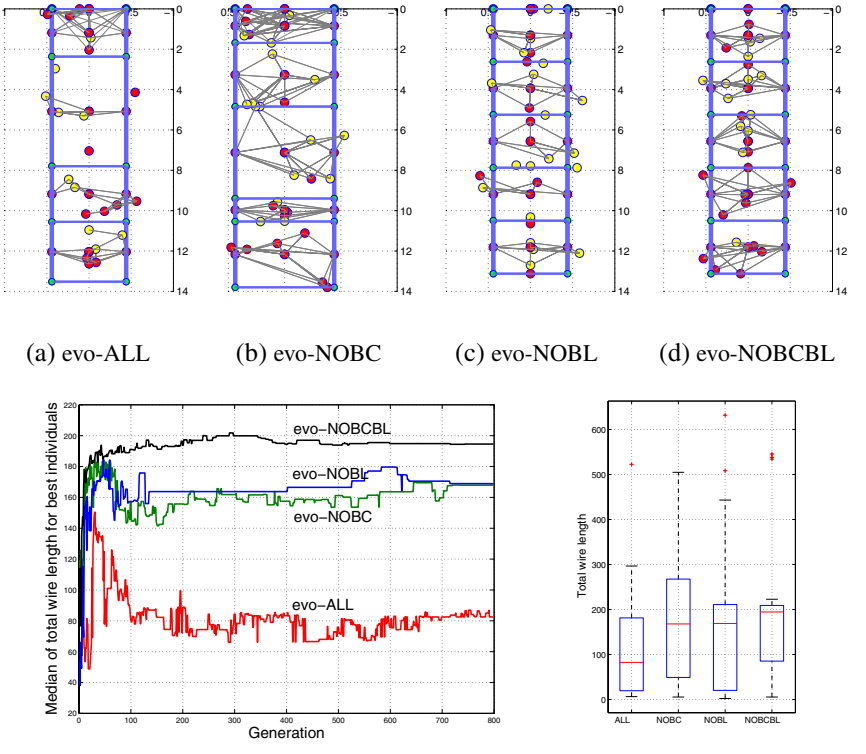
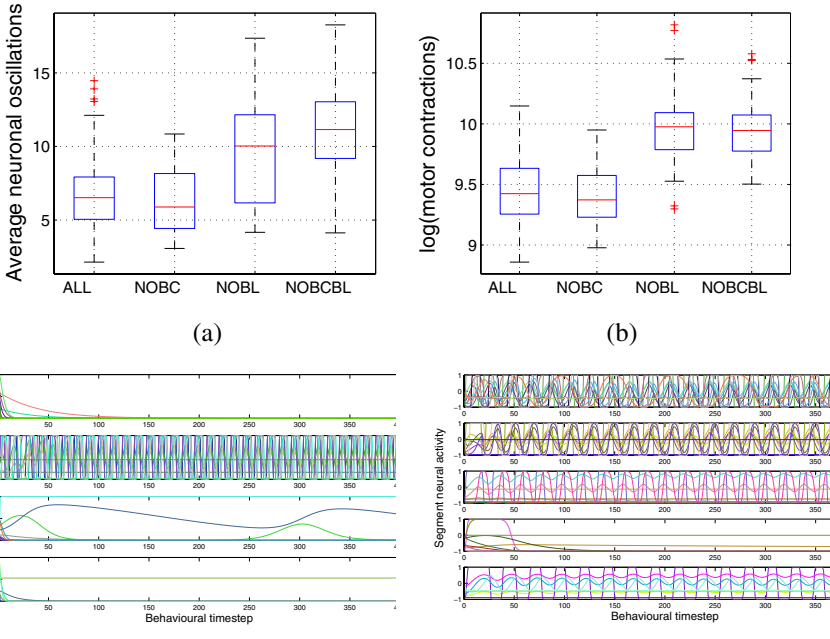


Fig. 5. (a-d) Best of the best neural architectures. In all architectures, there was a general tendency for to become decentralised with no interconnections between them. The two lower plots show wire lengths during and after (boxplot) evolution.

Table 2. Statistical p-values generated from comparisons in best agent wire length (ranksum test) at the end of evolution

Comparison	evo-ALL	evo-NOBC	evo-NOBL	evo-NOBCBL
evo-ALL	-	0.0072	0.1725	0.0018
evo-NOBC	-	-	0.1840	0.8590
evo-NOBL	-	-	-	0.2190

Effect on neural dynamics. Statistics of oscillation count (average number of times neuron activation moves from a negative to a positive state, and vice-versa) and motor contraction count ('motor activity', number of motor (spring) compressions), derived from the behaviours of all agents, are plotted in Figs. 6a and 6b. With regards to Fig. 6a, we can observe that neural activity is statistically greater in simulations evo-NOBL and evo-NOBCBL than in evo-NOBC and evo-ALL. Also, with regards to Figs. 6c and 6d, we can observe that actually, neural activity is minimal in all types of agent, given the limited number of neural circuits exhibiting at least *some* neural activity, but less prominently so in the evo-NOBL and evo-NOBCBL agents (note, for the sake of brevity, plots for evo-NOBC and evo-NOBCBL have been omitted). This signifies that for agents in which the lengths of the segments are not evolved, movement kinematics become constrained to the extent that neural activity has to compensate. Statistical differences in this property are given in Table 3.



(a) (b)

(c) evo-ALL (d) evo-NOBL

Fig. 6. Box plots at the end of evolution for (a) neural and (b) motor activity; (c&d) Neural activities of the very best individuals from simulations evo-ALL and evo-NOBL highlighting two extremes of neural activity. Horizontal bars are segmental neural activity.

Table 3. Statistical p-values generated from comparisons in best agent neuronal oscillation counts (ranksum test) at the end of evolution

Comparison	evo-ALL	evo-NOBC	evo-NOBL	evo-NOBCBL
evo-ALL	-	0.3919	0	0
evo-NOBC	-	-	0	0
evo-NOBL	-	-	-	0.0259

5 Discussion and Conclusions

The results presented in this paper demonstrate how neural organisation is constrained by body plan morphology. Successful animal or agent behaviour will only emerge if these two components interact appropriately so that the computational workload can be properly distributed. An emergence of this coupled process is complex due to the non-linear interactions between these components; but, depending on the constraints that we enforce, the emergence of optimal swimming behaviour is easier or harder.

At the beginning of the evolutionary process, all agent types are subject to environmental selection pressure. Then, as the evolutionary process progresses, these pressures change depending on the pre-existing constraints that we have enforced. In terms of those agents not endowed with segment length coevolution (which were demonstrated to be of significantly worse fitness than all other agent types), environmental pressure can only ever have a marginal impact on the body plan morphology; in other words, a body shape optimised for swimming is prevented from the outset and can never evolve. In order to make up for this, evolution strives to compensate by more rigorously adapting the architecture and computational properties of the nervous system.

The results signify several things:

- a tuning of the body plan morphology alone can have a very significant impact on fitness, and this is before we even consider the computational effort made by the nervous system.
- much of the computational workload can be offloaded to the body plan morphology, the distribution of which is mediated by the levels of permitted evolvability.
- neural circuits will evolve in order to compensate for a lack of body plan flexibility; this was demonstrated in terms of significantly greater levels of neural dynamic exhibited in such agents (see Fig. 6).
- neural circuitry had a tendency to emerge minimalistically and more so in the evo-ALL agents. This would suggest that when segmentation characteristics are co-evolved, less neural circuitry is required since behaviour generation can then be offloaded to an optimised body plan morphology.

The above points demonstrate that when the agent is evolved to reach reasonable levels of swimming behaviour, the distribution of computational workload is non-linearly affected given the complex coupling between body plan morphology and nervous system architecture. This was further demonstrated to have been automatic given a non pre-determined but evolutionary process.

In conclusion, the integrated coupling existing between body plan morphology and nervous system in an artificial agent lends to a dynamic distribution of computational workload between these two components. Tuning this coupling allows for better distribution; different aspects of the whole agent (nervous system and body plan) are endowed with the capacity to partake in the generation of behaviour. The agent survives; evolution becomes favourable.

Acknowledgements

This work was partially supported by grants from the Honda Research Institute Europe GmbH.

References

1. Bäck, T., Schwefel, H.-P.: An overview of evolutionary algorithms for parameter optimization. Evolutionary Computation 1(1), 1–23 (1993)
2. Blynel, J., Floreano, D.: Levels of dynamics and adaptive behavior in evolutionary neural controllers. In: From Animals to Animats: The Seventh International Conference on Simulation of Adaptive Behavior, pp. 272–281. MIT Press, Cambridge (2002)
3. Bongard, J.C.: Incremental Approaches to the Combined Evolution of a Robot's Body and Brain. PhD thesis, Mathematisch-naturwissenschaftlichen Fakultät der Universität Zürich, Zürich (2003)
4. Ekeberg, Ö.: An integrated neuronal and mechanical model of fish swimming. In: Computation in Neurons and Neural Systems, Washington, DC, pp. 217–222. Kluwer, Dordrecht (1994)
5. Ekeberg, Ö., Grillner, S.: Simulations of neuromuscular control in lamprey swimming. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 354, 895–902 (1999)
6. Gillis, G.B.: Undulatory locomotion in elongate aquatic vertebrates: Angulliform swimming since Sir James Gray. American Zoology 36, 656–665 (1996)
7. Ijspeert, A.J., Kodjabachian, J.: Evolution and development of a central pattern generator for the swimming of a lamprey. Artificial Life 5(3), 247–269 (1999)
8. Ijspeert, A.J., Arbib, M.: Visual tracking in simulated salamander locomotion. In: Sixth International Conference of the Society for Adaptive Behavior, pp. 88–97. MIT Press, Cambridge (2000)
9. Jones, B.: The Evolutionary Emergence of Neural Organisation in Computational Models of Primitive Organisms. PhD thesis, School of Computer Science, University of Birmingham, Birmingham, UK (2010)
10. Jones, B., Jin, Y., Sendhoff, B., Yao, X.: Evolving functional symmetry in a three dimensional model of an elongated organism. In: Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems, pp. 305–312. MIT Press, Cambridge (2008)
11. Müller, U.K., Smit, J., Stamhuis, E.J., Videler, J.J.: How the body contributes to the wake in undulatory fish swimming: flow fields of a swimming eel (*Anguilla Anguilla*). The Journal of Experimental Biology 204(16), 2751–2762 (2001)
12. Newman, S.A.: Is segmentation generic? BioEssays 15(4), 277–283 (1993)
13. Pfeifer, R., Iida, F.: Morphological computation: Connecting brain, body, and environment. In: Ijspeert, A.J., Masuzawa, T., Kusumoto, S. (eds.) BioADIT 2006. LNCS, vol. 3853, pp. 2–3. Springer, Heidelberg (2006)
14. Sarma, J., De Jong, K.: An analysis of the effects of neighborhood size and shape on local selection algorithms. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN IV 1996. LNCS, vol. 1141, pp. 236–244. Springer, Heidelberg (1996)
15. Sfakiotakis, M., Tsakiris, D.P.: Simuun: A simulation environment for undulatory locomotion. International Journal of Modelling and Simulation 26(4), 350–358 (2006)
16. Sims, K.: Evolving virtual creatures. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, pp. 15–22. ACM, New York (1994)
17. Soltoggio, A., Bullinaria, J.A., Mattiussi, C., Dürr, P., Floreano, D.: Evolutionary Advantages of Neuromodulated Plasticity in Dynamic, Reward-based Scenario. In: Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems. MIT Press, Cambridge (2008)

Multi-objective Evolutionary Algorithms to Investigate Neurocomputational Issues: The Case Study of Basal Ganglia Models*

Jean Liénard, Agnès Guillot, and Benoît Girard

Institut des Systèmes Intelligents et de Robotique
Université Pierre et Marie Curie-Paris 6, CNRS UMR 7222
4, place Jussieu
75252 Paris Cedex 05 - France
lienard@isir.upmc.fr

Abstract. The basal ganglia (BG) are a set of subcortical nuclei involved in action selection processes. We explore here the automatic parameterization of two models of the basal ganglia (the GPR and the CBG) using multi-objective evolutionary algorithms. We define two objective functions characterizing the supposed winner-takes-all functionality of the BG and obtain a set of solutions lying on the Pareto front for each model. We show that the CBG architecture leads to solutions dominating the GPR ones, this highlights the usefulness of the CBG additional connections with regards to the GPR. We then identify the most satisfying solutions on the fronts in terms of both functionality and plausibility. We finally define critical and indifferent parameters by analyzing their variations and values on the fronts, helping us to understand the dynamics governing the selection process in the BG models.

1 Introduction

In order to explore the characteristics and the critical parameter choices of two different models of the basal ganglia –a brain region involved in action selection– we use an empirical analysis method based on multi-objective evolutionary algorithms [1].

The Basal Ganglia (BG) is commonly defined as a set of subcortical interconnected nuclei, comprising the Striatum (Str), the Sub-Thalamic Nucleus (STN), both the external and internal Globus Pallidus (GPe and GPi) and both the Substantia Nigra par reticulata and pars compacta (SNr and SNc) [2]. The Str comprises mostly medium spiny neurons (MSN) and, more marginally, different types of interneurons, among which the fast-spiking interneurons (FSI) received special attention [3]. MSN are commonly subdivided on the basis of the presence of D1 or D2 dopamine receptors, distinguishing a striato-pallidal and a striato-nigral pathway [4]. The BG form a loop with the ventro-lateral thalamus (VL), the thalamic reticular nucleus (TRN) and parts of the frontal cortex (FC).

* This research was funded by the ANR, project EvoNeuro ANR-09-EMER-005-01.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 597–606, 2010.

The presented work is based on the hypothesis stipulating that the functional role of the BG is of generic action selection [2,5]. The BG are organized in multiple parallel segregated loops [6]. Within each of these loops, multiple striato-nigral channels interact, leading to a selection through disinhibition of the GPi/SNr, which is tonically active at rest [7]. This *base level* is strong enough to inhibit any action. The functionality of the BG would be to operate a "winner-takes-all" (WTA) algorithm, where the channel with the maximum input has the minimum output activity in the GPi/SNr (at least inferior to the *base level*), while the other ones have a maximal activity in the GPi/SNr (superior or equal to the *base level*). Based on a WTA hypothesis, many computational models have been proposed (see [8] for a recent review). These models are commonly set with hand-tuned parameters, with the goal of respecting biological constraints (for example, the STN has to be active at rest) while achieving a selection function, which is not necessarily expressed in precise quantitative terms.

Evolutionary Algorithms (EA) are designed to tackle optimization problems in a stochastic way. Inspired by the nature's evolutionary principle, the most striking difference to classical search is that EA evolve a population of solutions instead of a single one. Multi-Objective Evolutionary Algorithms (MOEA) are a subclass of EA conceived to optimize multiple conflicting objectives [9]. Analyzing a set of optimal and *a priori* equally acceptable solutions can lead to a better understanding of the dynamics governing the problem. Furthermore, one can eventually pick one particular solution, on the basis of expert knowledge.

There is no established method to parameterize a BG model. In a previous attempt at using EA to set the parameters, Wang et al. [10] used a simple genetic algorithm and evolved the "GPR" model described in [11]. The main difference with our approach lies in their characterization of a WTA. They used a binary measure to reflect whether the inputs were classified correctly (*i.e.* whether the channel that should be selected is under an arbitrary threshold). We use here multiple scores to describe the comparative levels of disinhibition of all the channels; the goal here is to understand the parameterization compromises that are made when simultaneously trying to inhibit the loosing channels as much as possible, as well as disinhibiting the winning one.

2 Materials and Methods

Building upon the recent "CBG" model [12] and the classical "GPR" model [11], we take their structures as a basis for two different evolutions. The connectivity of these models includes the knowledge on the macro biology of the BG [13]. Among others differences, the CBG permits the $D1 \rightarrow GPe$ connection, includes the modeling of FSI and grants a more global role of inhibition to the GPe [12]. Both of them incorporate the modeling of the thalamic loop, and both of them contain 6 concurrent striato-nigral channels. In the GPR, each nucleus contains 6 classical leaky integrator neurons, whereas in the CBG each nucleus contains 6 locally Projected Dynamical System neurons, a leaky integrator variant with dynamics described in [12]. The evolution of the CBG and of the GPR can modify

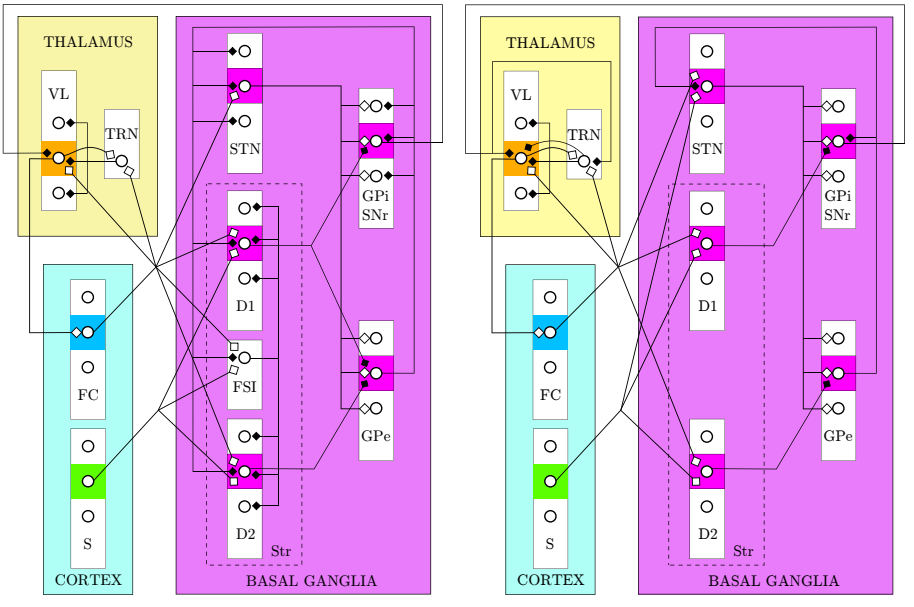


Fig. 1. The CBG (left) and GPR (right) architectures. Three channels in competition are represented. The outgoing connections of the shaded channel are the only one shown. White endings are excitatory, black are inhibitory. The inputs are the Saliences (S) in the cortex, and the outputs are the activity of the GPi/SNr neurons. See text for the meaning of the other abbreviations.

both the connections weights between nuclei (noted "nucleus → nucleus") and the tonic levels (noted "T_{nucleus}"), but leaves the structure of the circuits as shown in Figure 1. This accounts for a total of 25 parameters for the CBG, and 20 parameters for the GPR; see the results in Tables 1, 2 and 3 for the evolved parameters list.

We introduced some constraints on the parameters, in order to be sure that some levels of biological plausibility are respected but also to reduce the exploration space to make the convergence of the evolution more likely. Neurons from the GPe and GPi exhibit comparable baseline [14], hence they have the same tonic levels : $T_{\text{GPe}} = T_{\text{GPi}}$. Neurons of the Striatum with D1 and D2 receptors are thought to have comparable afferents from the Cortex [15], so $S \rightarrow D1 = S \rightarrow D2$ and $FC \rightarrow D1 = FC \rightarrow D2$. Afferents to both the GPe and GPi from the STN appear to be the same [16], consequently $STN \rightarrow GPe = STN \rightarrow GPi$. To be acceptable, a solution had to have a GPi output at rest larger than 0.09, as the GPi is known to be tonically active [2] and as this output is 0.10 for the CBG and 0.17 for the GPR. Finally, we chose to limit connection weights in the range $[0.05; 1]$. Indeed, a null connection weight would mean no connection at all, changing the connectivity of the circuit.

2.1 Evolving the Circuit

Our choice for simulating the evolution is the widely used NSGA-II algorithm [17]. The mutation rate was fixed at $\mu = 0.1$, a rather high value that serves to widely explore the space of parameters. We set the distribution index for mutation and cross-over operators in favor of mutation, with $\eta_m = 15$ and $\eta_c = 10$. We scheduled 10 runs comprising 1500 generations each, with a population size of 200. The computational framework used was SFERES [18]. Designed to work in a parallel fashion, it permits fast simulation on multi-cores machines; it is also fairly easy to set up and use.

The selection functionality of the BG has to be formally defined so as to allow the computation of some fitness criteria. To evaluate the quality of the circuits, they were submitted to $N = 500$ different inputs, each input being a vector of six random decimals drawn uniformly in $[0, 1]$. They were simulated during one second with each of these inputs, before the outputs are considered for fitness evaluation.

Two objectives have been defined for the evolution. First, the channel corresponding to the largest input (the *selected channel*) has to be disinhibited to the maximum. This translates straightforwardly as :

$$\text{First objective : minimise } f_1 = \frac{\sum_N GPi_{\text{selected channel}}}{N} \tag{1}$$

But this is not sufficient to obtain a WTA algorithm, as this could lead to the disinhibition of all channels. Therefore, we define the second objective as the mean of the five other channels :

$$\text{Second objective : maximise } f_2 = \frac{\sum_N \left(\sum_{\text{channel} \neq \text{selected channel}} \frac{GPi_{\text{channel}}}{5} \right)}{N} \tag{2}$$

3 Results

Each run resulted after 1500 generations in a set of non-dominated solutions, called a Pareto front [9]. We define the global Pareto-optimal front as the set of non-dominated solutions from all the runs (Figure 2). Every front contributes to it for both models, hence we assume they all have converged to the same optimal front for both experiments.

As the best solutions are within the global Pareto-optimal fronts, we will confine further development to them. Each of these solutions represents the best trade-off from a certain point of view; *a priori* there is no reason to chose one particular solution at this level of analysis. The Pareto-optimal front comprises 510 solutions for the CBG and 551 solutions for the GPR. Interestingly, solutions from the CBG front dominate those from the GPR front (Figure 2), the additional degrees of freedom of the CBG can thus be exploited to globally improve the performance on our task.

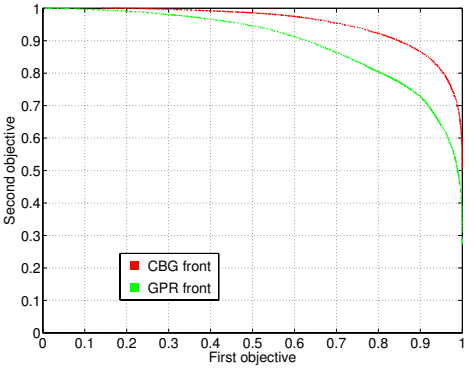


Fig. 2. The fitnesses for the global Pareto-front of the evolution of CBG and GPR

Overview of the parameters. For the CBG, 8 parameters are maximised or minimised for all the solutions (Table 1); for the GPR, only two parameters are in this case (Table 2). The dispersion of a parameter for close solutions on the front indicates how much this parameter contributes to the fitnesses (high dispersion reflect minor importance). The standard deviation (SD) does not suffice to express this, as it is a global indicator reflecting the dispersion of the values across the whole front. Hence we use the standard deviation of the first difference of the parameter values accross the front.

For both models, no parameter is at the boundaries in the thalamic loop, and the SD is high as it ranges from 0.23 to 0.29 (Table 3). The first difference SD is high too, ranging from 0.33 to 0.42. This shows high dispersion for close solutions. Hence the contribution of each of these parameters to the fitness scores appears as secondary, as the evolution process did not set them to precise values.

WTA functionality analysis. We designed the two objectives so that their combination favors the WTA functionality. Of course, maximizing only one of

Table 1. Mean ± SD and first difference SD for the CBG parameters (except the thalamic loop)

Minimised		
$FC \rightarrow FSI$	0.05 ± 0.01	0.01
$GPe \rightarrow D2$	0.06 ± 0.05	0.04
$GPe \rightarrow GPi$	0.07 ± 0.03	0.01
$-T_{D1/D2}$	0.06 ± 0.02	0.03

Maximised		
$S \rightarrow D1/D2$	1 ± 0.01	0.01
$D1 \rightarrow GPi$	1 ± 0.01	0.02
$D2 \rightarrow GPe$	0.99 ± 0.04	0.05
$T_{GPe/GPi}$	0.98 ± 0.03	0.04

Between the limits		
$S \rightarrow FSI$	0.74 ± 0.14	0.17
$GPe \rightarrow STN$	0.15 ± 0.04	0.03
$GPe \rightarrow D1$	0.47 ± 0.11	0.05
$GPe \rightarrow FSI$	0.34 ± 0.07	0.09
$STN \rightarrow GPe/GPi$	0.45 ± 0.35	0.14
$D1 \rightarrow GPe$	0.76 ± 0.22	0.12
$FSI \rightarrow D1/D2$	0.48 ± 0.22	0.09
$FC \rightarrow D1/D2$	0.43 ± 0.25	0.11
$FC \rightarrow STN$	0.24 ± 0.17	0.16
T_{STN}	0.91 ± 0.13	0.17

Table 2. Mean ± SD and first difference SD for the GPR parameters (except the thalamic loop)

Minimised		
$GPe \rightarrow GPi$	0.07 ± 0.04	0.04

Maximised		
$D1 \rightarrow GPi$	1.00 ± 0.00	0.01

Between the limits		
$S \rightarrow D1/D2$	0.86 ± 0.14	0.11
$FC \rightarrow D1/D2$	0.91 ± 0.11	0.10
$STN \rightarrow GPe/GPi$	0.62 ± 0.42	0.15
$GPe \rightarrow STN$	0.48 ± 0.37	0.13
$D2 \rightarrow GPe$	0.22 ± 0.19	0.15
$FC \rightarrow STN$	0.24 ± 0.17	0.21
$S \rightarrow STN$	0.48 ± 0.34	0.25
$-T_{D1/D2}$	0.91 ± 0.20	0.05
T_{STN}	0.29 ± 0.29	0.20
$T_{GPe/GPi}$	0.76 ± 0.29	0.06

Table 3. Mean ± SD and first difference SD for the thalamic loop parameters

CBG parameters		
$BG \rightarrow VL$	0.32 ± 0.24	0.33
$FC \rightarrow VL$	0.54 ± 0.28	0.38
$FC \rightarrow TRN$	0.52 ± 0.29	0.4
$VL \rightarrow FC$	0.47 ± 0.26	0.35
$VL \rightarrow TRN$	0.62 ± 0.27	0.38
$TRN \rightarrow VL$	0.63 ± 0.29	0.42
T_{VL}	0.44 ± 0.24	0.34

GPR parameters		
$BG \rightarrow VL$	0.53 ± 0.27	0.37
$BG \rightarrow TRN$	0.53 ± 0.24	0.34
$FC \rightarrow VL$	0.57 ± 0.27	0.38
$FC \rightarrow TRN$	0.44 ± 0.24	0.33
$VL \rightarrow FC$	0.62 ± 0.26	0.34
$VL \rightarrow TRN$	0.53 ± 0.25	0.34
$TRN \rightarrow VL$	0.56 ± 0.27	0.39
$TRN \rightarrow VL_{self}$	0.52 ± 0.23	0.33

these objectives is useless for that: a circuit systematically disinhibiting all the channels would have a score ($f_1 = 1$, $f_2 = 0$), while a circuit maximally inhibiting all the channels would have a score ($f_1 = 0$, $f_2 = 1$). As shown in Figure 2, the evolution found no solution that can maximise both of these objectives.

In order to more deeply characterize the solutions, we studied their relationship with the *base level*, defined as the output of the GPi/SNr with null inputs. Electrophysiological studies show that the GPi is tonically active in the absence of inputs, meaning that the *base level* should be high [2]. Furthermore, when there are inputs to the BG, the majority of GPi neurons have an increased activity compared to the resting state [2,14]. Hence, as mentioned in introduction, a biologically plausible selection should have a *base level* higher than the selected channel's output (expressed by objective 1) and lower than the mean of unselected channels output (expressed by objective 2). Figure 3 (top) represents the objectives functions altogether with the base level for each solution of the fronts. The values of the *base levels* seem to be structured with regard to fitness values, plateaus of specific values appear, especially a very constrained one for the CBG individuals in the 250-450 interval. With regards to the aforementioned biological plausibility criterion, the best solutions for the CBG are those ranging

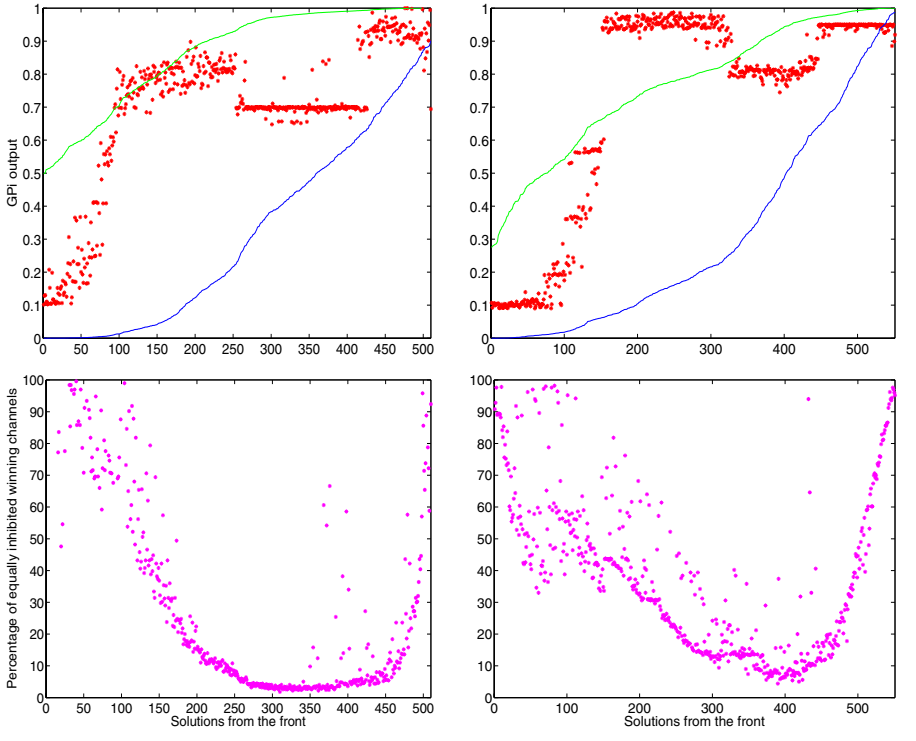


Fig. 3. Left graphics concern the CBG, right graphics concern the GPR. The horizontal abscissae of each graphic enumerate the solutions along the front, in ascending order for their first objective. Top : blue and green lines are the values of f_1 and f_2 respectively (equations 1 and 2); red dots represent the base level. Bottom : points represent the percentage of cases where two channels are equally disinhibited, leading to an abusive selection of both of them.

approximately from 1 to 100 and from 250 to 450. For the GPR, they are the ones from 1 to 150 and those larger than 325.

We can further refine the set of acceptable solutions by imposing an exclusive selection. Indeed, we did not explicitely forbid the designation of multiple winners; some of the solutions obtained often disinhibit more than one channel, instead of the one with the most important input only. The percentage of such multiple selections is represented in Figure 3 (bottom). Interestingly, the best selection is found for the solutions operating a biologically plausible WTA (solutions 250 to 450 for the CBG and 325 to 450 for the GPR), comforting us in the adequacy between the supposed function operated by the BG, the modeling of the BG (as described by the CBG or GPR) and electrophysiological data.

4 Discussion

In this work, we show that two existing models of BG [12,11] can be optimized through MOEA, with the goal of recreating a "winner-takes-all" behavior. The analysis of the variations of the parameters of the best solutions highlighted some regularities and helped identify critical and indifferent parameters. The study of the biological plausibility of the solutions helped identifying the most interesting parts of the Pareto front. Generality of the reasoning is exemplified by the application on two different models.

Parameterization exploration. On the basis of this case study, it seems that the MOEA permit to gain insight into the properties of models of a given neural structure. By generating multiple solutions fulfilling antagonist objectives, MOEA can be used as a mean for exploration. We divided the modeling of a WTA into two separate sub-problems (expressed by equations 1 and 2) to explore a wide variety of different WTA implementations. It was then possible to pick particular solutions on the basis of "expert knowledge" not expressed in the constraints, as we did by delineating a biologically plausible selection behavior.

Model comparison. Comparison between two architectures is also possible. The CBG Pareto front dominates the GPR one (Figure 2). Furthermore, the percentage of cases with two winning channels instead of one (bottom of Figure 3) is smaller for the CBG. Hence, the additional degrees of freedom permitted by the CBG (Figure 1) are useful for our purposes, as they lead to a better adequacy to a WTA function.

Parameter contribution. The MOEA help identifying the contribution of a given parameter to a particular behavior. Maximised parameters along the front give informations on the most important parameters needed to achieve a presupposed function. For the CBG, these parameters are $S \rightarrow D1/D2$, $D1 \rightarrow GPi$, $D2 \rightarrow GPe$ and $T_{GPe/GPi}$ (Table 1). This outlines some of the principal connections of the Direct / Indirect pathway hypothesis [19], comforting in some sense the fundamental pertinence of these two loops in order to achieve selection. Interestingly, the less complete GPR structure does not permit to see this, as the $D2 \rightarrow GPe$ connection is limited (Table 2).

Minimised parameters are in opposition with the desired behavior. In both models, the $GPe \rightarrow GPi$ connection is minimised. On the one hand, this is contradictory with anatomical data pointing out that one third of the projecting GPe neurons target the GPi/SNr [20], implying that the connection from GPe to GPi is not non-existent. On the other hand, electrophysiological data lead to the hypothesis that they might be weak [21]. Our contribution to this precise point is that, in all the variations of WTA obtained with both the CBG and GPR structures, this connection grieves the supposed functionality. To explain this result, we can conceive three non-contradictory hypothesis : (1) both the CBG and GPR lack in the modeling of a (perhaps unknown) fundamental connection or property enabling the $GPe \rightarrow GPi$ to contribute to a WTA algorithm; (2) the

BG is performing another function which gives meaning to this connection; (3) in accordance with electrophysiological data, this connection is effectively weak.

Some parameters are fixed by evolutionary pressure to rather constant values that are not at the bounds. This is for example the case of the $GPe \rightarrow STN$ and $GPe \rightarrow FSI$ connections of the CBG, exhibiting low standard deviations (Table 1). This could indicate the best ratio of relative strengths for these connections. However, the models used here are at a pretty high level of abstraction; consequently, biological interpretation for these relative values is not as straightforward as for minimised or maximised values, and should be made with caution.

The study of the parameter variances for solutions exhibiting comparable fitnesses permits to isolate parameters that are not needed for the desired behavior. The thalamic loop of the CBG and GPR models has a high first difference SD (Table 3), hence the modeling of these is not pertinent with regards to the defined objectives. This is coherent with the intuition of the modelers [22] who used it to modulate the time taken for the selection. Defining a third objective for the time of convergence could help to explore the thalamic loop role.

Future work will first involve examining in details what in the BG model structures or in the constraints underlies the appearance of a *base level* plateau for the best WTA solutions obtained, and its possible relevance for the functional neurobiology of the BG. We will also dig deeper into the biological plausibility of BG models. By adding connections and neuronal properties neglected in BG models and by polishing our set of constraints, we will evolve a more complete model with the goal of imitating data from electrophysiological experiments.

Acknowledgment

The authors thank Stéphane Doncieux and Jean-Baptiste Mouret for their valuable advices concerning MOEA.

References

1. Doncieux, S.: Evolutionary algorithms as exploration and analysis helper tools, application to a flapping wings aircraft. In: IROS Workshop Exploring New Horizons in Evolutionary Design of Robots, Saint Louis, USA, pp. 19–25 (2009)
2. Mink, J.: The basal ganglia: focused selection and inhibition of competing motor programs. Progress in Neurobiology 50(4), 381–425 (1996)
3. Tepper, J., Bolam, J.: Functional diversity and specificity of neostriatal interneurons. Current opinion in neurobiology 14(6), 685–692 (2004)
4. Gerfen, C., Engber, T., Mahan, L., Susel, Z., Chase, T., Monsma Jr., F., Sibley, D.: D1 and D2 dopamine receptor-regulated gene expression of striatonigral and striatopallidal neurons. Science 250(4986), 1429 (1990)
5. Redgrave, P., Prescott, T., Gurney, K.: The basal ganglia: a vertebrate solution to the selection problem? Neuroscience-Oxford 89, 1009–1024 (1999)
6. Alexander, G., DeLong, M., Strick, P.: Parallel organization of functionally segregated circuits linking basal ganglia and cortex. Annual review of neuroscience 9(1), 357–381 (1986)

7. Chevalier, G., Deniau, J.: Disinhibition as a basic process in the expression of striatal functions. Trends Neurosci. 13(7), 277–280 (1990)
8. Cohen, M., Frank, M.: Neurocomputational models of basal ganglia function in learning, memory and choice. Behavioural Brain Research 199(1), 141–156 (2009)
9. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, Chichester (2001)
10. Wang, Y., Li, S., Chen, Q., Hu, W.: Biology Inspired Robot Behavior Selection Mechanism: Using Genetic Algorithm. In: Li, K., Fei, M., Irwin, G.W., Ma, S. (eds.) LSMS 2007. LNCS, vol. 4688, pp. 777–786. Springer, Heidelberg (2007)
11. Gurney, K., Prescott, T., Redgrave, P.: A computational model of action selection in the basal ganglia. I. A New Functional Anatomy. Biological Cybernetics 84(6), 401–410 (2001)
12. Girard, B., Tabareau, N., Pham, Q., Berthoz, A., Slotine, J.: Where neuroscience and dynamic system theory meet autonomous robotics: a contracting basal ganglia model for action selection. Neural Networks 21(4), 628–641 (2008)
13. Tepper, J., Abercrombie, E., Bolam, J.: Basal ganglia macrocircuits. Progress in Brain Research 160, 3 (2007)
14. Turner, R., Anderson, M.: Pallidal discharge related to the kinematics of reaching movements in two dimensions. Journal of neurophysiology 77(3), 1051 (1997)
15. Ballion, B., Mallet, N., Bezard, E., Lanciego, J., Gonon, F.: Intratelencephalic corticostriatal neurons equally excite striatonigral and striatopallidal neurons and their discharge activity is selectively reduced in experimental parkinsonism. European Journal of Neuroscience 27(9), 2313–2321 (2008)
16. Shink, E., Bevan, M., Bolam, J., Smith, Y.: The subthalamic nucleus and the external pallidum: two tightly interconnected structures that control the output of the basal ganglia in the monkey. Neuroscience 73(2), 335–357 (1996)
17. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., Fast, A., Algorithm, E.: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2) (2002)
18. Mouret, J.B., Doncieux, S.: Sferes$_{v2}$: Evolvin' in the multi-core world. In: IEEE Congress on Evolutionary Computation, CEC 2010 (to appear 2010)
19. Albin, R., Young, A., Penney, J.: The functional anatomy of basal ganglia disorders. Trends in Neurosciences 12(10), 366–375 (1989)
20. Sato, F., Lavallee, P., Levesque, M., Parent, A.: Single-axon tracing study of neurons of the external segment of the globus pallidus in primate. Journal of Comparative Neurology 417(1), 17–31 (2000)
21. Nambu, A.: Seven problems on the basal ganglia. Current Opinion in Neurobiology 18(6), 595–604 (2008)
22. Girard, B., Cuzin, V., Guillot, A., Gurney, K.N., Prescott, T.J.: A basal ganglia inspired model of action selection evaluated in a robotic survival task. Journal of Integrative Neuroscience 2(2), 179–200 (2003)

Cooperative Stigmergic Navigation in a Heterogeneous Robotic Swarm

Frederick Ducatelle, Gianni A. Di Caro, and Luca M. Gambardella

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)
Galleria 2, 6928 Manno-Lugano, Switzerland
{frederick,gianni,luca}@idsia.ch

Abstract. We study self-organized cooperation in a heterogeneous robotic swarm consisting of two sub-swarms. The robots of each sub-swarm play distinct roles based on their different characteristics. We investigate how the swarm as a whole can solve complex tasks through a self-organized process based on local interactions between the sub-swarms. We focus on an indoor navigation task, in which we use a swarm of wheeled robots, called foot-bots, and a swarm of flying robots that can attach to the ceiling, called eye-bots. Foot-bots have to move back and forth between a source and a target location. Eye-bots are deployed in stationary positions against the ceiling, with the goal of guiding foot-bots. We study how the combined system can find efficient paths through a cluttered environment in a distributed way. The key component of our approach is a process of mutual adaptation, in which foot-bots execute instructions given by eye-bots, and eye-bots observe the behavior of foot-bots to adapt the instructions they give. The system is based on pheromone mediated navigation of ant colonies, as eye-bots function as stigmergic markers for foot-bots. Through simulation, we show that the system finds feasible paths in cluttered environments, converges onto the shortest of two paths, and spreads over different paths in case of congestion.

1 Introduction

We study self-organized cooperation in a heterogeneous robotic swarm consisting of two sub-swarms. The robots of each sub-swarm play distinct roles based on their different characteristics. We investigate how the swarm as a whole can solve complex tasks through a process based on local interactions between sub-swarms. We focus on an indoor navigation task. Our solution is based on stigmergic foraging in ant colonies [1]: we let one sub-swarm serve as active stigmergic markers for the other, so that the system as a whole learns efficient paths.

We consider the following problem setup. A swarm of wheeled robots, called foot-bots, is deployed in an indoor environment to solve a navigation task: they need to go back and forth between a source and a target location (e.g., to transport objects). They are assisted by a swarm of flying robots that can attach to the ceiling, called eye-bots. These are deployed beforehand to cover the area between source and target and take fixed positions at the ceiling. From these positions,

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 607–617, 2010.

they give directional instructions to the foot-bots, to guide them towards the source or the target. The use of a heterogeneous swarm provides flexibility. The flying eye-bots can quickly explore and cover an unknown indoor environment. Their ability to attach to the ceiling allows them to interact with the foot-bots, guide and observe them, without physically interfering with their movements.

We investigate how the combined system of foot-bots and eye-bots can find navigable and efficient paths in a cluttered environment. We focus on a fully distributed solution, which relies only on local communication between eye-bots and nearby foot-bots (using infrared and visual communication), without any interaction among eye-bots or foot-bots. We do not study how eye-bots are deployed. Given the limited communication requirements of our approach, we do not need the eye-bots to be in communication range or in a specific arrangement: we can use any algorithm that provides sufficient coverage of an area (e.g., [2]).

We propose a distributed learning algorithm, in which eye-bots start from a random decision policy to give instructions to foot-bots, and observe foot-bot behavior in order to adapt the instructions they give and learn a good policy. This way, eye-bots use foot-bots as sampling agents to learn about the environment. From a different point of view, the eye-bots form a set of discrete locations in the environment storing and updating a distributed navigation policy that is based on observed foot-bot behavior and in turn influences future foot-bot movements. In this sense, they form stigmergic markers for foot-bot navigation. The heterogeneous system of eye-bots and foot-bots is able to cooperatively find feasible paths for foot-bots through the environment. Moreover, it is capable of finding shortest paths and of spreading over multiple paths in case of congestion.

2 Robot Characteristics and Problem Setup

The foot-bot and eye-bot are under development in the Swarmanoid project (`http://www.swarmanoid.org`). The foot-bot (Fig. 1(a)) moves on the ground. It has two cameras, one omnidirectional and one pointing up. Foot-bots can communicate with each other and with eye-bots via visual signals (in this paper, we use only communication between foot-bots and eye-bots), using the 256 color LED ring that is placed around their body and the LED beacon they have on top. Moreover, they can exchange wireless messages locally (up to 3 m) at low bandwidth using an infrared range and bearing (IrRB) system.The eye-bot (Fig. 1(b)) is a flying robot, which can attach to the ceiling using a magnet (the design assumes ferromagnetic ceilings). It has a pan-and-tilt camera which it can point in any direction below. Like the foot-bot, it can communicate using a multi-color LED ring placed around its body, or using the IrRB system. Details for both robots can be found on the Swarmanoid website.

The eye-bots and foot-bots are placed in an indoor arena like the one shown in Fig. 1(c). The task of foot-bots is to find paths to go back and forth between a source (top right in the figure) and a target location (bottom left in the figure) (e.g., to transport objects). Eye-bots are attached to the ceiling in a formation that covers the area between source and target (we show a grid,

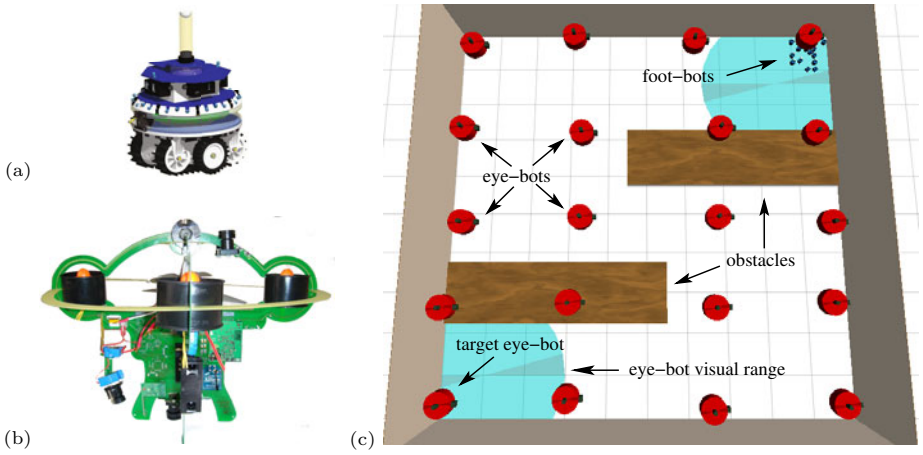


Fig. 1. (a) Foot-bot (CAD draw), (b) eye-bot (prototype), and (c) example scenario

but any formation that lets eye-bots approximately cover the area with their visual range could be used). They support the foot-bots in their task, by giving directional instructions to foot-bots that are within their visual range (the disk under selected eye-bots in the figure). The task is made difficult by the presence of obstacles that block the way for foot-bots. We study how the heterogeneous swarm of foot-bots and eye-bots can solve this task in a distributed way.

3 Related Work

We know of no other work that studies stigmergic cooperation between subswarms. However, given the ant colony inspiration, our work is related to research on pheromone based stigmergic foraging in swarm robotics [3,4,5,6]. A difficult issue in such systems is how to implement pheromone. Some authors use practically infeasible solutions, such as light encoding of pheromone using an overhead projector [3,5] or a map in a shared memory [6], assuming that the issue of pheromone implementation will be solved somehow in the future. Other authors experiment with chemical traces, e.g. using alcohol [4]. We use stigmergic communication points to store pheromone, which can be considered a practical alternative. A similar approach was followed in [7], where pheromone is stored in RFID tags embedded in the environment. Compared to that work, our approach has the advantage that communication points are mobile robots, so that the system also works in environments that have not been fitted with embedded tags. Moreover, that work was not concerned with adaptive path finding. One other work that employs robots to store pheromone is [8]. Also they do not study adaptive path learning. Moreover, they do not use distinct robot swarms to store pheromone and to solve a task. We believe our approach gives more flexibility. Finally, we point out that our work is to our knowledge the first that considers automatic traffic spreading in addition to shortest path finding.

In terms of problem setup, our work is related to research on the use of embedded sensor networks for robot navigation [9,10,11]. In such systems, sensor nodes spread in the environment are used to guide a mobile robot to a target. The sensor nodes play a role similar to that of our eye-bots. An important difference is the central role of network communication in those systems. Sensor nodes calculate the shortest path through the network formed among them, and use this to guide the robot. This requires all nodes to be connected in a network. Moreover, it assumes a one-to-one relation between communication links and navigable path segments: all communication links are expected to indicate navigable paths, and all navigable paths should be covered by communication links. Since our approach relies on local communication and foot-bot observation, rather than on network communication, none of these restrictions are present.

4 Cooperative Stigmergic Navigation

4.1 General Description

The main idea behind our approach is that eye-bots maintain stochastic policies, which they use to choose navigation instructions to broadcast to foot-bots, and that they update these policies based on visual observations of foot-bot behavior.

We limit the possible navigation instructions to 12 discrete directions, so one direction every $\pi/6$ radians. Each eye-bot maintains two different policies: policy P_t for the target and policy P_s for the source. Each policy consists of an array of 12 positive real valued numbers, expressing the preference for the discrete navigation directions. Periodically, at discrete time steps, the eye-bot selects two directions from the policies, θ_t for the target and θ_s for the source. These directions are broadcast locally to guide nearby foot-bots.

Foot-bots move towards the directions they receive from eye-bots. As they move, they use light signals to make their behavior visible for eye-bots. A foot-bot simultaneously switches on its LED beacon on top and one LED in front, to show eye-bots its movement direction. The color of the front LED is used to indicate whether the foot-bot's goal is the source or the target, whereas the color of the LED beacon shows whether it is doing obstacle avoidance.

Eye-bots use their camera to observe the behavior of foot-bots in their field of view, and based on this information they update P_t and P_s. They consider three aspects of foot-bot behavior: the foot-bot's current goal (whether it is going to the target or the source), the direction θ_f it is coming from (relative to the eye-bot's orientation), and whether it is performing obstacle avoidance.

Using this algorithm, eye-bots use foot-bots as sampling agents to explore the effect of different actions in the environment and learn efficient policies. From a different point of view, foot-bots use eye-bots as stigmergic communication points which store previous foot-bot experiences and influence future foot-bot movements. Seen in this way, eye-bots fulfill a similar role as pheromone in stigmergic foraging by ant colonies. Therefore, they could be seen as a practical way to physically implement pheromone in swarm robotic systems.

4.2 Updating and Using Eye-Bot Stochastic Policies

When an eye-bot observes a foot-bot that is going towards the target, it assumes that the foot-bot is coming from the source, so it increases the policy P_s for the direction θ_f that the foot-bot is coming from, and decreases the policy P_t for that same direction (equivalent updates are made for foot-bots going towards the source). The idea is that θ_f is a local sample of a feasible direction towards the source. The frequency with which a direction is observed is also expected to be related to its quality, as this effect has been observed in experiments with ant colonies [12]. When the eye-bot observes a foot-bot performing obstacle avoidance, it decreases both policies P_s and P_t for the direction in which it sees the foot-bot, assuming that direction is blocked by obstacles.

To update a policy P in a given direction θ, the eye-bot first associates θ with the closest of its 12 discrete policy directions, and derives the corresponding index i. Then, policy increases are performed using an additive constant c_a, while policy decreases are performed using a multiplicative constant $c_m \in]0, 1[$, as shown in (1). The multiplicative rule allows the system to learn fast about the presence of obstacles. All policy entries are initialized to $P[i] = 1/12$.

$$P[i] = \begin{cases} P[i] + c_a & \text{in case of a policy increase} \\ P[i] \cdot c_m & \text{in case of a policy decrease} \end{cases} \tag{1}$$

Eye-bots draw directions from the policies using a stochastic rule balancing exploitation and exploration: with a constant probability q, the direction with highest preference is chosen. Otherwise, a direction is chosen randomly, from a distribution proportional to the relative preferences of directions in the policy.

4.3 Directional Instructions from Eye-Bots to Foot-Bots

Eye-bots give instructions to foot-bots using a combination of visual signals with LEDs and wireless communication with the IrRB system. Periodically, they sample the directions θ_s and θ_t from their policies, and broadcast them over the IrRB system so foot-bots can locally receive them. To show foot-bots a reference direction θ_0, they switch on a red LED in front and a blue LED in the back. This communication scheme is scalable for the number of foot-bots and eye-bots, since wireless communication is limited to one periodic, local broadcast by each eye-bot. All other communication is via light signals.

IrRB communication from eye-bots to foot-bots is focused in a cone, so a foot-bot needs to move under an eye-bot to receive its messages. The foot-bot uses its upward camera to read θ_0, and extracts direction θ_s or θ_t (depending on whether its goal is the source or the target) from the received wireless message. It interprets θ_s or θ_t relative to θ_0, to derive a new travel direction θ_n. It turns into direction θ_n, and then moves forward for a distance d (enough to get out of view of the eye-bot it received the message from), or until it arrives under a different eye-bot. If after d no eye-bot is reached, the foot-bot moves towards the closest eye-bot in its camera view. If no eye-bot is seen, it starts a random movement: repeatedly make a random turn and move forward for a random distance.

4.4 Foot-Bot Navigation Behavior

Foot-bot movements are guided by the instructions of eye-bots, as outlined in Sect. 4.3. However, Foot-bots have a preference not to return where they come from, so exploration is directed away from where they come from. This preference is implemented as follows. When a foot-bot receives from an eye-bot e a travel direction θ_n that is forward (i.e., between $-\pi/2$ and $\pi/2$) with respect to the travel direction received from the previous eye-bot, the foot-bot follows θ_n and does not consider other directions received in subsequent time steps from e. If, however, the received travel direction θ_n is backward, the foot-bot follows θ_n but simultaneously keeps listening for other instructions from e. If e has a strong preference for the backward direction θ_n, it will send θ_n to the foot-bot again in the next time steps, so that the foot-bot keeps going in that direction. However, if in one of the subsequent time steps e sends a forward direction θ'_n, the foot-bot will use that without listening to other directions, and will not turn back. Besides this, foot-bots have an obstacle avoidance behavior, which makes them turn away reactively from obstacles detected using infrared proximity sensors.

Foot-bots use LED signals to show eye-bots their status and behavior. Besides changing the colors, as described in Sect. 4.1, they also switch off the front LED in certain occasions. This way, eye-bots can see where they are (through the LED beacon, which is not switched off), but not the direction they are coming from, θ_f. As a consequence, eye-bots cannot update their policy for θ_f. Foot-bots do this whenever their movement direction is not representative for the general direction they are following from source to target: when they are performing obstacle avoidance, when they are following an instruction that sends them backward, or when they are not following an eye-bot instruction (e.g., performing random movement). The goal is to reduce noise in eye-bot policies.

5 Finding Paths in a Cluttered Environment

We experimentally investigate whether our system can find paths in a cluttered environment. All tests in this section and in the rest of this paper are done with the ARGoS simulator, which was developed in the Swarmanoid project (see `http://www.swarmanoid.org/swarmanoid_simulation.php`). All experiments last 3000 s. We carry out 100 independent runs for each test. Some preliminary results for the behavior presented in this paper appeared in [13] (with slightly different settings for the behavior and the experimental setup). In all tests, we use the following parameter settings, which were defined empirically: $c_a = 0.5$, $c_m = 0.99$, $q = 0.5$, and $d = 2$ m.

We study the scenario of Fig. 1(c) and consider the effect of varying the number of foot-bots. We measure the time from the start of the experiment until the first foot-bot reaches the target, t_1, the average time needed by foot-bots to travel between source and target, t_a, and the average time needed by foot-bots in case we pre-program eye-bots to show the shortest path, t_s. The results are shown in Fig. 2. Error bars show one standard deviation.

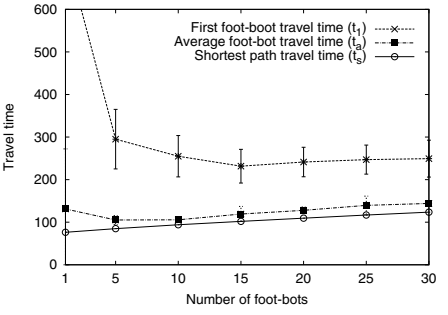


Fig. 2. Travel times for foot-bots in the scenario of Fig. 1(c)

At first, eye-bot policies are uniform, and foot-bots perform random exploration. Once the first foot-bot has reached the target, it can for its way back profit from updated policies. A comparison between t_a and t_1 shows that foot-bots need much less time on average to travel between source and target than during the first run. This shows that the system can learn a path from experience and guide foot-bots between source and target in a cluttered environment. Moreover, t_a is close to t_s, showing that the system finds efficient paths. For increasing numbers of foot-bots, t_1 decreases. This is because multiple foot-bots searching in parallel explore the environment more efficiently. The increase in t_a and t_s for higher numbers of foot-bots is due to congestion. We come back to this in Sect. 7. Finally, we point out that the theoretically best travel time for a foot-bot going between source and destination at maximum speed while passing under eye-bots is 73 s, which is very close to t_s for 1 robot (76 s). This time is dependent on the placement of eye-bots, which defines the possible paths followed by foot-bots. If eye-bots were placed above the shortest path around obstacles, this time could be reduced to 53 s, which indicates that optimal eye-bot placement can be an interesting direction for future research.

6 Shortest Path Finding

In our navigation system, eye-bots play the role of active stigmergic markers for foot-bots. We designed the system after pheromone-based stigmergic foraging in ant colonies. By laying and following pheromone trails, ant colonies are able to converge onto the shortest of multiple paths [12]. This is because the shortest path can be completed faster and more frequently by ants, and therefore receives more pheromone, which in turn attracts more ants. Given the similarities between the basic mechanisms in our system and in ant colonies, we investigate to which extent our swarm exhibits similar shortest path finding abilities.

We used the scenarios of Fig. 3. The source and target locations are connected by two corridors. We vary the ratio $r = l_l/l_r$, where l_l is the length of the left corridor and l_r of the right corridor: we use $r = 1$, $r = 1.5$ and $r = 2$ ($l_r = 20$ m in all scenarios). This setup is derived from the one used with ants in [12]. We use 15 foot-bots, which we deploy one by one with an interval of 30 s.

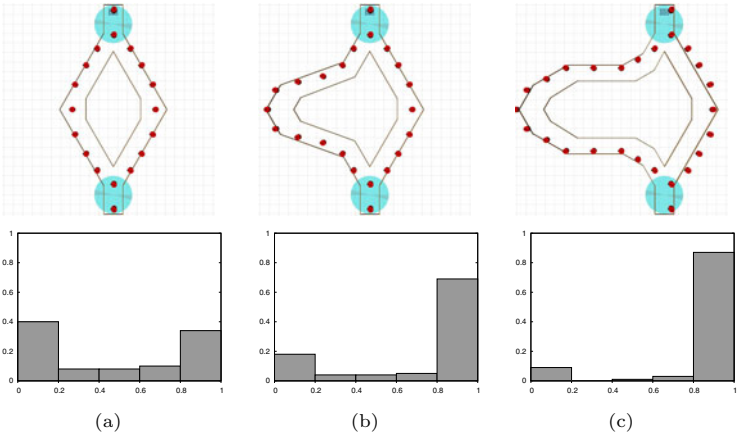


Fig. 3. Double corridor experiments: (a) $r = 1$, (b) $r = 1.5$ and (c) $r = 2$

We gather statistics in the last 1000 s of each experiment, when all foot-bots have been deployed and the system has had time to explore the area. We count over all remaining time steps how many foot-bots use the right corridor, c_r, and how many the left corridor, c_l. We calculate the ratio $\rho = c_r/(c_r + c_l)$, which is near 1 or near 0 if the foot-bots have converged onto respectively the right or the left corridor, and near 0.5 if they use both corridors in similar proportions. In Fig. 3, we show a histogram summarizing the values of ρ measured in 100 test runs (on the x-axis the values of ρ discretized into 5 intervals, on the y-axis the fraction of the 100 runs that falls into each interval).

In the case of equal corridors ($l_r = l_l$), the foot-bots converge on moving over one of them, which can be either the left or the right. In some cases, both corridors are used equally. This behavior is the same as for ants [12]. When corridors are of different length, foot-bots converge more often onto the shortest corridor, and this effect gets stronger as the difference between corridors increases, showing that our system can find shortest paths.

The shortest path behavior of the system is influenced by the number of foot-bots. When only one foot-bot is used, the system converges onto both corridors equally likely. Increasing the number of foot-bots, the system selects the shortest corridor with increasing preference. Finally, when the number of foot-bots increases further, congestion starts to play a role. This is investigated in Sect. 7.

7 Robot Congestion and Self-organized Spreading

In Sect. 5, we showed how the time t_a needed by foot-bots to go between source and target increases for increasing numbers of foot-bots, due to congestion (especially directly under the eye-bots, where foot-bots need to go to get directions). A way to handle congestion is to spread robot traffic over multiple paths. Here we investigate how spreading can be obtained in a distributed self-organized way.

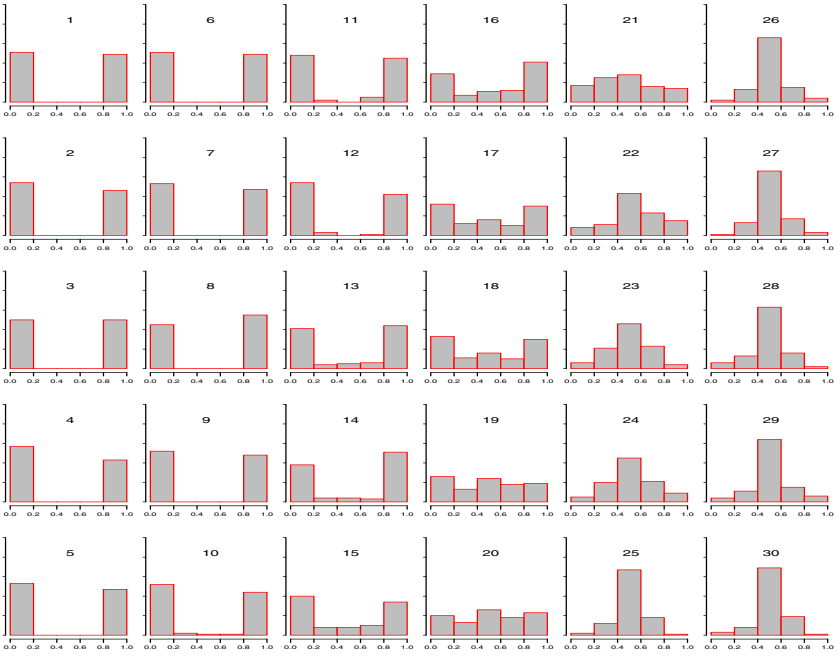


Fig. 4. Frequency histogram of the ratio ρ in the scenario of Fig. 3(a) for the number of foot-bots ranging from 1 to 30. The y-axis scale of all plots ranges from 0 to 1.

Interestingly, ants are capable of traffic spreading. When two paths of equal length are available, they converge onto one when ant traffic is low, and spread over both when traffic is high [14]. This behavior is based on direct interactions between ants: in crowded conditions, ants physically push each other onto different paths. Since robots, like ants, are embodied agents, physical interactions play an important role in their behavior. These interactions increase in case of congestion. A mechanism of traffic spreading similar to that of ants could therefore also be used for robots.

Experiments show that our behavior, without modifications, is capable of traffic spreading. We use the scenario with equal corridors of Fig. 3(a), with increased numbers of foot-bots. In Fig. 4, we show the distribution of the ratio ρ for tests with 1 up to 30 foot-bots. For low numbers of foot-bots, all tests have a ratio ρ of either 0 or 1, indicating that they focus on one of the corridors. As the number of foot-bots grows, the number of tests with intermediate values for ρ increases, indicating that the system increasingly spreads traffic over both corridors. For high numbers of foot-bots, traffic is always spread. The sequence of histograms in Fig. 4 shows how the system displays qualitatively different behavior for increasing numbers of robots: the distribution of ρ evolves from bimodal over uniform to unimodal.

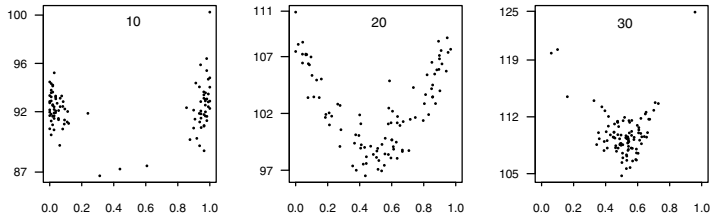


Fig. 5. The average foot-bot travel time (in seconds) vs. the ratio ρ in the scenario of Fig. 3(a) for 10, 20, and 30 foot-bots

Visual investigation revealed that traffic spreading is indeed caused by interactions between foot-bots. The driving factor is the fact that foot-bots execute obstacle avoidance when they bump into each other. As described in Sect. 4, eye-bots observe where foot-bots do obstacle avoidance, and reduce their policies in those directions. This directs foot-bots away from congested areas, so that traffic spreading emerges from the self-organized path finding behavior.

In Fig. 5, we investigate the relation between the ratio ρ and the travel time t_a. We show graphs for 10, 20 and 30 foot-bots. For each of these, we plot t_a against ρ for all 100 test runs. As discussed before, t_a grows for increasing numbers of foot-bots. However, when traffic is spread over two corridors, t_a is lower than when traffic is focused on one corridor, confirming the usefulness of traffic spreading. This difference grows for higher numbers of foot-bots, and the system increasingly chooses intermediate values of ρ.

Traffic spreading also takes place in scenarios with unequal corridors. In the scenario of Fig. 3(b), we observed that for high numbers of foot-bots (more than 15), traffic eventually spreads over both corridors. The system may even send more traffic over the longest corridor, as this can accommodate more foot-bots under less congestion. Reducing congestion this way, the system could reduce foot-bot travel time. However, given the limited information exchange (observed foot-bots form only an implicit feedback about path qualities for eye-bots), the system is not able to choose the point of operation that minimizes foot-bot travel time. In future work, we will improve this by including explicit travel time feedback from foot-bots to eye-bots.

8 Conclusions

We have described a cooperative behavior for heterogeneous swarm robotics to solve a navigation task in a distributed way. It is inspired by pheromone based stigmergic foraging in ant colonies: we let the robots of one sub-swarm function as active stigmergic markers for the other sub-swarm. We showed that our approach can find paths in a cluttered environment, find shortest paths, and spread robot traffic in case of congestion. The system also shows a practically feasible approach to implement pheromone in swarm robotics.

We will develop this system in two directions. First, we want to let eye-bots move and adapt their position based on foot-bot feedback. This lets them search the best locations to give instructions, and reduces the need for full eye-bot coverage of the area. Second, we want to let foot-bots give explicit feed-back about the quality of the paths they follow, to make policy learning more precise.

Acknowledgments

This work was supported by the SWARMANOID project, funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission under grant IST-022888. The information provided is the sole responsibility of the authors and does not reflect the Commission's opinion. The Commission is not responsible for any use made of data appearing in this publication.

References

1. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, Oxford (1999)
2. Stirling, T., Wischmann, S., Floreano, D.: Energy-efficient indoor search by swarms of simulated flying robots without global information. Swarm Intelligence 4(2) (June 2010)
3. Garnier, S., Tache, F., Combe, M., Grimal, A., Theraulaz, G.: Alice in pheromone land: An experimental setup for the study of ant-like robots. In: Proceedings of the IEEE Swarm Intelligence Symposium, SIS (2007)
4. Sharpe, T., Webb, B.: Simulated and situated models of chemical trail following in ants. In: Proceedings of SAB 1998 (1998)
5. Sugawara, K., Kazama, T., Watanabe, T.: Foraging behavior of interacting robots with virtual pheromone. In: Proceedings of IEEE/RSJ IROS (2004)
6. Vaughan, R., Støy, K., Sukhatme, G., Mataric, M.: Whistling in the dark: Cooperative trail following in uncertain localization space. In: Proceedings of the Fourth International Conference on Autonomous Agents (2000)
7. Johansson, R., Saffiotti, A.: Navigation by stigmergy: A realization on an rfid floor for minimalistic robots. In: Proceedings of IEEE ICRA (2009)
8. Payton, D., Daily, M., Estowski, R., Howard, M., Lee, C.: Pheromone robotics. Autonomous Robots 11(3) (November 2001)
9. Batalin, M., Sukhatme, G., Hattig, M.: Mobile robot navigation using a sensor network. In: Proceedings of IEEE ICRA (2004)
10. O'Hara, K., Balch, T.: Pervasive sensor-less networks for cooperative multi-robot tasks. In: Proceedings of DARS 2004 (2004)
11. Vigorito, C.: Distributed path planning for mobile robots using a swarm of interacting reinforcement learners. In: Proceedings of AAMAS (2007)
12. Goss, S., Aron, S., Deneubourg, J.L., Pasteels, J.M.: Self-organized shortcuts in the Argentine ant. Naturwissenschaften 76, 579–581 (1989)
13. Ducatelle, F., Di Caro, G.A., Gambardella, L.: Cooperative self-organization in a heterogeneous swarm robotic system. In: Proceedings of GECCO (2010)
14. Dussutour, A., Fourcassié, V., Helbing, D., Denebourg, J.L.: Optimal traffic organization in ants under crowded conditions. Nature 428, 70–73 (2004)

How to Pick the Right One: Investigating Tradeoffs among Female Mate Choice Strategies in Treefrogs

Matthias Scheutz, Jack Harris, and Sunny K. Boyd

[1] Cognitive Science Program, Indiana University, Bloomington, IN 47404
{mscheutz,jackh}@indiana.edu
[2] Department of Biology, University of Notre Dame Notre Dame, IN 46556
boyd.1@nd.edu

Abstract. Two main theories of female mate choice, that females either pick the best from the n closest males (*best-of-n*) or the closest with some minimum quality (*min-threshold*), make different behavioral predictions in some cases, yet both are supported by biological data. We present a computational agent-based model that is well-suited for investigating the differences between the two strategies for the biological model organism *Hyla versicolor* ("gray treefrog"). We show, based on results from systematic simulation studies, that *min-threshold* overall is the dominant strategy, even though *best-of-n* has some areas in parameter space where it dominates *min-threshold*.

Keywords: female choice, mating strategies, agent-based modeling.

1 Introduction

Most studies of mate choice assume that females "choose" a single mate from a group of eligible males based on some criteria. Female treefrogs, for example, show phonotaxis toward calls of males with higher pulse numbers [4,10]. Females are thus assumed to make an active choice [11], show a directional bias (more pulses are better [19]), and differentiate between individual males up to a maximum of 5 [9]. While there are several proposed rules for female sampling and decision making [14], the most prevalent theories suggest that females choose either the "best" of the closest $n = 1,\ldots,5$ – *best-of-n* theory [13] – or the first male they encounter whose quality is above a minimum threshold for acceptance – *min-threshold* theory [14].

Yet, as Jennions and Petrie [14] point out "it has proved difficult to distinguish which tactics are used by females in the field". In fact, there is empirical evidence for both theories, even though they predict different choices in some cases. In this paper, we attempt to contribute to resolving the apparent contradiction in the empirical data using a social agent-based simulation model of female choice in treefrogs.[1]

We start by introducing our agent-based model, which consists of frog and environmental models based as much as possible on biological data. Then we present a thorough analysis of the results and show how both *best-of-n* and *min-threshold* strategies dominate each other in different regions of parameter space for different performance

[1] This work was supported in part by NSF grant #0725187 to the first and third authors.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 618–627, 2010.

measures with *min-threshold* dominating in vast parts of parameter space. The subsequent discussion briefly reflects on some biological implications of our findings and proposes an empirically testable hypothesis to resolve the apparent contradiction in the biological data. The conclusion briefly summarizes our findings and proposes some model extensions for future work.

2 The Model

The aim of our agent-based model is to investigate the social interactions of male and female agents in a spatially explicit environment, using amphibians as the biological model organisms. It was specifically designed to generate and evaluate hypotheses about the dynamics of mating with focus on trade-offs among different female mating strategies in both artificial and natural conditions. The model includes only essential variables for the mating scenario: variables for *female choice strategies*, *male-female sex ratio*, and *male call quality* as well as parameters for the *distributions of males and females*. For simplicity, male amphibian social interactions with other males were not investigated at this stage.

We model male and female frogs using "male agents" and "female agents" in a continuous two-dimensional environment called the "swamp", with dimensions E_x and E_y.[2] While the goal for all agents is to mate, only females can choose their mate.

Environment. Each male agent i is placed in the swamp at location $\langle i_x, i_y \rangle$ in a territory of radius r_τ according to the male distribution m_δ. Male agents always have to be inside the swamp and cannot be placed in another male's territory. I.e., the position $\langle i_x, i_y \rangle$ of male agent i must satisfy the following restrictions for all positions $\langle j_x, j_y \rangle$ of males $j \neq i$: $[d(i,j) > 2 \cdot \tau] \wedge [i_x + r_\tau < E_x] \wedge [i_y + r_\tau < E_y] \wedge [i_x - r_\tau > 0] \wedge [i_y - r_\tau > 0]$ where $d(i,j) = \sqrt{(i_x - j_x)^2 + (i_y - j_y)^2}$ is the Euclidean distance between agents i and j (note that male agents do not move). Different from males, female agents are always placed in locations on the border of the swamp according to the female distribution f_δ. Mating between a female agent i and a male agent j always occurs when they are within *mating distance* d_{mate}, i.e., when $d(i,j) \leq d_{mate}$.

Agents. Both male and female agents have states that fully describe their make-up at any given time (Tab. 1 top). The state of a male agent i consists of its location in the swamp $\langle i_x, i_y \rangle$, its size i_{size}, as well as it pulse number i_{pn}. The state of a female agent i consists of its location in the swamp $\langle i_x, i_y \rangle$, its size i_{size}, its velocity i_v, its orientation α and and its mating strategy $i_{\pi(n)}$.

2.1 Model Parameters

We divide the model parameters into two sets: *fixed* and *free*. Fixed parameters (Tab. 1 middle) were set to known constant values from the literature and kept the same for all simulation experiments. In contrast, free parameters (Tab. 1 bottom) are parameters for

[2] To avoid confusion between biological and simulated entities, we will refer to simulated frogs as "agents". We will use the gender attributes "male" and "female" alone if it is clear from the context whether we refer to simulated or real (male or female) frogs.

Table 1. State variables of male and female agents (top), fixed model parameters kept at constant values for all simulations based on the literature (middle), and free model parameters and their variation (bottom)

Symbol	Parameter Meaning	Initial Value	Update/Rationale
Male agents (state var)			
$size$	size	given by m_{size}	constant
$\langle x,y \rangle$	position	based on m_δ	constant
pn	pulse number	based on μ_{pn} and σ_{pn}	constant
Female agents (state var)			
v	velocity	given by f_v	constant
$size$	size	given by f_{size}	constant
α	heading$\in$[0,360] degrees	based on f_δ	calculated
$\langle x,y \rangle$	position	based on f_δ	calculated
$\pi(n)$	mating strategy with strategy parameter n	given by f_π together with strategy parameter values	constant
Environment (fixed)			
E_x	swamp width	10m	see [17,6]
E_y	swamp height	25m	see [17,6]
d_{mate}	mating distance	4cm	model-specific
r_τ	(male) territory radius	50cm	model-specific
n_{male}	the number of males	25	model-specific
Individual (fixed)			
m_{size}	male size	4.72 cm	see [7]
f_{size}	female size	5.38 cm	see [18]
f_v	female velocity	1.86 cm/s	see [3]
Environment (free)			
n_{female}	the number of females	1..20	model-specific
f_δ	female distribution	random at swamp edge	model-specific
m_δ	male distribution	Gaussian	model-specific
Individual (free)			
μ_{pn}	mean male pulse number	6..24 in increments of 2	model-specific
$\pi(n)$	mating rule	*bestofn minthresh*	see [3,4,10]
Strategy (free)			
θ	minimum threshold	6..24 in increments of 2	model-specific
ν	number of sampled males	1..5	model-specific

which there is either no biological data available or no consensus on what the right value is supposed to be. They can be divided into *strategy parameters*, *individual* and *environmental* parameters. *Strategy parameters* concern only the strategy used by female agents in their selection of mates. *Individual parameters* concern the behavior of individual agents, and *environmental parameters* concern the make-up of the swamp, i.e., the number of agents, their locations, and their interactions (e.g., mating). Together, the free parameters form a *multidimensional parameter space* whose dimensions depend on the variations and particular samplings of each free parameter. We will first describe the variations of the free parameters and then provide details of the experimental setup.

Environmental parameters. The swamp is $E_x = 10$ by $E_y = 25$ meters which allows all female agents to sample every male agent. The male agents' territory radius r_τ was fixed at 50 cm. Males were placed in the swamp according to a Gaussian distribution with means $\mu_x = E_x/2$ and $\mu_y = E_y/2$ and standard deviations $\sigma_x = E_x/4$ and $\sigma_y = E_y/4$, respectively, resulting in a greater density of males in the center of the swamp. Since the number of males n_{males} was kept constant for all experiments at 25, the number of females was varied from 1 to 20 yielding different male-female sex ratios (from 0.04 to 0.82) at the beginning of each run (which, of course, changes throughout the run as females and males mate).

Individual parameters. The pulse number for individual males was assigned based on sampling a Gaussian distribution with mean μ_{pn} and standard deviation σ_{pn}, where $\mu_{pn} \in \{6,8,10,12,14,16,18,20,22,24\}$ and $\sigma_{pn} = 2$. The two biologically plausible strategies π (kept constant throughout each simulation run) are *best-of-closest-n* (*best-of-n*) and *closest-above-minimum-threshold* (*min-threshold*) [3,4,10]. Let $c(i,X) = \{j \in X | \neg\exists k \in X[d(k,i) < d(j,i)]\}$, which is the set of the closest agents from set X to the given agent i (usually this will only contain one agent, but there could be multiple agents that are equidistant to i). Let M denote the set of all male agents in the swamp.

- *Bestofn.* Let $c^n(i,X)$ denote the set of the n closest agents from set X with respect to the location of agent i.[3] Then the selected male agent is $\underset{j \in c^n(i,M)}{\operatorname{argmax}}(j_{pn})$ for the female i, i.e., the male with highest pulse number in the set of the closest n males.
- *Minthresh.* The selected male agent is $\underset{j \in c(i,\{l \in M | l_{pn} \geq i_\theta\})}{\operatorname{argmax}}(j_{pn})$, where i_θ is the minimum threshold of female agent i, i.e., the male with the highest pulse number above the minimum threshold among the closest males.

Note that we are only considering homogeneous populations of females, so that every female in a given simulation run has the same mating strategy.

Strategy parameters. For *bestofn*, we vary its parameter $\nu \in \{1,2,3,4,5\}$ and for *min-threshold*, we vary its parameter $\theta \in \{6,8,10,12,14,16,18,20,22,24\}$ (the upper limit of $\nu = 5$ was chosen based on prior model results [13] and empirical data [14]).

2.2 Model Algorithm and Implementation

The model was implemented in the Java programming language within the *Repast Simphony* agent-based simulation environment and run as a discrete-time simulation where each update cycle corresponds to one second in real-time (the pseudo-code of the main simulation loop is given below).

The males' positions and pulse numbers and females' positions and strategies are initialized based on a random seed (in *initializeMales* and *initializeFemales*).[4] In the

[3] $c^n(i,X)$ can be defined inductively as follows: $c^0(i,X) = \emptyset$ and $c^{n+1}(i,X) = c^n(i,X) \cup \{j \in X - C^n(i,X) | \neg\exists l \neq j(l \in (X - C^n(i,X)) \wedge d(l,i) < d(j,i)\}$.

[4] Females are incrementally placed, hence the location of the fourth female, for example, will be the same in all simulations with four or more females with the same random seed. This is to allow analyses of females across different male-female ratios.

Algorithm 1. Pseudo code of the model simulation algorithm.

```
Simulation(seed, maleList, femaleList)
  initializeMales(seed, maleList)
  initializeFemales(seed, femaleList)
  terminate ← false
  step ← 0
  while ¬terminate do
    step ← step + 1
    for all f ∈ femaleList, m ∈ maleList do
      if withinMatingDistance(f, m) then
        report_mated_and_remove(f, femaleList)
        report_mated_and_remove(m, maleList)
      end if
    end for
    for all f ∈ femaleList do
      updateState(f)
    end for
    terminate ← checkTerminationConditions()
  end while
```

main loop, female agents sample male agents at every time step, select a candidate male according to their strategy π (and strategy parameters n) and then approach the selected male at their *velocity* [3]. This sequence models the female frogs' behavior of sampling the males, selecting a potential mate, and leaping towards him [11]. The female's heading angle α is updated according to the direction to the chosen male and the female's new position $\langle x, y \rangle$ is updated based on the female's velocity v and heading α. At every step, for every male and female agent, their mating distance is compared to the mating range (*withinMatingDistance*): if a female is within d_{mate} of a male, both are reported as mated and removed from their respective lists (*report_mated_and_remove*), otherwise the female state is updated (*updateState*) (there is no need to update the male state because males neither move nor change their pulse number). Because there are always fewer females than males in all of our simulations [22], the termination condition for the simulation depends on the females' mating strategy: the simulation ends when either all females have mated (as will always be the case in *bestofn*) or when no more males above the female *min-threshold* θ are left (*checkTerminationConditions*), in which case none of the remaining females will ever mate.

3 Results

We ran 100 simulations with different random initial conditions (based on different random seeds) for each point in the multidimensional parameter space given by the free model parameters for a total of almost 1 million simulations. An analysis of variance (ANOVA) of the results with *number of females* (nf), *male pulse number* (pn), *strategy* (strat) and *strategy parameter* (sp) as independent variables, and *male quality* as dependent variable, shows highly significant main effects for all four variables,

Table 2. The results of the full ANOVA model for average mated male pulse number as dependent variable (see text for details). The bold-face values are highly significant p-values.

Variable	Df	F value	Pr(>F)	Variable	Df	F value	Pr(>F)
nf	1	406.17	<**.001**	nf×sp	1	289.73	<**.001**
pn	1	1098300	<**.001**	pn×sp×strat	1	1028.2	<**.001**
sp	1	168090	<**.001**	nf×sp×strat	1	248.73	<**.001**
strat	1	8384.5	<**.001**		—	—	—
nf×strat	1	265.25	<**.001**	nf×pn	1	1.12	.29
pn×strat	1	3765.2	<**.001**	nf×pn×strat	1	0.002	.96
sp×strat	1	10972	<**.001**	nf×pn×sp	1	0.72	.39
pn×sp	1	45679	<**.001**	nf×pn×sp×strat	1	0.01	.92

highly significant two-way interactions (except for nf×pn, which was not significant as expected), and two highly significant three-way interactions (pn×strat×sp, as well as nf×strat×sp); the four-way interaction was not significant (see Tab. 2).

Overall, the average quality of mated males using *best-of-n* is sensitive to the male-female sex ratio (Fig. 1 top), different from *min-threshold* where females do not mate below their threshold (leading to no matings in setups where the *min-threshold* exceeds the male pulse number). Both strategies show (almost) linear increase in mated male quality as a function of average male pulse number (Fig. 1 middle), with *best-of-n* having a steeper slope than *min-threshold*.

To be able to compare the two strategies directly, we introduce the notion of "dominance" where a strategy S with parameter value set P is said to *dominate* a strategy T with parameter value set Q if there exists a parameter value p for strategy S such that for all parameter values q for strategy T it is the case that S_p (i.e., S with parameter value p) leads to significantly higher performance (at a given α level) than T_q (i.e., T with any of its parameter values q); symbolically, $\exists p \in P \forall q \in Q$ **Perf**$(S_p) >_\alpha$**Perf**(T_q), where **Perf**(S_p) is the performance of S with parameter value p (in the mating task) and $>_\alpha$ denotes a statistically significant ordering at the α level (e.g., p-values $< \alpha = 0.05$ in a paired t-test). This notion of dominance is of particular interest from an evolutionary perspective because a dominant strategy is one that will likely evolve in competition with other strategies (because there is no parameter value for the non-dominant strategy that will lead to even equal performance). Comparing the dominance of *min-threshold* and *best-of-n*, we can see (Fig. 1 bottom left) that for average mated male quality *min-threshold* dominates *best-of-n* for most male pulse number and male-female ratios, only leaving the small area for the highest average male pulse number and low to medium male-female ratios for *best-of-n* to dominate.[5] Moreover, *min-threshold* is also superior in terms of time-to-mating (Fig. 1 bottom right), with no place for *best-of-n* to dominate. Note, however, that both parameter spaces have areas where neither strategy dominates.

[5] Note that the small region of domination for *best-of-n* seems to be in conflict within the top plots of Fig. 1 as *min-threshold* seems to lead to overall better average mated male quality than *best-of-n* for all sex ratios. However, the numbers for *min-threshold* are as high because in the averages across the 100 random conditions we excluded those runs where *min-threshold* did not lead to any mating. Yet, in the dominance plots those simulations are taken into account in the statistical comparison.

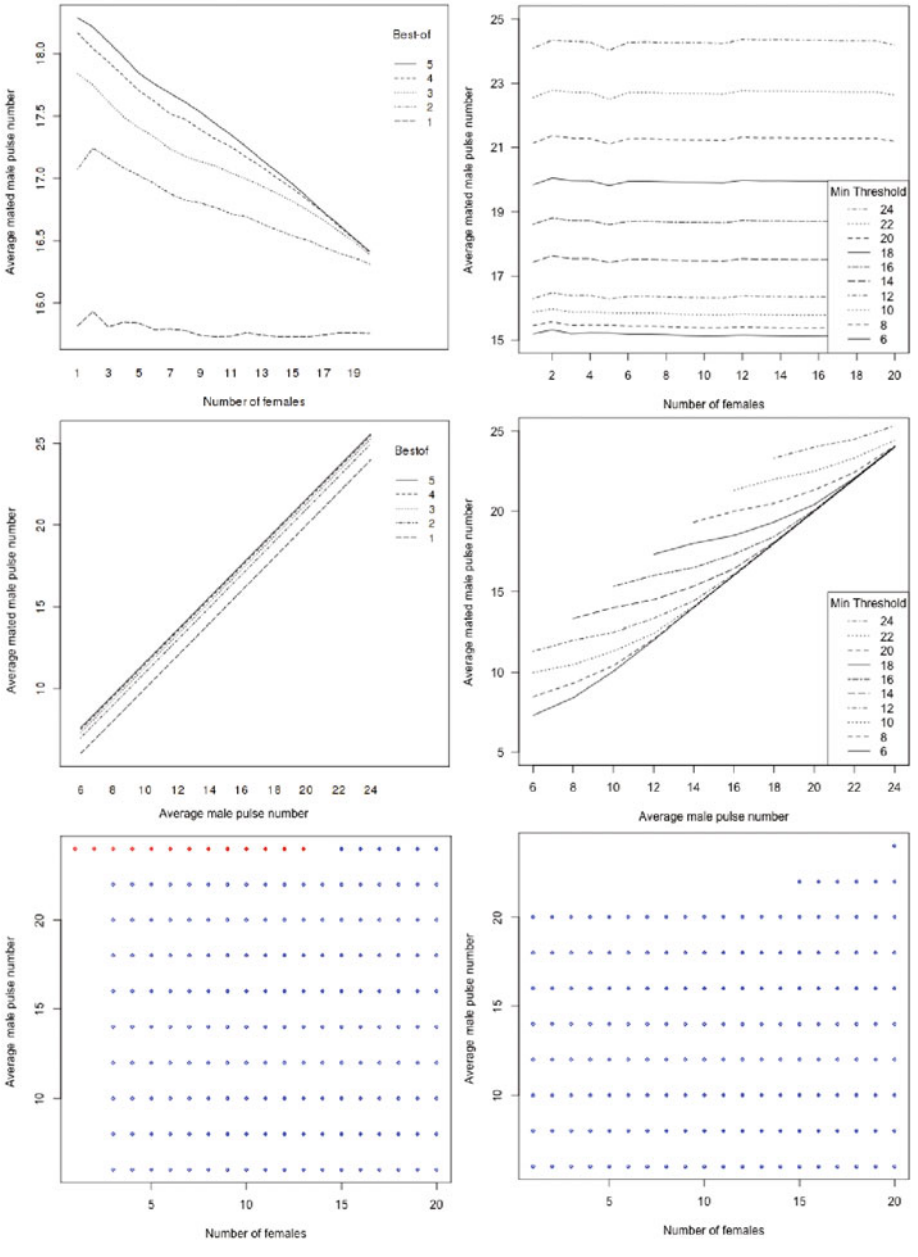


Fig. 1. Interactions for *best-of-n* (top) and *min-threshold* (middle) for average quality of mated males. The missing line segments in *min-threshold* (middle right) for some x-values indicate that there were scenarios without matings because *min-threshold* exceeds the male pulse numbers. The bottom shows dominance plots for *best-of-n* (red) and *min-threshold* (blue) for average male quality (bottom left) and average time-to-mating (bottom right).

4 Discussion

The above results suggest that the *min-threshold* strategy is very attractive from an evolutionary perspective. Females should be able to adjust their level of choosiness, given the lack of predictability in available mates and other aspects of natural environments [1]. The *min-threshold* strategy allows animals to change the threshold without changing the strategy itself, while requiring little complexity in terms of the cognitive architecture. In fact, empirical evidence in female frogs suggests that thresholds are not set via an active cognitive process (e.g., not via comparison of memories of calls heard in the past), but by hormonal changes [15]). This may explain why female frogs sometimes do not find mates – hormone levels may vary independently of the quality of the male population. On the other hand, the memory and neural integration requirements of *best-of-n* (e.g., in the case of $n = 5$) would be substantial, requiring females to remember the quality and location of males heard in the past [13].

The *min-threshold* strategy should also be adopted when the costs of comparing multiple mates is high [24]. Our simulation included no costs to sampling. Therefore, our study shows that the *min-threshold* strategy can be superior even in an environment without sampling costs. Another critical aspect is the time an animal spends searching for a mate. Not only are energy resources used during the search, but the individual may be at greater risk (e.g., from predation) and/or physiological processes might limit fertile times [14]. A unique feature of our spatially-explicit model is that time-to-mating is an outcome of the simulation (rather than a cost predicted a priori).

So why is it then that there is empirical evidence for *best-of-n* when *min-threshold* is superior in all the above respects? We believe the answer might lie in the fact that female treefrogs are usually only present at the breeding site on the night they mate and lay eggs. Decisions made on a single night will thus have significant impact on male fitness. And if a female's threshold is set too high so that none of the males in the swamp can meet it, she will not be able to mate. This is the point where switching strategies might help, i.e., if a female even after lowering her threshold over time can still not find an eligible male, then switching to *best-of-n* (for $n > 1$) will at least guarantee that she will find a partner close to the (remaining) male population's average pulse number (Fig. 1 top left). Mating with one of the remaining "average males" might in the end still be better than not mating at all, for females can only mate over a very limited time period, or they lay their eggs unfertilized. If females did switch strategies for the above reason, then it would not be surprising that some past empirical studies have found female choices to be consistent with the *best-of-n* strategy.

5 Related Work

Female choice has been extensively studied in a variety of scenarios, ranging from simulated mating strategy in animals such as fruit flies [16] to humans [21,12,5]. In the context of frogs, for example, Baugh and Ryan [2] investigated the propensity of a female to follow one mate selection strategy versus another. Tárano [23] researched the spacial orientation of male treefrogs during mating time. Our research leverages such empirical work to build models of our agents and the factors that play a critical role in the mating scenario.

There have also been attempts using neural network simulations to understand general mechanisms dictating mating preferences based on evolved sensory bias. Fuller [8], for example, shows modest connections between sensory bias and mate selection. Similarly, our work explores the range of sensory bias level (*min-threshold*) of female treefrogs when selecting a mate based on male call features. However, our study is less focused on how such biases develop and more focused on the ramification of such biases on overall utility of the resulting mate selection strategy.

While these studies focus on different aspects and properties of the mate choice at different levels of detail for different reasons, our study is the first that investigates female choice in the context of treefrogs in great detail over a large parameter space in order to contribute to the resolution of an open biological question, namely which strategy – *min-threshold* or *best-of-n*– female treefrogs might use to select their mates.

6 Conclusion and Future Work

The spatially-explicit frog mating simulation provides a unique and powerful method for testing predictions about mate choice behavior. The comprehensive investigation of two female choice strategies, *min-threshold* and *best-of-n*, across a large parameter space demonstrated that *min-threshold* is a superior strategy with respect to average mated male quality and time-to-mating. The main downside of *min-threshold* seems to be the lack of mating in cases where male call rates are lower than the female's threshold. We speculated that frogs might at some point switch strategies in order to be able to mate at all, which would explain the apparent inconsistency found in the empirical data. With the current results in hand, it is now possible to design empirical experiments with real frogs and arrays of speakers mimicking male callers in order to test specific predictions of behavior in a complex environment.

In the future, we plan to extend the agent-based model in several ways. For one, we only investigated one male distribution here, but it might be interesting to see if the above results will still be valid for other possible male (and also female) distributions. Moreover, we will also extend the model to include a male aggression model together with strategies for handling aggressive encounters (e.g., based on our previous general aggression model [20]). Finally, it would also be interesting to investigate evolutionary scenarios with multiple generations of frogs and possibly additional female strategies in order to compare the outcomes to the results of this study.

References

1. Ball, G., Sockman, K., Duffy, D., Gentner, T.: A neuroethological approach to song behavior and perception in European starlings: interrelationships among testosterone, neuroanatomy, immediate early gene expression, and immune function. Adv. Stud. Behav. (36), 59–121 (2006)
2. Baugh, A.T., Ryan, M.J.: Female tungara frogs vary in commitment to mate choice (2009)
3. Beckers, O.M., Schul, J.: Phonotaxis in Hyla versicolor (Anura, Hylidae): the effect of absolute call amplitude. Journal of Comparative Physiology a – Neuroethology Sensory Neural and Behavioral Physiology 190(11), 869–876 (2004)

4. Bush, S.L., Gerhardt, H.C., Schul, J.: Pattern recognition and call preferences in treefrogs (Anura: Hylidae): a quantitative analysis using a no-choice paradigm. Animal. Behaviour 63, 7–14 (2002)
5. Eriksson, K., Strimling, P.: Partner search heuristics in the lab: Stability of matchings under various preference structures. Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems 17(6), 524–536 (2009)
6. Fellers, G.M.: Aggression, territoriality, and mating-behavior in North-American treefrogs. Animal Behaviour 27(FEB), 107–119 (1979)
7. Fellers, G.M.: Mate selection in the gray treefrog, Hyla-versicolor. Copeia (2), 286–290 (1979)
8. Fuller, R.C.: A test of the critical assumption of the sensory bias model for the evolution of female mating preference using neural networks. Evolution 63(7), 1697–1711 (2009)
9. Gerhardt, H.C.: Female mate choice in treefrogs - static and dynamic acoustic criteria. Animal. Behaviour 42, 615–635 (1991)
10. Gerhardt, H.C., Tanner, S.D., Corrigan, C.M., Walton, H.C.: Female preference functions based on call duration in the gray tree frog (Hyla versicolor). Behavioral Ecology 11(6), 663–669 (2000)
11. Gibson, R.M., Langen, T.A.: How do animals choose their mates? Trends in Ecology & Evolution 11(11), 468–470 (1996)
12. Hooper, P.L., Miller, G.F.: Mutual mate choice can drive costly signaling even under perfect monogamy. Adaptive Behavior 16(1), 53–70 (2008)
13. Janetos, A.C.: Strategies of female mate choice - a theoretical analysis. Behavioral Ecology and Sociobiology 7(2), 107–112 (1980)
14. Jennions, M.D., Petrie, M.: Variation in mate choice and mating preferences: A review of causes and consequences. Biological Reviews of the Cambridge Philosophical Society 72(2), 283–327 (1997)
15. Lynch, K.S., Crews, D., Ryan, M.J., Wilczynski, W.: Hormonal state influences aspects of female mate choice in the Tungara Frog (Physalaemus pustulosus). Hormones and Behavior 49(4), 450–457 (2006)
16. Noor, M.A.: Effects of dominance on runaway sexual selection. Adaptive Behavior 8(3-4), 255–265 (2000)
17. Ptacek, M.B.: Calling sites used by male gray treefrogs, Hyla versicolor and Hyla chrysoscelis, in sympatry and allopatry in Missouri. Herpetologica 48(4), 373–382 (1992)
18. Ritke, M.E., Semlitsch, R.D.: Mating-behavior and determinants of male mating success in the gray treefrog, Hyla-chrysoscelis. Canadian Journal of Zoology-Revue Canadienne De Zoologie 69(1), 246–250 (1991)
19. Ryan, M.J., Keddyhector, A.: Directional patterns of female mate choice and the role of sensory biases. American Naturalist 139, S4–S35 (1992)
20. Scheutz, M., Schermerhorn, P.: The more radical, the better: Investigating the utility of aggression in the competition among different agent kinds. In: Proceedings of SAB 2004. MIT Press, Cambridge (2004)
21. Simão, J., Todd, P.M.: Modeling mate choice in monogamous mating systems with courtship. Adaptive Behavior 10(2), 113–136 (2002)
22. Sullivan, B.K., Hinshaw, S.H.: Female choice and selection on male calling behavior in the gray treefrog Hyla versicolor. Animal Behaviour 44(4), 733–744 (1992)
23. Tárano, Z.: Structure of transient vocal assemblages of Physalaemus fischeri (Anura, leiuperidae): Calling site fidelity and spatial distribution of males. South American Journal of Herpetology 4(1), 43–50 (2009)
24. Wittenberger, J.: Tactics of mate choice in mate choice. In: Bateson, P. (ed.) Mate Choice, pp. 435–447. Cambridge University Press, Cambridge (1983)

Autonomous Development of Social Referencing Skills

Sofiane Boucenna[1], Philippe Gaussier[1,2], Laurence Hafemeister[1], and Kim Bard[3]

[1] ETIS, CNRS UMR 8051, ENSEA, Univ Cergy-Pontoise
[2] IUF
[3] Psychology University of Portsmouth
{boucenna,gaussier,hafemeister}@ensea.fr, kim.bard@port.ac.uk

Abstract. In this work, we are interested in understanding how emotional interactions with a social partner can bootstrap increasingly complex behaviors such as social referencing. Our idea is that social referencing as well as facial expression recognition can emerge from a simple sensori-motor system involving emotional stimuli. Without knowing that the other is an agent, the robot is able to learn some complex tasks if the human partner has some "empathy" or at least "resonate" with the robot head (low level emotional resonance). Hence we advocate the idea that social referencing can be bootstrapped from a simple sensori-motor system not dedicated to social interactions.

1 Introduction

How can a robot or a human learn more and more complex tasks? This question is becoming central in robotics and psychology. In this work, we are interesting in understanding how emotional interactions with a social partner can bootstrap increasingly complex behaviors. This study is important both for robotics application and understanding development. In particular, we propose that social referencing, gathering information through emotional interaction, fulfills this goal. Social referencing, a developmental process incorporating the ability to recognize, understand, respond to and alter behavior in response to the emotional expressions of a social partner, allows an infant to seek information from another individual and use that information to guide his behavior toward an object or event [14].

Gathering information through emotional interaction seems to be a fast and efficient way to trigger learning. This is especially evident in early stages of human cognitive development, but also evident in other primates [19]. Social referencing ability might provide the infant, or a robot, with valuable information concerning the environment and the outcome of its behavior, and is particularly useful since there is no need for verbal interactions. In social referencing, a good (or bad) object or event is identified or signaled with an emotional message, not with a verbal label. The emotional values can be provided by a variety of modalities of emotional expressions, such as facial expressions, voice, gestures, etc. We

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 628–638, 2010.

choose to use facial expressions since they are an excellent way to communicate important information in ambiguous situations but also because they can be learned autonoumously very quickly [4]. Our idea is that social referencing as well as facial expression recognition can emerge from a simple sensori-motor system. All the work is based on the idea of the perception ambiguity: the inability at first to differentiate our own body from the body of others if they are correlated with our own actions. This perception ambiguity associated to a homeostatic system is sufficient to trigger first facial expression recognition and next to learn to associate an emotional value to an arbitrary object. Without knowing that the other is an agent, the robot is able to learn some complex tasks. Hence we advocate the idea that social referencing can be bootstrapped from a simple sensori-motor system not dedicated to social interactions.

2 Overview

Our social referencing experiment (fig. 1,2) has the following set-up: a robotic head having one camera is able to recognize facial expressions and another camera is turned toward a workspace where a Katana arm is able to reach an object. As a consequence, the robot (head plus arm) can interact with the environment (human partner) and can manipulate objects. In the developed architecture, the robot learns to handle positive objects and to avoid negative objects as a direct consequence of emotional interactions with the social partner. The robotic head learns to recognize emotional facial expressions (sadness, joy, anger, suprise and neutral face) autonomously [4]. The internal emotional state of the robot triggers one specific expression and the human mimicks the robot face to face. The robot can learn to associate its internal emotional state with the human's facial expression. The robot associates what it is doing with what it is seeing. After few minutes of real time learning (typically less than 3 minutes), the robot is

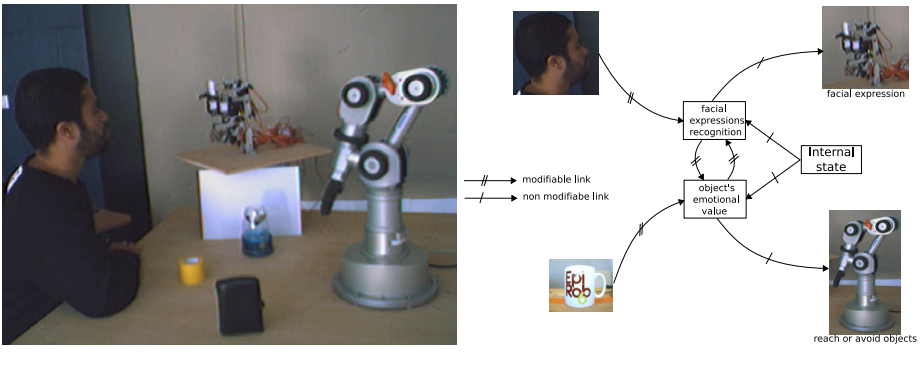


Fig. 1. Experimental set-up for social referencing. The robot relies upon the use of its expressive head which is also able to recognize facial expressions. the robotic arm will reach the positive objects and avert the negative objects after emotional interactions with a human partner.

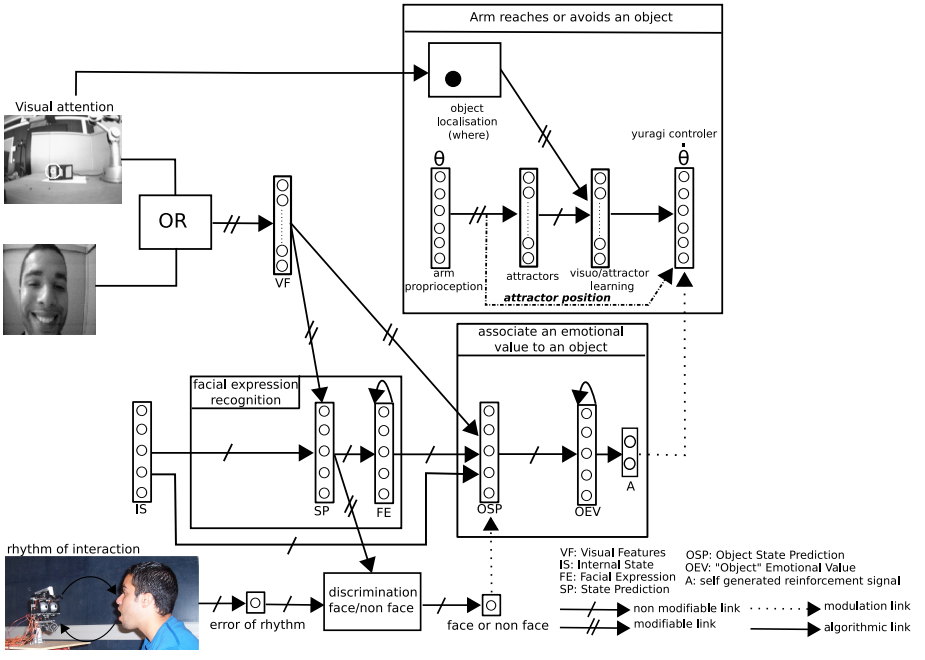


Fig. 2. Social referencing model. Social referencing emerging from the sensori-motor interactions between facial expression recognition, objects emotional value and visuo-motor learning. A simple sensori-motor architecture is able to learn and recognize the facial expressions, and then to discriminate between facial/non facial stimuli. Using a simple chain of conditioning, the robot learns the emotional value of an object as a result of the interactions with the human (face discrimination). The robot focuses on an object using a visual attention processus (Gabor filters, color). After a visuo-motor learning, the robot arm reaches or avoids some objects in the workspace thanks to the self generated reinforcement signal A(emotional value coming from the facial expression recognition). A is built as the result of the facial expression recognition (with A_1 neuron corresponding to happy facial expression, the A_2 neuron corresponding to angry facial expression)

able to recognize the human facial expressions as well as to mimick them. In parallel, the eye-arm system can learn throw visuo-motor asociation to reach several positions in the workspace [1]. A reinforcing signal is used to reach or avoid a position in the workspace [9]. The signal can be an emotional signal (e.g joy facial expression is a positive signal and an angry facial expression is a negative signal).

The tested scenario is the following: The robot is in a neutral emotional state, a human displays a joy facial expression in the presence of an object, consequently the robot moves to a joy state and associates a positive value to the object. On the contrary if the human displays an anger facial expression, the value associated to this object is negative. The robot arm can handle or avoid the objects according to their associated emotional value. In other words, the

emotional value associated to the object is the reinforcing signal that the arm uses so as to move.

In the following paper, we will see a developmental approach of the social referencing namely: the development of facial expressions recognition, the association of emotional value to an object and finally an aspect of motor control according to emotional stimuli.

3 Online Learning of Facial Expression Recognition

At this stage of the development, the robot must be able to recognize and understand the caregiver facial expressions. We summarize here an architecture that we developped for online learning of facial expression recognition. A simple sensory-motor architecture is able to express several emotions and to recognize online the facial expression of a caregiver if this latter naturally tends to imitate the system or to resonate with it. In particular, we showed that autonomous learning of face/non face discrimination is more complex than the facial expression recognition [4].

Using the cognitive system algebra [11], we showed that a simple sensory-motor architecture based on a classical conditioning paradigm [20,2] can learn to recognize facial expressions online. Furthermore, the dynamics of the human-robot interaction bring important but non explicit signals, such as the interaction rhythm that helps the system to perform the face/non face discrimination. The interaction rhythm is used to allow first a robust learning of the facial expression without face tracking and next to perform the learning of the face/non face discrimination. Psychologists underline the importance of the synchrony during the interaction between the mother and the baby [7]. If a rhythmic interaction between baby and mother involves positive feelings and smiles (positive reward), a social interaction interuption involves negative feelings (negative reward). In our case (following [1]), the rhythm is used as a reward signal. It provides an interesting reinforcement signal to learn to recognize an interacting partner(face/non face).

We adopt the following experimental protocol: the facial expressions of the robotic head have been calibrated by FACS experts [8]. In the first phase of interaction, the robot produces a random facial expression during 2s (among the following: sadness, happiness, anger, surprise), then returns to a neutral face during 2s to avoid human misinterpretations of the robot facial expression (same procedure as in psychological experiments). The human subject is explicitely asked to mimic the robot head (even without any instruction, psychologist have shown that the human subject resonates with the facial expressions of the robot head [17]). This first phase lasts between 2 and 3 minutes depending on the subject "patience". Then, in the second phase, the random emotional states generator is stopped. After the N.N (Neural Network) has learned, the robot mimics the human partner facial expressions. This architecture (see fig. 3) allows the robot to recognize the subjects visual features and to learn if these features are correlated with the robot own facial expressions. Moreover, another

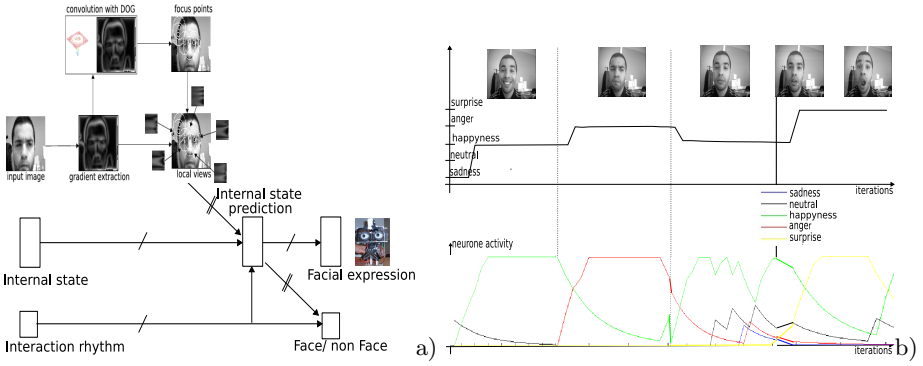


Fig. 3. a)The global architecture is able to recognize and imitate a facial expression and to perform a face/non face discrimination. A visual processing allows to extract sequentially the local views. The *internal state prediction* learns the associations between the local views and the internal state. b)Temporal activity of the neurons associated to the triggering of the different facial expressions when the robot imitates the human (after learning).

sub network learns to predict the interaction rhythm allowing the robot to detect if an interacting agent (a human) faces the robot head. In this case, the facial expression recognition is a bootstrap to discriminate face from non face images. At the end of this development stage, the robot head is able to recognize and understand the emotional facial expressions. They can now be seen as a way to communicate.

4 Associating an Emotional Value to an Object

After the human partner has imitated during 2 to 3 minutes the robot head, the robot is able to recognize and display the human facial expressions. As soon as this learning is performed, the human can interact with the robotic head to associate an emotional value to an object (positive or negative). The emotional expression is a way to communicate, that will help the robot to interact with objects according to the human will.

The N.N processes (see fig. 2) in the same way signals from the robot's internal state and infomation correlated with this internal state. An internal state can trigger a robot facial expression and a human facial expression can trigger also the robot facial expression. In case of conflict, the weights from the internal state to control the facial expression are higher than those coming from the facial expression recognition. That allows to prefer the display of the internal state rather than facial expression recognition (this is an apriori to avoid the use of much more complex structures that could be useful to allow a volontary control of the facial expression). In the absence of the internal state, the recognized facial expression induces an internal state which is associated with the object (a simple conditionning chain: fig. 2). Classical conditioning is used to

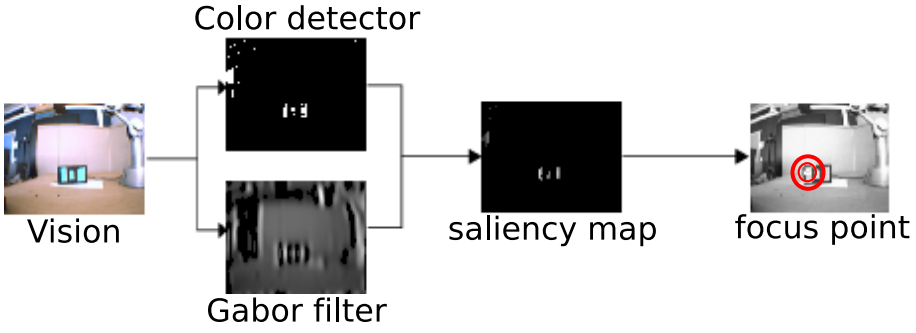


Fig. 4. Visual attention. The system focuses on some relevant features of the image. A saliency map is performed in order to focus an interesting area in the image. Visual primitives are calculated independently (gabor filters, color detector), a fusion of these primitives is performed in order to find the area that the robot must analyze.

perform the association between the emotional value that the human transmits and some areas of the image. The attentional process used in this model is very simple (see [12,6] for more information), the robot focuses on colored patches and textures (fig. 4). When focusing on an object, the robot extracts some focus points and associates the recognition of the local view surrounding each focus point with the emotional value of the robot. The focus points are the result of a DOG (Difference of gaussian) filter convolved with the gradient of the input image. This process allows the system to focus more on corners or end of lines in the image. Its main advantages over the SIFT [15] method are its computational speed and the few number of needed focus points. One after another, the most active focus points are used to compute local views (a log polar[1] transform centered on the focus point and its radius is 20 pixels). Each local view is learned by a VF_j (Visual Features) neuron:

$$VF_j = net_j.H_\theta(net_j) \qquad \theta = max(\gamma, \overline{net} + \sigma_{net}) \tag{1}$$

$$net_j = 1 - \frac{1}{N}\sum_{i=1}^{N} |W_{ij} - I_i| \tag{2}$$

VF_j is the activity of neuron j in the group VF. $H_\theta(x)$ is the Heaviside function[2]. $\gamma = 0.95$ is the vigilance (if the prototype recognition is below γ then a new neuron is recruited). $\overline{net}$ is the average of the output, σ_{net} is the standard deviation, I is the input image (N size of I) and W are the weights between

[1] The local polar transform increases the robustness of the extracted local views to small rotations and scale variations.

[2] Heaviside function:

$$H_\theta(x) = \begin{cases} 1 \text{ if } x > \theta \\ 0 \text{ otherwise} \end{cases}$$

I and VF. The learning rule for the local view categorization allows both one shot learning and long term averaging. The modifications of the weights W are computed as follow:

$$\Delta W_{ij} = {\delta_j}^k(a_j(t)I_i + \epsilon(I_i - W_{ij})(1 - VF_j)) \tag{3}$$

with $k = ArgMax(VF_j)$, $a_j(t) = 1$ only when a new neuron is recruited otherwise $a_j(t) = 0$, ${\delta_j}^k$ is the Kronecker symbol[3] and $\varepsilon = 0.001$ is a positive constant inferior to 1. When a new neuron is recruited, the weights are modified to match the input (term $a_j(t)I_i$). The other part of the learning rule $\varepsilon(I_i - W_{ij})(1 - VF_j)$ is used to average the already learned prototypes. The more the input will be close to the weights, the less the weights are modified. Conversely the less the inputs will be close to the weights, the more they are averaged. If ε is chosen too small then it will have a small impact. Conversely, if ε is too big, the previously learned prototypes can be forgotten. With this learning rule, the neurons in the VF group learn to average the prototypes of objects.

The object state prediction (OSP) group associates the activity of VF with the recognized facial expression (FE) by the robot (simple conditioning mechanism using the Least Mean Square rule [22]):

$$OSP_j = \sum_i w_{ij}.VF_i \qquad \Delta w_{ij} = \epsilon_1.VF_i.(FE_j - OSP_j) \tag{4}$$

OSP corresponds to the object emotional value for one focus point and w_{ij} is the synaptic weights between VF and OSP. OEV (object emotinal value) corresponds to a short term memory. OEV is used to recursively sum and filter on a short period ($N < 1$), the emotional value OSP associated with each explored local view. OEV corresponds to the emotional value to object (accumulation of all focus points), the OEV_i highest activity triggers the i^{th} ($0 < i \leq 5$) emotional value (WTA mechanism). After learning, the associations between VF the view recognition and OSP the emotional state are strong enough to bypass the low level reflex activity coming from the FE. Each focus points has an emotional value (OSP) and OEV is the accumulation of all focus points which corresponds to the object emotional value:

$$OEV_i = OSP_i + N.OEV_i \tag{5}$$

At this stage of development, the robot is able to use the emotional facial expression of the human partner in order to assign an emotional value to an object. As a result of the interaction with the partner, the robot recognizes and understands the human's expression in the aim of disambiguating some situations (a new object in the workspace).

[3] Kronecker function:

$${\delta_j}^k(x) = \begin{cases} x \text{ if } j = k \\ 0 \text{ otherwise} \end{cases}$$

5 Visuo-motor Learning and Yuragi Controler

At this stage of the development, the robot must be able to modulate his behavior as the result of the emotional interaction. After visuo-motor learning (learning between the extremity of the arm and the proprioception), several positions in the workspace are reached by the robot arm [1]. One visual position corresponds to one or several motor configurations (e.g attractors). These attractors pull the arm in an attraction basin (the position target). This control is performed with a dynamical system to smooth the trajectory [9]. This dynamical system also uses a reinforcing signal in the aim of attaching a lot of or little importance to some attractors, for instance a reward can be given if the arm follows the right direction, otherwise a punition. The reinforcing signal can be emotional (joy facial expression as a positive signal and angry facial expression as negative signal). Following[9] attractor selection model can be represented by Langevin equation as:

$$\tau_x \dot{x} = f(x) * A + \epsilon \tag{6}$$

where x and f(x) are the state (arm proprioception)and the dynamics of the attractor selection model, $\tau_x = 0.1$ is time constant and ϵ represents noise. A is the reinforcing signal which indicates the fitness of the state x to the environment and controls the behavior of the attractor selection model. That is to say, $f(x)*A$ becomes dominant when the activity is large, and the state transition approaches deterministic behavior (converge towards the goal). On the other hand, the noise ϵ becomes dominant when the activity is small and the state transition becomes more probabilistic.

$$f(x) = \sum_{i=1}^{n_a} N_i \frac{(X_i - x)}{||X_i - x||} \tag{7}$$

$$N_i = \frac{g_i(x)}{\sum_{j=1}^{n_a} g_j(x)} \tag{8}$$

$$g_i(x) = exp\{-\beta ||X_i - x||^2\} \tag{9}$$

With n_a the number of selected attractors, X_i (i=1, ... , n_a) a vector reprensenting the center of the i-th attractor and the function N_i a normalized Gaussian. The behavior of this system is such that the arm approaches to the nearest attractor.

Figure 5 shows the important steps of the social referencing model. Figure 5a shows the object's emotional value associated with the facial expressions of the human partner. Before T_1, the partner displays a happy facial expression in presence of the object,the human associates a positive emotional value to this object (A_1 is activated). We can see (fig. 5b,5c) more the distance between the gripper and the object decreases more the speed of the arm's motors decreases in order to tend to 0 when the object is reached. After T_1, the human partner displays an angry facial expression (transmits a negative value), the object value is modified (negative emotional value, A_2 is activated). We can see that the emotional value is now negative although, due to noise, the positive emotional

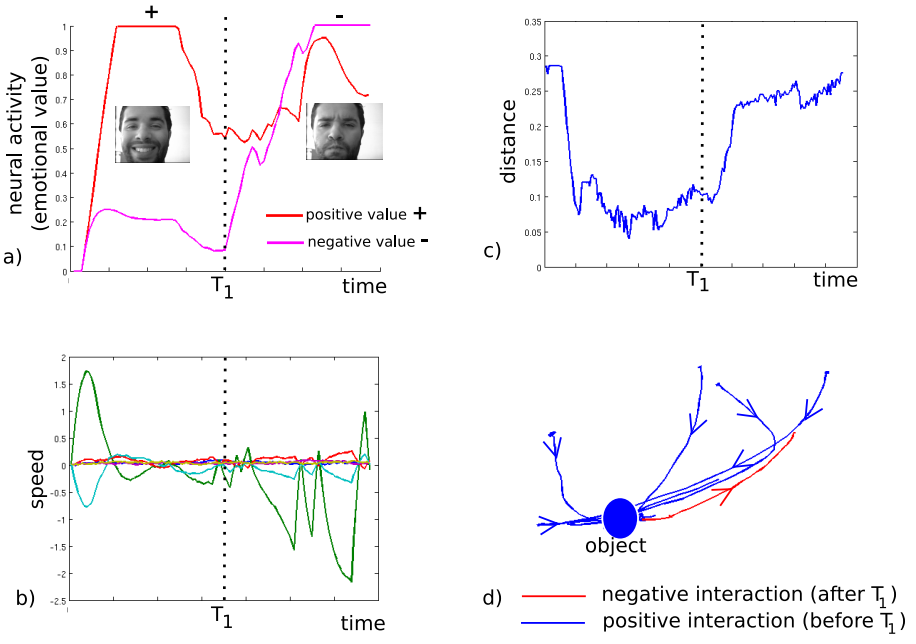


Fig. 5. These curves show: a) the emotional value transmits to the object thanks to the interaction with the human's partner (before T_1 human transmits a positive value after T_1 the human transmits a negative value) b)the speeds of each arm's motor (6 degrees of freedom) c) the distance to the object d) the robotic arm trajectories from different starting points: the arm is able to reach the object associated with the happy facial expression and avoid the object when it is associated with the angry facial expression.

value is high. This shows the learning robustness to the noise. Now, the arm avoids the object as if the object appears to be "dangerous" to the robot.

At this development stage, the robot can reach an object if the self generated reinforcing signal A is positive (the emotional value is positive) and avoid an object if A is negative (the emotional value is negative). The human emotional expression is able to communicate an emotional value to an object (for instance a dangereous object or a interested object) and moreover can modulate the robot behavior.

6 Conclusion

This work suggests the robot/partner system is an autopoietic social system [16] in which the emotional signal and empathy are important elements of the network to maintain the interaction and to allow the learning of more and more complex skills for instance the social referencing. The emotional facial expression is an excellent way to communicate in some ambiguous situations. The relationship between the robot and the partner is improved because an emotional communication can exist. It allows the robot to learn and manipulate

an object. This work also emphasizes that the recognition of the other is built through interaction.

Social cognition, including social referencing, may have a stronger emotional foundation and less of a need for complex cognition, than previously thought (e.g. [3]). New neuropsychological studies of the mirror system in emotions[13], the neural basis of intersubjectivity (e.g. [10]) and the current study highlight the important role played by emotion in the developmental emergence of social referencing.

To our knowledge, this is the first system that autonomously learns a coupling between emotion (facial expression recognition) and sensory-motor skills. We developped a real self-supervised developmental sequence contrary to others autors [5,21]. Here, we don't solve the question of joint attention which is an social referencing skill. Joint attention may also be reached using a learning protocol similar to Nagai[18] (developmental model for the joint attention). We think this approach can provide new interesting insights about how humans can develop social referencing capabilities from sensorimotors dynamics. In contrast to current developmental theory that social referencing is a complex cognitive process of triadic relations, the current work suggests 1) the primacy of emotion in learning, 2) the simple classical conditionning mechanisms by which anothers emotional signal assumes identity with internal emotional states, and 3) a simple system of pairing internal emotional state with object-directed behavior. To improve the functioning of the system, there may be a need to modulate the internal emotional state as a function of intensity of emotional expressions, and to modulate the behavior to the object in accordance, e.g an intense angry expression might involve withdrawing, an intense happy expression might involve picking up more quickly. On going work suggest it might be possible.

Acknowledgments. The authors thank J. Nadel, M. Simon and R. Soussignan for their help to calibrate the robot facial expressions and P. Canet for the design of the robot head. Many thanks also to L. Canamero for the interesting discussions on emotion modelling. This study was supported by the European project "FEELIX Growing" IST-045169 and also the French Region Ile de France (Digiteo project). P. Gaussier thanks also the Institut Unisersitaire de France for its support.

References

1. Andry, P., Gaussier, P., Moga, S., Banquet, J.P., Nadel, J.: Learning and communication in imitation: An autonomous robot perspective. IEEE Transactions on Systems, Man and Cybernetics, Part A 31(5), 431–444 (2001)
2. Balkenius, C., Moren, J.: Emotional learning: a computational model of the amygdala. Cybernetics and Systems 6(32), 611–636 (2000)
3. Bard, K.A., Leavens, D.A., Custance, D., Vancatova, M., Keller, H., Benga, O., Sousa, C.: Emotion cognition: Comparative perspectives on the social cognition of emotion. Cognitie, Creier, Comportament (Cognition, Brain, Behavior), Special Issue: Typical and atypical development 8, 351–362 (2005)

4. Boucenna, S., Gaussier, P., Andry, P.: What should be taught first: the emotional expression or the face? In: Epirob (2008)
5. Breazeal, C., Buchsbaum, D., Gray, J., Gatenby, D., Blumberg, B.: Learning from and about others: Towards using imitation to bootstrap the social understanding of others by robots. Artif. Life 11(1-2), 31–62 (2005)
6. Chevallier, S., Tarroux, P.: Covert attention with a spiking neural network. In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (eds.) ICVS 2008. LNCS, vol. 5008, pp. 56–65. Springer, Heidelberg (2008)
7. Devouche, E., Gratier, M.: Microanalyse du rythme dans les échanges vocaux et gestuels entre la mère et son bébé de 10 semaines. Devenir 13, 55–82 (2001)
8. Ekman, P., Friesen, W.V.: Facial action coding system: A technique for the measurement of facial movement. Consulting Psychologists Press, Palo Alto (1978)
9. Fukuyori, I., Nakamura, Y., Matsumoto, Y., Ishiguro, H.: Flexible control mechanism for multi-dof robotic arm based on biological fluctuation. In: Asada, M., Hallam, J.C.T., Meyer, J.-A., Tani, J. (eds.) SAB 2008. LNCS (LNAI), vol. 5040, pp. 22–31. Springer, Heidelberg (2008)
10. Gallese, V.: The roots of empathy: The shared manifold hypothesis and neural basis of intersubjectivity. Psychopathology 36, 171–180 (2003)
11. Gaussier, P., Prepin, K., Nadel, J.: Toward a cognitive system algebra: Application to facial expression learning and imitation. In: Iida, F., Pfeiter, R., Steels, L., Kuniyoshi, Y. (eds.) Embodied Artificial Intelligence. LNCS (LNAI), pp. 243–258. Springer, Heidelberg (2004)
12. Itti, L., Koch, C.: Computational modelling of visual attention. Nature Reviews Neuroscience 2(3), 194–203 (2001)
13. Keyser, C., Bastiaansen, J., Thioux, M.: Evidence for mirror systems in emotions. Phil. Trans. R. Soc. B 364, 2391–2404 (2009)
14. Klinnert, M.D., Campos, J.J., Sorce, J.F., Emde, R.N., Svejda, M.: The development of the social referencing in infancy. Emotion in Early Development 2, 57–86 (1983)
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 2, 91–110 (2004)
16. Mataruna, H.R., Varela, F.J.: Autopoiesis and Cognition: the realization of the living. Reidel, Dordrecht (1980)
17. Nadel, J., Simon, M., Canet, P., Soussignan, R., Blanchard, P., Canamero, L., Gaussier, P.: Human responses to an expressive robot. In: Epirob 2006 (2006)
18. Nagai, Y., Hosoda, K., Morita, A., Asada, M.: A constructive model for the development of joint attention. Connect. Sci. 15(4), 211–229 (2003)
19. Russell, C.L., Bard, K.A., Adamson, L.B.: Social referencing by young chimpanzees (pan troglodytes). Journal of Comparative Psychology 111(2), 185–193 (1997)
20. Schmajuk, N.A.: A neural network approach to hippocampal function in classical conditioning. Behavioral Neuroscience 105(1), 82–110 (1991)
21. Thomaz, A.L., Berlin, M., Breazeal, C.: An embodied computational model of social referencing. In: IEEE International Workshop on Human Robot Interaction, RO-MAN (2005)
22. Widrow, B., Hoff, M.E.: Adaptive switching circuits. In: IRE WESCON, New York, pp. 96–104 (1960); Convention Record

A Model of Symmetry Breaking in Collective Decision-Making

Heiko Hamann[1], Bernd Meyer[2], Thomas Schmickl[1], and Karl Crailsheim[1]

[1] Artificial Life Lab of the Dep. of Zoology, Karl-Franzens University Graz, Austria
{heiko.hamann,thomas.schmickl,karl.crailsheim}@uni-graz.at
[2] FIT Centre for Research in Intelligent Systems, Monash University, Melbourne
bernd.meyer@acm.org

Abstract. Symmetry breaking is commonly found in self-organized collective decision making. It serves an important functional role, specifically in biological and bio-inspired systems. The analysis of symmetry breaking is thus an important key to understanding self-organized decision making. However, in many systems of practical importance available analytic methods cannot be applied due to the complexity of the scenario and consequentially the model. This applies specifically to self-organization in bio-inspired engineering. We propose a new modeling approach which allows us to formally analyze important properties of such processes. The core idea of our approach is to infer a compact model based on stochastic processes for a one-dimensional symmetry parameter. This enables us to analyze the fundamental properties of even complex collective decision making processes via Fokker–Planck theory. We are able to quantitatively address the effectiveness of symmetry breaking, the stability, the time taken to reach a consensus, and other parameters. This is demonstrated with two examples from swarm robotics.

1 Introduction

Self-organization is one of the fundamental mechanism used in nature to achieve flexible and adaptive behavior in unpredictable environments [1]. Particularly collective decision making in social groups is often driven by self-organizing processes. Some of the most prominent examples of this are found in social insects, for example the choice of nest sites [2] and food sources [3,4] by ant colonies and the aggregation behavior of bees [5]. Countless other examples of self-organized collective decision making exist in both biological and human social systems, such as in quorum sensing in bacteria colonies [6] and in trend setting and following in economic markets [7]. Despite this variability the fundamental principles that govern self-organized collective behavior appear to be universal across the range [8].

Understanding the fundamental properties of self-organized collective decision making is thus of central importance in a broad spectrum of disciplines, ranging from biology and social science to bio-inspired engineering.

An important property commonly found in self-organized collective decision making is symmetry breaking. This refers to the fact that a system choosing between a number of options may make a majority (or even an unanimous) decision

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 639–648, 2010.

for one of these options even if all of them have the same utility. From a biological perspective, symmetry breaking serves an important function. This is immediately obvious when we think about decisions that must be unanimous, such as the choice of a new nest site [9]. In other cases symmetry breaking still serves an important function even though its benefit may be less obvious. Consider food source exploitation by an ant colony. When faced with a choice between two equally good food sources, many species will converge on only one of them instead of exploiting both simultaneously [1]. There are several benefits of concentrating the colony's foragers on a single source. Arguably the most important one is that it is significantly easier to defend a single source against potential competition and aggressors. Further benefit may, for example, arise from the fact that collaborative transport of large prey items can be an efficient form of food retrieval [10] compared with the transport of small items by individual foragers. Similar considerations transfer immediately to technical applications of self-organized collective decision making, for example in swarm robotics.

Because of its central functional role, the analysis of symmetry breaking is one of the keys to a better understanding of collective decision making. This is widely accepted and a substantial body of work on the topics exists, much of which is based on bifurcation analysis. Recruitment in ant colonies is a well-known example where this approach has yielded a significant amount of insight [1].

Symmetry breaking in self-organized collective decision-making usually arises from the interplay of positive and negative feedback loops. For example, in mass recruiting ant colonies pheromone communication is the main mechanism for guiding foragers to food sources. Roughly speaking, pheromones attract potential foragers to a food source, and successful foragers deposit pheromones on paths to good food sources. This positive feedback is counterbalanced by negative feedbacks, namely the evaporation of pheromone and overcrowding. The balance between these two influences leads to a stable yet flexible decision system [3,4].

Arguably the most common approach to the analysis of such coupled feedback systems is via differential equation systems. In sufficiently simple cases, bifurcation analysis can then be used to establish properties of the symmetry breaking process. However, as the complexity of the scenario and the individual behavior increases, an analysis of the full differential equation model soon becomes extremely involved (see e.g. [11]) and often is simply impossible. Moreover, finding such a model can be difficult in the first place, specifically in cases where the behavior of individuals is relatively complex.

Thus, alternative ways to model the symmetry breaking properties in such systems are required. In this paper we propose such a method. The core idea of our approach is to dispense with complex mechanistic models which capture feedback loops or even individual behavior explicitly. We replace these with significantly simpler phenomenological models that focus exclusively on the description of symmetry breaking. The advantage of doing this is that we obtain a compact mathematical model which is amenable to a formal analysis. In this way we can analytically obtain information about the fundamental properties of the process, such as the stability of decisions. We will illustrate this approach and such an analysis

with two examples from swarm robotics. The first one is the aggregation behavior of a swarm whose control algorithms are modeled on honeybee behavior [12,13,14], the second one is an emergent density classification task [15].

2 Overall Approach

To introduce our approach we consider the simplest possible scenario of a binary choice between two options A, B with equal utility. We will say that a system with N individuals (agents) exhibits symmetry breaking, if a significant majority of it decides for either A or B regardless of their equal utilities. We define a symmetry parameter (w.l.o.g. with respect to option A) as $s(t) = L(t)/N$, where $L(t)$ is the number of agents that have chosen option A at time t. A majority decision is any outcome in which the process has converged to a state where at least $L \geq \delta N$ individuals have chosen option A with $0.5 \ll \delta \leq 1.0$. In terms of s this simply means $s \geq \delta$. $s(t)$ essentially captures all information required to analyze the symmetry breaking properties of the process.

If $s(t)$ converges we can in principle obtain the steady state probability density function (PDF) for $s(t)$. This steady state PDF $\rho^*(s)$ tells us how likely it is that a certain proportion of agents decides for option A. In a binary symmetry breaking scenario, $\rho^*(s)$ will be bimodal and (possibly) symmetric (see Fig. 3 for the steady state PDFs of our two example scenarios). Given $\rho^*(s)$ we can thus calculate the expected proportion P of experiments in which a majority decision with at least δ majority occurs as

$$\int_0^{\delta} \rho^*(s)ds + \int_{1-\delta}^{1} \rho^*(s)ds = P. \tag{1}$$

Provided $s(t)$ converges we can in principle always obtain $\rho^*(s)$ at least approximately by statistical evaluation of a large number of sample simulations. The same is true of other properties of $s(t)$, for example the time to convergence. This is indeed the standard approach to the analysis of such systems if no analytic description is available: A symmetry parameter is defined, the full system is simulated, and the statistics of the symmetry parameter is measured from the results of these (typically high-dimensional) simulations. This approach has two drawbacks: Firstly, it can be computationally extremely costly as obtaining numerical values for individual properties requires the simulation of a large number of sample developments. Secondly, this form of statistics only gives us numeric results for individual properties and does not allow us to perform a more general analysis of these properties.

An explicit representation of $s(t)$ would obviously enable us to perform significantly more powerful analysis provided it is simple enough to stay mathematically tractable. This idea is at the very center of our approach.

We simply and somewhat boldly postulate that the development of $s(t)$ can be described by a 1-d Langevin equation, a particular form of stochastic differential equation:

$$\frac{ds}{dt} = \alpha(s,t) + \beta(s,t)\xi(t), \tag{2}$$

where α describes the deterministic development (so-called drift), ξ a Gaussian noise $|\xi(t)| = 1$, with mean $\langle \xi(t) \rangle = 0$, and uncorrelated in time $\langle \xi(t)\xi(t') \rangle = \delta(t - t')$, and β captures the fluctuation of the noise amplitude. Having a definition of $s(t)$ in this form unlocks a whole repertoire of formal tools for the analysis of stochastic differential equations that allow us to calculate most of the properties that we are interested in [16].

Two questions arise immediately: (1) Does this description indeed exist for a given system, i.e. is it possible at all to correctly reduce the behavior of the (potentially high-dimensional) system to a one-dimensional system? (2) Even if it is possible, how can we infer α and β?

The first question cannot generally be answered, and some systems will not admit such a description. However, our case studies give some indication that it should be possible to approximate the behaviour of many interesting system quite well. Our methodology approaches the question optimistically by assuming the existence of such a description: We first attempt to infer α and β. This can be done via a heuristic argument (as evidenced by the case studies) or with standard numerical fitting techniques. Once candidates for α and β are obtained the crucial step is their verification. This is performed by obtaining a large number of sample developments for $s(t)$ in two complementary ways: One set of samples is generated by simulating the full original (mechanistic) model and measuring $s(t)$ at each simulation step, another set of samples is obtained by numeric forward integration of $s(t)$ according to Eq. 2 for given α and β. If the two sample sets are not in statistical agreement we have to dismiss the candidate functions α, β. However, if the sets agree statistically, we are justified in our choice of α and β and the simplified model Eq. 2 captures the relevant aspects of the process statistically correctly. In this case we may discard the full model and simply proceed with an analysis of the simplified model.

3 Investigated Scenarios

We will now illustrate this approach with two example scenarios from swarm robotics. Both are instances of homogeneous multi-agent systems (MAS), where the agents move in a two dimensional rectangular arena surrounded by walls.

Our first scenario is collision-based adaptive aggregation. The task of the robot swarm is to aggregate at the brightest spot in the arena. This is essentially a physically embodied simulation of the behavior of young honeybees, that typically aggregate at areas of a certain temperature. The system and its distributed control algorithm, termed BEECLUST, have been described in full detail in [13,17], and we only give a brief summary here.

The robots are equipped with sensors for distance measurements as well as a sensor that allows them to measure a special inhomogeneous property of the arena (e.g., light). In addition, they are able to identify other robots as such. Each robot moves in a straight line (with initial random heading) until it perceives an obstacle Ω within sensor range. If Ω is a wall it turns away and moves straight again. If Ω is another robot, it counts the number of other robots K

in the vicinity. If $K \geq \sigma$ the robot measures the local luminance. The higher the luminance the longer the agent stays stopped. After the waiting period has elapsed, the agent turns away from the other agent and moves straight again.

The collective aggregation at the brightest spot is a consequence of positive feedback. Waiting times are longer at bright spots and the agent density is increased through clustering in these regions. Negative feedback is induced by the saturation of the limited space with high brightness.

Initially, the agents have random headings, are in the state 'moving', and are random uniformly distributed in the whole arena (i.e., on average we have initially the same number of robots in the left and in the right half of the arena). The luminance distribution in the test arena is bimodal with maxima of the same value and shape in the left and right half of the arena (for details, see [12,14]). As a measure of symmetry we use $s_b(t) = L(t)/N$ ('b' for BEECLUST) where $L(t)$ is the number of robots in the left half of the arena, and N the swarm size.

Our second scenario is an emergent density classification task. Robots in the swarm exist in two states 'red' and 'green'. The task of the robot swarm is to estimate whether there are initially more green or more red swarm members, i.e. to converge on a majority decision. This problem is derived from a well known example of emergent computation in cellular automata [18]. N robots are randomly distributed in the arena and randomly initialized to be red or green in a given proportion. The robots move and perform collision avoidance, i.e. if another robot comes too close they turn away to increase the distance. At each encounter the robots remember the color of each other. After five robot encounters, each robot changes its color to the one it encountered most often. We are interested in the question whether (a majority of) the swarm stably converges on one color and whether this is the original majority color (for details, see [15]).

As a measure of symmetry we use $s_d(t) = R(t)/N$ ('d' for 'density classification') where $R(t)$ is the number of red robots and N the total swarm size.

Both scenarios include a stochastic component. The density classification scenario includes a stochastic component as we explicitly account for errors in the color recognition. We assume that a robot recognizes the color of the encountered robot correctly only with a given probability $\gamma = 0.8$. While noise is explicitly incorporated in the case of the density classification scenario, it implicitly enters the BEECLUST scenario. This is because the basic movement and collision-avoidance mechanism is a billiard-like system that introduces pseudo-randomness through deterministic chaos.

4 Dynamics of the Symmetry Parameter

As reported in [12], most macroscopic characteristics of the collective decision processes of these systems can approximately be captured by two features of the symmetry parameter s. First, the mean of the absolute changes

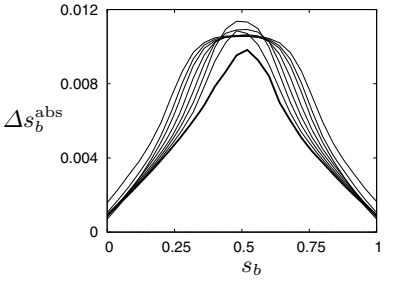
$$\Delta s^{\text{abs}}(s,t) = \frac{1}{K} \sum_i |s_i(t) - s_i(t-1)|, \tag{3}$$

averaged over K samples $s_i(t)$ obtained from many independent simulation runs. Second, the mean of the relative changes

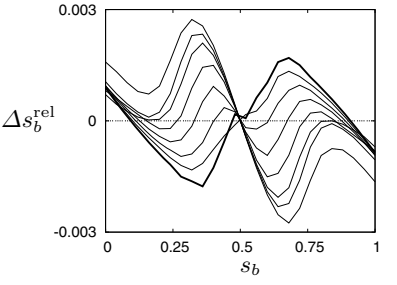
$$\Delta s^{\text{rel}}(s,t) = \frac{1}{K}\sum_i s_i(t) - s_i(t-1), \tag{4}$$

which is an approximation of the derivative. In Fig. 1, we show the results of the measurements for both scenarios. Note that Δs^{rel} at $s = 0.5$ averages to $\Delta s^{\text{rel}}(s = 0.5, t) \approx 0$ as expected (Fig. 1(b), 1(d)), while the converged absolute changes keep a maximum at $s = 0.5$ (Fig. 1(a), 1(c)). Fig. 1 shows that Δs^{abs} and Δs^{rel} are indeed time-variant. Δs^{abs} basically keeps its shape and is simply scaled down with time, whereas Δs^{rel} even changes its shape. It only converges after some time to a function that can cause multiple stable (or meta-stable) decisions (as it exhibits multiple zero-crossings corresponding to fixpoints).

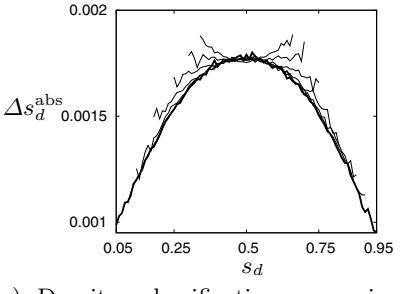
For numeric calculations we will subsequently simply use the (time-invariant) values for Δs^{abs} and Δs^{rel} which are reached with an error exponentially decreasing in time as shown in Fig. 1. While this is clearly a rather drastic simplification, our choice will be justified by the verification step of our basic methodology.



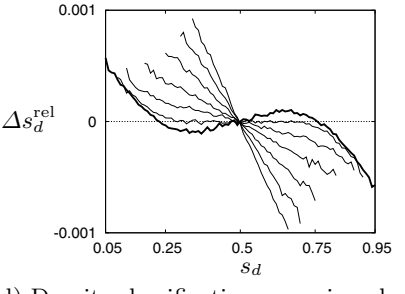
(a) BEECLUST scenario, abs. change of Δs for times 10; 20; 40; 80; 160; 320; 640; 1,280 (bold line).



(b) BEECLUST scenario, rel. change of Δs for times 10; 20; 40; 80; 160; 320; 640; 1,280 (bold line).

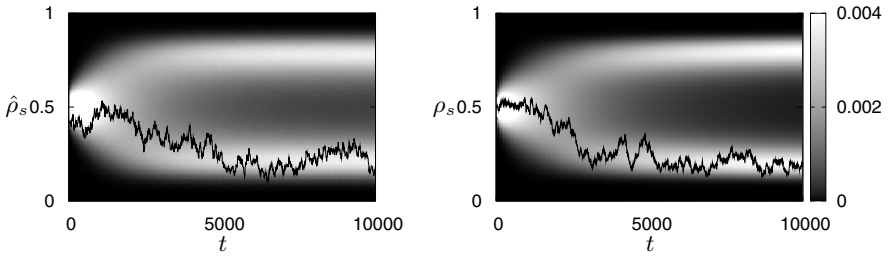


(c) Density classification scenario, abs. change for 100; 200; 400; 800; 1,600; 3,200; 6,400 (bold line).



(d) Density classification scenario, rel. change for 100; 200; 400; 800; 1,600; 3,200; 6,400 (bold line).

Fig. 1. Measured dynamics of the symmetry parameter; 2×10^5 samples per scenario



(a) simulation (b) macro-model.

Fig. 2. Density classification scenario; PDF measured in the simulation ($\hat{\rho}_s$, based on 2×10^5 samples) and obtained by solving Eq. 6 (using the converged measurements) and superimposed samples of trajectories of s from the simulation and by sampling Eq. 2

Based on Δs^{rel} and Δs^{abs} we suggest the following heuristics to infer candidates for α and β. Consider the development of s in the time-discrete simulation. We suggest to approximate this as

$$s_{t+1} = s_t + \Delta s^{\text{rel}}(s_t) + (\Delta s^{\text{abs}}(s_t) - |\Delta s^{\text{rel}}(s_t)|)\xi_t, \tag{5}$$

for Δs^{rel} and Δs^{abs} as defined above, and for a Gaussian white noise ξ_n. The assumption of Gaussian white noise is of course only an approximation which could be improved by measuring the distributions of the Δs for each time step.

To justify Eq. 5, we consider the three points $s \in \{0, 0.5, 1\}$ in Fig. 1. At $s = 0$ and $s = 1$ we have $|\Delta s^{\text{rel}}| = \Delta s^{\text{abs}}$ because changes are only possible in one direction at the borders. Hence, the influence of the noise term should be zero there. At $s = 0.5$ we have, as expected, $\Delta s^{\text{rel}} = 0$. Hence, the only influence is due to Δs^{abs}, which is fulfilled by Eq. 5.

We make the transition to a continuous time version by using Eq. 2 with drift $\alpha(s) = \Delta s^{\text{rel}}(s)$ and diffusion coefficient $\beta(s) = \Delta s^{\text{abs}}(s) - |\Delta s^{\text{rel}}(s)|$. For this continuous model we can now use the Fokker–Planck equation

$$\frac{\partial \rho_s}{\partial t} = \frac{\partial}{\partial s}(\alpha(s,t)\rho_s) + \frac{1}{2}\frac{\partial^2}{\partial s^2}(\beta^2(s,t)\rho_s) \tag{6}$$

to obtain the time development of the probability density function for s, and specifically its steady-state PDF. As outlined earlier for our methodology, we must verify whether this candidate model is valid by comparing two versions of the PDF for $s(t)$ obtained in different ways: once by simulating the full system and once by solving the Fokker–Planck equation for the 1-d system.

Fig. 2 shows the densities obtained via simulation and Fokker–Planck equation, respectively, together with a typical sample trajectory. These plots were obtained by solving Eq. 6 numerically for the initial value problem of an initial peak at $s = 0.5$ for $t = 0$ and with the corresponding full simulation. It shows that there is good qualitative correspondence between the two systems and thus that the simplified model with the chosen α, β on this level adequately describes the evolution of the symmetry parameter for these systems. We are thus

justified in dispensing with the full model and conducting an analysis of symmetry breaking properties based only on the simplified model, which is amenable to a formal analysis.

5 Deriving Properties of the Collective Decision System

The simplified model can now be used to analyze the symmetry breaking properties of the system. To begin, consider the effectivity of symmetry breaking. It can be quantified via the steady state of the Fokker–Planck equation. Assuming reflecting boundaries at $s = 0$ and $s = 1$, the steady state $\rho^*(s)$ is given by

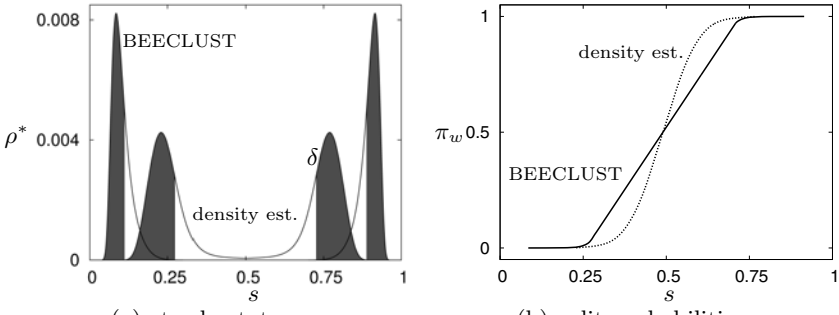
$$\psi(s) = \exp\left(\int_0^s 2\frac{\alpha(y)}{\beta^2(y)}dy\right), \quad \rho^*(s) = C\frac{\psi(s)}{\beta(s)}, \tag{7}$$

with a normalization constant C [16,4]. In Fig. 3(a), the results obtained from Eq. 7 are shown for both scenarios using the functions α and β as defined above in Eq. 2 and with the data shown in Fig. 1 (with noise). In addition, the positions of δ as defined by Eq. 1 are given and the associated areas are marked in gray. For both scenarios $\rho^*(s)$ is clearly bimodal and exhibits very low densities around $s \approx 0.5$, hence the symmetry breaking is effective.

As a further example of a possible analysis we quantifying the stability of decisions. An appropriate measure for this is the probability of revising a majority decision. This is similar to the concept of 'splitting probabilities' [16]. The splitting probability $\pi_w(x)$ gives the probability that the system initialized at $s = x$ will reach the state $s = w$ before the state $s = u$, i.e. it will make a majority decision for w. The corresponding probability $\pi_u(x)$ is defined symmetrically. The splitting probability can be calculated as

$$\pi_w(x) = \int_u^x \rho^*(s)^{-1}ds \left(\int_u^w \rho^*(s)^{-1}ds\right)^{-1}, \tag{8}$$

if we take u and w to correspond to the positions of the two peaks in the PDF (i.e. the decision states), we can immediately read off various important properties



(a) steady states. (b) split probabilities.

Fig. 3. Analytically and numerically obtained measures of symmetry breaking

from a plot of $\pi_w(x)$ particularly how likely it is that the system reaches a given decision from a given start state and thus also how likely it is that a decision, once made, will be reversed giving a measure of the stability of the decision process.

The numerically obtained results for $\pi_w(x)$ are shown in Fig. 3(b). Based on this, the stability of the density classification scenario has to be classified as higher than that of the BEECLUST scenario, because its slope around the center point ($\pi = 0.5$) is steeper and it reaches saturation ($\pi = 0$ or $\pi = 1$) more quickly.

In the same way a host of other properties can be analyzed using the Fokker–Planck equation. In some cases, such as for the convergence time, a full analytic treatment may be difficult, particularly for time-variant functions α and β. However, a numeric estimate can still be obtained as shown in Fig. 2(b).

6 Conclusion

We have proposed a generalized approach to analyzing symmetry breaking in collective decision making. It is based on a simple stochastic process model that describes the development of a one-dimensional symmetry parameter. The model and its parameters are inferred by measurements from (real or virtual) experiments. The models obtained are simple enough to be amenable to powerful analytic techniques using Fokker-Planck equations. This allows us to calculate important properties, such as the effectiveness of the decision process, likelihood of decisions to be reached and to be revised, time to reach a consensus etc. without having to rely on further simulations. We suspect that our heuristics for inferring the model parameters (drift and diffusion coefficients) based on Δs^{rel} also applies to other self-organized collective decision making processes. This assumption receives some support from their successful use in the analysis of the two substantially different example scenarios. It is crucial where this heuristics cannot be used, the core of our approach remains applicable as it allows us to use a range of other methods to identify and verify candidates for α and β. We thus expect our approach to be in principle useful for a wide range of self-organized collective decision making scenarios.

Acknowledgments

This work is supported by: EU-IST-FET project ‘SYMBRION’, no. 216342; EU-ICT project ‘REPLICATOR’, no. 216240; FWF research grant ‘Temperature-induced aggregation of young honeybees’, no. P19478-B16 and ARC DP0879239.

References

1. Camazine, S., Deneuenbourg, J.L., Franks, N.R., Sneyd, J., Theraulaz, G., Bonabeau, E.: Self-Organization in Biological Systems. Princeton Univ. P., Princeton (2001)
2. Franks, N.R., Mallon, E.B., Bray, H.E., Hamilton, M.J., Mischler, T.C.: Strategies for choosing between alternatives with different attributes: exemplified by house-hunting ants. Animal Behavior 65, 215–223 (2003)

3. Dussutour, A., Beekman, M., Nicolis, S.C., Meyer, B.: Noise improves collective decision-making by ants in dynamic environments. Proceedings of the Royal Society London B 276, 4353–4361 (2009)
4. Meyer, B., Beekman, M., Dussutour, A.: Noise-induced adaptive decision-making in ant-foraging. In: Asada, M., Hallam, J.C.T., Meyer, J.-A., Tani, J. (eds.) SAB 2008. LNCS (LNAI), vol. 5040, pp. 415–425. Springer, Heidelberg (2008)
5. Kernbach, S., Thenius, R., Kornienko, O., Schmickl, T.: Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic swarm. Adaptive Behaviour 17, 237–259 (2009)
6. Reading, N.C., Sperandio, V.: Quorum sensing: the many languages of bacteria. FEMS Microbiol. Lett. 254(1), 1–11 (2006)
7. Weisbuch, G., Stauffer, D.: Hits and flops dynamics. Physica A 287, 563–576 (2000)
8. Vicsek, T.: A question of scale. Nature 411, 421 (2001)
9. Jeanson, R., Deneubourg, J.L., Grimal, A., Theraulaz, G.: Modulation of individual behavior and collective decision-making during aggregation site selection by the ant messor barbarus. Behav. Ecol. Sociobiol. 55, 388–394 (2004)
10. Garnier, S., Gautrais, J., Theraulaz, G.: The biological principles of swarm intelligence. Swarm Intelligence 1, 3–31 (2007)
11. Nicolis, S.C., Deneubourg, J.L.: Emerging patterns and food recruitment in ants: an analytical study. Journal of Theoretical Biology 198, 575–592 (1999)
12. Hamann, H., Schmickl, T., Wörn, H., Crailsheim, K.: Analysis of emergent symmetry breaking in collective decision making. Neural Comp. & Appl. (2010) (in Press)
13. Schmickl, T., Thenius, R., Möslinger, C., Radspieler, G., Kernbach, S., Crailsheim, K.: Get in touch: Cooperative decision making based on robot-to-robot collisions. Autonomous Agents and Multi-Agent Systems 18(1), 133–155 (2008)
14. Schmickl, T., Hamann, H.: BEECLUST: A swarm algorithm derived from honeybees. In: Xiao, Y., Hu, F. (eds.) Bio-inspired Computing and Communication Networks. Routledge, New York (August 2010)
15. Hamann, H., Wörn, H.: Embodied computation. Parallel Processing Letters 17(3), 287–298 (2007)
16. Gardiner, C.W.: Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences. Springer, Heidelberg (1985)
17. Hamann, H., Wörn, H., Crailsheim, K., Schmickl, T.: Spatial macroscopic models of a bio-inspired robotic swarm algorithm. In: IEEE/RSJ 2008 Int. Conf. on Intel. Robots and Syst. (IROS 2008), pp. 1415–1420. IEEE Press, Los Alamitos (2008)
18. Packard, N.H.: Adaptation toward the edge of chaos. In: Dynamic Patterns in Complex Systems, pp. 293–301. World Scientific, Singapore (1988)

Simulation of How Neuromodulation Influences Cooperative Behavior

Andrew Zaldivar, Derrik E. Asher, and Jeffrey L. Krichmar

Department of Cognitive Sciences
University of California, Irvine
Irvine, CA 92697-5100
{azaldiva,dasher,jkrichma}@uci.edu

Abstract. Neuromodulators can have a strong effect on how organisms cooperate and compete for resources. To better understand the effect of neuromodulation on cooperative behavior, a computational model of the dopaminergic and serotonergic systems was constructed and tested in games of conflict and cooperation. This neural model was based on the assumptions that dopaminergic activity increases as expected reward increases, and serotonergic activity increases as the expected cost of an action increases. The neural model guided the behavior of an agent that played a series of Hawk-Dove games against an opponent. The agent adapted its behavior appropriately to changes in environmental conditions and to changes in its opponent's strategy. The neural agent tended to engage in Hawk-like behavior in low-risk situations and Dove-like behavior in high-risk situations. When the simulated dopaminergic activity was greater than the serotonergic activity, the agent tended to escalate a fight. These results suggest how the neuromodulatory systems shape decision-making and adaptive behavior in competitive and cooperative situations.

Keywords: Dopamine; Serotonin; Cooperation; Game Theory; Computational Neuroscience; Decision-Making.

1 Introduction

Neuromodulators, such as dopamine (DA) and serotonin (5-HT), are known to be important in predicting rewards, costs, and punishments. Dopamine activity (DA), which originates in the ventral tegmental area (VTA) and the substantia nigra (SN), appears to be linked to expected reward [1], and incentive salience or "wanting" [2]. Serotonin (5-HT), which originates in the Raphe nucleus, appears to be related to cognitive control of stress, social interactions, and risk taking behavior [3], [4]. The structures that are innervated by 5-HT and their connecting circuits modulate the behavioral response to threats and risks, that is, behaviors that are typically thought to reflect the anxiety state of the organism [3]. Whereas DA is related to the expected reward of a given decision, 5-HT could be related to the expected cost of a decision.

S. Doncieux et al. (Eds.): SAB 2010, LNAI 6226, pp. 649–660, 2010.

Game theory has been useful for understanding risk-taking and cooperation [5]. Of particular interest are studies in which neuromodulators were depleted or altered, while subjects play games. In one study, subjects, who were 5-HT depleted through dietary changes, cooperated less in a Prisoner's Dilemma game [6]. In an Ultimatum game study, 5-HT depleted subjects tended to reject monetary offers more than control subjects when they deemed the offers to be unfair [4]. Moreover, a recent study has shown that individuals with lower levels of dopamine in the prefrontal cortex tended to take less risks in a gambling task [7].

To better understand the roles of dopamine and serotonin during decision-making in games of conflict, we developed a computational model of neuromodulation and action-selection, based on the assumption that DA levels are related to the expected reward of an action, and 5-HT levels are related to the expected cost of an action. An agent, whose behavior was guided by the neural model, played the Hawk-Dove game, where players must choose between confrontational and cooperative tactics [5], [8]. The model makes predictions of how neuromodulatory activity can shape behavior under different environmental and competitive situations.

2 Methods

Game Playing. A *game* consisted of two agents (*Neural* and *Opponent*) taking a single action in response to a territory of interest (TOI). At the start of each game, the agents were randomly placed in a square grid (not occupying the same area) and were modeled to approach the neutral TOI at the same speed. The agent that arrived at the neutral TOI first had the opportunity to take either of the two possible actions: *Escalate* (i.e., an aggressive, confrontational tactic) or *Display* (i.e., a nonviolent, cooperative tactic). The agent that arrived second responded with one of the two aforementioned actions. After each game, payoff was calculated and plastic connections were updated. The payoff matrix for this game is given in Table 1. If both agents *Escalate*, they received a penalty that was either a serious injury (large penalty) or just a scratch (small penalty). The probability of serious injury was set to 0.25 or 0.75 at the start of the game. If both agents *Display*, they share the TOI resource. If one agent escalated and the other displayed, the agent that escalated gets the entire resource. A *series* consisted of 100 games with a given parameter set. At the start of each series, the neural network was initialized and the *Neural* agent was considered "naïve", that is, the weights of the network were set to their initial values (see next section). For each parameter set, the two agents played 100 Hawk-Dove series with a different random number seed.

Table 1. Payoff matrix for Hawk-Dove game between players A and B. V is the value of the resource and is set to 0.60. D is the damage incurred when both players escalate. D is set to 1.60 for serious injury and 0.62 for a scratch. The probability of a serious injury is 0.25 or 0.75.

	B. Escalate	**B. Display**
A. Escalate	A: (V–D)/2, B: (V–D)/2	A: V, B: 0
A. Display	A: 0, B: V	A: V/2, B: V/2

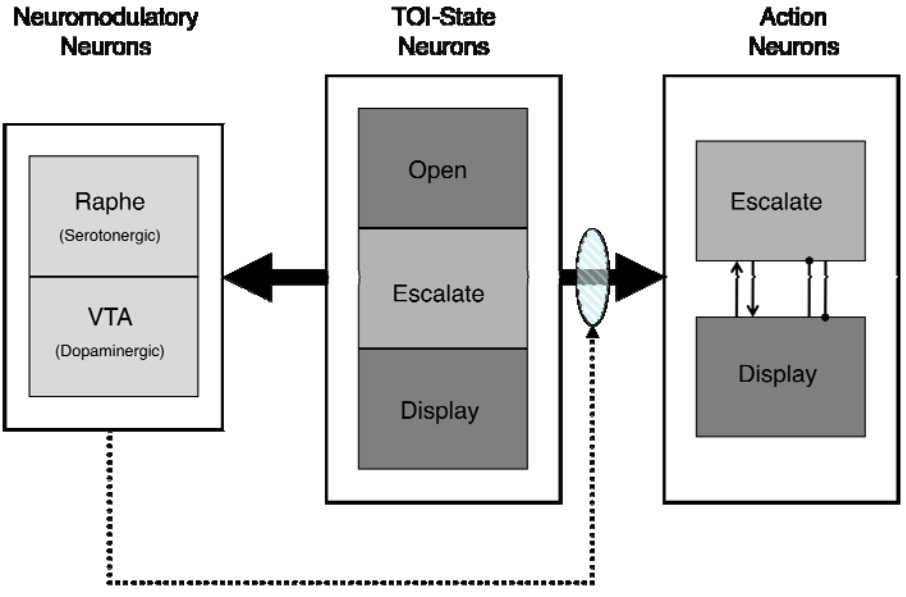


Fig. 1. The diagram shows the architecture of the neural model (two *Neuromodulatory*: Raphe and VTA; three *TOI-State*: Open, Escalate, and Display; and two *Action*: Escalate and Display). The solid arrows extending from the *TOI-State* neurons represent all-to-all connections. The thick arrows represent plastic pathways. The dotted arrows and shaded ovals represent neuromodulatory pathways. Within the *Action* neurons, the line with the arrow at the end represent excitation, and the line with the dot at the end represent inhibition.

Neural Agent. A neural network controlled the behavior of the *Neural* agent. The neural network had three areas: *TOI-State*, *Action*, and *Neuromodulatory* (Fig. 1). The *TOI-State* included three neurons that corresponded to the possible states of the TOI the *Neural* agent may observe: 1) *Open*. The *Neural* agent reached the TOI first. 2) *Escalate*. The *Opponent* agent reached the TOI first and escalated a conflict. 3) *Display*. The *Opponent* agent reached the TOI first but did not start a conflict. The equation for the activity of each of these neurons (n_i) was set based on the current state of the TOI:

$$n_i = \begin{cases} 0.75 + rnd(0.0,\ 0.25); & i = TOIState \\ rnd(0.0,\ 0.25); & Otherwise \end{cases} \quad (1)$$

where rnd(0.0,0.25) was a random number uniformly distributed between 0.0 and 0.25. The *Action* area included two neurons: 1) *Escalate*. The *Neural* agent escalated a conflict. 2) *Display*. The *Neural* agent did not start a conflict or retreated if the *Opponent* agent escalated. The neural activity was based on input from *TOI-State* neurons and neuromodulation. Lastly, the *Neuromodulatory* area included two neurons: 1) *Raphe*. A simulated raphe nucleus, which is the source of serotonergic

neuromodulation. 2) *VTA*. A simulated ventral tegmental area, which is the source of dopaminergic neuromodulation. The synaptic connectivity of the network is shown in Fig. 1 and in Table 2, and was all-to-all. Some of these connections were subject to synaptic plasticity and phasic neuromodulation, where the activity of *Neuromodulatory* neurons affected the synaptic efficacy.

Table 2. Synaptic connections between neural areas

From	To	Initial Weight	Plastic	Phasic Neuromodulation
TOI-State	*Action*	0.1	Y	Y
TOI-State	*Neuromodulatory*	0.1	Y	N
Action-Escalate	*Action-Display*	0.1	N	N
Action-Escalate	*Action-Display*	-0.1	N	Y
Action-Display	*Action-Escalate*	0.1	N	N
Action-Display	*Action-Escalate*	-0.1	N	Y

The neural activity was simulated by a mean firing rate neuron model, where the firing rate of each neuron ranged continuously from 0 (quiescent) to 1 (maximal firing). The equation for the mean firing rate neuron model was:

$$s_i(t) = \rho_i s_i(t-1) + \left(1-\rho_i\right)\left(\frac{1}{1+\exp\left(-5I_i(t)\right)}\right) \tag{2}$$

where t was the current time step, s_i was the activation level of neuron i, ρ_i was a constant set to 0.1 and denoted the persistence of the neuron, and I_i was the synaptic input. The synaptic input of the neuron was based on pre-synaptic neural activity, the connection strength of the synapse, and the amount of neuromodulator activity:

$$I_i(t) = rnd(-0.5,0.0) + \sum_j nm(t-1)w_{ij}(t-1)s_j(t-1) \tag{3}$$

where w_{ij} was the synaptic weight from neuron j to neuron i, and nm was the level of neuromodulator at synapse ij. Phasic neuromodulation had a strong effect on action selection and learning. During phasic neuromodulation, synaptic projections from sensory systems and inhibitory neurons are amplified relative to recurrent or associational connections [9]. In our model, the *TOI-State* to *Action* neurons represented sensory connections and the excitatory *Action*-to-*Action* neurons represented the associational connections. To simulate the effect of phasic neuromodulation, inhibitory and sensory connections were amplified by setting *nm* (equation 3) to ten times the combined average activity of the simulated Raphe, and VTA neurons. Otherwise, *nm* was set to 1 for recurrent or association connections. The last column of Table 2 lists connections amplified by phasic neuromodulation. In simulation studies [10] and robotic experiments [11], this mechanism was shown to be effective in making the network exploitive when neuromodulation levels were high and exploratory when neuromodulation levels were low.

Action selection depended on the summed activity of the *Action* neurons after the neural agent reached the TOI. When the *Neural* agent reached the TOI, neural activities of the *Action* and *Neuromodulator* neurons were calculated for ten time-steps

(equations 1-3). The Action neuron with the largest total activity during those ten time-steps dictated the action taken (e.g. if the total *Display* activity was greater then *Escalate*, the agent displayed).

After both the *Neural* and *Opponent* agents chose a tactic, a learning rule, which depended on the current activity of the pre-synaptic neuron, the post-synaptic neuron, the overall activity of the neuromodulatory systems and the payoff from the game, was applied to the equation for the plastic connections (see Table 2):

$$\Delta w_{ij} = \alpha * nm(t-1)s_j(t-1)\left(s_i(t-1)\right) * R \tag{4}$$

where s_j was the pre-synaptic neuron activity level, s_i was the post-synaptic neuron activity level, α was a learning rate set to 0.1, *nm* was the average activity of all neuromodulatory neurons, and *R* was the level of reinforcement based on payoff and cost (equation 5). The pre-synaptic neuron (s_j) in equation 4 was the most active *TOI-State* neuron. The post-synaptic neuron (s_i) could either be the most active *Action* neuron, the *Raphe* neuron, or the *VTA* neuron. Weights were normalized by the square root of sum of squared weights. The level of reinforcement (R, equation 4) was:

$$R = \begin{cases} (\text{Reward - } VTA) - (Cost - Raphe); & \text{TOI - State} \rightarrow \text{Action connection} \\ \text{Reward - VTA}; & \text{TOI - State} \rightarrow \text{VTA connection} \\ Cost - Raphe; & \text{TOI - State} \rightarrow \text{Raphe connection} \end{cases} \tag{5}$$

where the Reward was the *Neural* agent's payoff from Table 1 divided by the maximum possible reward. It was assumed that 5-HT plasticity was based on the predicted cost of an action and DA plasticity was based on the predicted reward of an action. If there was an error in this prediction, weights changed according to equations 4 and 5. If the Raphe or VTA accurately predicted the respective cost or payoff of an action, learning ceased. The *Neural* agent's cost was 1 if seriously injured, the ratio of scratch to serious injury (i.e., 0.3875, Table 1) if scratched, or zero otherwise. The *Neural* agent's reward was set to 1 if it won the resource, 0.5 if it split the resource, and zero otherwise.

Opponent Agent. The *Opponent* followed one of three strategies. In one strategy, referred to as the *Statistical* model, the agent had a probability of escalation independent of the *Neural* agent's tactics, which was set at the beginning of the game to 0.25 or 0.75. In the second strategy, referred to as *Tit-For-Tat* (*TT*), the computer model always repeated the *Neural* agent's previous move. The only exception to this rule was if the *Opponent* agent reached the TOI first in the opening game, in which the *Opponent* opened with a Display. *TT* is a simple, yet effective strategy in game theory, which has shown to be successful in game playing tournaments [8]. In the third strategy, referred to as *Win-Stay, Lose-Shift (WSLS)*, the *Opponent* agent would win and stay with the same action in the following situations: the *Opponent* agent's *Escalate* is met with the *Neural* agent's *Display* or the *Opponent* agent's *Display* is matched by a *Neural* agent's *Display*, otherwise the *Opponent* agent resorted to a lose and shift action [12]. As with the *TT* strategy, the *WSLS* opponent would open with a *Display* action if it arrived at the TOI first on the first game.

3 Results

Adopted Strategies. During the course of a series, the *Neural* agent learned to adopt different strategies depending on the chance of serious injury and its *Opponent's* strategy. To ensure that these strategies did not occur by chance, 100 randomly behaving agents played against all three *Opponents*. The random agents had lesions (i.e. activity set to zero) of both the simulated VTA and Raphe, which resulted in no learning occurring (equation 4). The 95% confidence interval was used as the cutoff for gauging non-random behavior in the random agents. This cutoff corresponded to the probability of selecting a particular action in response to a given *TOI-State* greater than 65% or less than 35% of the time.

The *Neural* agent adapted its behavior depending on its opponent's strategy and environmental conditions (Fig. 2). In response to a given *TOI-State*, the agent could respond randomly (i.e. within the 95% confidence), or significantly tend toward escalation or displaying. There are a total of 27 possible outcomes the *Neural* agent can take with respect to the three different states of the *TOI*. Only a few of these outcomes emerged in the simulations, and these outcomes are represented in Fig. 2 as a triplet pairing (i.e., EEE, DDE, UDE, etc.). The first value in the triplet pairing corresponds to the expected action when the *TOI-State* was *Open*. The second represents the anticipated action when the *TOI-State* was *Escalate*. The third value denotes the expected outcome when the *TOI-State* was *Display*. These triplets are associated with a color spectrum, where aggressive outcomes ('E' in the triplet) are denoted red, passive outcomes ('D' in the triplet) are denoted in blue, and values that do not fall within either outcome ('U' in the triplet) are denoted in yellow.

Against all three opponents, the *Neural* agent adopted *Hawk-like* behavior in "safe" environments, where the probability of serious injury was 0.25 (top row, Fig. 2), and *Dove-Like* behavior in "harsh" environments, where the probability of serious injury was 0.75 (bottom row, Fig. 2). Figure 2 shows an increase in the adoption of 'DDE' strategy (*Neural* agent displayed when the *TOI-State* was *Open* and *Escalate*, and escalated when the *TOI-State* was *Display*) as the probability of serious injury or an opponent escalating increased. This demonstrates that in situations where the *Neural* agent was in a competitive, antagonistic environment, the *Neural* agent tended to behave in a *Dove-like* way (displaying a large proportion of the games in a series). Conversely, Figure 2 also shows an increase in aggressive strategies (i.e., EEE, *Neural* agent escalated when the *TOI-State* was *Open*, *Escalate* and *Display*) as the probability of serious injury or an opponent escalating decreased. This illustrates that in circumstances where the *Neural* agent was in a cooperative, forgiving environment, it tended to adopt more *Hawk-like* behavior (escalating in a larger proportion of the games in a series).

Simulated lesion experiments were carried out to test the effect of neuromodulation on behavior. An intact neuromodulatory system was necessary for appropriate behavior (see Table 3). When the serotonin was removed from the system, by simulated lesions to the Raphe, the *Neural* agent's behavior became more *Hawk-like,* even when the chance of serious injury was high (*Harsh* column in Table 3). When the simulated VTA was lesioned, effectively removing dopaminergic input to the system, the *Neural* agent's behavior became more *Dove-like* (fewer escalations) in all environments.

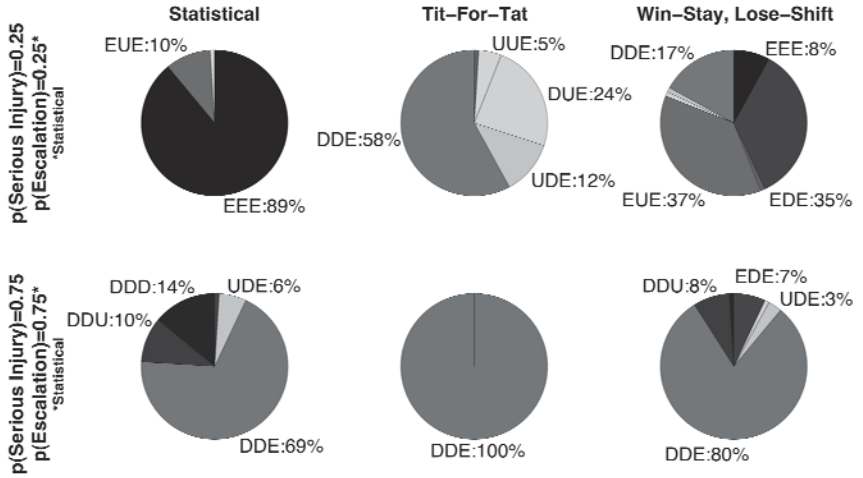


Fig. 2. The pie charts show the proportion of probable actions taken by the *Neural* agent in 100 series of games. There are three *TOI-State* areas (Open, Escalate, and Display), and three outcomes the *Neural* agent can commit to: Escalate (E), Display (D) or Undecided (U). Undecided represents random choice between 'E' and 'D'. The labels represent the *Neural* agent's response to the three *TOI-State* areas. Strategies that are *Dove-like* are displayed in blue, *Hawk-like* are displayed in red, and arbitrary strategies displayed in yellow.

Table 3. Percentage of Escalation for the *Neural* agent

	Control		**Raphe Lesion**		**VTA Lesion**	
	Safe	*Harsh*	*Safe*	*Harsh*	*Safe*	*Harsh*
Statistical	97.65%	10.00%	99.06%	92.86%	34.79%	7.14%
TT	34.15%	13.64%	81.82%	81.82%	24.74%	12.50%
WSLS	93.22%	9.09%	96.88%	96.88%	20.93%	8.22%

The *Neural* agent adapted its behavior to its *Opponent's* strategy. Against the *TT* opponent, the *Neural* agent oscillated between escalating and displaying in successive games. In essence, the *Neural* agent learned to adopt a *TT* strategy against this opponent, which yielded approximately equal reward to both agents. The oscillating neuromodulatory activity corresponded to the alternating actions taken by both agents (Fig. 3A). Against the *WSLS* opponent, the *Neural* agent created opportunities for high payoffs. The high-expected cost and reward were reflected in the serotonergic and dopaminergic activity when both agents escalated (see Fig. 3B: bottom plot, games 79, 82, or 86). In these examples, the *Neural* agent escalated first and its *Opponent* escalated second (Fig. 3B: top plot, games 79, 82, or 86). The *Neural* agent learned that this tactic caused the *Opponent* agent to 'lose-shift' towards *Display* in the following game, which could be taken advantage of by escalating (Fig. 3B: top plot, games 80, 83, or 87). This tactic resulted in a maximal reward to the *Neural*

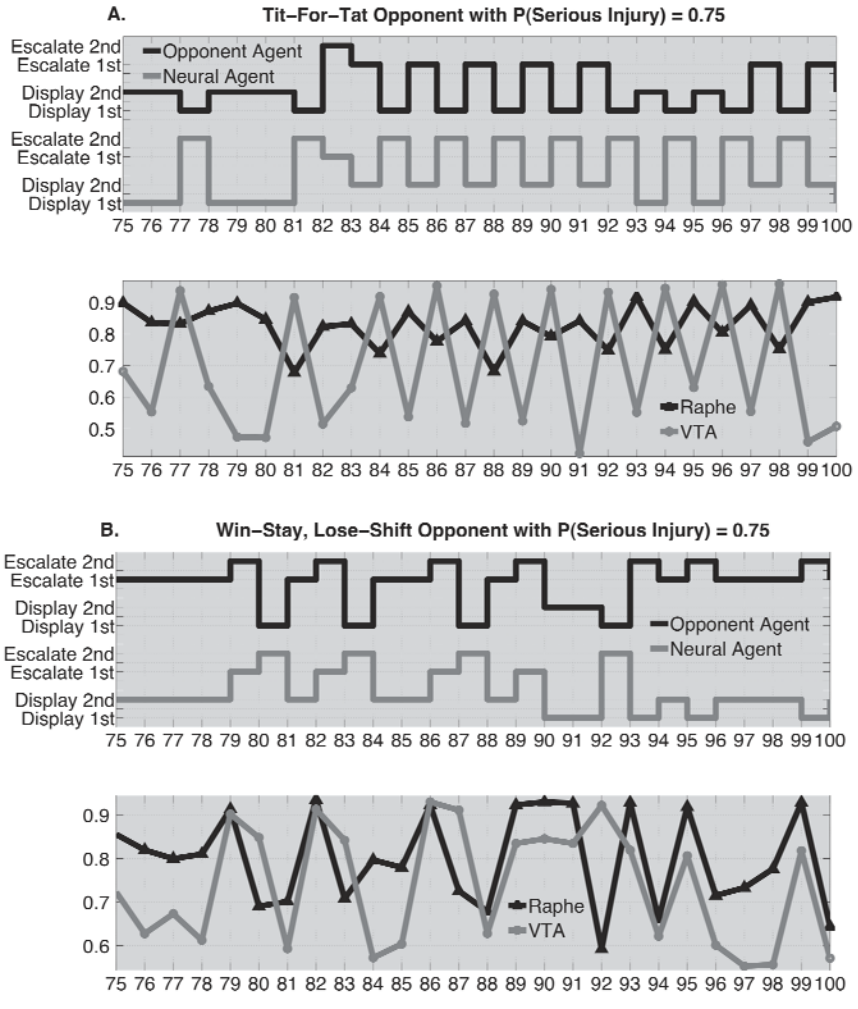


Fig. 3. Actions taken by the *Neural* and *Opponent* agents during the last 25 games of a single series, and the corresponding neuromodulatory activity for the *Neural* agent. The stair plots located on the top half of A and B, are the actions taken by both the *Neural* (green) and *Opponent* (black) agents. The line plots located in the bottom of A and B represent the neuromodulatory activity for the *Neural* agent during the same 25 games of the same series. The red line represents the Raphe activity, and the blue line represents the VTA activity. **A.** *Control* agent versus the *TT* opponent. **B.** *Control* agent versus *WSLS* opponent.

agent but caused the *Opponent* agent to 'lose-shift' back to *Escalate* in the following game (see Figure 3B: top plot, games 81, 84, or 88).

The neural response of the simulated neuromodulators appears to govern the *Neural* agent's actions (Fig. 3). When the VTA activity dropped below the Raphe activity, the neural agent displayed. That is, Raphe activity may be acting as a threshold for the expected cost of upcoming actions, whereas the VTA activity rises and falls based on

the expected reward. When the expected reward is lower than the expected cost, the *Neural* agent tended to display. For example, when a *Neural* agent behaved *Dove-like*, its serotonin activity was high relative to the dopamine activity due to the low expected reward from displaying (see Fig. 3A: games 78-80). In addition, the oscillatory actions taken by the *Neural* agent (see top Fig. 3A: games 84-99), are exactly matched by the oscillatory VTA neuromodulatory activity (see Fig. 3A: games 84-99) rising above and falling below the Raphe neuromodulatory activity. The low fluctuation in Raphe values from one game to the next in Fig. 3A result from the precision of predicted cost when playing a highly predictable opponent using the *TT* strategy. Predicted cost was not as regular for the *Neural* agent when playing against *the WSLS* opponent, which is why the Raphe neuromodulatory activity fluctuated more in Fig. 3B (bottom plot). Although the Raphe activity fluctuated more when playing against the *WSLS* opponent, the actions taken by the *Neural* agent were consistent with the neuromodulatory activity. Thus, the results from the simulated neuromodulatory activity of Fig. 3 suggest that the Raphe neural activity acts as a threshold for aggressive (escalate) or non-aggressive (display) actions taken by the *Neural* agent.

4 Discussion

In the present paper, we showed that an agent, whose behavior was guided by a computational model of the neuromodulatory system, learned to adjust its strategy appropriately depending on environmental conditions and its opponent's strategy in the Hawk-Dove game. The model makes several predictions on how the activity of neuromodulatory systems can lead to appropriate action selection in competitive and cooperative environments.

In constructing the model, it was assumed that DA activity increased as expected reward increased, and that 5-HT activity increased as the expected cost of an action increased. DA appears to be important for reward anticipation [1], and the "wanting" of things, that is, the motivation process in acquiring an object [2]. Thus, having DA activity related to payoff in a game appears to be a reasonable assumption. 5-HT activity appears to modulate behavioral response to risks, stress, threats [3], [13]and social anxiety in primates [14], all of which have a cost associated with them. Moreover, reduced 5-HT transmission is associated with a release of aversive or punishing responses [15]. These assumptions are similar to a model proposed by Daw Kakade and Dayan in which dopamine and serotonin levels track predicted rewards and punishments [16]. However, our model differs in that punishments and rewards are not necessarily mutually inhibitory. Our model takes into consideration that an action could have independent costs and rewards associated with it (i.e., an action may have a high predicted reward, and a high predicted cost).

Given these assumptions, the *Neural* agent adjusted its strategy depending on environmental conditions and on its *Opponent's* strategy (Fig. 2). For example, in situations where it was more likely to sustain a serious injury, the *Neural* agent's behavior became more *Dove-like*. Because the *Neural* agent learned that there was an increased cost and decreased reward to be expected by escalating a confrontation in these harsher conditions, it adapted its strategy to increase in *Display* actions (Fig. 2). No matter which *Opponent* the *Neural* agent faced, it learned to alter its strategy to take

advantage of a no cost escalation in response to its *Opponent's* displaying first. This can be seen in Fig. 2 for all tactics that end in 'E' (e.g., DDE or EDE).

The adaptive behavior demonstrated by the *Neural* agent required an intact neuromodulatory system in which the agent could evaluate the expected cost and the expected reward of a given action. Lowering the simulated serotonin levels resulted in *Hawk-like* tactics that were similar to uncooperative behavior seen in human studies where serotonin levels were lowered [4], [6]. Lowering dopamine levels resulted in the *Neural* agent avoiding risks that lead to a higher payoff. These results are in agreement with a study in which a blockade of dopamine resulted in rats not making an extra effort of climbing over a barricade to get a high reward [17], and a study in which individuals with a polymorphism that lowers levels of dopamine in the prefrontal cortex tended to take less risks in a gambling task [7].

The model makes the following predictions: 1) The interaction between the DA and 5-HT neuromodulatory systems allows for appropriate decision making in games of conflict. In our model, when the VTA activity, which tracked expected reward, exceeded the Raphe activity, which tracked the expected cost, the agent would tend to escalate a fight (see Fig. 3). 2) Impairment to either the dopaminergic or serotonergic system will lead to perseverant, uncooperative behavior. In our model, impairment of the dopaminergic system resulted in risk-averse behavior (*Dove-like)* caused by an inability to assess reward, and impairment of the serotonergic system resulted in risk-taking behavior (*Hawk-like*) due to an inability to assess cost (see Table 3). 3) Although dopamine and serotonin activity appears to be related to different expectations (e.g., predictive reward, anticipated cost), the action of these neuromodulators on downstream targets is similar in that it governs decision-making. That is, phasic neuromodulation shifts an agent's behavior from random and exploratory to decisive and exploitive through differentially modulating synaptic pathways.

The model constructed for the present experiments is based on the notion that all neuromodulators have the same effect on downstream targets, but that specific neuromodulator levels are driven by environmental stimuli [10]. Large, phasic increases in neuromodulator activity cause an organism's behavior to be more exploitive or decisive, whereas lower levels of neuromodulatory activity result in the organism being more exploratory or indecisive. This is in agreement with the idea of cholinergic modulation of attention [18] and noradrenergic modulation of decision-making [19], but extends it to other neuromodulators such as dopamine and serotonin. Our model differs somewhat from the behavioral and neuroscience literature that suggests the role of dopamine is to calculate the reward prediction error, and that serotonin controls the timescale of the evaluation of delayed rewards in reinforcement learning [20], [21]. Instead it may be more in agreement with the proposal that neuromodulators, such as dopamine and serotonin are involved with the discovery of new actions to outcome mappings [22].

We designed our model to investigate how neuromodulation shapes behavior during competitive and cooperative situations. Our model has similarities to other computational models of neuromodulatory processes during decision-making [16], [20]; however, it tests a specific hypothesis of phasic neuromodulation, and applies it to game theory. Other computational models such as Evolutionary Algorithms and Reinforcement Learning have been effective in developing optimal strategies in games of

conflict [23], [24]. It may be of interest in the future to pit our neurobiologically inspired model against reinforcement learning and evolutionary algorithms.

Acknowledgements. This work was supported by the National Science Foundation (EMT/BSSE Award No.: 0829752) and the Office of Naval Research (Award No.: N000140910036).

References

1. Schultz, W., Dayan, P., Montague, P.R.: A neural substrate of prediction and reward. Science 275, 1593–1599 (1997)
2. Berridge, K.C.: Motivation concepts in behavioral neuroscience. Physiol. Behav. 81, 179–209 (2004)
3. Millan, M.J.: The neurobiology and control of anxious states. Prog. Neurobiol. 70, 83–244 (2003)
4. Crockett, M.J., Clark, L., Tabibnia, G., Lieberman, M.D., Robbins, T.W.: Serotonin modulates behavioral reactions to unfairness. Science 320, 1739 (2008)
5. Maynard Smith, J.: Evolution and the theory of games. Cambridge University Press, Cambridge (1982)
6. Wood, R.M., Rilling, J.K., Sanfey, A.G., Bhagwagar, Z., Rogers, R.D.: Effects of tryptophan depletion on the performance of an iterated Prisoner's Dilemma game in healthy adults. Neuropsychopharmacology 31, 1075–1084 (2006)
7. Roussos, P., Giakoumaki, S.G., Pavlakis, S., Bitsios, P.: Planning, decision-making and the COMT rs4818 polymorphism in healthy males. Neuropsychologia 46, 757–763 (2008)
8. Axelrod, R., Hamilton, W.D.: The evolution of cooperation. Science 211, 1390–1396 (1981)
9. Hasselmo, M.E., McGaughy, J.: High acetylcholine levels set circuit dynamics for attention and encoding and low acetylcholine levels set dynamics for consolidation. Prog. Brain Res. 145, 207–231 (2004)
10. Krichmar, J.L.: The Neuromodulatory System – A Framework for Survival and Adaptive Behavior in a Challenging World. Adaptive Behavior 16, 385–399 (2008)
11. Cox, B.R., Krichmar, J.L.: Neuromodulation as a Robot Controller: A Brain Inspired Design Strategy for Controlling Autonomous Robots. IEEE Robotics & Automation Magazine 16, 72–80 (2009)
12. Nowak, M., Sigmund, K.: A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. Nature 364, 56–58 (1993)
13. Amat, J., Paul, E., Zarza, C., Watkins, L.R., Maier, S.F.: Previous experience with behavioral control over stress blocks the behavioral and dorsal raphe nucleus activating effects of later uncontrollable stress: role of the ventral medial prefrontal cortex. J. Neurosci. 26, 13264–13272 (2006)
14. Watson, K.K., Ghodasra, J.H., Platt, M.L.: Serotonin transporter genotype modulates social reward and punishment in rhesus macaques. PLoS ONE e4156 (2009)
15. Cools, R., Roberts, A.C., Robbins, T.W.: Serotoninergic regulation of emotional and behavioural control processes. Trends Cogn. Sci. 12, 31–40 (2008)
16. Daw, N.D., Kakade, S., Dayan, P.: Opponent interactions between serotonin and dopamine. Neural Netw. 15, 603–616 (2002)

17. Denk, F., Walton, M.E., Jennings, K.A., Sharp, T., Rushworth, M.F., Bannerman, D.M.: Differential involvement of serotonin and dopamine systems in cost-benefit decisions about delay or effort. Psychopharmacology (Berl) 179, 587–596 (2005)
18. Pauli, W.M., O'Reilly, R.C.: Attentional control of associative learning-A possible role of the central cholinergic system. Brain Res. 1202, 43–53 (2008)
19. Aston-Jones, G., Cohen, J.D.: An integrative theory of locus coeruleus-norepinephrine function: adaptive gain and optimal performance. Annual Review of Neuroscience 28, 403–450 (2005)
20. Doya, K.: Metalearning and neuromodulation. Neural Netw. 15, 495–506 (2002)
21. Schweighofer, N., Tanaka, S.C., Doya, K.: Serotonin and the evaluation of future rewards: theory, experiments, and possible neural mechanisms. Annals of the New York Academy of Sciences 1104, 289–300 (2007)
22. Redgrave, P., Gurney, K.: The short-latency dopamine signal: a role in discovering novel actions? Nature Reviews 7, 967–975 (2006)
23. Sandholm, T.W., Crites, R.H.: Multiagent reinforcement learning in the Iterated Prisoner's Dilemma. Biosystems 37, 147–166 (1996)
24. Harrald, P.G., Fogel, D.B.: Evolving continuous behaviors in the Iterated Prisoner's Dilemma. Biosystems 37, 135–145 (1996)

Author Index