Gianluca Baldassarre · Marco Mirolli
Editors

Computational and Robotic Models of the Hierarchical Organization of Behavior

Springer

Computational and Robotic Models of the Hierarchical Organization of Behavior

Gianluca Baldassarre • Marco Mirolli
Editors

Computational and Robotic Models of the Hierarchical Organization of Behavior

Springer

Editors
Gianluca Baldassarre
Marco Mirolli
National Research Council
Institute of Cognitive Sciences and
Technologies
Rome, Italy

ISBN 978-3-642-39874-2 ISBN 978-3-642-39875-9 (eBook)
DOI 10.1007/978-3-642-39875-9
Springer Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Contents

Computational and Robotic Models of the Hierarchical Organization of Behavior: An Overview 1
Gianluca Baldassarre and Marco Mirolli

Part I Hierarchical Organization of Behavior in Robots

Behavioral Hierarchy: Exploration and Representation 13
Andrew G. Barto, George Konidaris, and Christopher Vigorito

Self-Organized Functional Hierarchy Through Multiple Timescales: Neuro-dynamical Accounts for Behavioral Compositionality 47
Yuichi Yamashita and Jun Tani

Autonomous Representation Learning in a Developing Agent 63
Jonathan Mugan and Benjamin Kuipers

Hierarchies for Embodied Action Perception 81
Dimitri Ognibene, Yan Wu, Kyuhwa Lee, and Yiannis Demiris

Learning and Coordinating Repertoires of Behaviors with Common Reward: Credit Assignment and Module Activation 99
Constantin A. Rothkopf and Dana H. Ballard

Part II Hierarchical Organization of Animal Behavior

Modular, Multimodal Arm Control Models 129
Stephan Ehrenfeld, Oliver Herbort, and Martin V. Butz

Generalization and Interference in Human Motor Control 155
Luca Lonini, Christos Dimitrakakis, Constantin A. Rothkopf, and Jochen Triesch

A Developmental Framework for Cumulative Learning Robots 177
Mark Lee, James Law, and Martin Hülse

The Hierarchical Accumulation of Knowledge in the Distributed Adaptive Control Architecture 213
Encarni Marcos, Milanka Ringwald, Armin Duff, Martí Sánchez-Fibla, and Paul F.M.J. Verschure

Part III Hierarchical Organization of Animal Brain

The Hierarchical Organisation of Cortical and Basal-Ganglia Systems: A Computationally-Informed Review and Integrated Hypothesis 237
Gianluca Baldassarre, Daniele Caligiore, and Francesco Mannella

Divide and Conquer: Hierarchical Reinforcement Learning and Task Decomposition in Humans 271
Carlos Diuk, Anna Schapiro, Natalia Córdova, José Ribas-Fernandes, Yael Niv, and Matthew Botvinick

Neural Network Modelling of Hierarchical Motor Function in the Brain 293
Juan M. Galeazzi and Simon M. Stringer

Restoring Purpose in Behavior 319
Henry H. Yin

Index 349

Computational and Robotic Models of the Hierarchical Organization of Behavior: An Overview

Gianluca Baldassarre and Marco Mirolli

Abstract The hierarchical organisation of behaviour is a fundamental means through which robots and organisms can acquire and produce sophisticated and flexible behaviours that allow them to solve multiple tasks in multiple conditions. Recently, the research on this topic has been receiving increasing attention. On the one hand, machine learning and robotics are recognising the fundamental importance of the hierarchical organisation of behaviour for building robots that scale up to solve complex tasks, possibly in a cumulative fashion. On the other hand, research in psychology and neuroscience is finding increasing evidence that modularity and hierarchy are pivotal organisation principles of behaviour and of the brain. This book reviews the state of the art in computational and robotic models of the hierarchical organisation of behaviour. Each contribution reviews the main works of the authors on this subject, the open challenges, and promising research directions. Together, the contributions give a good coverage of the most important models, findings, and challenges of the field. This introductory chapter presents the general aims and scope of the book and briefly summarises the contents of each chapter.

1 Modelling the Hierarchical Organization of Behaviour

The performance of flexible behaviour to accomplish multiple goals requires a hierarchical organisation of actions. Each action consists in a sensorimotor mapping that associates a flow of motor commands with the flow of sensory inputs. The mappings related to different actions can be substantially different. When this happens, different actions have to be encoded in distinctive portions of the

G. Baldassarre (✉) · M. Mirolli
Laboratory of Computational Embodied Neuroscience, Institute of Cognitive Sciences and Technologies, National Research Council, Via San Martino della Battaglia 44, I-00185 Rome, Italy
e-mail: gianluca.baldassarre@istc.cnr.it; marco.mirolli@istc.cnr.it

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_1,

architecture of the control system to avoid a crosstalk or *catastrophic interference* between them (French 1999; McCloskey and Cohen 1989). On the other hand, when sensorimotor mappings are very similar their encoding in common structures facilitates *generalisation* and the *reuse of knowledge* for the accomplishment of different purposes (Meunier et al. 2010; Singh 1992). Hierarchical control architectures can allow both the avoidance of catastrophic interference and the exploitation of previously acquired skills to accomplish new tasks. Furthermore, they can also allow the decomposition of complex control problems into smaller tractable problems (Hart and Grupen 2011) and the *chunking* of simpler actions into higher level actions so that increasingly complex behaviours can be acquired cumulatively (Bakker and Schmidhuber 2004; Balleine and Dickinson 1998).

For all these reasons, hierarchical architectures are becoming increasingly important in robotics, in particular when robots are requested to solve not only one task but also multiple tasks, and not only in one condition but also in multiple conditions. Hierarchical architectures are now generally considered as the necessary condition to allow robots to undergo a prolonged autonomous development (Baldassarre and Mirolli 2010) and to scale up robot behaviour to address real-life problems (e.g., Demiris and Khadhouri 2006; Yamashita and Tani 2008). As shown in various chapters presented in this book, state-of-the-art robotics uses hierarchical control architectures to solve multiple tasks, to facilitate the reuse of acquired knowledge to solve other tasks, to facilitate human-to-robot transfer of knowledge, to avoid interference, and so on.

If the adoption of hierarchical architectures is rather new in robotics, the recognition that animal behaviour is hierarchically organised is quite older, dating back at least to the early 1960s (Miller et al. 1960). Today, the hierarchical organisation of behaviour is given for granted in psychology, where it is generally conceived that humans cumulatively build a repertoire of skills that can be flexibly composed to form increasingly complex action programs. Empirical, theoretical, and computational research has been carried out in psychology to understand the details of such a hierarchical organisation of behavior (Botvinick and Plaut 2004; Cooper and Shallice 2000; Fischer 1980; Schneider and Logan 2006; Zacks et al. 2007).

Recent research has also been demonstrating that the hierarchical organisation of behaviour is supported by a hierarchical organisation of the brain (Fuster 2001; Meunier et al. 2010). In particular, animals' brains have been suggested to exploit hierarchy in order to chunk pieces of behaviour so as to reuse them in new tasks (Graybiel 1998), to easily recall behaviours at later times (e.g., to pursue goals associated with them; Redgrave and Gurney 2006), to avoid crosstalk, and to exploit the compositionality allowed by a modular organisation of information (Graziano 2006). Furthermore, recent neuroscientific research is revealing that the brain is hierarchically organised at multiple levels both within cortical (Miller and Cohen 2001) and sub-cortical regions (Yin and Knowlton 2006). In this respect, it can be said that much of the behavioural flexibility exhibited by real organisms depends on the fine hierarchical organisation of the underlying brain structure (Meunier et al. 2010).

The aim of the present book is to review the state of the art in computational and robotic models of the hierarchical organisation of behaviour. The books covers the full spectrum of models: from (scientific) models that try to explain behavioural and/or neural phenomena found in real animals to (technological) models that aim to endow robots with increasingly powerful controllers (including models that try to do both things). Indeed, we are convinced that the cross-fertilisation between different disciplines and approaches is of the most importance both if we want to better understand human behaviour and if we want to construct more useful artificial systems (in the case of hierarchies as in any other). And that computational modelling is the lingua franca that can help to bridge different disciplines that have different concepts, methods, and traditions.

In order to provide a wide overview of current computational research on the hierarchical organisation of behaviour we asked the authors of the various contributions, which represent some of the most active researchers in the field today, to: (1) offer a broad survey of their main works on hierarchical behaviour, (2) highlight the problems and open challenges they see in their area, and (3) when possible, point to the most promising research direction for tackling those challenges. We hope that, taken together, the resulting contributions, collected in one unique venue and following homogeneous guidelines, will aid the reader novel to the field to have a comprehensive panoramic knowledge on hierarchical behaviour, and the more expert reader to be informed of the latest advancements in the field.

2 Book Overview

The chapters of this book have been organised as follows. Part I presents the contributions that address the issue of the hierarchical organisation of behaviour by mainly addressing the problem of how to build skilled robots. Some of these contributions are "bio-inspired", but their distinctive feature is to aim to build useful intelligent technological artefacts. Part II presents computational models directed to understand the hierarchical organisation of behaviour in real animals. The distinctive feature of the models of this part is to aim to answer scientific questions on the organisation of natural intelligence. Although some contributions are robotic and/or refer also to elements of the brain organisation, its major focus is, however, the study of animal behaviour. Finally, Part III presents computational models focussed on understanding the hierarchical organisation of an animal's brain. As those of the previous part, these contributions aim to understand natural intelligence. Although these models refer heavily to behaviour, given their system-level approach to the study of brain their distinctive feature is their close reference to the empirical evidence on the organisation and functioning of real nervous systems. Together, the models illustrated in the chapters represent a solid basis from which to depart to build future robots solving multiple problems or to foster further investigations of the hierarchical organisation of action in the behaviour and brain of real animals. We now present brief summaries of the contents of each book chapter.

2.1 Part I: Hierarchical Organization of Behaviour in Robots

Behavioral Hierarchy: Exploration and Representation. In this chapter, *Andrew G. Barto*, *George Konidaris*, and *Christopher Vigorito* discuss the advantages of behavioural hierarchy from the point of view of hierarchical reinforcement learning. In particular, the authors explore two kinds of benefits that are often overlooked: the benefits given by behavioural hierarchies with respect to the problem of how to efficiently *explore* the environment, and those related to the problem of how to appropriately *represent* the state space on which an agent works. Each of the two kinds of benefits is exemplified by reviewing two couples of computational experiments: the first two experiments show how a behavioural hierarchy can improve exploration in structured environments so as to significantly improve learning speed; the other two experiments show how behavioural hierarchies can allow the use of low complexity function approximation methods for complex problems and how they can allow the selection of different state abstractions for different skills to be learned, thus facilitating learning in high dimensional state spaces. Finally, the authors discuss the generality of their findings and some promising directions for future research in hierarchical reinforcement learning.

Self-Organized Functional Hierarchy Through Multiple Timescales: Neuro-dynamical Accounts for Behavioral Compositionality. In this chapter, *Yuichi Yamashita* and *Jun Tani* review their work on functional hierarchies of behaviors in distributed neural systems that work at multiple timescales. They first discuss the problems encountered in scaling up a modular architecture to complex robotic systems with many degrees of freedom and then present two kinds of systems based on distributed representations. In the first system, the *recurrent neural network with parametric biases* (RNNPB), different sensory-motor sequences are encoded in the fixed activation patterns of the parametric bias units. In the more recent *multiple timescale recurrent neural network* model (MTRNN), different classes of context units have different time constants thus working at different timescales. Through a supervised learning processes, fast units learn to encode different primitives while slow units learn to appropriately switch between those primitives so as to perform higher level behaviors. The authors conclude by discussing the challenge of overcoming the limits of both distributed and localist systems, and how these might be tackled in future work.

Autonomous Representation Learning in a Developing Agent. In this chapter, *Jonathan Mugan* and *Benjamin Kuipers* present a hierarchical computational model, called *Qualitative Learner of Action and Perception* (QLAP), that has been built for developing high-level qualitative representations from low-level continuous sensor and actuator representations while autonomously interacting with its environment. The system has been built on the basis of four principles, devoted in particular to facing the problem of learning useful representations: (1) the exploitation of the synergy between created representations and the agent's development; (2) the generation of new qualitative representations that capture in an abstract form

relevant "phenomena" in the environment; (3) the decomposition of the environment into many sub-parts; (4) the creation of representations making learning more efficient. After explaining these principles, the authors present QLAP in detail and show how such systems can effectively learn useful representations and hierarchical actions in a simulated robotic environment with realistic physics.

Hierarchies for Embodied Action Perception. In this chapter, *Dimitri Ognibene*, *Yan Wu*, *Kyuhwa Lee*, and *Yiannis Demiris* present the *Hierarchical Attentive Multiple Models for Execution and Recognition* (HAMMER) architecture, a bio-inspired learning framework for endowing robots with the ability to understand and imitate human actions. The model, which is based on a repertoire of paired inverse and forward models, is based on three key principles: (1) human knowledge is hierarchically structured, both at the perceptual and at the execution level; (2) perception is active, and driven by task requirements and context; (3) learning is pivotal for action perception, permitting the continuous acquisition of new behaviours. After discussing these principles and describing the general architecture and functioning of HAMMER, the authors review some robotic experiments directed to test the architecture. The experiments demonstrate how the hierarchical organisation and the repertoire of inverse and forward models of the architecture facilitate its autonomous acquisition of complex behaviours and also the acquisition of sequences of actions on the basis of imitation processes.

Learning and Coordinating Repertoires of Behaviors with Common Reward: Credit Assignment and Module Activation. In this chapter, *Constantin A. Rothkopf* and *Dana H. Ballard* present a reinforcement learning computational model that autonomously learns different simultaneous tasks on the basis of different computational modules and learns how to appropriately mix them to maximise a common indistinct reward. The model is composed by many modules that have different state representations and that collectively control the behaviour of the system by sharing the same action space. The proposed algorithm learns to use the modules that are more appropriate for each task at hand by combining each module's reward estimate with an error signal depending on the difference between the unique reward estimate and the sum of the reward estimates of other co-active modules. The efficacy of the model is demonstrated through different experiments involving both simple abstract grid-world tasks and a more complex navigation task in a virtual 3D environment.

2.2 Part II: Hierarchical Organization of Animal Behaviour

Modular, Multimodal Arm Control Models. In this chapter, *Stephan Ehrenfeld*, *Oliver Herbort*, and *Martin V. Butz* discuss some important challenges faced by human motor control and present modular and hierarchical computational architectures that are able to meet those challenges. In particular, the authors identify three main challenges for flexible human motor control: (1) sensory redundancy,

related to the fact that different sources of information about the state of the body and of the environment must be considered for motor control; (2) motor redundancy, related to the fact that different postures and trajectories can be used for the same goal, thus requiring the resolution of behavioural alternatives; (3) uncertainty, related to the fact that many features of the motor task can change from time to time and even during the execution of a movement while both movement execution and sensory processing are always noisy. After identifying the modularity of representations and the hierarchical organisation of planning and control as the two main mechanisms that the brain may exploit to meet these challenges, the authors present a computational model, based on direct inverse modelling mechanisms, that has been used to account for the human behavioural flexibility in motor planning, the SURE-REACH model. Furthermore, since the representational scheme of SURE-REACH is computationally very expensive, a new model, called the Modular Modality Frame (MMF) model, is presented. This model aims to cope with the problem of scalability by further modularising the representational space and by exploiting modular and hierarchical representations at several levels.

Generalization and Interference in Human Motor Control. In this chapter, *Luca Lonini*, *Christos Dimitrakakis*, *Constantin A. Rothkopf*, and *Jochen Triesch* identify the problem of generalization without interference as a fundamental issue in modeling human motor control. Indeed, a cumulative learning system that is learning a new motor skill must be able to efficiently exploit previously acquired abilities while avoiding that catastrophic interference disrupts old knowledge, as is typically the case in simple neural systems. The authors first review the available empirical data on consolidation of procedural memories and then discuss the different computational models that have been proposed for learning multiple tasks in bio-inspired learning architectures.

A Developmental Framework for Cumulative Learning Robots. In this chapter, *Mark Lee*, *James Law*, and *Martin Hülse* provide an extensive review of their work on the Lift-Constraint, Act, Saturate (LCAS) approach, a framework for building robot controllers inspired by developmental psychology (developmental robotics). The basic idea behind LCAS is that development proceeds in a staged fashion, thanks to the presence of learning constraints of various sorts (anatomical, maturational, computational, environmental, etc.). These facilitate skill learning and are progressively released as soon as learning is saturated, thus allowing the cumulative development of increasingly complex abilities. The authors first present their approach to learn sensorimotor mappings: these constitute the fundamental building block of all their systems. Then the authors show how such sensorimotor mappings can be acquired from autonomous experience and how the LCAS framework can lead to a staged development similar to those observed in children. Finally, the authors discuss the role of novelty in the LCAS framework, a possible developmental program for humanoid robots based on it, the research challenges facing the framework, and its relations with other works.

The Hierarchical Accumulation of Knowledge in the Distributed Adaptive Control Architecture. In this chapter, *Encarni Marcos*, *Milanka Ringwald*, *Armin Duff*, *Martí Sánchez-Fibla*, and *Paul F.M.J. Verschure* present the *Distributed Adaptive Control* (DAC) architecture as a biologically-motivated cumulative learning system. In this hierarchical framework, a reactive layer stores built-in stimulus-response associations, an adaptive layer co-adapts behavioral responses and perceptual representations according to reinforcements, and a contextual layer stores sensory-motor chains and uses them to perform memory-based goal-directed behaviors. The authors review several recent studies on this architecture focused on the interactions and arbitration between the higher contextual layer and the lower-level ones and discuss how important challenges in cumulative learning can be tackled within the DAC framework.

2.3 Part III: Hierarchical Organization of Animal Brain

The Hierarchical Organisation of Cortical and Basal-Ganglia Systems: A Computationally-Informed Review and Integrated Hypothesis. In this chapter, *Gianluca Baldassarre*, *Daniele Caligiore*, and *Francesco Mannella* discuss an important problem existing in the neuroscientific literature, related to the presence of two research frameworks focussed on either the hierarchical organisation of cortex or the hierarchical organisation of sub-cortical structures, in particular basal ganglia. The problem is that these two research threads proceed quite in isolation from each other, so failing to account for the integrative nature of hierarchical brain and encountering important limitations in its explanation. To better illustrate this problem, the authors review in detail two of their computational models developed, respectively, in each of the two research frameworks. This allows them to exemplify in detail the problems, brain areas, experiments, main concepts, and limitations of the two frameworks. On this basis, the authors then propose a theoretical integration of the two perspectives and show how this leads to solve most problems found by the two accounts when taken in isolation. Overall, the chapter shows that the cortex and the basal ganglia form a whole highly-integrated system solving all the challenges of choice, selection, and behaviour implementation posed by adaptive behaviour on the basis of a sophisticated hierarchical organisation.

Divide and Conquer: Hierarchical Reinforcement Learning and Task Decomposition in Humans. In this chapter, *Carlos Diuk*, *Anna Schapiro*, *Natalia Córdova*, *José Ribas-Fernandes*, *Yael Niv*, and *Matthew Botvinick* present recent empirical research that investigates the brain mechanisms underlying complex human behaviour requiring task decomposition by using hierarchical reinforcement learning computational models as interpretative tools. In particular, after briefly reviewing the field of hierarchical reinforcement learning, the authors summarise two recent experiments that demonstrate the existence of neural correlates of key predictions of hierarchical reinforcement learning, i.e. the presence of

prediction error signals at different levels of abstraction, and the presence of pseudo-reward signals generated in the presence of sub-goal accomplishment. The authors then focus on the important problem, currently still open in both robotics and neuroscience/psychology, of how a system can autonomously acquire goals/sub-goals that can guide the acquisition of repertoires of skills, essential for supporting hierarchical behaviour. Finally, the authors review three other behavioural and neuroimaging experiments devoted to investigating the brain mechanisms underlying the solution of this problem in animals.

Neural Network Modelling of Hierarchical Motor Function in the Brain. In this chapter, *Juan M. Galeazzi* and *Simon M. Stringer* review their recent work on modelling of the hierarchical organisation of motor function in the brain. This work is based on biologically-plausible neural network architectures in which hierarchically organised dynamical and recurrent neural populations learn through biologically-plausible Hebbian synaptic learning rules. The authors review various simulations in which they demonstrate how their system can learn to perform both low-level motor primitives and high-level motor programs, how novel movement sequences can be learned through the modulation of high-level motor programs by context, and how motor sequences can be learned on the basis of delayed reward signals, thanks to an associative learning rule. Finally, the authors discuss what they consider the most important future challenges in the modelling of hierarchical motor function.

Restoring Purpose in Behavior. In this chapter, *Henry H. Yin* provides an original perspective towards behaviour. Yin is not a computational modeller but a behavioural neuroscientist and has in particular contributed to the understanding of the hierarchical organisation of the brain as expressed in goal-directed behaviour. In this contribution, the author criticises the dominant paradigm in the behavioural and brain sciences, which views behaviour as determined by its antecedent causes (i.e. external or internal stimuli), on the ground that organisms' behaviour is teleologic, and hence must be understood by considering the animal's *goals*. The author's proposal, which builds on the works of cybernetics and its concept of negative feedback, views behaviour as the manifestation of control in systems that act to make inputs match their goals. The author tries to demonstrate the flaws of previous theories in the explanation of behaviour and then discusses the kind of control allowed by a hierarchical organisation of behaviour. Finally, the contribution ends with a discussion of the possible experimental protocols that may be used to exploit the proposal for improving our understanding of animal behaviour.

3 Conclusions

The present book offers a broad overview on the major works and open problems in the field of the hierarchical organisation of behaviour in robots and organisms. The contributions presented here, coming from some of the most active and important

researchers of the field, can surely give both the expert and the non-expert reader a panoramic knowledge on what can be found in this area and this can hopefully prompt new ideas. We thus hope that the book will attract new researchers and foster further investigations in this exciting front-edge field of research at the core of robotics and cognitive science.

Acknowledgements This chapter and a large part of the effort that led to this book have been supported by the Project "IM-CLeVeR—Intrinsically Motivated Cumulative Learning Versatile Robots" funded by the European Commission under the 7th Framework Programme (FP7/2007–2013), "Challenge 2—Cognitive Systems, Interaction, Robotics", Grant Agreement No. ICT-IP-231722. Support or co-support from other institutions, where present, is described in the "Acknowledgments" section of each chapter. The editors of the book thank the EU reviewers (Benjamin Kuipers, Luc Berthouze, and Yasuo Kuniyoshi) and the EU Project Officer (Cécile Huet) for their valuable advice and encouragement. For more information on the IM-CLeVeR project, and for additional multimedia material, see the project web site: http://www.im-clever.eu/. We also thank Simona Bosco for her editorial help with some contributions.

References

Bakker, B., & Schmidhuber, J. (2004). Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In F. Groen, N. Amato, A. Bonarini, E. Yoshida, B. Kruse (Eds.), *Proceedings of the 8-th conference on intelligent autonomous systems (IAS-8)* (pp. 438–445).

Baldassarre, G., & Mirolli, M. (2010). What are the key open challenges for understanding the autonomous cumulative learning of skills? *The Newsletters of the Autonomous Mental Development Technical committee (IEEE CIS AMD Newsletters), 7*(1), 11.

Balleine, B. W., & Dickinson, A. (1998). Goal-directed instrumental action: contingency and incentive learning and their cortical substrates. *Neuropharmacology, 37*(4–5), 407–419.

Botvinick, M., & Plaut, D.C. (2004). Doing without schema hierarchies: a recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review, 111*(2), 395–429.

Cooper, R., & Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology, 17*(4), 297–338.

Demiris, Y., & Khadhouri, B. (2006). Hierarchical attentive multiple models for execution and recognition of actions. *Robotics and Autonomous Systems, 54*(5), 361–369.

Fischer, K. W. (1980). A theory of cognitive development: the control and construction of hierarchies of skills. *Psychological Review, 87*(6), 477–531.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences, 3*(4), 128–135.

Fuster, J. M. (2001). The prefrontal cortex–an update: time is of the essence. *Neuron, 30*, 319–333.

Graybiel, A. M. (1998). The basal ganglia and chunking of action repertoires. *Neurobiology of Learning and Memory, 70*(1–2), 119–136.

Graziano, M. (2006). The organization of behavioral repertoire in motor cortex. *The Annual Review of Neuroscience, 29*, 105–134.

Hart, S., & Grupen, R. (2011). Learning generalizable control programs. *IEEE Transactions on Autonomous Mental Development, 3*(1), 216–231.

McCloskey, M., & Cohen, N. (1989). Catastrophic interference in connectionist networks: the sequential learning problem. In G. H. Bower (Ed.), *The psychology of learning and motivation* (vol. 24, pp. 109–165). San Diego: Academic.

Meunier, D., Lambiotte, R., Bullmore, E. T. (2010). Modular and hierarchically modular organization of brain networks. *Front Neuroscience, 4*, 200.

Miller, E. K., & Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *The Annual Review of Neuroscience, 24*, 167–202.

Miller, G. A., Galanter, E., Pribram, K. H. (1960). *Plans and the structure of behavior*. New York: Adams-Bannister-Cox.

Redgrave, P., & Gurney, K. (2006). The short-latency dopamine signal: a role in discovering novel actions? *Nature Reviews Neuroscience, 7*(12), 967–975.

Schneider, D. W., & Logan, G. D. (2006). Hierarchical control of cognitive processes: switching tasks in sequences. *Journal of Experimental Psychology. General, 135*(4), 623–640.

Singh, S. (1992). Transfer of learning by composing solutions of elemental sequential tasks. *Machine Learning, 8*(3), 323–339.

Yamashita, Y., & Tani, J. (2008). Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. *PLoS Computational Biology, 4*(11), e1000220.

Yin, H. H., & Knowlton, B. J. (2006). The role of the basal ganglia in habit formation. *Nature Reviews Neuroscience, 7*(6), 464–476.

Zacks, J. M., Speer, N. K., Swallow, K. M., Braver, T. S., Reynolds, J. R. (2007). Event perception: a mind-brain perspective. *Psychological Bulletin, 133*(2), 273–293.

Part I
Hierarchical Organization of Behavior in Robots

Behavioral Hierarchy: Exploration and Representation

Andrew G. Barto, George Konidaris, and Christopher Vigorito

Abstract Behavioral modules are units of behavior providing reusable building blocks that can be composed sequentially and hierarchically to generate extensive ranges of behavior. Hierarchies of behavioral modules facilitate learning complex skills and planning at multiple levels of abstraction and enable agents to incrementally improve their competence for facing new challenges that arise over extended periods of time. This chapter focuses on two features of behavioral hierarchy that appear to be less well recognized: its influence on exploratory behavior and the opportunity it affords to reduce the representational challenges of planning and learning in large, complex domains. Four computational examples are described that use methods of hierarchical reinforcement learning to illustrate the influence of behavioral hierarchy on exploration and representation. Beyond illustrating these features, the examples provide support for the central role of behavioral hierarchy in development and learning for both artificial and natural agents.

1 Introduction

Many complex systems found in nature or that humans have designed are organized hierarchically from components—modules—that have some degree of independence. Herbert Simon called such systems "nearly decomposable" and suggested that complex systems tend to take this form because it enhances evolvability due to module stability (Simon 1996, 2005). The subject of modularity has attracted

A.G. Barto (✉) · C. Vigorito
School of Computer Science, University of Massachusetts Amherst, Amherst, MA 01003, USA
e-mail: barto@cs.umass.edu

G. Konidaris
Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139, USA

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_2,

significant attention in psychology, neuroscience, evolutionary biology, computer science, and philosophy (Callebaut and Rasskin-Gutman 2005). Although it is widely accepted that modularity is critically important for understanding and constructing complex systems, no single concept of modularity adequately covers all the cases. Callebaut (2005), for instance, distinguishes modularity of structure from modularity of process, further pointing out significant differences between developmental, evolutionary, neural, and cognitive modularity. This chapter focuses on a type of modularity in which modules are viewed as "building blocks" that can be combined and connected to span large spaces of structured entities, whether they are physical entities or behaviors.

Specifically, this chapter focuses on *behavioral hierarchy*. A behavioral hierarchy is composed of behavioral modules: units of behavior by which an agent interacts with its environment. Hierarchical structure results when a module is itself composed of other modules. Human behavior has long been recognized to exhibit hierarchical structure, with tasks being comprised of subtask sequences, which in turn are built out of simpler actions. Behavioral hierarchy has been an enduring theme in psychology, from the advent of cognitive views (e.g., Lashley 1951; Miller et al. 1960; Newell et al. 1963) to more recent models and cognitive architectures (e.g., Anderson 2004; Botvinick and Plaut 2004; Langley et al. 2009; Langley and Rogers 2004; Schneider and Logan 2006). Behavioral hierarchy has also been of long-standing interest in artificial intelligence (e.g., Sacerdoti 1974), control engineering (e.g., Antsaklis and Passino 1993), software design and analysis (e.g., Alur et al. 2002), and neuroscience (e.g., Botvinick et al. 2009).

In this chapter we use the framework of hierarchical reinforcement learning (HRL) to illustrate the benefits of behavioral hierarchy in addressing problems of *exploration* and *representation* that arise in designing capable learning agents. Although these benefits are well recognized by some, they are not as widely appreciated as are more obvious benefits, and yet they also lie at the heart of the utility of behavioral hierarchy. The relevance of behavioral hierarchy to exploration and representation may also help explain why we see hierarchical structure in the behavior of humans and other animals.

After discussing behavioral hierarchy and HRL, we describe four computational experiments. The first two experiments illustrate how the influence of behavioral hierarchy on exploration can greatly improve learning in structured environments (Sects. 4.3 and 4.4). The third and fourth experiments, respectively, illustrate how behavioral hierarchy can address representational challenges by allowing the use of low-complexity function approximation methods (Sect. 5.1) and how an agent can select different abstractions for each behavioral module to make learning easier in problems with high-dimensional state spaces (Sect. 5.2). Our conclusion emphasizes the generality of the principles illustrated by these examples and their relevance to the design of agents that accumulate competence over extended time periods. The chapter ends with a discussion of prospects for future research. Material in this chapter has appeared previously in Konidaris and Barto (2009a,b), Konidaris (2011), and Vigorito and Barto (2010).

2 Behavioral Hierarchy

The most widely appreciated aspect of behavioral hierarchy is that actions can make use of lower-level actions without concern for the details of their execution. This facilitates both learning complex skills and planning at multiple levels of abstraction. If an agent can construct a useful hierarchy of behavioral modules, which here we think of as *skills*, then the search space of behaviors effectively shrinks. This is because selecting between alternate higher-level skills allows the agent to take larger, more meaningful steps through the search space of behavioral strategies than does selecting between more primitive actions.

Another salient feature of behavioral modularity and hierarchy is that it facilitates the transfer of results of learning in one task to other related tasks. This has been called *transfer learning* and is recognized to be significant for both artificial and natural agents (Taylor and Stone 2009). Rather than acquiring skills from scratch for each problem it faces, an agent can acquire "portable skills" that it can deploy when facing new problems. Illustrating this benefit of behavioral modularity is beyond the scope of this chapter, and we refer the reader to Guestrin et al. (2003), Konidaris et al. (2012a), Konidaris and Barto (2007), Liu and Stone (2006), Mehta et al. (2008), Taylor et al. (2007), and Torrey et al. (2008).

The influence of behavioral hierarchy on exploratory behavior is less well recognized. As new skills are added to an agent's behavioral repertoire, they become available as atomic behavioral modules that may be used when computing behavioral strategies and models of more complex skills. The agent's growing skill set allows it to reach increasingly many areas of its state space that were previously not easily accessible. This in turn allows for learning about more complex environmental dynamics and consequently enables further skill discovery. In this sense, behavioral hierarchy can provide a means for continual, developmental learning in which the acquisition of new skills is bootstrapped on existing structural and procedural knowledge.

Another aspect of behavioral hierarchy, and of behavioral modularity in general, is the opportunity it affords to reduce the representational challenges of planning and learning in large, complex domains. Each module can incorporate its own module-specific representation that includes what is needed for its operation while excluding information that is irrelevant to that operation. When we perform a skill like throwing a ball, for instance, we do not have to take into account the vast range of information available to us that is relevant to other skills, but not to ball throwing. This can make it feasible to plan and learn successfully in complex domains. This benefit of behavioral modularity is well recognized across artificial intelligence, including the various approaches to HRL (e.g., Barto and Mahadevan 2003; Dietterich 2000a; Parr 1998; Parr and Russell 1998; Sutton et al. 1999), but its profound importance may not be widely appreciated.

3 Hierarchical Reinforcement Learning

This chapter is concerned with behavioral hierarchies implemented and learned using computational reinforcement learning (RL, Sutton and Barto 1998). RL algorithms address the problem of how a behaving agent can learn to approximate an optimal behavioral strategy while interacting with its environment. More technically, RL is about the online approximation of solutions to stochastic optimal control problems, usually under conditions of incomplete knowledge of the system being controlled. By emphasizing incremental online algorithms instead of batch-style algorithms, RL not only relates well to the kind of learning we see in animals, but it is also useful for engineering control problems where it can have some advantages over more conventional approaches (Lewis and Vrabie 2009).

RL problems consist of four elements: a set of environmental states; a set of actions available to the agent in each state; a transition function, which specifies the probability of the environment transitioning from one state to another in response to each of the agent's actions; and a reward function, which indicates the amount of reward associated with each such transition. Given these elements, the objective for learning is to discover a behavioral strategy that maximizes cumulative long-term reward. Behavioral strategies are called *policies*, which are rules, or functions, that associate with each possible state an action to be taken in that state. Policies are like stimulus-response rules of learning theories except that a state is a broader concept than a stimulus. A state characterizes relevant aspects of the learning system's environment, which includes information about the world with which the learning agent interacts as well as information about the internal status of the agent itself.

In the usual RL scenario, an agent learns how to perform a "task" specified by a given reward function—for instance, learning how to win at playing backgammon (Tesauro 1994), where a win is rewarded and a loss punished.[1] In these scenarios, there is a conflict between *exploitation* and *exploration*: in deciding which action to take, the agent has to exploit what it has already learned in order to obtain reward, and it has to behave in new ways—explore—to learn better ways of obtaining reward. RL systems have to somehow balance these objectives. However, in other scenarios the goal is to learn a predictive model of the environment, the environment's causal structure, or a collection of widely useful skills. In these scenarios, exploration is itself part of an agent's task. Scenarios like this play a role in some of the illustrations described below.

A key focus of RL researchers is the problem of scaling up RL methods so they can be effective for learning solutions to large-scale problems. Artificial Intelligence researchers address the need for large-scale planning and problem solving by introducing various forms of abstraction into problem solving and planning systems

[1] In RL, the reward signal usually handles both rewards and punishments, which are, respectively, represented by positive and negative values of the numerical reward signal. This abstraction is widely used despite the fact that it is at odds with the differences between appetitive and aversive systems in animals.

(e.g., Fikes et al. 1972; Korf 1985; Sacerdoti 1974). Abstraction allows a system to ignore details that are irrelevant for the task at hand. For example, a macro—a sequence of operators or actions that can be invoked by name as if it were a primitive operator or action—is one of the simplest forms of abstraction. Macros form the basis of hierarchical specifications of action sequences because macros can include other macros in their definitions: a macro can "call" other macros. A macro is a type of module: a reusable component of a program.

HRL uses a generalization of a macro that is often referred to as a "temporally-abstract action," or just an "abstract action" (Sutton et al. 1999). Conventional macros are *open-loop* behavioral policies in the sense that their unfolding does not depend on information sensed during their execution. In contrast, HRL uses *closed-loop* policies as modules, meaning that the course of execution can depend on input from the agent's environment. Unlike a macro, then, which specifies a specific action sequence, an abstract action in HRL generates a sequence of actions that depends on how the actions influence the environment. Thus, in addition to specifying a single "primitive action" to execute in a given state, a policy in HRL can specify a multi-step abstract action to execute in a state, which is characterized by its own policy that can specify both primitive actions and other abstract actions. Once a temporally abstract action is initiated, execution of its policy continues until a specified termination condition is satisfied. Thus, the selection of an abstract action ultimately results in the execution of a sequence of primitive actions as the policy of each component abstract action is "expanded."

Although several approaches to HRL have been developed independently, they all use closed-loop policies as behavioral modules: the *options* formalism of Sutton et al. (1999), the *hierarchies of abstract machines* (HAMs) approach of Parr and Russell (Parr 1998; Parr and Russell 1998), and the MAXQ framework of Dietterich (2000a). These approaches are reviewed by Barto and Mahadevan (2003).

The illustrations in this chapter are based on the theory of options, and we use the terms option and skill interchangeably. An option consists of (1) an *option policy* that directs the agent's behavior when the option is executing, (2) an *initiation set* consisting of all the states in which the option can be initiated, and (3) a *termination condition*, which specifies the conditions under which the option terminates. In addition, a system can maintain, for each option, an *option model*, which is a probabilistic description of the effects of executing an option. As a function of an environment state where the option is initiated, it gives the probability with which the option will terminate at any other state, and it gives the total amount of reward expected over the option's execution. Option models can be learned from experience (usually only approximately) using standard methods. Option models allow algorithms to be extended to handle learning and planning at higher levels of abstraction (Sutton et al. 1999).

The question of where options come from has occupied a number of researchers. In many cases a system designer can define a collection of potentially useful options by hand based on prior knowledge about the agent's environment and tasks. To do this, the designer specifies the policy, initiation set, and termination condition for each of these "native options," which are analogous to action patterns and their

triggering conditions given to an animal through evolution, such as swallowing, the fight-or-flight-or-freeze response, and many others. In other instances, it may be possible to define an option by creating only a reward function for that option, and let the agent learn a policy, together with its initiation and termination conditions, through RL.[2] This is most commonly done by identifying potentially useful option goal states and rewarding the agent for reaching them. These goal states are potential subgoals for problems the agent will face over its future.

Option goal states have been selected by a variety of methods, the most common relying on computing visit or reward statistics over individual states to identify useful subgoals (Digney 1996; McGovern and Barto 2001; Şimşek and Barto 2004, 2009). Graph-based methods (Mannor et al. 2004; Menache et al. 2002; Şimşek et al. 2005) build a state-transition graph and use its properties (e.g., local graph cuts, Şimşek et al. 2005) to identify option goals. Other methods create options that allow the agent to alter otherwise infrequently changing features of its environment (Hengst 2002; Jonsson and Barto 2006). Mugan and Kuipers (2009) developed a related system, which they called Qualitative Learner of Actions and Perception (QLAP), that creates a discrete qualitative representation of a continuous state space using "landmarks" to partition the space. Options are created to change the values of the resulting qualitative variables. A related method was presented by Bakker and Schmidhuber (2004) that relies on unsupervised clustering of low-level sensations to define subgoals. Clustering was also used to cluster subgoals to prevent the creation of multiple options that all correspond to the same underlying skill (Niekum and Barto 2011). Konidaris and colleagues (Konidaris and Barto 2009b; Konidaris et al. 2011a, 2012b) illustrated the utility of setting the goal of an option to be reaching the initiation set of an already-formed option in a process called "skill chaining." This method is used in the example described in Sect. 5.1 below. Still other methods extract options by exploiting commonalities in collections of policies over a single state space (Bernstein 1999; Perkins and Precup 1999; Pickett and Barto 2002; Thrun and Schwartz 1995).

Barto and colleagues (Barto et al. 2004; Singh et al. 2005) proposed that option goals can be identified through reward signals unrelated to a specific task, such as signals triggered by unexpected salient stimuli. This allows an agent to create options that have the potential to be useful for solving *many different tasks* that the agent might face in its environment over the future (see Sect. 4.4 below). Hart and Grupen (2011) proposed a comprehensive approach in which a reward signal identifies behavioral affordances (Gibson 1977) that expose possibilities for action in the environment. Mugan and Kuiper's (2009) QLAP system similarly adopts this "developmental setting" in creating options outside the context of a specific task. Reward signals that do not specify an explicit problem to solve in an environment are related to what psychologists call *intrinsic motivation*. Whereas *extrinsic*

[2]Option reward functions have been called "pseudo reward functions" (Dietterich 2000a) to distinguish them from the reward function that defines the agent's overall task, a distinction not emphasized in this chapter.

motivation means doing something because of some specific rewarding outcome, intrinsic motivation means "doing something because it is inherently interesting or enjoyable" (Ryan and Deci 2000). Intrinsic motivation leads organisms to engage in exploration, play, and other behavior driven by curiosity in the absence of externally-supplied rewards. Schmidhuber (1991a,b) proposed that an RL agent could be given a kind of curiosity by being rewarded whenever it improved its environment model. Although he didn't use the term "intrinsic reward," this was the first contribution to what we now call "intrinsically-motivated RL." An extended discussion of intrinsically-motivated RL is beyond the scope of this chapter, and the reader is referred to Baldassarre and Mirolli (2012) for multiple perspectives on the subject.

Overall, the problem of how to create widely useful options—or more generally, how to create widely useful behavioral modules—is of central importance for progress in HRL, and much research remains to be done. The illustrations described in this chapter are relevant to creating useful options, but since our purpose here is to highlight the utility of behavioral modularity, we do not deeply discuss option creation algorithms.

4 Exploration in Structured Environments

Exploration is indispensable for an RL system's operation. To learn in the absence of explicit instructional information, meaning information that directly tells the agent how it should act, an agent has to try out actions to see what effect they have on reward signals and other aspects of the agent's environment. In other words, behavior has to exhibit some form of *variety*. The simplest RL systems inject randomness into their action-generation procedures so that, for example, they sometimes choose an action uniformly at random from the set of all possible actions instead of taking an action that appears to be one of the best based on what they have learned so far. But exploration does not have to be random: trying something new can be directed intelligently, the only requirement being that an exploratory action is not one of the actions currently assessed to be best for the current situation. Behavioral hierarchy provides an important means for exploring more intelligently than acting randomly because it allows exploration to take advantage of an environment's structure.

In this section we present two examples that illustrate how behavioral hierarchy is implicated in intelligent exploration in structured environments. The first example illustrates how hierarchically-organized skills can be discovered that facilitate learning an environment's structure. The second example shows how a skill hierarchy can make it possible to solve a collection of learning problems that would be essentially impossible to solve otherwise. Both examples use learning problems posed in what Vigorito and Barto (2010) called the "Light Box Environment," described next.

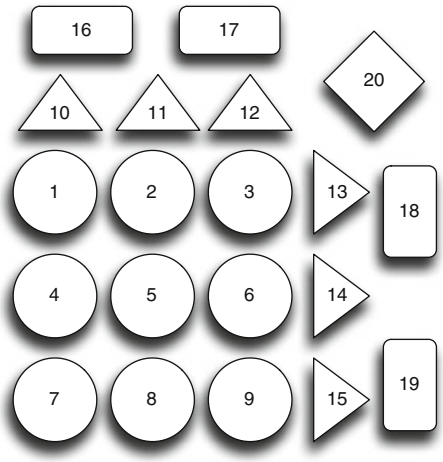

Fig. 1 The Light Box Environment. ©2010 IEEE. Reprinted, with permission, from Vigorito and Barto (2010)

4.1 The Light Box Environment

Figure 1 shows the Light Box Environment in which many different problems can be posed. There is a set of 20 "lights," each of which can be on or off. For each, the agent has an action that toggles the light on or off. Thus there are 20 actions, $2^{20} \approx 1$ million states, and approximately 20 million state-action pairs.

The nine circular lights are simple toggle lights that can be turned on or off by executing their corresponding action. The triangular lights are toggled similarly, but only if certain configurations of circular lights are active, with each triangular light having a different set of dependencies. Similarly, the rectangular lights depend on certain configurations of triangular lights being active, and the diamond-shaped light depends on configurations of the rectangular lights. In this sense there is a strict hierarchy of dependencies in the structure of this environment.

Figure 2 is the causal graph of the Light Box Environment, showing the dependencies between the variables that represent the states (on or off) of the lights. To remove clutter, the dependencies of the variables on themselves are not drawn, but the state of each light obviously depends on its own value at the previous time step. With the exception of these reflexive dependencies, each link in the causal graph indicates that the parent light must be "on" in order to satisfy the dependency.[3]

[3]More technically, the complete structure of an environment like the Light Box can be represented as a set of Dynamic Bayesian Networks (DBNs, Dean and Kanazawa 1989), one for each of the agent's actions. A DBN is a directed acyclic graph with nodes in two layers representing the environment's features at time steps t and $t + 1$, respectively. A DBN also includes conditional probability tables that give the state-transition probabilities. A causal graph as shown in Fig. 2 summarizes the feature dependencies by including a directed edge from one feature's node to another's if and only if there is an agent action whose DBN has an edge from that feature's node at step t to the second feature's node at step $t + 1$ (Jonsson and Barto 2006).

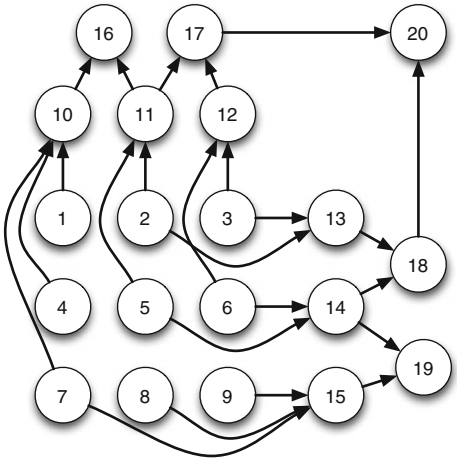

Fig. 2 The causal graph of the Light Box Environment. ©2010 IEEE. Reprinted, with permission, from Vigorito and Barto (2010)

The Light Box Environment is also stochastic (each action taken fails to produce its intended effect with probability 0.1), and it is even more complicated because if an action is taken to toggle a light whose dependencies are not currently satisfied, the environment's entire state is reset to all lights being off.

This environment emulates scenarios in which accurate lower-level knowledge is essential for successfully learning more complex behaviors and their environmental effects. Because of the "reset" dynamics, random action selection is extremely unlikely to successfully turn on any of the lights at the top of the hierarchy. An agent must learn and make use of specific skills in order to reach and remain in the more difficult-to-reach areas of the state space. We emphasize that the agent does not perceive any structure directly: it only senses strings of 20 bits. The structure must be discovered solely from the state transitions the agent experiences, which is a nontrivial problem.

Skills, in the form of options, discovered in the Light Box Environment may have nested policies, the relationship between two of which is shown in Fig. 3. The policies are represented as trees, with internal nodes representing state variables and leaves representing action choices, which may be either primitive actions or options. Branches are labeled with the possible values of their parent variables. In the example shown, the policy for the option to turn on light number 16 (O_{16}) contains at one of its leaves another option (O_{10}) to turn on light number 10, which is one of the dependencies for light number 16. This nesting of policies is a direct result of the hierarchical nature of the environment.

4.2 Exploration for Learning Structure

If the causal structure of an agent's environment is known, in the form of a causal graph like that shown in Fig. 2, it can be used to create a hierarchical collection

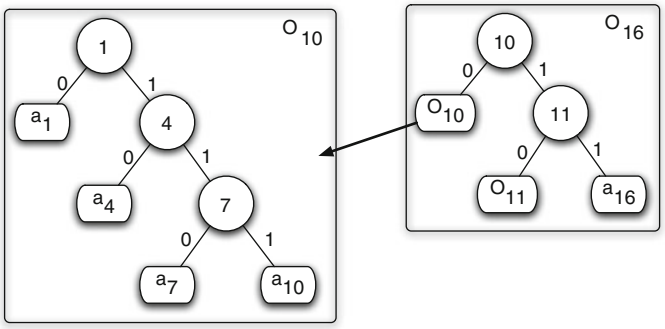

Fig. 3 Examples of option policies in the Light Box Environment. Internal nodes represent state variables, leaves represent action (option) choices. Branches are labeled with state variable values. Notice the nested policies. ©2010 IEEE. Reprinted, with permission, from Vigorito and Barto (2010)

of skills that decompose a task into sub-tasks solved by these skills. Jonsson and Barto (2006) presented an algorithm called Variable Influence Structure Analysis (VISA) that creates options by analyzing the causal graph of an environment. VISA identifies context-action pairs, called *exits* (Hengst 2002), that cause one or more variables to be set to specific values when the given action is executed in the corresponding context. A context is a setting of a subset of the environment's descriptive variables to specific values from which the exit's action has the desired effect; it is like a production system's precondition (Waterman and Hayes-Roth 1978). By searching through the structured representation of the environment, VISA constructs exit options that allow the agent to reliably set collections of variables to any of their possible values. Variables for which such options have been formed are called *controllable variables*. The result of executing VISA is a hierarchy of skills that together represent a solution to the original task. VISA also takes advantage of structure in the environment to learn compact policies for options by eliminating variables on which a policies do not depend, a topic we address in detail in Sect. 5 below.

But it is unrealistic to assume that a causal model of the environment is available to an agent before it has to tackle specific tasks in that environment. Although discovering causal structure is a very subtle problem in general situations (Pearl 2000), the assumptions built into an environment such as the Light Box, in which actions are the complete causes of state changes, make it possible to apply a number of different approaches that have been developed to learn the structure of Bayesian networks. Many of these apply to the case where there is a training set of possible observations (assignments of values to the environment's descriptive variables, or sequences of such assignments) that can be accessed without restriction (e.g., Buntine 1991; Friedman et al. 1998; Heckerman et al. 1995). Some algorithms accelerate learning by selecting the most informative data instances from the training set through a process called active learning (e.g., Murphy 2001; Steck

and Jaakkola 2002; Tong and Koller 2001). These methods effectively perform experiments by fixing subsets of the variables to specific values and sampling over the remaining variables.

In an RL setting where an agent learns while interacting with its environment, it may not be easy for the agent to access data necessary for learning the environment's causal structure. For example, a mobile robot attempting to learn the structure of a new environment cannot transport itself to any location instantaneously, making it difficult to perform all the experiments that might be useful. A collection of exit options formed by an algorithm like VISA would be valuable for allowing the agent to perform useful experiments, but such an algorithm requires knowledge of the environment's causal structure to begin with. Degris et al. (2006), Diuk et al. (2009), Jonsson and Barto (2007), Mugan and Kuipers (2009), and Strehl et al. (2007) proposed methods for learning causal graphs of certain types of dynamic environments as part of the RL process, where it is not possible to sample the process in arbitrary states but only along trajectories.

Here we focus on the structure learning method of Jonsson and Barto (2007), which is an active learning method that takes advantage of the growing structural knowledge to guide behavior so as to collect useful data for refining and extending this knowledge. It follows other structure-learning methods by estimating the probability of observing the data given a particular graphical representation of the system's causal structure, and incrementally searches the space of structures by adding and deleting the edges in the graph in an attempt to find the structure that maximizes this probability. This is not guaranteed to find the best graph (the general problem is NP-complete; Heckerman et al. 1995), but it usually succeeds in finding high-scoring graphs.

Details of Jonsson and Barto's (2007) active learning scheme are fairly complicated and beyond the scope of this chapter, but for current purposes the essential point is that the agent collects sample experiences so that the amount of evidence for each possible graph refinement is roughly equalized. This is accomplished by the agent choosing actions that maximize the entropies of the probability distributions the system uses for its refinement criteria. This is advantageous because having a more uniform distribution over the input values of a given refinement variable makes the evaluation of that refinement more accurate. Thus, correct refinements get discovered more quickly than they do with uniformly random action selection.

This approach does produce faster learning in some environments, but in more complex environments it can fail to discover a significant portion of the environment's structure. This is because this active learning scheme is myopic, only considering the effects of primitive actions at each step, and thus can cause the agent to become stuck in "corners" of the state space that are difficult to get out of. To partially remedy this the agent can learn skills, in the form of options, while it is exploring. Selecting skills can allow the agent to reach configurations of environmental variables that would be difficult to reach using only primitive actions. In this approach, an agent uses its current skill collection to perform experiments so as to expedite structure learning. An experiment in this scheme, like an exit, is composed of a context and an associated primitive action. Vigorito and Barto (2010)

studied a method that selects actions according to Jonsson and Barto's (2007) active structure learning method, but while interacting with its environment it constructs options with hierarchically-defined policies like that shown in Fig. 3 and uses them to improve active structure learning. Instead of selecting only from primitive actions, it selects from the agent's entire suite of primitive actions and available options.

Although this method takes advantage of acquired skills to explore more broadly than possible with just primitive actions, in complex tasks with large state spaces it still may fail to explore many regions of the state space, preferring to maintain uniformity in the current region. This happens because the agent still uses only local information in selecting actions: it only considers how the distributions change locally as a result of executing single actions or single options. This method is illustrated in the Light Box Environment in Sect. 4.3 below, where we call the agent using it the LOCAL agent.

One way to produce a more global method is to allow the agent to use its current environment model to *plan* to reach configurations of environmental variable values that will likely yield more relevant information. Here, executing a plan means executing the plan's policy that specifies actions—primitive actions or options—in response to observations from the environment. Vigorito and Barto (2010) produced an algorithm that does this by introducing a reward function that rewards the agent for achieving a context consisting only of controllable variables. This method is also illustrated in the Light Box Environment in Sect. 4.3 below, where we call the agent using it the GLOBAL agent.

Using this reward function, a policy is computed (using Structured Value Iteration, Boutilier et al. 2000) to reach the rewarding context and execute the action associated with it. The policy is executed to completion before the next best experiment is computed. In this sense, the agent defines its own problems as it continues to learn and explore, becoming "bored" with things that it understands well and focusing its attention on the parts of its environment about which it is uncertain. This is a property of intrinsically-motivated curiosity as pointed out by Schmidhuber (1991b). We think of the reward function used by the GLOBAL agent as producing intrinsic reward signals.

Since the GLOBAL agent starts out with no controllable variables, initial exploration is carried out according to the local active learning scheme like that described above. However, as enough structure is discovered and certain variables become controllable via construction of low-level options, the agent can use these new skills to reliably set contexts about which it has limited experience. When options happen to be created prematurely and are malformed, their lack of utility is discovered fairly quickly by the agent when it attempts to use these options in its experimental plans and they fail repeatedly. These options are removed from the agent's skill collection until the agent performs more experiments relevant to discovering their structure, at which point they will be re-created and tested in further experiments. Once a correct option is learned, its empirical success rate will on average match its expected success rate, and the option will remain in the agent's skill collection to be used in all further experiments. In this way, structure learning is bootstrapped on existing structural and procedural knowledge.

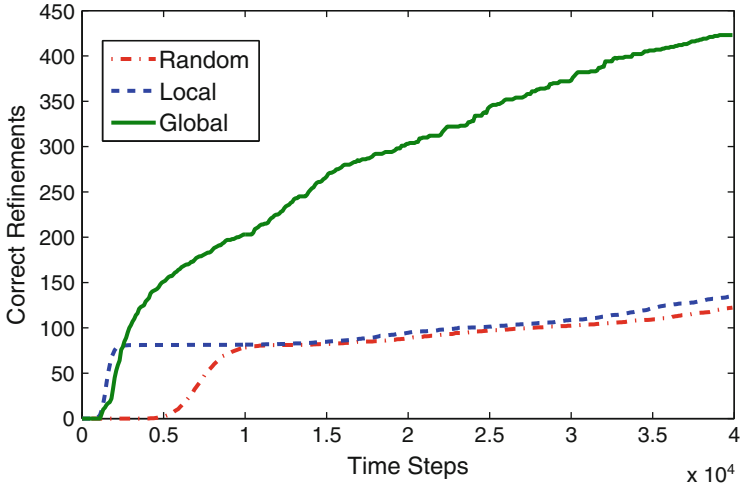

Fig. 4 Structure learning performance for three exploration policies. ©2010 IEEE. Reprinted, with permission, from Vigorito and Barto (2010)

The next section describes a study by Vigorito and Barto (2010) that compares the performance of agents using different types of exploration strategies to guide behavior while learning the structure of the Light Box Environment. Although this environment is relatively simple and was designed with an explicit hierarchical structure, it clearly illustrates the advantages that skill hierarchies can confer to exploration for learning an environment's structure.

4.3 Active Learning of Light Box Structure

Three agents were compared in Vigorito and Barto's (2010) illustration of the behavior of structure-learning agents in the Light Box Environment. Agent RANDOM always selected a random action from its set of available actions, which included options previously acquired, and executed each to completion before choosing another. Agent LOCAL also selected from its set of primitive actions and previously acquired options, but employed the local active learning scheme described above. Agent GLOBAL, on the other hand, used the global active learning scheme of described above. This agent was able to compute plans (via Structured Value Iteration) in order to select among primitive actions and previously acquired options so as to maximize future reward. It therefore used more global information than did agent LOCAL and so could reach more informative areas of the state space.

Since Vigorito and Barto (2010) knew the true transition structure of the Light Box, they could compare the refinements made by each agent at any given time step to the set of refinements that define the correct model. Figure 4 shows the number

of correct refinements discovered by each agent as a function of the number of time steps. The learning curves presented are averages of 30 runs for each agent. Clearly the hierarchical nature of the environment made structure learning very difficult for agents that could not plan ahead in order to reach more informative areas of the state space. Both RANDOM and LOCAL were able to learn what is essentially the bottom layer of the hierarchy, but once this structure was discovered they continually sampled the same areas of the state space and their learning rate leveled out.

GLOBAL, on the other hand, used the options constructed from this initial structure to plan useful experiments in its environment, allowing it to reach areas of the state space that the other agents could not reach reliably. This allowed it to uncover more of the environment's structure, which in turn allowed it to generate new skills that enabled further exploration not possible with only the previous set of options. This bootstrapping process continued until all of the domain structure had been discovered, at which point the agent possessed options to set each light to either on or off. The structured representation of the environment allowed the agent to uncover the transition dynamics without ever visiting a vast majority of the environment states, with GLOBAL reliably finding the correct structure in under 40,000 time steps.

This experiment clearly demonstrates the utility of behavioral modularity in a developmental framework. Using an initially learned set of skills as a basis for further exploration in hierarchically-complex environments like the Light Box is necessary for the discovery and learning of new, more complex skills. But the initial learning of low-level skills may not be sufficient for learning complex tasks. An agent must continue to learn new skills using those it has already discovered in order to make full use of a behavioral hierarchy. Unlike agents RANDOM and LOCAL, GLOBAL used those skills in a planning framework that drove it towards the more informative areas of the state space, and thus it performed much better than the others.

The learning curves for RANDOM and LOCAL in Fig. 4 eventually flatten out, indicating that the agents reach a complexity plateau and are unable to learn beyond that point. By contrast, agent GLOBAL *is* able to learn more complex behavior, using its already acquired skills as building blocks. This kind of open-ended learning, where the agent is able to continually acquire ever more complex skills by progressively building on the skills it has already acquired, characterizes the utility of this approach in complex hierarchically-structured environments.

4.4 Skills for Multiple Tasks

Another benefit of behavioral modularity for exploration occurs when an agent has to face multiple tasks in an environment. Consider the following scenario. Suppose an agent finds itself in an environment in which it will—over the future—face a collection of tasks, where each task is specified by a different reward function. Further suppose that the environment has structure that can be exploited to

accumulate reward, but that the agent does not know anything about this structure to begin with, or how it can be exploited. Assume also that at the start the agent does not know what tasks it will have to face. This scenario captures some features of the situation in which an infant finds itself, or in which an adult finds itself when confronting an environment in which it has little experience and where there is the luxury of a "developmental period" during which relatively few demands are made. What should an agent do to prepare for future challenges? Clearly, it should explore its environment and discover features of its structure that might be useful later on. Our perspective is that it should not only accumulate knowledge of the environment's structure, but it should also acquire behavioral modules in the form of skills that will be on call as it faces multiple tasks over the future, thereby enabling it to learn to perform those tasks more efficiently than would be possible without those skills.

Vigorito and Barto (2010) conducted experiments to illustrate this scenario in the Light Box Environment. They compared the time it took to compute policies for various tasks (defined by different reward functions) for an agent with only primitive actions to the time taken by one with a full hierarchy of options (including primitives). For each of the 20 lights they computed a policy for a task whose reward function was 1 when that light was on and -1 otherwise. They averaged together the computation times of the tasks at each level of the Light Box hierarchy, i.e., all times for circular lights were averaged together, and similarly for triangular and rectangular lights, with only one task for the diamond light.

Results in Fig. 5 show that for the lowest level of the hierarchy, where each task could be accomplished by one primitive action, the two agents took very little time to compute policies, with the options agent being slightly slower due to having a larger action set through which to search. However, once the tasks required longer sequences of actions to perform, there was a significant increase in the computation time for the primitives-only agent, and little or no increase for the options agent. The overhead of computing the options in the first place was thus compensated for once the agent had been confronted with just a few different higher-level tasks. The savings became very substantial above level 2 (note the log scale).

These results illustrate that even with as few as two or three dependencies-per-variable, the benefits of exploring with options can be dramatic. Lacking a hierarchically-structured search, the probability of the agent "stumbling upon" the solution to a higher-level task is extremely small. Indeed, in some environments, this probability is essentially zero. Caching previous experience in the form of reusable skills effectively restructures the search space, enabling the agent to experience the consequences of behavior that would otherwise be very unlikely.

5 Representational Advantages of Behavioral Modularity

In addition to facilitating searches for highly rewarding behavior by restructuring the search space, behavioral modularity presents the opportunity to greatly reduce the

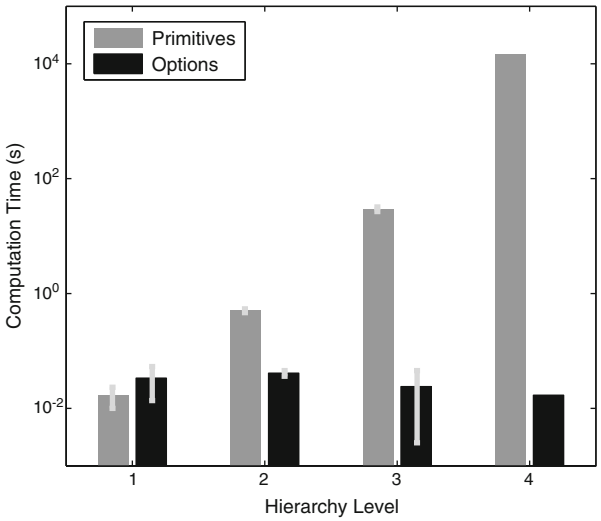

Fig. 5 Policy computation times. Times are given for tasks at varying levels of the Light Box hierarchy for an agent with primitive actions only and for one with options plus primitives. Note the log scale. ©2010 IEEE. Reprinted, with permission, from Vigorito and Barto (2010)

representation problem that is a major challenge in applying RL to problems with large or infinite state and/or action sets. RL algorithms work by learning certain functions of a problem's set of states. Depending on the type of RL algorithm, these functions can include value functions, which for each state (or possibly each state-action pair) provide a prediction of the reward the agent can expect to receive over the future after observing that state (and possibly performing a specific action in that state). Or an algorithm may require directly learning a policy, a function that for each state specifies an action that the agent should take when observing that state.

Learning in environments with large state sets requires compact representations of the needed functions when it is not feasible to explicitly store each function value. This is especially an issue when state sets are multi-dimensional since the size of the state set increases exponentially with the number of dimensions: the well-known "curse of dimensionality" (Bellman 1957). The problem is compounded when states are represented by multiple descriptive variables that can take real numbers as values, which is the case for many applications of RL. Hierarchical organizations of behavioral modules can be especially advantageous in these types of problems.

An important component of all the approaches to HRL is that each behavioral module, or skill, can operate with its own representations of states and actions, as well as its own manner of representing any needed functions. The representations on which a skill depends must be sufficient only for successfully learning and performing that skill, and thus can exclude wide ranges of information that, while

perhaps relevant to other skills, are irrelevant to the skill in question. This implies that the problem of learning an individual skill can be much simpler than learning to solve a larger problem.

To be more specific about how skill-specific representations can differ from one another, consider the following common approach to representing the functions a learning agent may need to maintain. Suppose each environmental state has a "native representation" that is a vector of the values of a set of descriptive variables.[4] For example, the native representation of a state of the Light Box Environment of Sect. 4.1 is a tuple of 20 bits, each indicating the state of one of the lights. In the Pinball Task described in Sect. 5.1 below, the native representation of a state is a 4-tuple of real numbers giving the two-dimensional position of the ball and its two-dimensional velocity. These descriptive variables are usually called features, and we can call them "native features." Of course, what are considered native features are not immutable properties of an environment. In designing artificial agents, these features depend on design decisions and the nature of the problems the agent is supposed to handle; in nature, they depend on the animal's sensory repertoire and perceptual processes, both exteroceptive and interoceptive.

Given a native state representation, it is necessary to implement a function approximation method that receives as input a state's native representation and produces as output a function value, for example, an estimate of the amount of reward expected over the future after the agent visits that state. Linear function approximation is a simple and very well-studied way to do this. In this method, a state's native representation is transformed into another vector specified by a set of *basis functions*, and this new vector's components are linearly combined to produce the function's output value. Each basis function maps the space of native representations to the real numbers. The weights used to combine the components are adjusted by a learning algorithm.

The reason for using a set of basis functions is that although any function implemented by this method is a linear function of the weights—which makes it relatively easy to design and analyze learning algorithms—the overall function can be a very complex nonlinear function of the native representation because the basis functions can be nonlinear functions of the native representation. Further, it is usual practice to use many more basis functions than native features. For example, in continuous domains with many native features, an overall problem's value function or policy approximation may require hundreds of even thousands of basis functions to represent it. The examples described in Sects. 5.1 and 5.2 below use Fourier bases consisting of varying numbers of multivariate sinusoidal basis functions (Konidaris et al. 2011b). Mahadevan (2009) provides a thorough account of the use of basis functions in RL.

Skill-specific representations can therefore differ in terms of native features, function approximation methods, basis functions, or all of these. For example,

[4] Actions similarly have native representations, but we restrict attention to state representations to keep things simple.

a skill may depend only on a subset of the environment's full set of native features, the rest being irrelevant to the skill. More generally, a skill may depend only on an *abstraction* of the full native representation produced by transforming the native representation in a many-to-one manner to produce a more compact description that removes some of the distinctions originally present while preserving relevant information (Dietterich 2000b; Li et al. 2006; Ravindran and Barto 2002). Abstraction is a key approach to solving high-dimensional RL problems, but it is typically difficult to find a single abstraction that applies to the entirety of a complex problem: the problem itself may simply be intrinsically high-dimensional and therefore hard to solve monolithically. Nevertheless, it may at the same time consist of several subproblems that can be captured as skills, each of which can be learned efficiently using a suitable abstraction.

Skill-specific representations can also differ in the function approximation methods they use. In the case of linear function approximation, for example, each skill may use its own set of basis functions. Alternatively, each skill might use the same type of basis functions, but can use fewer of them than would be required to provide sufficiently accurate function approximation over the environment's entire state space. This applies when skills generate activity that is concentrated on a subset of the environmental state space.

In the next section we illustrate the case in which all the options rely on the same native features, but the options are only required to achieve accuracy on subsets of the state space. Each option therefore uses a function approximator based on many fewer basis functions than would be required to achieve adequate accuracy over the entire state space. Konidaris and Barto (2009b) called these "lightweight options," in reference to the relative simplicity of their function approximation methods. In Sect. 5.2, we present an example in which each skill can select a different abstraction of the task's native representation.

5.1 Representing Complex Policies Using a Collection of Lightweight Options

Konidaris and Barto (2009b) use a "Pinball Task" to illustrate lightweight options. A Pinball Task is a dynamic navigation task with a 4-dimensional continuous state space. Figure 6 shows an instance of a Pinball Task. Other instances have different obstructions and different goal locations.

The goal of a Pinball Task is to maneuver a small ball from its starting location into the large red hole in the upper right corner in Fig. 6. The ball's movements are dynamic so its state is described by four real-valued variables: x, y, $\dot{x}$, and $\dot{y}$. Collisions with obstacles are fully elastic and cause the ball to bounce, so rather than merely avoiding obstacles the agent may choose to use them to efficiently reach the red hole. There are five primitive actions: incrementing or decrement $\dot{x}$ or $\dot{y}$ by a small amount (which incurs a reward of -5 per action), or leaving them unchanged

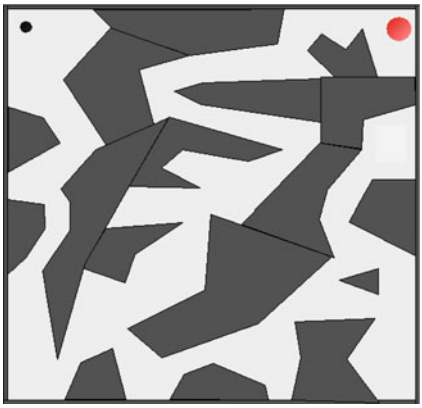

Fig. 6 An instance of a Pinball Task. The goal is to maneuver a small ball from a starting location into the large red hole in the upper right corner. One possible starting location is shown in the upper left corner

(which incurs a reward of −1 per action); reaching the goal obtains a reward of 10,000.

Learning policies for a Pinball Task is very difficult because its dynamic aspects and sharp discontinuities make it difficult for both control and for function approximation, and it requires very accurate control for long periods of time. Konidaris and Barto (2009b) compared Pinball agents that attempted to learn flat policies (that is, policies not hierarchically defined) with agents that used hierarchical policies. The latter agents used *skill chaining* to create skills such that from any point in the state space, the agent could execute a sequence of skills, implemented as options, to reach a given goal region of state space. This algorithm both discovers the skills and determines their initiation sets by learning the local region from which their policies consistently allow the agent to reach its goal. Skill chaining adaptively breaks the problem into a collection of subproblems whose sizes depend on the complexity of policy that they can represent. Its key mechanism is to treat the initiation sets of options already acquired as goal regions for forming new options.[5]

We omit details of skill chaining and refer the reader to Konidaris and Barto (2009b). It is related to what is known in robotics as pre-image backchaining or sequential composition (Burridge et al. 1999). In related work by Neumann et al. (2009), an agent learns to solve a complex task by sequencing motion templates. The most recent related work is by Tedrake (2010) in a model-based control setting.

The options created by skill chaining were lightweight because they used a function approximation scheme with many fewer basis functions than required to approximate the value function adequately over the entire state space. Specifically, whereas the flat policies were derived from value functions combining 1,296 basis functions for each action for the Pinball instance shown in Fig. 6, the value functions

[5] In this respect, it is closely related to the algorithm used by agent GLOBAL described in Sect. 4.3 in which an intrinsic reward is generated for reaching a context consisting only of controllable variables, although that algorithm does not immediately extend to problems with continuous state spaces.

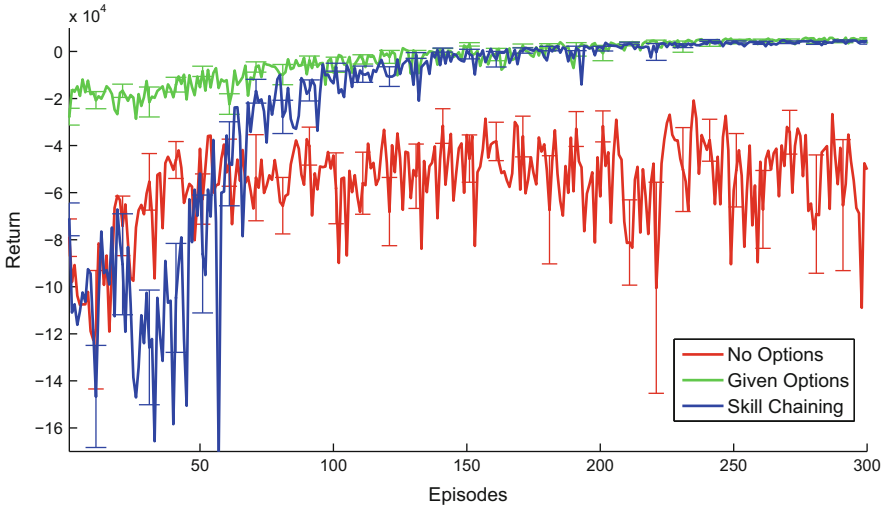

Fig. 7 Performance in the Pinball Task. Graphs show performance for agents employing skill chaining, agents using given pre-learned options, and agents without options. Performance is shown as an agent's return for each episode, which is inversely related to the amount of time the agent took to complete the task. Results are averages of 100 runs

of the options each combined only 256 basis functions for each action.[6] Each option therefore implemented a "lightweight policy"—a much simpler policy than the overall task policy.

Figure 7 shows the performance (averaged over 100 runs) in the Pinball Task shown in Fig. 6 for agents using a flat policy (without options) compared with agents employing lightweight options obtained through skill chaining, and agents starting with given (pre-learned) options. Performance is shown as an agent's return for each episode, which is the total amount of reward it received over that episode. Given the task's reward function, return is inversely related to the number of time steps required to maneuver the ball into the hole. Pre-learned options were obtained using skill chaining over 250 episodes in the same Pinball Task instance. The figure shows that the skill chaining agents performed significantly better than flat agents by 50 episodes, and obtained consistently better solutions by 250 episodes, whereas the flat agents did much worse and were less consistent. Konidaris and Barto (2009b) observed similar results for another instance of the Pinball Task.

Agents that started with pre-learned options did very well initially—with an initial episode return far greater than the average solution eventually learned by

[6] Konidaris and Barto (2009b) used Sarsa with linear function approximation using a 5th-order Fourier basis (1,296 basis functions per action). Option policy learning was done using Q-learning with a 3rd-order Fourier basis (256 basis functions per action). See Konidaris et al. (2011b) for details about using Fourier bases in RL.

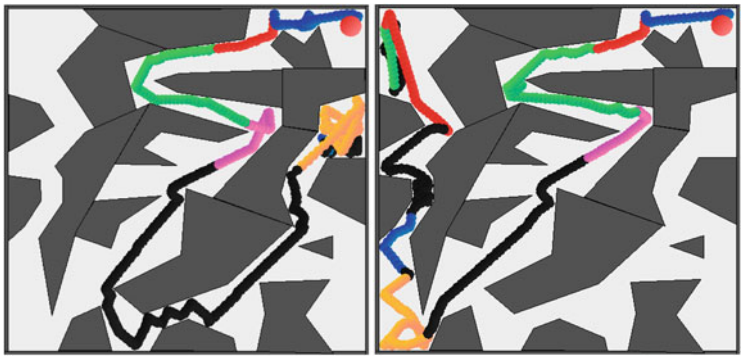

Fig. 8 Sample solution trajectories from different start states in the Pinball Task. Acquired options executed along each sample trajectory are shown in different colors; primitive actions are shown in black

agents without options—and proceeded quickly to the same quality of solution as the agents that discovered the options themselves. This shows that it was the options acquired, and not a by-product of acquiring them, that were responsible for the increase in performance. Given this set of lightweight options, the problem became relatively easy to solve.

Figure 8 shows two sample solution trajectories from different start states accomplished by an agent performing skill chaining in the Pinball Task, with the options executed shown in different colors. The figure illustrates that this agent discovered options corresponding to simple, efficient policies covering segments of the sample trajectories.

The primary benefit of behavioral modularity in this problem is that it reduces the burden of representing the task's value function, allowing each option to focus on representing its own local value function and thereby achieving a better overall solution. Furthermore, using the strategy described here, an agent can create as many options as necessary, thereby adapting to fit the problem difficulty. Although these results illustrate a behavioral hierarchy consisting of only two levels (the level of the options and the level of the primitive actions) the benefit of lightweight options clearly extends to deeper hierarchies.

5.2 Selecting Skill-Specific Abstractions

The lightweight options described above illustrate the fact that a behavioral building block can have lower complexity than would be required for a monolithic solution to a given problem. Although the lightweight options in that example acquired different policies applicable to different parts of the problem's state space, they were all built using the same, complete, set of native environmental features. It is

also possible for each skill to be based on its own abstraction of the problem's state and action spaces. Here we describe an example due to Konidaris and Barto (2009a) and Konidaris (2011) in which each skill selects its own abstraction, which in this case is a subset of the environment's native features.[7] The goal is similar to that of earlier work by Jonsson and Barto (2002) in which each option implemented a separate instance of McCallum's (1996) U-Tree algorithm designed to synthesize state representations from past histories of observations and actions. The goal is also similar to that of Seijen et al. (2007) who studied a method that included special abstraction-switching actions.

In abstraction selection (Konidaris and Barto 2009a; Konidaris 2011), when an agent creates a new option it also selects an appropriate abstraction for learning that option from a library of existing abstractions. Once the agent has determined the option's termination condition—but not yet its policy or initiation set—the agent uses trajectories that reach the option's termination condition as sample trajectories to determine the appropriate abstraction for the new option. The new option policy is then defined using only the native features relevant to its selected abstraction.

Briefly, abstraction selection is treated as a model selection problem, where the agent aims to find the highest likelihood model to fit the returns it receives along its sample trajectories. For the example described below, the class of models considered consists of linear combinations of a set of basis functions defined over the sets of native features corresponding to the abstractions in the library. Konidaris and Barto (2009a) used the Bayesian Information Criterion (BIC), a relatively simple metric that balances an abstraction's fit and its size while incorporating prior beliefs as to which abstractions might be suitable. The process returns a selection probability as output. These properties allow the agent, for example, to select an abstraction when it is only 95 % sure it is the correct one. The abstraction selection algorithm runs on each abstraction in parallel and is incremental and online. It compiles information from the sample transitions into sufficient statistics and does not require them to be stored. For details, see Konidaris and Barto (2009a), Konidaris (2011).

Konidaris and Barto (2009a) illustrate abstraction selection using the Continuous Playroom, an environment where skill-specific abstractions can significantly improve performance. The environment, a continuous version of the environment of Barto et al. (2004) and Singh et al. (2005), consists of three effectors (an eye, a marker, and a hand), five objects (a red button, a green button, a light switch, a bell, a ball, and a monkey) and two environmental variables (whether the light is on, and whether the music is on). The agent is in a 1-unit by 1-unit room, and may move any of its effectors 0.05 units in one of the usual four directions. When both its eye and hand are over an object, the agent may additionally interact with the object, but only if the light is on (unless the object is the light switch). Figure 9 shows an example configuration.

[7] More technically, each abstraction is a projection of the state space onto the subspace spanned by a subset of the set of native features.

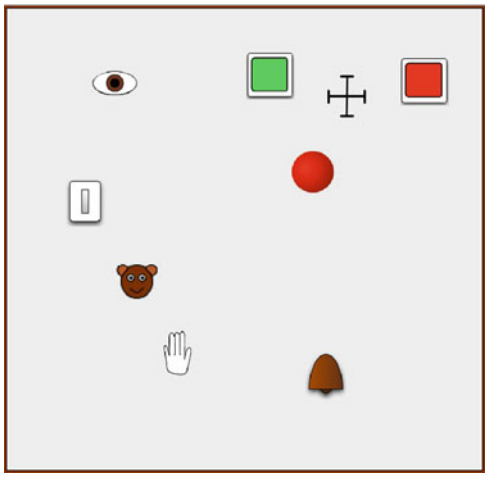

Fig. 9 An example Continuous Playroom. The agent can move its eye, hand, and marker (shown as the cross). The objects are randomly relocated at the start of each episode

Interacting with the green button switches the music on, while the red button switches the music off. The light switch toggles the light. Finally, if the agent interacts with the ball and its marker is over the bell, then the ball hits the bell. Hitting the bell frightens the monkey if the light and music are both on and causes it to squeak, whereupon the agent receives a reward of 100,000 and the episode ends. All other actions cause the agent to receive a reward of -1. At the beginning of each episode the objects are arranged randomly in the room so that they do not overlap.

The agent has 13 possible actions (3 effectors with 4 actions each, plus the interact action), and a full description of the Continuous Playroom requires 18 state variables: x and y pairs for three effectors and five objects (since the position of the monkey may be omitted) plus a variable each for the light and the music. Because the environment is randomly rearranged at the beginning of each episode, the agent must learn the relationships between its effectors and each object, rather than simply the absolute location for its effectors. Moreover, the settings of the light and music are crucial for decision making and must be used in conjunction with object and effector positions. Thus, for task learning the agent used 120 native state features—for each of the four settings of the lights and music it used a set of 30 features representing the difference between each combination of object and effector (Δx and Δy for each object-effector pair, so 5 objects $\times 3$ effectors $\times 2$ differences $= 30$).

The Continuous Playroom is a good example of a problem that should be easy—and is easy for humans—but is made difficult by the large number of features and the interactions between them that cannot all be included in an overall task function approximator: a 1st order Fourier basis over 120 variables that does not treat each variable as independent has 2^{120} basis functions. Thus, it is a problem in which options can greatly improve performance, but only if those options are themselves feasible to learn.

Konidaris and Barto (2009a) assumed that the agent starts with primitive actions only, and that a new option is created for moving each effector over each object when the agent first successfully does so. The task is then to efficiently learn the policies for these options using abstraction selection.

The agent was given an abstraction library consisting of 17 abstractions. Since the environment consists of objects and effectors, abstractions were included for each of the 15 object-effector pairs, with each abstraction consisting of just two features: Δx and Δy for the object-effector pair. Also included in the library was a random abstraction (two features with values selected uniformly at random over [0, 1] at each step), and a null abstraction, which used all 120 native features.

For overall task learning in the Continuous Playroom, Konidaris (2011) used a 10th-order independent Fourier basis over the 120 native features (1,320 basis functions per action). For learning an option policy with an option-specific abstraction, a full 10th-order Fourier basis was used for each action (121 basis functions per option). For option learning without abstraction, the lights and music features were discarded and the agent used the 30 difference features and a 10th-order independent Fourier basis (330 basis functions per action). For the abstraction selection process, a 2nd-order Fourier basis was used.

Figure 10 compares the overall learning curves in the Continuous Playroom for agents that used abstraction selection with those that did not use abstractions when learning option policies. The figure also includes curves for agents that were given pre-learned optimal option policies, and agents that were immediately given the correct abstraction so that performance can be compared to the ideal case. The abstraction selection agents were constrained to only perform selection after 5 episodes.

Figure 10 shows that agents performing abstraction selection performed identically to those that did not use abstractions initially, until about 5 episodes, whereafter the agents were allowed to perform abstraction selection and their performance improved relative to agents that did not use abstractions, matching the performance of agents that were given the correct abstractions in advance by 10 episodes. The difference in performance between agents that did not use abstractions and agents that performed abstraction selection was substantial.

This example illustrates advantages of using abstractions, but it does not explain why it makes sense for an agent to select abstractions from a library instead of developing them itself from its experiences. According to Konidaris and Barto (2009a), the key advantage of abstraction selection is that it shifts the state-space representation problem out of the agent's control loop: instead of having to design a relevant abstraction for each skill as it is discovered, a library of abstractions can be provided and the agent can select a relevant one for each new skill. As the Continuous Playroom example shows, this can significantly bootstrap learning in high-dimensional state spaces, and it allows one to easily incorporate background knowledge about the problem into the agent.

However, providing a library of abstractions to the agent in advance requires both extra design effort and significant designer knowledge of the environment the agent is operating in. This immediately suggests that the abstraction library should

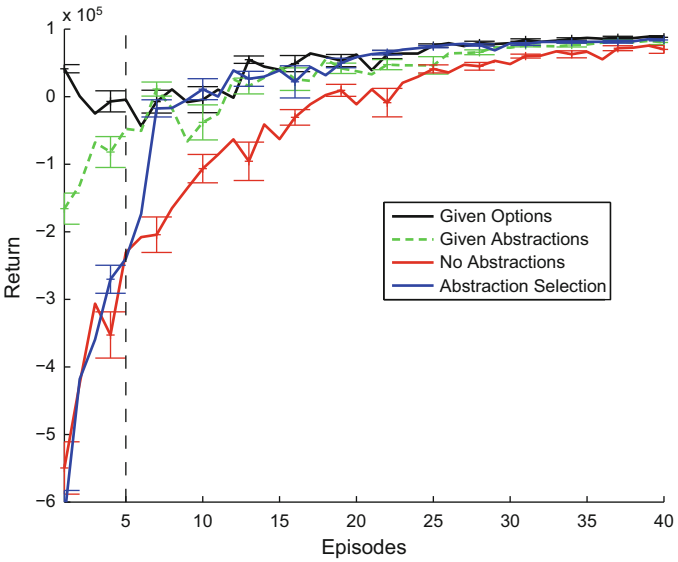

Fig. 10 Learning with abstraction selection. Learning curves for agents given optimal option policies, agents given the correct abstraction in advance, agents using no abstractions, and agents that performed abstraction selection. The dashed line at 5 episodes indicates the first episode where abstraction selection was allowed; before this line, agents using abstraction selection learned option policies without abstractions

be *learned*, rather than given. Just as an agent should learn a library of applicable skills over its lifetime, so should it also learn a library of suitable abstractions over its lifetime—because an agent's abstraction library determines which skills it can learn quickly. We refer the interested reader to Konidaris (2011) for more on these issues.

6 Summary and Prospects

Modularity, in one form or another, is widely observed in both artificial and natural systems. This chapter has focused on behavioral modularity in which "units of behavior," which we call skills (or in the HRL framework, options), are reusable building blocks that can be composed to generate extensive ranges of behavior. Perhaps the most widely appreciated feature of behavioral modularity becomes apparent when hierarchical composition is possible, in which modules can be composed of other, lower-level modules. Behavioral hierarchies facilitate both learning complex skills and planning at multiple levels of abstraction. Another salient feature of behavioral modularity is that it facilitates transfer learning, in which results of learning in one task are transferred to other related tasks.

In a developmental framework, this allows agents to incrementally improve their competence for facing new challenges that arise over extended time periods.

This chapter focuses on two features of behavioral hierarchy that appear to be less well recognized: its influence on exploratory behavior and the opportunity it affords to reduce the representational challenges of planning and learning in large, complex domains. Behavioral hierarchy not only allows exploration to take "bigger steps," but it can also make areas of a state space accessible that would be effectively inaccessible otherwise. Our example using the Light Box Environment illustrates this: without the ability to cache learned behavior into reusable skills, the agent would have very little chance of ever causing "higher-level" events to occur, and thus would have very little chance to learn about these aspects of its environment. This additionally illustrates how behavioral hierarchy can provide a means for continual, developmental learning in which the acquisition of new skills is bootstrapped on previously acquired structural and procedural knowledge.

Our other examples using the Pinball and Continuous Playroom environments illustrate the utility of allowing each module to incorporate its own module-specific representation. Skills typically involve circumscribed aspects of an agent's environment, which allows irrelevant features to be omitted in the representations underlying the skill. Because each skill in a Pinball Task applies to a restricted subset of the problem's state space, skills can use mechanisms of lower complexity than they would need if they were more widely applicable. This was illustrated through the idea of a "lightweight option." In the Continuous Playroom, an example with a higher dimensional state space, the skill-creation mechanism selected the subset of state variables considered most relevant to learning and performing that skill. This is an example of how individual skills can take advantage of skill-specific abstractions. In this case, abstractions were selected from a pre-specified small library of abstractions, but the principle would be similar with other methods for creating and selecting abstractions.

The examples described in this chapter were developed to illustrate methods designed to scale RL to larger and more complex problems. However, we believe that key features of these examples are relevant to modularity in human and animal behavior as well. A guiding principle of all of this research has been that behavioral modules, especially when hierarchically organized, provide a powerful means for continual, developmental learning. It is plausible that the benefits of behavioral hierarchy that we have illustrated through these examples have counterparts that influenced the evolution of hierarchical modularity in humans and other animals.

Although the examples presented in this chapter are based on the options framework of Sutton et al. (1999), it would be a mistake to conclude that this framework is fully developed, that it provides an adequate account of all the issues associated with behavioral hierarchy, or that it is the only framework in which our main points could have been framed. The options framework has the advantages of resting on a strong theoretical base and affording a collection of principled algorithms, but much more development is needed to make it better suited to engineering applications and to improve its account of salient features of behavioral modularity observed in animals. We conclude by briefly describing several avenues for continued development.

1. *Option Creation.* The question of where options come from was briefly discussed in Sect. 3. Although many different approaches are being studied, significant challenges remain. For example, most of the methods that have been developed focus on identifying state space regions that act as useful option goals, and therefore as subgoals for higher-level tasks. These options terminate when the goal region is entered. How to identify useful option goals is far from settled. Further, many skills involve ongoing repetitive behavior, such as running, swinging, and stirring. These kinds of skills are not characterized by goal regions but rather by desired dynamic behavior, such as desired limit cycles. Little work has been done on creating options of this type, although much work in robotics is relevant, such as the methods used in the biped walker of Tedrake et al. (2004) and the control-basis approach of Grupen and colleagues (Hart and Grupen 2011, 2012; Huber and Grupen 1997).
2. *Parameterized Options.* One limitation of the option model of a skill is that most skills, as the term is ordinarily used, seem more flexible than options. What we may think of as a single skill, for instance, throwing a ball, would correspond to many different options whose policies would differ depending on the type of ball, its desired trajectory, etc. Because option policies are closed-loop, the behaviors that options specify are reactive to state information, which could include desired trajectory specifics. However, it might be better to model a skill as a *family* of options that are parametrically related to one another in the sense that they share a basic structure and differ by the settings of only a few parameters. For instance, the ball's speed might be a parameter for a ball-throwing skill. Initial investigations of parameterized options have been undertaken by Soni and Singh (2006) and da Silva et al. (2012).
3. *Representation and Abstraction.* It is well known that how a problem is represented critically influences the performance of problem solving methods Amarel (1981). The problems of designing and/or learning representations are intimately related to the problem of forming useful abstractions: all representations involve abstraction in one guise or another. The forms of representation and abstraction illustrated by the examples in this chapter only touch the surface of these topics, which have long occupied researchers in many fields. There is ample opportunity for integrating the approaches to representation and abstraction taken in RL and HRL, such as those described by Mahadevan (2009) and Osentoski and Mahadevan (2010), with those developed by researchers in other areas.
4. *Many-Level Skill Hierarchies.* There is a shortage of examples in which an agent automatically constructs a many-level skill hierarchy. The GLOBAL agent in the Light Box illustration constructed a four-level skill hierarchy for turning on the highest-level light, and the agent solving a Continuous Playroom problem constructed a multi-level skill hierarchy, but these illustrations were accomplished in relatively simple environments with explicitly-designed hierarchical structure. Other examples, such as the Pinball example described here, create two-level hierarchies (primitives actions plus options that do not invoke other options). Compelling support remains to be developed for the claim that truly

open-learning can emerge from an HRL system with the ability to autonomously form deep skill hierarchies.

5. *Ensembles of Tasks: Competence and Transfer.* A theme only briefly touched upon in this chapter is the importance of considering ensembles of tasks instead of single tasks. The example in Sect. 4.4 illustrates how appropriate skills can make it much easier for an agent to learn any one of several tasks posed by different reward functions in the Light Box Environment. The right kinds of skills—based on the right kinds of representations and abstractions—can make an agent "competent" (White 1959) in an environment, meaning that it is able to efficiently solve many different problems that can come up in that environment. Beyond this, competence can extend from multiple tasks in a single environment to multiple tasks in multiple environments, where the environments have features in common that make it possible to transfer knowledge and skills from one to another. Developing algorithms for learning and problem solving across task ensembles continues to be a challenge. Some perspectives on this theme are discussed by Barto (2012) and Singh et al. (2010).
6. *Real-World Applications.* The illustrations described in this chapter all involved simulated environments that were artificially structured to test and demonstrate the capabilities of various HRL algorithms. Although these are considered to be "toy" environments, the problems they pose are not trivial. For example, a Pinball Task can be very difficult to solve for both humans and conventional control methods. But in many respects HRL theory has far outpaced its applications. Although there are many applications of related techniques to real-world problems (too many to attempt to cite here), there is a shortage of instances that demonstrate the full potential of HRL in important real-world tasks. This is a direction in which progress is much needed.

Acknowledgments The authors thank Sridhar Mahadevan, Rod Grupen, and current and former members of the Autonomous Learning Laboratory who have participated in discussing behavioral hierarchy: Bruno Castro da Silva, Will Dabney, Anders Jonsson, Scott Kuindersma, Scott Niekum, Özgür Şimşek, Andrew Stout, Phil Thomas, and Pippin Wolfe. This research has benefitted from Barto's association with the European Community 7th Framework Programme (FP7/2007–2013), "Challenge 2—Cognitive Systems, Interaction, Robotics", grant agreement No. ICT-IP-231722, project "IM-CLeVeR—Intrinsically Motivated Cumulative Learning Versatile Robots." Some of the research described here was supported by the National Science Foundation under Grant No. IIS-0733581 and by the Air Force Office of Scientific Research under grant FA9550-08-1-0418. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the views of the sponsors.

References

Alur, R., McDougall, M., Yang, Z. (2002). Exploiting behavioral hierarchy for efficient model checking. In E. Brinksma & K. G. Larsen (Eds.), *Computer aided verification: 14th international conference, proceedings (Lecture notes in computer science)* (pp. 338–342). Berlin: Springer.

Amarel, S. (1981). Problems of representation in heuristic problemsolving: related issues in the development ofexpert systems. Technical Report CBM-TR-118, Laboratory for Computer Science, Rutgers University, New Brunswick NJ.

Anderson, J. R. (2004). An integrated theory of mind. *Psychological Review, 111*, 1036–1060.

Antsaklis, P. J., & Passino, K. M. (Eds.), (1993). *An introduction to intelligent and autonomous control*. Norwell MA: Kluwer.

Bakker, B., & Schmidhuber, J. (2004). Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In F. Groen, N. Amato, A. Bonarini, E. Yoshida, B. Kröse (Eds.), *Proceedings of the 8-th conference on intelligent autonomous systems, IAS-8* (pp. 438–445). Amsterdam, The Netherlands: IOS.

Baldassarre, G., & Mirolli, M. (Eds.), (2012). *Intrinsically motivated learning in natural and artificial systems*. Berlin: Springer.

Barto, A., Singh, S., Chentanez, N. (2004). Intrinsically motivated learning of hierarchical collections of skills. In J. Triesch & T. Jebara (Eds.), *Proceedings of the 2004 international conference on development and learning* (pp. 112–119). UCSD Institute for Neural Computation.

Barto, A. G. (2012). Intrinsic motivation and reinforcement learning. In G. Baldassarre & M. Miroll (Eds.), *Intrinsically motivated learning in natural and artificial system*. Berlin: Springer.

Barto, A. G., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamcal Systems: Theory and Applications, 13*, 341–379.

Bellman, R. E. (1957). *Dynamic programming*. Princeton: Princeton University Press.

Bernstein, D. S. (1999). Reusing old policies to accelerate learning on new MDPs. Technical Report Technical Report UM-CS-1999-026, Department of Computer Science, University of Massachusetts Amherst.

Botvinick, M. M., & Plaut, D. C. (2004). Doing without schema hierarchies: a recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review, 111*, 395–429.

Botvinick, M. M., Niv, Y., Barto, A. G. (2009). Hierarchically organized behavior and its neural foundations: a reinforcement-learning perspective. *Cognition, 113*, 262–280.

Boutilier, C., Dearden, R., Goldszmdt, M. (2000). Stochastic dynamic programming with factored representations. *Artificial Intelligence, 121*, 49–107.

Buntine, W. (1991). Theory refinement on Bayesian networks. In B. D'Ambrosio & P. Smets (Eds.), *UAI '91: proceedings of the seventh annual conference on uncertainty in artificial intelligence* (pp. 52–60). San Francisco: Morgan Kaufmann.

Burridge, R. R., Rizzi, A. A., Koditschek, D. E. (1999). Sequential composition of dynamically dextrous robot behaviors. *International Journal of Robotics Research, 18*, 534–555.

Callebaut, W. (2005). The ubiquity of modularity. In W. Callebaut & D. Rasskin-Gutman (Eds.), *Modularity: understanding the development and evolution of natural complex systems* (pp. 3–28). Cambridge: MIT.

Callebaut, W., & Rasskin-Gutman, D. (Eds.) (2005). *Modularity: understanding the development and evolution of natural complex systems*. Cambridge: MIT.

da Silva, B. C., Konidaris, G., & Barto, A. G. (2012). Learning parameterized skills. In J. Langford & J. Pineau (Eds.), *Machine learning, proceedings of the 29th international conference (ICML 2012)* (pp. 1679–1686). Omnipress: Edinburgh.

Dean, T. L., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence, 5*, 142–150.

Degris, T., Sigaud, O., Wuillemin, P. H. (2006). Learning the structure of factored Markov decision processes in reinforcement learning problems. In W. W. Cohen & A. Moore (Eds.), *Machine learning, proceedings of the twenty-third international conference (ICML 2006). ACM international conference proceeding series* (vol. 148, pp. 257–264). New York: ACM.

Dietterich, T. G. (2000a). Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research, 13*, 227–303.

Dietterich, T. G. (2000b). State abstraction in MAXQ hierarchical reinforcement learning. In S. A. Solla, T. K. Leen, K.-R. Müller (Eds.), *Advances in neural information processing systems 12* (pp. 994–1000). Cambridge: MIT.

Digney, B. (1996). Emergent hierarchical control structures: learning reactive/hierarchical relationships inreinforcement environments. In P. Meas, M. Mataric, J.-A. Meyer, J. Pollack, S. W. Wilson (Eds.), *From animals to animats 4: proceedings of the fourth international conference on simulation of adaptive behavior* (pp. 363–372). Cambridge: MIT.

Diuk, C., Li, L., Leffler, B. (2009). The adaptive k-meteorologists problems and its application to structure learning and feature selection in reinforcement learning. In A. P. Danyluk, L. Bottou, M. L. Littman (Eds.), *Proceedings of the 26th annual international conference on machine learning, ICML 2009. ACM international conference proceeding series* (vol. 382, pp. 249–256). New York: ACM.

Fikes, R. E., Hart, P. E., Nilsson, N. J. (1972). Learning and executing generalized robot plans. *Artificial Intelligence, 3*, 251–288.

Friedman, N., Murphy, K., Russell, S. (1998). Learning the structure of dynamic probabilistic networks. In G. F. Cooper & S. Moral (Eds.), *UAI '98: proceedings of the fourteenth conference on uncertainty in artificial intelligence* (pp. 139–147). San Francisco: Morgan Kaufmann.

Guestrin, C., Koller, D., Gearhart, C., Kanodia, N. (2003). Generalizing plans to new environments in relational MDPs. In *IJCAI-03, Proceedings of the eighteenth international joint conference on artificial intelligence* (pp. 1003–1010). San Francisco: Morgan Kaufmann.

Hart, S., & Grupen, R. (2011). Learning generalizable control programs. *IEEE Transactions on Autonomous Mental Development, 3*, 216–231. Special Issue on Representations and Architectures for Cognitive Systems.

Hart, S., & Grupen, R. (2012). Intrinsically motivated affordance discovery and modeling. In G. Baldassarre & M. Mirolli (Eds.), *Intrinsically motivated learning in natural and artificial systems*. Berlin: Springer.

Heckerman, D., Geiger, D., Chickering, D. (1995). Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning, 20*, 197–243.

Hengst, B. (2002). Discovering hierarchy in reinforcement learning with HEXQ. In C. Sammut & A. G. Hoffmann (Eds.), *Machine learning, proceedings of the nineteenth international conference (ICML 2002)* (pp. 243–250). San Francisco: Morgan Kaufmann.

Huber, M., & Grupen, R. A. (1997). A feedback control structure for on-line learning tasks. *Robotics and Autonomous Systems, 22*, 303–315.

Gibson, J. (1977). The theory of affordances. In R. Shaw & J. Bransford (Eds.), *Perceiving, acting, and knowing: toward an ecological psychology* (pp. 67–82). Hillsdale: Lawrence Erlbaum.

Jonsson, A., & Barto, A. G. (2002). Automated state abstraction for options using the U-tree algorithm. In T. G. Dietterich, S. Becker, Z. Ghahramani (Eds.), *Advances in neural information processing systems 14: proceedings of the 2001 neural information processing systems (NIPS) conference* (pp. 1054–1060). Cambridge: MIT.

Jonsson, A., & Barto, A. G. (2006). Causal graph based decomposition of factored mdps. *Journal of Machine Learning Research, 7*, 2259–2301.

Jonsson, A., & Barto, A. G. (2007). Active learning of dynamic Bayesian networks in Markov decision processes. In I. Miguel & W. Rumi (Eds.), *Proceedings of Abstraction, reformulation, and approximation, 7th international symposium, SARA 2007, Whistler, Canada, July 18–21, 2007. Lecture notes in computer science: abstraction, reformulation, and approximation* (vol. 4612, pp. 273–284). Berlin: Springer.

Konidaris, G., & Barto, A. (2007). Building portable options: Skill transfer in reinforcement learning. In M. Veloso (Ed.), *IJCAI 2007, proceedings of the 20th international joint conference on artificial intelligence, Hyderabad, India, 6–12 January 2007* (pp. 895–900). Menlo Park: AAAI Press.

Konidaris, G., & Barto, A. (2009a). Efficient skill learning using abstraction selection. In C. Boutilier (Ed.), *IJCAI 2009, Proceedings of the 21st international joint conference on artificial intelligence, Pasadena, California, USA, 11–17 July 2009* (pp. 1107–1112). Menlo Park: AAAI Press.

Konidaris, G., & Barto, A. (2009b). Skill discovery in continuous reinforcement learning domains using skill chaining. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, A. Culotta (Eds.), *Proceedings of the 2009 conference of Advances in neural information processing systems 22* (pp. 1015–1023). NIPS Foundation.

Konidaris, G., Barto, A., Scheidwasser, I. (2012a). Transfer in reinforcement learning via shared features. *Journal of Machine Learning Research, 13*, 1333–1371.

Konidaris, G., Kuindersma, S., Grupen, R., Barto, A. (2011a). Autonomous skill acquisition on a mobile manipulator. In W. Burgard & D. Roth (Eds.), *Proceedings of the twenty-fifth AAAI conference on artificial intelligence, AAAI 2011* (pp. 1468–1473). San Francisco: AAAI.

Konidaris, G., Kuindersma, S., Grupen, R., Barto, A. (2012b). Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research, 31*, 360–375.

Konidaris, G., Osentoski, S., Thomas, P. (2011b). Value function approximation in reinforcement learning using the Fourier basis. In W. Burgard & D. Roth (Eds.), *Proceedings of the twenty-fifth AAAI conference on artificial intelligence, AAAI 2011* (pp. 380–385). San Francisco: AAAI.

Konidaris, G. D. (2011). *Autonomous robot skill acquisition*. PhD thesis, Computer Science, University of Massachusetts Amherst.

Korf, R. E. (1985). *Learning to solve problems by searching for macro-operators*. Boston: Pitman.

Langley, P., Choi, D., Rogers, S. (2009). Acquisition of hierarchical reactive skills in a unified cognitive architecture. *Cognitive Systems Research, 10*, 316–332.

Langley, P., & Rogers, S. (2004). Cumulative learning of hierarchical skills. In J. Triesch & T. Jebara (Eds.), *Proceedings of the 2004 international conference on development and learning* (pp. 1–8). UCSD Institute for Neural Computation.

Lashley, K. S. (1951). The problem of serial order in behavior. In L. A. Jeffress (Ed.), *Cerebral mechanisms in behavior: the Hixon symposium* (pp. 112–136). New York: Wiley.

Lewis, F. L., & Vrabie, D. (2009). Reinforcement learning and adaptive dynamic programming for feedback control. In *IEEE circuits and systems magazine* (vol. 9, pp. 32–50). IEEE Circuits and Systems Society.

Li, L., Walsh, T., Littman, M. (2006). Towards a unified theory of state abstraction for MDPs. In *International symposium on artificial intelligence and mathematics (ISAIM 2006), Fort Lauderdale, Florida, USA, 4–6 January 2006*.

Liu, Y., & Stone, P. (2006). Value-function-based transfer for reinforcement learning using structure mapping. In *Proceedings, the twenty-first national conference on artificial intelligence and the eighteenth innovative applications of artificial intelligence conference* (pp. 415–420). San Francisco: AAAI.

Mahadevan, S. (2009). *Learning representation and control in Markov decision processes: new frontiers*. *Foundations and trends in machine learning* (vol. 1). Hanover: Now Publishers Inc.

Mannor, S., Menache, I., Hoze, A., Klein, U. (2004). Dynamic abstraction in reinforcement learning via clustering. In C. E. Brodley (Ed.), *Machine learning, proceedings of the twenty-first international conference (ICML 2004). ACM international conference proceeding series* (vol. 69, pp. 560–567). New York: ACM.

McCallum, A. K. (1996). *Reinforcement learning with selective perception and hidden state*. PhD thesis, University of Rochester.

McGovern, A., & Barto, A. (2001). Automatic discovery of subgoals in reinforcement learning using diverse density. In C. E. Brodley & A. P. Danyluk (Eds.), *Proceedings of the eighteenth international conference on machine learning (ICML 2001)* (pp. 361–368). San Francisco: Morgan Kaufmann.

Mehta, N., Natarajan, S., Tadepalli, P. (2008). Transfer in variable-reward hierarchical reinforcement learning. *Machine Learning, 73*, 289–312.

Menache, I., Mannor, S., Shimkin, N. (2002). Q-Cut – Dynamic discovery of sub-goals in reinforcement learning. In *Machine learning: ECML 2002, 13th European conference on machine learning*. *Lecture notes in computer science* (vol. 2430, pp. 295–306). Berlin: Springer.

Miller, G. A., Galanter, E., Pribram, K. H. (1960). *Plans and the structure of behavior*. New York: Holt, Rinehart & Winston.

Mugan, J., & Kuipers, B. (2009). Autonomously learning an action hierarchy using a learned qualitative state representation. In C. Boutilier (Ed.), *IJCAI 2009, Proceedings of the 21st international joint conference on artificial intelligence, Pasadena, California, USA, 11–17 July 2009* (pp. 1175–1180). Menlo Park: AAAI Press.

Murphy, K. (2001). Active learning of causal Bayes net structure. Technical report, Computer Science Division, University of California, Berkeley CA.

Neumann, G., Maass, W., Peters, J. (2009). Learning complex motions by sequencing simpler motion templates. In A. P. Danyluk, L. Bottou, M. L. Littman (Eds.), *Proceedings of the 26th annual international conference on machine learning, ICML 2009. ACM international conference proceeding series* (vol. 382, pp. 753–760). New York: ACM.

Newell, A., Shaw, J. C., Simon, H. A. (1963). GPS, a program that simulates human thought. In J. Feldman (Ed.), *Computers and thought* (pp. 279–293). New York: McGraw-Hill.

Niekum, S., & Barto, A. G. (2011). Clustering via Dirichlet process mixture models for portable skill discovery. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Weinberger (Eds.), *Advances in neural information processing systems 24 (NIPS)* (pp. 1818–1826). Curran Associates.

Osentoski, S., & Mahadevan, S. (2010). Basis function construction for hierarchical reinforcement learning. In W. van der Hoek, G. A. Kaminka, Y. Lespérance, M. Luck, S. Sen (Eds.), *9th international conference on autonomous agents and multiagent systems (AAMAS 2010)* (pp. 747–754). International Foundation for Autonomous Agents and MultiAgent Systems (IFAAMAS).

Parr, R. (1998). *Hierarchical control and learning for Markov decision processes*. PhD thesis, University of California, Berkeley CA.

Parr, R., & Russell, S. (1998). Reinforcement learning with hierarchies of machines. In M. I. Jordan, M. J. Kearns, S. A. Solla (Eds.), *Advances in neural information processing systems 10: proceedings of the 1997 conference* (pp. 1043–1049). Cambridge: MIT.

Pearl, J. (2000). *Causality: models, reasoning, and inference*. Cambridge: Cambridge University Press.

Perkins, T. J., & Precup, D. (1999). Using options for knowledge transfer in reinforcement learning. Technical Report UM-CS-1999-034, University of Massachusetts Amherst.

Pickett, M., & Barto, A. G. (2002). PolicyBlocks: an algorithm for creating useful macro-actions in reinforcement learning. In C. Sammut & A. Hoffmann (Eds.), *Machine learning, proceedings of the nineteenth international conference (ICML 2002)* (pp. 506–513). San Francisco: Morgan Kaufmann.

Ravindran, B., & Barto, A. G. (2002). Model minimization in hierarchical reinforcement learning. In S. Koenig & R. C. Holte (Eds.), *Abstraction, reformulation and approximation, 5th international symposium, SARA 2002, Kananaskis, Alberta, Canada, 2–4 August 2002, proceedings. Lecture notes in computer science* (vol. 2371, pp. 196–211). Berlin: Springer.

Ryan, R. M., & Deci, E. L. (2000). Intrinsic and extrinsic motivations: classic definitions and new directions. *Contemporary Educational Psychology, 25*, 54–67.

Sacerdoti, E. D. (1974). Planning in a hierarchy of abstraction spaces. *Artificial Intelligence, 5*, 115–135.

Schmidhuber, J. (1991a). Adaptive confidence and adaptive curiosity. Technical Report FKI-149-91, Institut für Informatik, Technische Universität München, Arcisstr. 21, 800 München 2, Germany.

Schmidhuber, J. (1991b). A possibility for implementing curiosity and boredom in model-building neural controllers. In J.-A. Meyer & S. W. Wilson (Eds.), *From animals to animats: proceedings of the first international conference on simulation of adaptive behavior (complex adaptive systems)* (pp. 222–227). Cambridge: MIT.

Schneider, D. W., & Logan, G. D. (2006). Hierarchical control of cognitive processes: switching tasks in sequences. *Journal of Experimental Psychology: General, 135*, 623–640.

Simon, H. A. (1996). *The sciences of the artificial*, 3rd edn. Cambridge: MIT.

Simon, H. A. (2005). The structure of complexity in an evolving world: the role of near decomposability. In W. Callebaut & D. Rasskin-Gutman (Eds.), *Modularity: understanding the development and evolution of natural complex systems* (pp. ix–xiii). Cambridge: MIT.

Şimşek, Ö., & Barto, A. (2004). Using relative novelty to identify useful temporal abstractions in reinforcement learning. In C. E. Brodley (Ed.), *Machine learning, proceedings of the twenty-first international conference (ICML 2004) ACM international conference proceeding series* (vol. 69, pp. 751–758). New York: ACM.

Şimşek, Ö., & Barto, A. (2009). Skill characterization based on betweenness. In D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), *Advances in neural information processing systems 21, Proceedings of the twenty-second annual conference on neural information processing systems* (pp. 1497–1504). Red Hook: Curran Associates, Inc.

Şimşek, Ö., Wolfe, A. P., Barto, A. (2005). Identifying useful subgoals in reinforcement learning by local graph partitioning. In L. D. Raedt & S. Wrobel (Eds.), *Machine learning, proceedings of the twenty-second international conference (ICML 2005) ACM international conference proceeding series* (vol. 119, pp. 816–823). New York: ACM.

Singh, S., Barto, A. G., Chentanez, N. (2005). Intrinsically motivated reinforcement learning. In L. K. Saul, Y. Weiss, L. Bottou (Eds.), *Advances in neural information processing systems 17: proceedings of the 2004 conference* (pp. 1281–1288). Cambridge: MIT.

Singh, S., Lewis, R. L., Barto, A. G., Sorg, J. (2010). Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development, 2*, 70–82. Special issue on Active Learning and Intrinsically Motivated Exploration in Robots: Advances and Challenges.

Soni, V., & Singh, S. (2006). Reinforcement learning of hierarchical skills on the Sony Aibo robot. In L. Smith, O. Sporns, C. Yu, M. Gasser, C. Breazeal, G. Deak, J. Weng (Eds.), *Fifth international conference on development and learning (ICDL)*. Bloomington IN.

Steck, H., & Jaakkola, T. (2002). Unsupervised active learning in large domains. In A. Darwiche & N. Friedman (Eds.), *UAI '02, Proceedings of the 18th conference in uncertainty in artificial intelligence* (pp. 469–476). San Francisco: Morgan Kaufmann.

Strehl, A. L., Diuk, C., Littman, M. L. (2007). Efficient structure learning in factored-state MDPs. In *Proceedings of the twenty-second AAAI conference on artificial intelligence*. San Francisco: AAAI.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: an introduction*. Cambridge: MIT.

Sutton, R. S., Precup, D., Singh, S. (1999). Between MDPs and semi-MDPs: a framework for temporal abstraction inreinforcement learning. *Artificial Intelligence, 112*, 181–211.

Taylor, M. E., & Stone, P. (2009). Transfer learning for reinforcement learning domains: a survey. *Journal of Machine Learning Research, 10*, 1633–1685.

Taylor, M. E., Stone, P., Liu, Y. (2007). Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research, 8*, 2125–2167.

Tedrake, R. (2010). LQR-Trees: feedback motion planning on sparse randomized trees. In J. Trinkle, Y. Matsuoka, J. A. Castellanos (Eds.), *Robotics: science and systems V: proceedings of the fifth annual robotics: science and systems conference* (pp. 17–24). Cambridge: MIT.

Tedrake, R., Zhang, T. W., Seung, H. S. (2004). Stochastic policy gradient reinforcement learning on a simple 3D biped. In *Proceedings of the IEEE international conference on intelligent robots and systems (IROS)* (vol. 3, pp. 2849–2854). Japan: Sendai.

Tesauro, G. J. (1994). TD–gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation, 6*, 215–219.

Thrun, S. B., & Schwartz, A. (1995). Finding structure in reinforcement learning. In G. Tesauro, D. S. Touretzky, T. Leen (Eds.), *Advances in neural information processing systems 7: proceedings of the 1994 conference* (pp. 385–392). Cambridge: MIT.

Tong, S., & Koller, D. (2001). Active learning for structure in Bayesian networks. In B. Nebel (Ed.), *Proceedings of the seventeenth international joint conference on artificial intelligence, IJCAI 2001* (pp. 863–869). San Francisco: Morgan Kaufmann.

Torrey, L., Shavlik, J., Walker, J., Maclin, R. (2008). Relational macros for transfer in reinforcement learning. In H. Blockeel, J. Ramon, J. Shavlik, P. Tadepalli (Eds.), *Inductive logic programming 17th international conference, ILP 2007. Lecture notes in computer science* (vol. 4894, pp. 254–268). Berlin: Springer.

van Seijen, H., Whiteson, S., Kester, L. (2007). Switching between representations in reinforcement learning. In R. Babuska & F. C. A. Groen (Eds.), *Interactive collaborative information systems*. *Studies in computational intelligence* (vol. 281, pp. 65–84). Berlin: Springer.

Vigorito, C., & Barto, A. G. (2010). Intrinsically motivated hierarchical skill learning in structured environments. *IEEE Transactions on Autonomous Mental Development, 2*, 83–90. Special issue on Active Learning and Intrinsically Motivated Exploration in Robots: Advances and Challenges.

Waterman, D. A., & Hayes-Roth, F. (1978). *Pattern-directed inference systems*. New York: Academic.

White, R. W. (1959). Motivation reconsidered: the concept of competence. *Psychological Review, 66*, 297–333.

Self-Organized Functional Hierarchy Through Multiple Timescales: Neuro-dynamical Accounts for Behavioral Compositionality

Yuichi Yamashita and Jun Tani

Abstract Based on the ideas of self-organized functional hierarchy in dynamics of distributed neural activities, we introduce a series of neural modeling studies. The models have been examined through robot experiments for the purpose of exploring novel phenomena appearing in the interaction between neural dynamics and physical actions, which could provide us new insights to understand nontrivial brain mechanisms. Those robot experiments successfully showed us how a set of behavior primitives can be learned with distributed neural activity and how functional hierarchy can be developed for manipulating these primitives in a compositional manner.

1 Introduction

Functional hierarchy, defined broadly as the principle by which complex entities may be segmented into simpler elements and these simple elements may be integrated into a complex entity, is a ubiquitous feature of information processing in biological neural systems (Botvinick 2008; Felleman and Van Essen 1991; Fuster 2001; Hilgetag et al. 2000). For example, in primary sensory areas such as V1 and S1, the receptive field of neurons is relatively small, and these neurons respond to features of the stimulus that are simpler than those responded to by higher associative areas. Such functional hierarchy allows humans to perceive various

Y. Yamashita
Department of Functional Brain Research, National Institute of Neuroscience, National Center of Neurology and Psychiatry (NCNP), 4-1-1 Ogawa-Higashi, Kodaira, Tokyo 187-8502, Japan
e-mail: yamay@ncnp.go.jp

J. Tani (✉)
Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea
e-mail: tani1216jp@gmail.com

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_3,

complex sensations and to produce countless patterns of behavior. This productivity of hierarchical organization of neural systems is inevitably related to the perspective of cumulative learning. Therefore, investigating how these functional hierarchies are implemented in neural systems is a fundamental challenge not only in the field of neuroscience but also in the studies of artificial intelligent agents.

The human behavior generation system is a representative example of a system with functional hierarchy. Humans acquire a number of skilled behaviors through the experience of repeatedly carrying out the same movements. Certain components of such movements, through repetitive experiences, are segmented into reusable elements referred to as "primitives." In adapting to various situations, a series of motor primitives are in turn also integrated into diverse sequential behaviors. For example, the action of drinking a cup of coffee may be broken down into a combination of motor primitives such as the motion of reaching for the cup on the table, and the motion of grasping the cup and bringing it to one's mouth. Ideally, these motor primitives should be represented in a generalized manner, in the sense that the representation should be adaptive for differences in locations and in shapes of the cup. Primitives must also be flexible with respect to changes in the sequence of actions; for example, after grasping a cup, one sometimes brings the cup to one's mouth to drink, but one also sometimes takes the cup off the table to wash up. It is this adaptability (intra-primitive level) and flexibility (inter-primitive level) of primitives that allow humans to generate countless patterns of sequential behaviors. The idea underlying this basic process was proposed by Arbib et al. (1998) in terms of "schema theory" and has since been used as the basis for many studies (e.g., Kuniyoshi and Sangawa 2006; Mussa-Ivaldi and Bizzi 2000).

A number of biological observations suggest the existence of motor primitives in real animals. At the behavioral level, Thoroughman and Shadmehr (2000), for example, showed that humans learn the dynamics of reaching motions through a flexible combination of movement elements. Sakai et al. (2003) showed that, in visuomotor sequential learning, human subjects spontaneously segmented motor sequences into elementary movements. At the level of animal muscle movement, Giszter et al. (1993), through observations of muscle movement in the frog's leg, found that there are a finite number of linearly combinable modules, organized in terms of muscle synergies on limbs. At the brain level, meanwhile, it has been shown that electrical stimulation in the primary motor and premotor cortex of the monkey brain evokes coordinated movements, such as reaching and grasping (Graziano et al. 2002). These observations strongly suggest that the diversity of behavior sequences in animals is made up of flexible combinations of reusable movement elements, i.e. motor primitives. From the perspective of cognitive science, such productivity of the animal behavior generation system is also referred to as behavioral "compositionality." The concept of compositionality originates in the principle of compositionality in linguistics (Evans 1981), describing that the meaning of complex expression is determined by the meaning of its constituents and the rules that combine them. This principle can be naturally transferred to our problem such that complex goal-directed behavior sequences can be generated by combining a set of re-usable behavior primitives by following corresponding

rules. Although this is a minimal notion of the compositionality, the functional hierarchy of behavior organization systems plays a key role in achieving behavioral compositionality. The idea of compositionality is also related to the higher cognitive functions of human and artificial agents, such as executive control, meta-cognition, and logical thought. The functional hierarchy in neural systems could be considered as one possible implementation of such compositional structures of cognitive system. Therefore, investigating the underlying mechanisms of functional hierarchy in behavior generating systems may contribute to further understanding of the higher cognitive functions of human and artificial agents.

In the field of conventional cognitive science, it has been considered that functional hierarchy is achieved by means of symbol representation and their manipulations in the higher cognitive level. In contrast, we have assumed an alternative account that the functional hierarchy of behavior generation systems could be self-organized in the neural dynamics of distributed activity through repetitive learning of sensory-motor experiences. In the following, we will introduce a series of neural modeling studies based on these ideas. The models have been examined through robot experiments for the purpose of exploring novel phenomena appearing in the interaction between neuro-dynamics and physical actions which could provide us with new insights to understand nontrivial brain mechanisms. Those robot experiments successfully showed us how a set of behavior primitives can be learned with distributed neural activity and how functional hierarchy can be developed for manipulating these primitives in a goal-directed manner.

2 Self-Organized Segmentation of Motor Primitives in Distributed Representation

First, we show how a set of behavior primitives can self-organize with distributed neural activity through repetitive learning of behavioral experiences. For this problem, we (Tani and Nolfi 1999) first proposed a model of gated modular networks based on a "local representation" scheme, representative examples being the "mixture of expert" model (Baldassarre 2002) and the "MOSAIC" model (Haruno et al. 2001).

In the local representation schema, functional hierarchy is realized through the use of explicit hierarchical structure, with local modules representing motor primitives in the lower level, and a higher module representing the order of motor primitives switched via additional mechanisms such as gate-selection (Fig. 1).

The modular networks as forward models compete to predict current sensory-motor sequence and the gate of the winner, established on the basis of the minimum prediction error opens exclusively. The winner is entitled to learn the pattern to become an expert for it. Although the local representation model worked successfully in a simple simulation experiment of navigation learning, it could not scale easily to more complex problems dealing with robots with larger degrees of freedom (DOF)(Tani et al. 2008a).

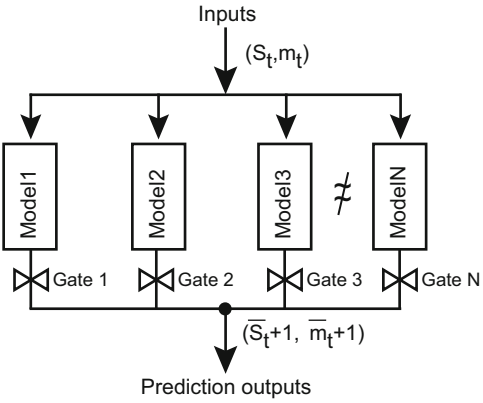

Fig. 1 Gated modular network. Motor primitives are represented in local modules. By selecting the gates in different order, different behavior sequences can be generated. s_t and m_t are current sensory-motor inputs, and s_{t+1} and m_{t+1} are predictions of the network for the next time step, respectively

For this reason, we proposed an alternative scheme based on distributed representation, which is referred to as the recurrent neural network with parametric biases (RNNPB; for details of the modeling, please refer to our prior publications (Sugita and Tani 2004; Tani 2003; Tani and Ito 2003)). The PB is static vector input to the network which acts as the bifurcation parameters of nonlinear dynamical systems (Tani 2003). Owing to this characteristic of the PB, the proposed network was able to generate multiple patterns of sensory-motor sequences through the self-organized associations between a specific PB activity and different dynamic patterns in the network (i.e., different combinations of behavior primitives). The learning of RNNPB is conducted by means of usual back-propagation through time (BPTT) algorithm. In the learning phase, a set of movement patterns are learned through the forward model of the RNNPB by self-organized both the PB vectors, which are assigned differently for each movement pattern, and a synaptic weight matrix, which is common for all the patterns. After learning, learned sensory-motor sequence patterns can be regenerated by setting corresponding PB values (Fig. 2a).

The robot experiment (Tani and Ito 2003) showed that a set of discrete movements and periodic movements can be learned simultaneously as fixed point and limit cycling attractors, respectively (Fig. 3). These results suggest that, continuous behavior sequences are segmented into chunks of spatio-temporal patterns of the sensory-motor flow (primitives), each of which corresponds to different values of PB. The results also showed that the system has certain generalization capability in learning, in the sense that similar patterns were learned with those PB values that were close to each other (Tani et al. 2004). For example, the changes in the amplitude and frequencies of cyclic movements can be extracted into the two-dimensional PB space, suggesting that those movements were generated by interpolating the set of actions in training data (Ito and Tani 2004).

Another important feature of the RNNPB is that the system works as both a behavior recognizer and generator (Fig. 2). When given a fixed PB vector, the RNNPB generates the corresponding dynamic patterns. On the other hand, when given target patterns to be recognized, the corresponding PB vectors are obtained

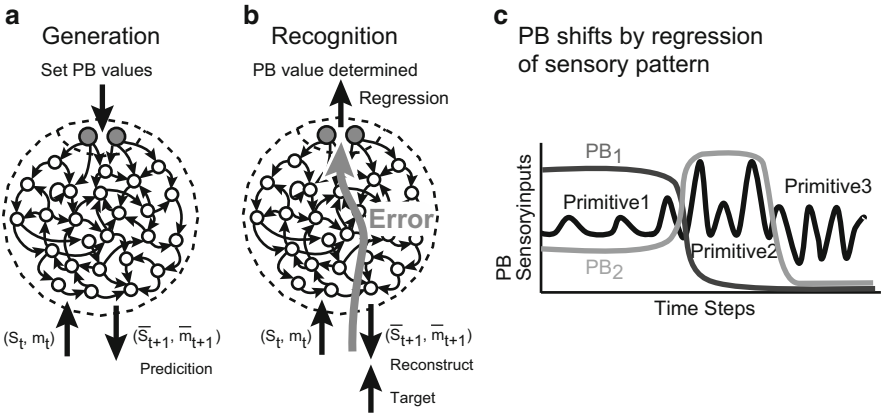

Fig. 2 RNNPB
(**a**) Generation mode. By setting a particular value of PB, spatio-temporal patterns of the sensory-motor sequence (primitives) can be generated based on the self-organized associations between a specific PB activity and different dynamic patterns in the network. *Gray-filled circle* corresponds to PB units which give static vector inputs to the network. (**b**) Recognition mode. When a target patterns are given, the corresponding PB vectors can be obtained through an iterative inverse computation. (**c**) Combination of behavior primitives through the switching of PB activities. s_t and m_t are current sensory-motor inputs, and s_{t+1} and m_{t+1} are predictions of the network for the next time step, respectively

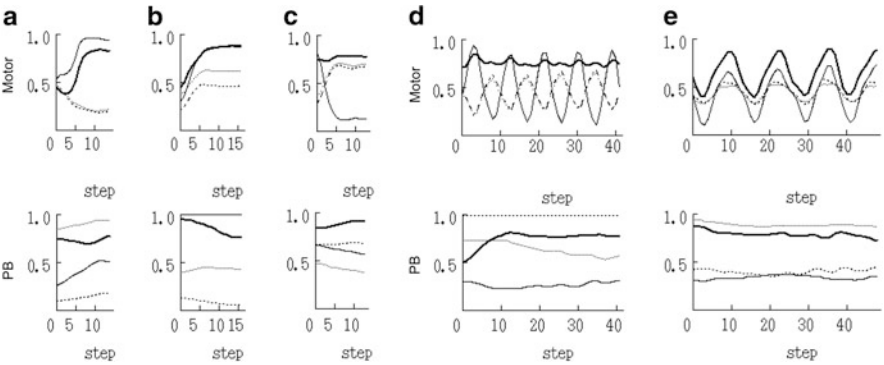

Fig. 3 Examples of learned movements, each of which corresponds to different values of PB. Discrete movements (**a**), (**b**), and (**c**), and periodic movements (**d**) and (**e**). *Lines* correspond to 4 out of a total of 8 dimensional arm joint angles. Values of joint angles were mapped to values ranging from 0.0 to 1.0

through an iterative inverse computation (Regression: Fig. 2b). As an illustrative example, on the topics of imitation learning (Ito and Tani 2004), a humanoid robot with 8 DOF arms was trained for multiple movement patterns demonstrated by a human experimenter. More specifically, the robot, facing with the experimenter, learned to predict how the positions of the experimenter's both hands change in

time based on visual sensory sequences and also it learned corresponding own arm movements in terms of motor sequences by imitation. It was shown that the robot can successfully follow sudden shifts from a learned pattern to another demonstrated by the experimenter by utilizing the PB regression. In this experiment, the functional role of PB can be associated with that of mirror neurons found in the ventral premotor area of monkeys, in the sense that the same PB activation accounts for both recognizing others' particular movement patterns and generating own corresponding movement patterns (Tani et al. 2004).

3 Self-Organized Functional Hierarchy Through Multiple Timescale

In this section, we describe the self-organized functional hierarchy of the behavior generation system, within which series of reusable primitives are integrated into a variety of goal-directed behavior sequences through sensory-motor experiences. In order to overcome difficulties associated with the explicit hierarchical structure of the local representation model, we introduce a different type of representation for functional hierarchy. The representation we use neither uses separate local modules to represent primitives nor introduces explicit hierarchical structure to manipulate these primitives. Instead of setting up an explicit hierarchy, we attempt to realize the self-organization of a functional hierarchy by means of neural activity with multiple timescales ("hierarchy in time"). This functional hierarchy is made possible through the use of two distinct types of neurons, each with different temporal properties. The first type of neuron is the "fast" unit, whose activity changes quickly over the short term. The second type of neuron is the "slow" unit, whose activity changes over the long term.

The idea that multiple timescales may carry advantages for neural systems in interacting with complex environments is intuitively understandable. Indeed, the importance of multiple timescales in neural systems has been emphasized in a number of earlier studies from various different fields. For example, in a study of reinforcement learning, Sutton's group showed that learning speed can be increased by using not only short-term rewards but also long-term rewards (Precup and Sutton 1997). At the level of behavior, it has been shown that the process of acquiring motor skills develops through multiple timescales (Huys et al. 2004; Newell et al. 2001). Biological observations on motor adaptation, such as saccade adaptation and force field adaptation, likewise suggest that these processes involve distinct subsystems with differing timescales (Kording et al. 2007; Smith et al. 2006). At the level of neural synchrony, meanwhile, it is thought that differing timescales in neural synchrony are involved at different levels of information processing, such as in local and global interactions of brain regions (Honey et al. 2007; Varela et al. 2001). These previous studies strongly suggest the possibility that multiple timescales may be essential for the emergence of functional hierarchy in neural systems.

At the neuron level, the use of timescale variation has also been proposed as a means of representing different levels of functionality. In a study of auditory perception, for example, Poeppel et al. (2008) hypothesized that different temporal integration windows in neural activities correspond to a perceptual hierarchy between formant transition level and syllable level. In a study of an evolutional neural network model using a mobile robot, Nolfi (2002) showed that a model with differing temporal integration windows is superior to the normal model in cases in which the robot is required to achieve two different tasks: collision avoidance, which requires short-term sensory-motor control, and self-localization, which requires long-term sensory integration. Furthermore, Paine and Tani (2005) showed that, using a similar evolutional neural network model with a mobile robot, it was possible to achieve hierarchical functionality of motor primitives (wall avoidance) and execution of a given sequence of primitives (global goals) through a particular constraint on neural connectivity. In this model, one part of the network evolved so as to be responsible for primitives with fast dynamics, whereas another part of the network evolved so as to be responsible for sequences of primitives with slower dynamics.

Based on the idea of the functional hierarchy through multiple timescale, we proposed a continuous time RNN model referred to as a "multiple timescale RNN (MTRNN)" (Arie et al. 2009; Namikawa et al. 2011; Nishimoto and Tani 2009; Yamashita and Tani 2008), in which a network is made up of two different types of context units, each type with its own distinct time constant: large τ (slow context) and small τ (fast context). Continuous time characteristic of the MTRNN is described by the following differential equation, which uses a parameter τ referred to as the time constant:

$$\tau_i \dot{u}_{i,t} = -u_{i,t} + \sum_j w_{ij} a_{j,t} \tag{1}$$

where $u_{i,t}$ is the membrane potential, $a_{i,t}$ is the neural state of the ith unit, and w_{ij} is synaptic weight from the jth unit to the ith unit. The second term of Eq. (1) corresponds to synaptic inputs to the ith unit. The time constant τ is defined as the decay rate of the unit's membrane potential, analogous to the leak current of membrane potential in real neurons. One might consider this decay rate to correspond to an integrating time window of the neuron, in the sense that the decay rate indicates the degree to which the earlier history of synaptic inputs affects the current state. When the τ value of a unit is large, the activation of the unit changes slowly, because the internal state potential is strongly affected by the history of the unit's potential. On the other hand, when the τ value of a unit is small, the effect of the history of the unit's potential is also small, and thus it is possible for activation of the unit to change quickly.

The lower level network receives current visuo-proprioceptive (VP) state (v_t, p_t) as inputs and generates its prediction outputs (v_{t+1}, p_{t+1}) for the next time step (Fig. 4).

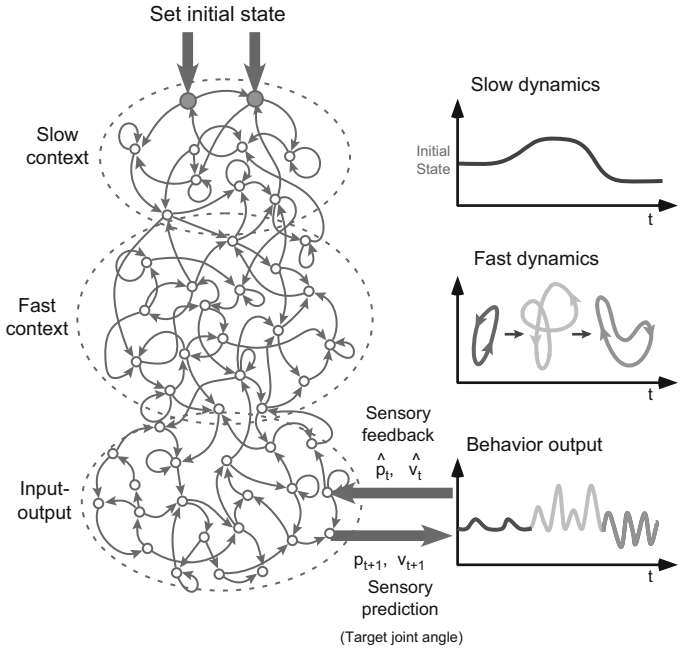

Fig. 4 MTRNN

Here, proprioceptive state means the current body posture in terms of joint positions. The lower level network can generate variety of VP sequences depending on the synaptic weights of the whole network as well as the initial states set in the higher level network. The network is trained to regenerate a set of VP sequences by determining optimal synaptic weights and specific initial states of some of the higher level units corresponding to each of the sequences. The initial states of other neural units are set as neutral, i.e. the internal state of each neuron is set to 0. This means, in other words, that if initial states of the slow context units had not been set, the network would not have been able to produce multiple behavior sequences.

Training of the network is conducted by means of supervised learning (BPTT) using teaching sequences obtained through tutoring by the experimenter. Through the training process, the network can also generate motor imagery of the learned sequences without receiving real sensation for the current VP state but by feeding own prediction to next sensory inputs by closing the loop.

The MTRNN was tested through the interaction of a humanoid robot with a physical environment (Yamashita and Tani 2008). In the experiment, the robot learned to reproduce five different types of learned behavior each of which is composed of sequences of various behavior primitives, including reaching to the object, moving the object up and down, moving the object left and right, moving the object backward and forward, touching the object with one hand and clapping hands. Through training, the robot was able to reproduce perfectly

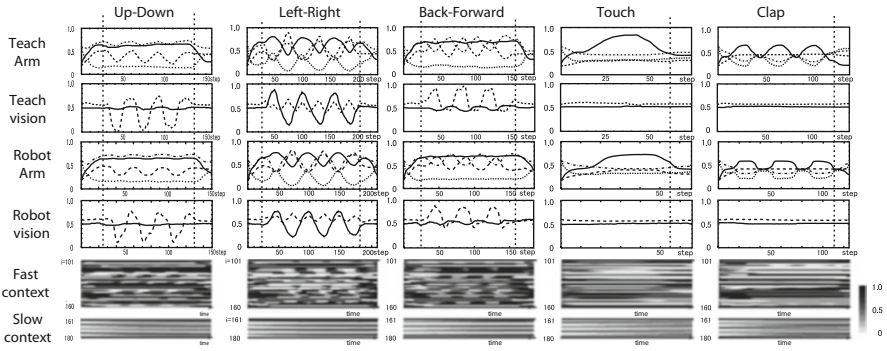

Fig. 5 Examples of behavior sequences generated by the robot after training.
Proprioception (arm), vision sense (vison), fast and slow context activation of teaching signal (teach), and actual sensory feedback in physical environment (robot) during four different patterns of behavior 3 are shown. In proprioception, 4 out of a total of 8 dimensions were plotted (*full line*: left arm pronation, *dashed*: left elbow flexion, *dot-dash-dot-dash*: right shoulder flexion, *dotted*: right arm pronation). In the case of vision, two lines correspond to the relative position of the object (*full line*: X-axis, *dashed line*: Y-axis). Values for proprioception and vision were mapped to the range from 0.0 to 1.0. In the graph of context unit activity, a long sideways *gray-rectangle* indicates the single unit activity over many time steps. Colors of rectangles indicate activation level (cf. color bar). Fast context and Slow context indicates activity of units in the lower level and the higher level of the network

learned behavior sequences generalized across object locations, and also managed to successfully interact with the physical environment. Figure 5 illustrates examples of sensory-motor sequences, as well as examples of teaching signal, activation of the slow units and the fast units, of the trained model network interacting with a physical environment through the body of the robot. In the analysis of the internal network representations for each pattern of behavior, activity of the slow context units exhibited very little location-dependent variation, and no patterns corresponding to repetitive movements were observed (Fig. 6). On the other hand, in the fast context units, trajectories for each behavior exhibited a particular structure which shifted with the object position. These observations suggest that functional hierarchy of primitives and sequence of primitives was self-organized in the model network. That is, in the task behavior sequences, movements that appeared repeatedly (e.g., cyclic patterns) were segmented into reusable "primitives." These primitives were represented in fast context dynamics in a form that was generalized across object locations. On the other hand, the slow context units appeared to be more abstract in nature, representing sequences of primitives in a way that was independent of the object location.

If the fast context units and slow context units encode, respectively, motor primitives and sequences of primitives, one would anticipate that novel combinations of primitives would be generated only by altering the activity of the slow context units. In order to test this idea, the network was trained to additionally generate novel sequences of behavior assembled out of new combinations of primitives. During

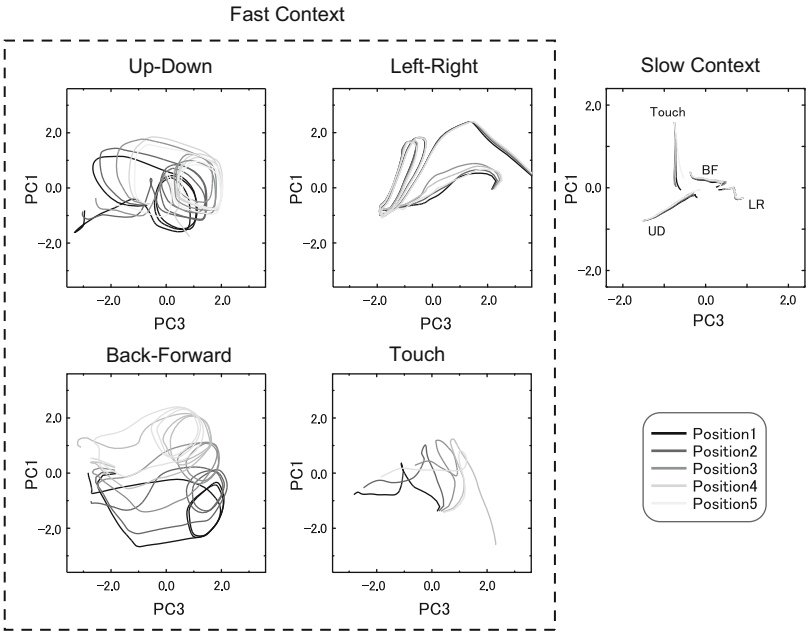

Fig. 6 Position-dependent changes in context state space.
Changes of context activation during each behavior at every position are shown in a 2-dimensional space based on the results of PCA analysis. The four graphs on the left side and single graph on the right side correspond to fast context activities and slow context activities, respectively. State changes of the fast context units for each behavior exhibit a particular structure which shifts with the object position. On the other hand, activity of the slow context units for a particular behavioral task exhibited very little location-dependent variation

the additional training, only connections of the slow context units were allowed to change, weights of the other units remaining fixed at the values that were set through the basic training. As a result of the additional training, the robot was able to reproduce perfectly the novel behavior sequences (Yamashita and Tani 2008).

These results indicate the role of functional differentiation in the current model: motor primitives, such as reaching for the object, moving the object up and down, and moving the object left and right, were represented in the dynamics of fast context units, whereas activities of the slow context units represented switching of these primitives. By changing activities of slow context units, segmented primitives moreover were integrated into new behavior sequences by combining them in different orders.

In other studies using the same model (Nishimoto and Tani 2009), the MTRNN also succeeded in incremental learning of additional primitives, which can be integrated into novel patterns of behavior sequences. Moreover, our study (Arie et al. 2009) also showed that MTRNN can generate action plans for achieving given goals. Action programs in terms of the initial state are searched such that

imagery VP sequences generated from them can reach to the goal states in the distal steps. It was shown that such planning can generate even novel combinations of learned episodic sequences.

4 Discussion and Summary

The current chapter reviewed our attempts that functional hierarchy of behavior generation system could be self-organized in neural dynamics of distributed activity through repeative learning of sensory-motor experiences. First, we introduced RNNPB schema, in which reusable behavior primitives can be represented in distributed neural activity in a generalized manner. In the RNNPB, by manipulating values of PB, behavioral compositionality could be achieved, in the sense that by setting PB values in different order, diverse patterns of behavior sequences could be generated.

We next introduced MTRNN that is an alternative way for realizing the functional hierarchy in behavior generating system. In the MTRNN, through the introduction of multiple timescales, continuous sequences of behavior are segmented into reusable primitives, and the primitives, in turn, are flexibly integrated into novel sequences. It is important to mention that what had been achieved by the MTRNN is not just ordinal compositionality of combining primitives. Elaboration between the slow and the fast dynamics during the learning process achieved quite smooth transitions between one primitive to another rather than just connecting discrete objects of primitives. Moreover, the slow dynamics carries contextual information, for example, counting of cycles of periodic movements. Although this counting of times could be often imprecise with plus-minus one in our experiment, the transitions never take through in the middle of primitives. When combinatorial action sequences can be generated with carrying context, with smooth transitions of primitives and with enough generalization, the appeared structures are not only just compositional but also as fluid and "organic" (Nishimoto and Tani 2009; Tani et al. 2008b).

4.1 Multiple Scales in Space and Time: General Mechanisms for Hierarchy

At the conceptual level, it is intuitively understandable that forms of hierarchy can be realized through differing scales in space and time. In a photo image, for example, elemental information in a narrow space, such as the edges of an object and the color of pixels, is integrated into complex features of the image in a larger space. In speech sounds, syllable-level information on short timescale is integrated into word-level information over a longer timescale. It is not unrealistic to think that

the mechanisms of multiple scales in space and time, which are responsible for generating these hierarchies, might also be at work in the neural systems of animals.

Information processing in the visual cortex, investigated extensively in the study of visual perception, is thought to occur on multiple spatial scales (Hubener et al. 1997; Tootell et al. 1981; Vuilleumier et al. 2003). It is as such considered that functional hierarchy in visual information processing operates on the basis of the spatial structure of visual cortices, such as connections between local modules at a narrow spatial scale, and connections between brain regions at a wider spatial scale. This observation of functional hierarchy based on spatial hierarchy leads naturally to the idea of the local representation model.

On the other hand, there also exists a hypothesis claiming that hierarchical functional differentiation is caused by different timescales of neural activities, specifically, difference in the temporal integration window of neural activities. Based on the observation that speech perception requires multi-time resolution at the formant transition level (20–50 ms) and at the syllable level (200–300 ms), Poeppel et al. (2008) hypothesized that different temporal integration windows in neural activities correspond to a perceptual hierarchy between formant transition level and syllable level. In a neuroimaging study using auditory stimuli, Poeppel and his colleagues found that different brain regions responded in a way which corresponded to differences in the temporal properties of stimuli: one stimulus required precise temporal resolution and activated one particular brain region, while the other stimulus modulated the sound stimulus slowly and activated a different region (Boemio et al. 2005).

It is also intuitively understandable that spatial scales of neural connectivity and timescales of neural activity work in concert with each other. Certain biological observations suggest that multiple scales in space and time in neural systems play an important role in giving rise to functional differentiation. For example, visual cortices of primates, considered to be organized according to a spatial hierarchy, also exhibit functional differentiation that is based on the timescales of neural activity. Many neurons in area V4, which is considered to process wavelength domains, fire in a sustained fashion (possibly to integrate longer timescale information); firing patterns in area MT/V5, on the other hand, which is considered to process visual motion perception, are phasic and brief in duration (possibly in order to achieve precise time resolution) (Schiller and Logothetis 1990).

There also exist studies emphasizing the relationship between spatial organization (neural connectivity) and the presence of multiple timescales in neural activity. For example, Honey et al. (2007) showed that, in simulations of a neural network that captured interregional connections of the macaque neocortex, neurons spontaneously synchronized at multiple timescales corresponding to local and global interactions in regions of the brain. This study can be considered to have shown that multiple timescales can emerge in neural activity through constraints on connectivity. As mentioned earlier, Paine and Tani (2005) showed that a particular constraint on connections encouraged the emergence in neural activity of functional hierarchy with multiple timescales. These facts strongly suggest that the spatial connections between neurons and the timescales of neural activity are strongly

related to each other, and that both act as essential mechanisms leading to functional hierarchy in neural systems.

4.2 Current Problem and Future Challenge

From the perspective of biological plausibility, one may be concerned about the use of the BPTT algorithm. However, in the current chapter, BPTT was used not for mimicking the learning process of biological neural systems, but rather as a general learning rule. Results obtained reflect characteristic features of the proposed network architecture, and not of the learning algorithm. Similar results could be obtained using a different learning algorithm, such as the biologically plausible algorithm, a kind of reinforcement learning (Seung 2003; Xie and Seung 2004).

Among dynamical systems models, the use of reservoir computing (e.g. echo state network, Jaeger and Haas 2004 and liquid state machine, Maass et al. 2002) has become popular in studying the mechanisms of complex sequential behavior. In reservoir computing, complex temporal patterns of sequences can be acquired based on the simple associations between complex dynamics of reservoir network and readouts. The main advantage of reservoir computing is that the training is performed only at the readout stage with a simple learning algorithm. However, it is nonetheless still difficult for this type of model to reproduce hierarchical organization of sequential behavior in robots starting at the level of sensory motor interactions.

Although we have emphasized advantages of distributed representation schema, there are a number of possible advantages to the local representation. First, the learning of one module would seem not to affect other modules. Second, based on this independence in the learning process, it would seem that increasing the number of local modules would lead to an increase in the number of acquirable primitives. An earlier study using multiple sensory-motor sequences, however, demonstrated that difficult problems arise in the local representation model as a result of its local nature (Tani et al. 2008a). Similarities in learned sensory-motor sequences create competition in the learning process between corresponding modules. Generalization requires similar patterns to be represented in the same module as the same primitive, even subtle differences exist in the treatment of sets between such patterns. On the other hand, for the purposes of achieving "crisp" segmentation of sensory-motor flow, different patterns must be represented as separate primitives in distinct modules. This conflict between generalization and segmentation poses serious problems in the treatment of sets of multiple sensory-motor sequences within which there are similarities and overlaps. On the other hand, distributed representation schema also have difficulties in cumulative learning as a result of their distributed nature, namely temporal interference and spatial interference.

To overcome such disadvantages of these types of models could be the next challenge in the study of hierarchical actions for cumulative learning. The following

directions may be possible. For example, introduction of "topological structure" among local modules may increase generalization capability of the local representation model. Tokunaga and Furukawa (2009) proposed a modular network self-organizing map (mnSOM), in which an array of multi-layer perceptrons is used as local modules, instead of the vector units of the conventional SOM. Through the introduction of this topological structure, in the mnSOM, function space can be self-organized with generalization capability (Tokunaga and Furukawa 2009). On the other hand, the introduction of sparseness of connectivity would lead to certain localization in the distributed representation model, which may reduce the interference of learning, and may result in an increase in the capability for cumulative learning.

References

Arbib, M., Erdi, P., Szentagothai, J. (1998). *Neural organization: structure, function, and dynamics.* Cambridge: MIT.

Arie, H., Endo, T., Arakaki, T., Sugano, S., Tani, J. (2009). Creating novel goal-directed actions at criticality: a neuro-robotic experiment. *New Mathematics and Natural Computation, 5*, 307–334.

Baldassarre, G. (2002). A modular neural-network model of the basal ganglia's role in learning and selecting motor behaviours. *Cognitive Systems Research, 3*(1), 5–13.

Boemio, A., Fromm, S., Braun, A., Poeppel, D. (2005). Hierarchical and asymmetric temporal sensitivity in human auditory cortices. *Nature Neuroscience, 8*, 389–395.

Botvinick, M. (2008). Hierarchical models of behavior and prefrontal function. *Trends in Cognitive Sciences, 12*, 201–208.

Evans, G. (1981). chapter Semantic Theory and Tacit Knowledge. In *Wittgenstein: to follow a rule* (pp. 118–137). London: Routledge and Kegan Paul.

Felleman, D., & Van Essen, D. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex, 1*, 1–47.

Fuster, J. (2001). The prefrontal cortex–an update: time is of the essence. *Neuron, 30*, 319–333.

Giszter, S., Mussa-Ivaldi, F., Bizzi, E. (1993). Convergent force fields organized in the frog's spinal cord. *Journal of Neuroscience, 13*, 467–491.

Graziano, M., Taylor, C., Moore, T., Cooke, D. (2002). The cortical control of movement revisited. *Neuron, 36*, 349–362.

Haruno, M., Wolpert, D., Kawato, M. (2001). Mosaic model for sensorimotor learning and control. *Neural Computation, 13*, 2201–2220.

Hilgetag, C., O'Neill, M., Young, M. (2000). Hierarchical organization of macaque and cat cortical sensory systems explored with a novel network processor. *Philosophical Transactions of the Royal Society of London B, 355*, 71–89.

Honey, C., Kotter, R., Breakspear, M., Sporns, O. (2007). Network structure of cerebral cortex shapes functional connectivity on multiple time scales. *Proceedings of the National Academy of Sciences USA, 104*, 10240–10245.

Hubener, M., Shoham, D., Grinvald, A., Bonhoeffer, T. (1997). Spatial relationships among three columnar systems in cat area 17. *Journal of Neuroscience, 17*, 9270–9284.

Huys, R., Daffertshofer, A., Beek, P. (2004). Multiple time scales and multiform dynamics in learning to juggle. *Motor Control, 8*, 188–212.

Ito, M., & Tani, J. (2004). On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system. *Adaptive Behavior, 12*, 93–115.

Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science, 304*, 78–80.

Kording, K., Tenenbaum, J., Shadmehr, R. (2007). The dynamics of memory as a consequence of optimal adaptation to a changing body. *Nature Neuroscience, 10*, 779–786.

Kuniyoshi, Y., & Sangawa, S. (2006). Early motor development from partially ordered neural-body dynamics – experiments with a cortico-spinal-musculo-skeletal model. *Biological Cybernetics, 95*, 589–605.

Maass, W., Natschlaeger, T., Markram, H. (2002). Real-time computing without stable states: s new framework for neural computation based on perturbations. *Neural Computation, 14*, 2531–2560.

Mussa-Ivaldi, F., & Bizzi, E. (2000). Motor learning through the combination of primitives. *Philosophical Transactions of the Royal Society of London B, 355*, 1755–1769.

Namikawa, J., Nishimoto, R., Tani, J. (2011). A neurodynamic account of spontaneous behaviour. *PLoS Computational Biology, 7*, e1002221.

Newell, K., Liu, Y., Mayer-Kress, G. (2001). Time scales in motor learning and development. *Psychological Review, 108*, 57–82.

Nishimoto, R., & Tani, J. (2009). Development of hierarchical structures for actions and motor imagery: a constructivist view from synthetic neuro-robotics study. *Psychological Research, 73*, 545–558.

Nolfi, S. (2002). Evolving robots able to self-localize in the environment: the importance of viewing cognition as the result of processes occurring at different time scales. *Connection Science, 14*, 231–244.

Paine, R., & Tani, J. (2005). How hierarchical control self-organizes in artificial adaptive systems. *Adaptive Behavior, 13*, 211–225.

Poeppel, D., Idsardi, W., van Wassenhove, V. (2008). Speech perception at the interface of neurobiology and linguistics. *Philosophical Transactions of the Royal Society of London B, Biology Science, 363*, 1071–1086.

Precup, D., & Sutton, R. (1997). Multi-time models for temporally abstract planning. In *Advances in neural information processing systems* (vol. 10, pp. 1050–1056). Cambridge: MIT.

Sakai, K., Kitaguchi, K., Hikosaka, O. (2003). Chunking during human visuomotor sequence learning. *Experimental Brain Research, 52*, 229–242.

Schiller, P., & Logothetis, N. (1990). The color-opponent and broad-band channels of the primate visual system. *Trends Neuroscience, 13*, 392–398.

Seung, H. (2003). Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron, 40*, 1063–1073.

Smith, M., Ghazizadeh, A., Shadmehr, R. (2006). Interacting adaptive processes with different timescales underlie short-term motor learning. *PLoS Biology, 4*, e179.

Sugita, Y., & Tani, J. (2004). Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. *Adaptive Behavior, 13*, 33–52.

Tani, J. (2003). Learning to generate articulated behavior through the bottom-up and the top-down interaction processes. *Neural Networks, 16*, 11–23.

Tani, J., & Ito, M. (2003). Self-organization of behavioral primitives as multiple attractor dynamics: a robot experiment. *IEEE Transactions on Systems, Man, and Cybernetics. Part A – Systems and Humans, 33*, 481–488.

Tani, J., Ito, M., Sugita, Y. (2004). Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using rnnpb. *Neural Networks, 17*, 1273–1289.

Tani, J., Nishimoto, R., Namikawa, J., Ito, M. (2008a). Codevelopmental learning between human and humanoid robot using a dynamic neural-network model. *Systems, Man, and Cybernetics, Part B: Cybernetics, 38*, 43–59.

Tani, J., Nishimoto, R., Paine, R. (2008b). Achieving 'organic compositionality' through self-organization: reviews on brain-inspired robotics experiments. *Neural Networks, 21*, 584–603.

Tani, J., & Nolfi, S. (1999). Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. *Neural Networks, 12*, 1131–1141.

Thoroughman, K., & Shadmehr, R. (2000). Learning of action through adaptive combination of motor primitives. *Science, 407*, 742–747.

Tokunaga, K., & Furukawa, T. (2009). Modular network som. *Neural Networks, 22*, 82–90.

Tootell, R., Silverman, M., De Valois, R. (1981). Spatial frequency columns in primary visual cortex. *Science, 214*, 813–815.

Varela, F., Lachaux, J., Rodriguez, E., Martinerie, J. (2001). The brainweb: phase synchronization and large-scale integration. *Nature Reviews Neuroscience, 2*, 229–239.

Vuilleumier, P., Armony, J., Driver, J., Dolan, R. (2003). Distinct spatial frequency sensitivities for processing faces and emotional expressions. *Nature Neuroscience, 6*, 624–631.

Xie, X., & Seung, H. (2004). Learning in neural networks by reinforcement of irregular spiking. *Physical Review E, 69*, 041909.

Yamashita, Y., & Tani, J. (2008). Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. *PLoS Computational Biology, 4*, e1000220.

Autonomous Representation Learning in a Developing Agent

Jonathan Mugan and Benjamin Kuipers

Abstract Our research goal is to design an agent that can begin with low-level sensors and effectors and autonomously learn high-level representations and actions through interaction with the environment. This chapter focuses on the problem of learning representations. We present four principles for autonomous learning of representations in a developing agent, and we demonstrate how these principles can be embodied in an algorithm. In a simulated environment with realistic physics, we show that an agent can use these principles to autonomously learn useful representations and effective hierarchical actions.

1 Introduction

A developing agent is one who learns continually in an environment so that its capabilities expand over time. Such an agent receives a flood of information from its sensors, and it must abstract its sensor input to create high-level representations that allow it to learn to act effectively. Many algorithms exist for learning actions using a given representation, but there has been less work on learning representations. In this chapter, we present four principles for creating representations and we present an algorithm, called the Qualitative Learner of Action and Perception (QLAP), that embodies those principles.

The first principle is to *exploit the synergy between the created representations and the development of the agent*. The agent's goals should drive the learning of new representations, and new representations should, in turn, expand what the agent can

J. Mugan (✉)
21CT, Austin, TX 78730, USA
e-mail: jmugan@21ct.com

B. Kuipers
Computer Science and Engineering, University of Michigan, Ann Arbor, MI 48109, USA
e-mail: kuipers@umich.edu

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_4,

learn. The developmental psychologist Jean Piaget described learning in humans as a process of assimilation and accommodation (Piaget 1952). For Piaget, assimilation is the process of fitting new information into what the learner already knows. This information may have to be warped to fit into the learner's current knowledge, and the more the agent does this warping, the less the learned knowledge fits with reality. The agent must therefore also engage in accommodation, which he described as the modification of this knowledge to fit new information.

Most machine learning algorithms do not take this developmental learning approach. Instead, they first create a representation and then have the agent learn using that representation. One example of this approach is clustering, for example of time-series data (Cohen et al. 2002). Clustering is an important abstraction tool when the statistics of the sensory input are such that clustering can find meaningful states for the robot. However, clustering can miss subtle but important distinctions that may be small in the statistics of the data, but large with respect to the goals of the robot. For example, consider a Skinner box (Skinner 1961). To a clustering algorithm, all of the small details of the box may be equally salient, but the important feature indicating the coming electrical shock may be a small red light (a similar point was made by Goodman et al. 2007).

The developmental approach embodied in this first principle is to allow the agent's current developmental goals to drive the learning of new representations. This creates a synergy between representations and learning because new representations are learned to better help the agent achieve its current goals, and these new representations open up further possibilities for learning. An implementation of this principle was provided by Mugan and Kuipers (2007). Additional examples of simultaneously learning a structure and a discretization where each constrains the other are provided by Friedman and Goldszmidt (1996) and Goodman et al. (2007).

The second principle is to *generate representations to stand for phenomena in the environment*. A phenomenon can be described as something, such as an object or event, that an agent can identify in the environment. Once an agent can identify a phenomenon in the environment, it can then track statistics on its occurrence. This stands in contrast to clustering or state-based representations where the goal is to map the environment onto one of a set number of identified states. By contrast, generating representations to stand for phenomena is concerned with identifying the existence or nonexistence of particular phenomena and then tracking statistics.[1] This helps to manage the flood of information because the agent can process the input through the intermediary step of the representations.

One example is learning what a chair is. Once an agent has created a separate representation to stand for a chair, it can identify when there are chairs in the environment and track statistics on their occurrence and location. The agent can

[1]Of course, for any fixed number of phenomena and any fixed number of statements about the phenomena, this is equivalent to a state-based representation. The differences are that the number of phenomena will change over time, and phenomena are considered largely independent of other phenomena.

then generate hypotheses based on these phenomena to create self-supervised learning problems and watch the environment for evidence of these hypotheses. An implementation of this principle was provided by Drescher (1991). He proposed a method whereby an agent learns predictive rules of the environment and then tracks statistics on their reliability. This principle is also related to reification, the process of converting sensory input to symbolic information (Gunderson and Gunderson 2008).

The third principle is to *break the environment into many small parts*. The environment can be too big to be represented in its entirety and representing it with one big model or one big plan does not scale well. An alternative approach is to represent the environment using subsets that are well understood and contained. This allows each model or plan to be small, and the agent can learn which to choose in which situation. This idea can be traced back to Piaget (1952). In his theory of child development, children organize knowledge into self-contained units called schemas.

The fourth principle is to *create representations so learning is easy*. Often, researchers focus on creating increasingly sophisticated algorithms to achieve better performance on a given representation. This principle states that an alternative approach is to represent the problem so that simple algorithms can be used.

In this chapter, we present a developmental learning algorithm that embodies these four principles. We call it QLAP. QLAP builds on the groundwork laid by Pierce and Kuipers (1997) and Modayil and Kuipers (2007). This previous work allows an agent to identify objects in its sensory input and to learn a set of orthogonal motor primitives, enabling the agent to sense the world through a set of continuous variables and affect the world using a set of continuous effectors. QLAP begins with this low-level representation of continuous variables, and through autonomous exploration learns a set of hierarchical actions.

QLAP is able to make this transition from low-level perception to high-level actions by learning a sequence of representations that build on each other. This sequence of representations includes a discretization of the environment that allows QLAP to build predictive models. These models are used both to refine the discretization and to build plans to perform actions. This chapter focuses on how QLAP creates representations. For a full technical description of QLAP, see Mugan (2010); Mugan and Kuipers (2012).[2]

2 Landmarks Discretize Continuous Variables and Define Events

QLAP bridges the gap between a continuous and a discrete representation by creating landmarks. A landmark allows the agent to represent the value of a continuous variable $\tilde{v}(t)$ as being less than, greater than, or equal to that landmark.

[2]A video describing QLAP can be seen at http://www.youtube.com/watch?v=xJ0g-NoerZ0.

In this way, landmarks discretize the input so that the continuous variable $\tilde{v}(t)$ can be converted to the qualitative variable $v(t)$. A qualitative variable v with two landmarks (v_1^*, v_2^*) would have five distinct qualitative values, $\mathcal{Q}(v) = \{(-\infty, v_1^*), v_1^*, (v_1^*, v_2^*), v_2^*, (v_2^*, +\infty)\}$. This type of representation is called a qualitative representation (Kuipers 1994).

Creating a landmark also allows the agent to create a second type of representation called an *event*. We say that an event takes place when a qualitative variable changes qualitative value. More specifically, if q is a qualitative value of a qualitative variable v, meaning $q \in \mathcal{Q}(v)$, then the event $v_t \rightarrow q$ is defined by $v(t-1) \neq q$ and $v(t) = q$.

Initially, the agent has no landmarks for each magnitude input variable $\tilde{v}(t)$. But for each such variable, QLAP creates a direction of change variable $\dot{v}$ that has a single intrinsic landmark at 0, so its quantity space is $\mathcal{Q}(\dot{v}) = \{(-\infty, 0), 0, (0, +\infty)\}$, which can be abbreviated as $\mathcal{Q}(\dot{v}) = \{[-], [0], [+]\}$. Motor variables are also given an initial landmark at 0. The agent must learn new landmarks for magnitude and motor variables. Each additional landmark allows the agent to perceive or affect the world at a finer level of granularity.

A qualitative representation affords two important advantages. The first is generalization. In a qualitative representation, many different real values map to the same qualitative value. For example, if a noise occurs each time a variable value is less than some landmark, then the agent can learn the association between the variable value and the noise because each instance maps to the same state. Without a qualitative representation, each instance would appear unique. This is an example of creating a representation so learning is easy.

The second advantage of a qualitative representation is that the learner can focus on important events. As the fire hose of sensory input comes in, the agent needs some way to know what to pay attention to and what to track statistics on. The goal of a QLAP agent is to predict and control the environment. We will see in Sect. 3.2 that QLAP can identify landmarks that are useful for making predictions. By defining these landmarks, the agent can focus on important events. This is an example of generating representations to stand for phenomena in the environment.

3 Models Use One Event to Predict Another

Identifying events allows the agent to learn when one event leads to another. Such a pairing of events is called a *contingency*. For example, flicking a light switch leads to the light coming on. It has been proposed that humans have an innate contingency detection module (Gergely and Watson 1999), and it has been shown that human infants can detect contingencies in their environment shortly after birth (DeCasper and Carstens 1981).

Contingencies have two important advantages. First, they are easy to learn since the agent needs only to look for pairs of events. Second, they are a natural representation for planning. The models in QLAP are based on contingencies.

Mathematically, these models are represented as dynamic Bayesian networks (Dean and Kanazawa 1989).

3.1 Creating Models

QLAP searches for contingencies by examining pairs of events. Once a contingency is found, a new representation is created in the form of a model based on that contingency. A model is created if a contingency is found linking the occurrence of an *antecedent event* E_1 with the occurrence of a *consequent event* E_2. Specifically, a model is created if observing the antecedent event E_1 makes it more likely that the consequent event E_2 will soon be observed than otherwise:

$$\Pr(\text{soon}(t, E_2)|E_1(t)) > \Pr(\text{soon}(t, E_2)) \tag{1}$$

where $\Pr(\text{soon}(t, E_2))$ is the probability of event E_2 occurring within a window of k time steps beginning at a random time t. The statement $\Pr(\text{soon}(t, E_2)|E_1(t))$ is the conditional probability of event E_2 occurring within a window of k time steps beginning at time t, given that event E_1 has occurred at time t. The time window size k is learned based on the environment.

Creating models based on contingencies differs in an interesting way from other methods that create a single model for each variable (Hester and Stone 2009) or create a model for each variable and action combination (Strehl et al. 2007). QLAP tracks statistics of almost all possible pairs of events in a large matrix and retrieves all pairs of events that may form a contingency. Because QLAP is considering large numbers of possible contingencies, QLAP potentially learns multiple models to predict each event on each variable, and the models can be small and be applicable in different situations. For example, the agent might also learn that the sound of clapping hands leads to the light going on. Then, instead of one model that would have to account for observing a flicking of a switch or hearing a clapping noise, QLAP would have two small models. Also note that the flicking of the switch or the clapping would not have to be given as actions in QLAP, the agent could just observe that these events lead to the light going on. QLAP could then learn that anything that brought these events about could be used to turn the light on.

3.2 Improving Models

Once a new representation is created in the form of a model, the agent can track statistics on when the model successfully predicts the environment and when it does not, resulting in self-supervised learning. Since each model uses one event E_1 to predict another event E_2, when the antecedent event E_1 occurs, the agent can watch the environment to determine under which conditions the consequent event E_2 will occur.

As was shown by Drescher (1991), self-supervised learning allows the agent to learn context variables to make each model more reliable by enabling QLAP to note correlations between values of qualitative variables and the reliability of the model. For example, by tracking statistics, the agent could learn that flicking the switch only leads to the light going on in rooms with white walls. It could then add room color as a context variable in the model. QLAP adds context variables one at a time as they are found to improve the model.

In QLAP, self-supervised learning also allows the agent to learn new landmarks by enabling the search for regions of real values that are correlated with the reliability of the model. For the light switch example, the agent might observe that the light only came on when the temperature v was below $v = 100$ degrees. QLAP could learn this by noting the value of v each time the switch was flicked. It could then observe that the light went on when $v = 99$, $v = 42$, and $v = 57$; and it could observe that the light did not go on when $v = 101$, $v = 119$, and $v = 109$. This would allow QLAP to improve the model by adding a landmark on v at $v = 100$ and adding v to the context. Since there is often a lot of noise in the environment, we use the information-theoretic method of Fayyad and Irani (1992) to find the best landmark value. New landmarks lead to new events, and new events can lead to new models. This leads to a synergy between models and landmarks.

The goal of improving each model is to identify a situation (a value in the Cartesian product of the context variables) where the antecedent event E_1 almost always leads to the consequent event E_2. Specifically, when a situation is found where the probability of event E_1 leading to event E_2 is greater than 0.75, the model is said to be *sufficiently reliable*. Sufficiently reliable models are converted into plans to bring about their consequent events, as described in the next section.

4 Plans are Different Ways to Perform Actions

Actions are how the QLAP agent brings about changes in the world. QLAP creates an action $a(v, q)$ for each combination of qualitative variable v and qualitative value $q \in \mathcal{Q}(v)$. When it performs an action $a(v, q)$, the agent strives to set $v = q$ in the environment. QLAP performs actions using plans. As mentioned in the last section, a plan is added to an action $a(v, q)$ when QLAP learns a sufficiently reliable model to predict the event $v{\rightarrow}q$.

Since there may be many models that predict the same event, there may be many plans to perform the action to bring about that event. This is good, because it keeps the plans small and the agent can learn which plan to choose in which situation. And according to the principle of breaking up the world into pieces, we want each plan to be like a Piagetian schema so the plan can be small and self-contained.

4.1 Plans in QLAP are Inspired by Schemas

How can an agent convert an observed model based on a contingency into a schema that can be used as a plan to bring about changes in the world? There has been much research into how humans consolidate knowledge. Within developmental psychology, Mandler (2004a) proposed a theory of *perceptual meaning analysis*, which is an experimentally grounded theory that explains how infants can learn concepts and how these concepts can be represented. She describes perceptual meaning analysis as "the central attentive process that redescribes attended perceptual information into a simpler and conceptual (accessible) form" (Mandler 2004b). In Mandler's theory, concepts are represented with *image schemas*. Mandler (2004a) says that "the image-schemas that perceptual meaning analysis creates are analog representations that summarize spatial relations and movements in space" (p. 79). The notion of image schemas and the idea of using them as a foundation for understanding was advanced by Lakoff and Johnson (1980). Johnson (1987) writes that an image schema consists of a set of components that are related by a definite structure.

While Mandler (2004a) focuses mostly on perception in her discussion of image schemas, Arbib (1992) has suggested that schemas can provide a type of "action-oriented memory." In QLAP, action-oriented memory is represented with a Markov Decision Process (MDP).

4.2 Plans in QLAP are Implemented as MDPs

In QLAP, each plan is implemented as an MDP. MDPs are a framework for understanding decision making over time (Puterman 1994). An MDP is a four-tuple of the form $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, T, R \rangle$ where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $T(s, a, s') = P(s'|s, a)$ is the transition function over states and actions, and $R(s, a, s')$ is a reward function.

MDPs are often used as the structure for decision making for learning agents (Sutton et al. 1999; Vigorito and Barto 2010). An MDP breaks the environment up into states: it represents what is in the environment, and it represents everything that can be perceived about the environment. Additionally, an MDP represents how the environment can change, including how the agent's actions affect the environment. And it also includes what is "good" for the agent in the reward function.

MDPs are not analog representations and are missing the "image" part of the image schemas, but MDPs do encapsulate knowledge about the world in a principled framework, and QLAP takes advantage of MDPs to represent how the agent can interact with the world. QLAP assumes that the world is too large to be represented with a single MDP and therefore QLAP does not assume a single, underlying MDP. Instead, QLAP creates many small MDPs, where each MDP represents some small aspect of the environment. These MDPs are not assumed to objectively exist in the

world—like the discretization, they are created by the agent. For example, to grab a block, a QLAP agent might learn an MDP that encodes that its hand must be centered over the block at a certain height. If its hand is to the left of the block, the MDP specifies that the agent must move its hand to the right. The only part of the world that is relevant is the location of the hand relative to the block.

4.3 Converting Models to Plans

When a model r is converted into a plan, the plan reflects the structure of the model. The model is first converted into an MDP $\mathcal{M}_r = \langle \mathcal{S}_r, \mathcal{A}_r, T_r, R_r \rangle$ where the state space $\mathcal{S}_r$ of the MDP comes from the variables identified as important in the model. These variables are those in the antecedent event, the consequent event, and the context. The action space $\mathcal{A}_r$ is the set of QLAP actions to reach values in the state space $\mathcal{S}_r$, the transition function T_r is computed using observed statistics, and the reward function R_r gives a reward for reaching the goal state of achieving the consequent event on r.

For each MDP $\mathcal{M}_r$, QLAP learns a policy π_r that serves as the plan. The policy specifies which action the agent should choose for each state $s \in \mathcal{S}_r$. Because QLAP knows the transition function T_r and the reward function R_r for the MDP $\mathcal{M}_r$, and because each MDP consists of only a small subset of the possible variables, QLAP can learn the policy using dynamic programming (Sutton and Barto 1998).

Arbib (1992) has described that schemas can be put together into coordinated control programs that consist of recursively defined schemas. In QLAP, the set of actions and plans becomes a linked network and results in a set of hierarchical actions. Plans are linked to actions because plans are ways to perform actions. When an action is to be performed, QLAP first chooses the best plan for the situation. The interesting thing is that the actions in each plan are QLAP actions. This is what creates the hierarchy. It is also worth noting that once a plan is created, QLAP can treat it similarly to how models are treated and learn statistics on when it will be reliable. These statistics are used by the action to pick a plan in the current state, and they are also used to add additional state variables in the same way that context variables are added to models.

A full summary of the learned representations can be seen in Fig. 1.

5 Exploration

A QLAP agent is not given any external task; it is driven by a desire to control its environment, and it explores the world autonomously based on that drive. As the agent explores, it is continually learning new landmarks, events, models, plans, and actions. Initially, the agent motor babbles by picking random values for the motor variables. Once the agent begins to learn plans for actions, it explores by continually

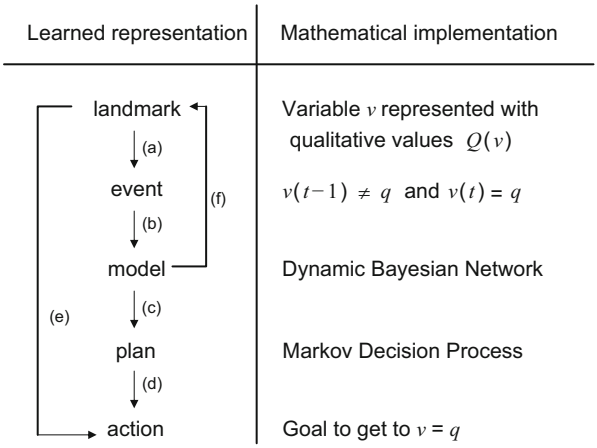

Fig. 1 Progression of learned representations created by QLAP. (**a**) QLAP creates *landmarks* to discretize each continuous input variable into a set of qualitative values. These landmarks lead to *events*, which occur when a variable changes its qualitative value. (**b**) Events allow QLAP to learn *models* that predict when one event will follow another. (**c**) When these models become sufficiently reliable, they can be converted into *plans*. (**d**) Plans are different ways to perform *actions*. (**e**) QLAP creates an action to achieve each qualitative value of each variable. This means that new landmarks also lead to new actions. (**f**) QLAP creates a new landmark when it estimates that adding that landmark would make a model more reliable

choosing actions to practice. As the agent develops, it will have a growing set of actions that it can perform. How can it best choose which actions to practice?

Ideally, the agent should choose actions where it can learn the most, that is, actions that are neither too hard nor too easy. Schmidhuber (1991) proposed a method whereby an agent learns to predict the decrease in the error of the model that results from taking each action. The agent can then choose the action that will cause the biggest decrease in prediction error. Oudeyer et al. (2007) apply this approach with a developing agent and have the agent explore regions of the sensory-motor space that are expected to produce the largest decrease in predictive error. Their method is called Intelligent Adaptive Curiosity (IAC). QLAP uses a modified version of IAC to choose actions that are expected to produce the largest increase in reliability.

6 Evaluation

We want to evaluate how well QLAP can use the learned representations to act in the world. Evaluating autonomous learning is difficult because there is no pre-set task on which to evaluate performance. The approach we take is to first have the

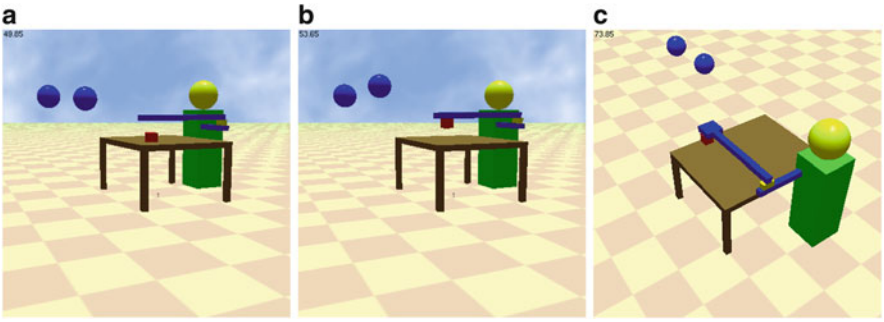

Fig. 2 The evaluation environment (shown here with floating objects) (**a**) Not grasping, (**b**) Grasping, (**c**) Above view

agent learn autonomously in an environment; we then evaluate if the agent is able to perform a set of tasks. It is important to note that during learning the agent does not know on which tasks it will be evaluated.

6.1 Evaluation Environment

The evaluation environment is implemented in Breve (Klein 2003) and has realistic physics. Breve simulates physics using the Open Dynamics Engine (ODE) (Smith 2004). The simulation consists of a robot at a table with a block and floating objects. The robot has an orthogonal arm that can move in the x, y, and z directions. The environment is shown in Fig. 2, and the variables perceived by the agent for the core environment are shown in Table 1. The block has a width that varies between 1 and 3 units. The block is replaced when it is out of reach and not moving, or when it hits the floor. See Mugan (2010) for further details.

The robot can grasp the block in a way that is reminiscent of both the palmar reflex (Payne and Isaacs 2007) and having a sticky mitten (Needham et al. 2002). The palmar reflex is a reflex that is present from birth until the age 4–6 months in human babies. The reflex causes the baby to close its hand when something touches the palm. In the sticky mittens experiments, three-month-old infants wore mittens covered with Velcro that allowed them to more easily grasp objects.

Grasping is implemented on the robot to allow it to grasp only when over the block. Specifically, the block is grasped if the hand and block are colliding, and the Euclidean 2D distance from the center of the block in the x and y directions is less than half the width of the palm, $3/2 = 1.5$ units.

In addition to the core environment, QLAP is also evaluated with distractor objects. This is done using the *floating extension environment*, which adds two floating objects that the agent can observe but cannot interact with. The purpose of this environment is to see if the robot can learn in the presence of distractor objects.

Table 1 Variables of the core environment

Variable	Meaning
u_x, u_y, u_z	Force in x, y, and z directions
u_{UG}	Ungrasp the block
h_x, h_y, h_z	Global location of hand in x, y, and z directions
$\dot{h}_x, \dot{h}_y, \dot{h}_z$	Derivative of h_x, h_y, h_z
$y_{\mathrm{TB}}, \dot{y}_{\mathrm{TB}}$	Top of hand in frame of reference of bottom of block (y direction)
$y_{\mathrm{BT}}, \dot{y}_{\mathrm{BT}}$	Bottom of hand in frame of reference of top of block (y direction)
$x_{\mathrm{RL}}, \dot{x}_{\mathrm{RL}}$	Right side of hand in frame of reference of left side of block (x direction)
$x_{\mathrm{LR}}, \dot{x}_{\mathrm{LR}}$	Left side of hand in frame of reference of right side of block (x direction)
$z_{\mathrm{BT}}, \dot{z}_{\mathrm{BT}}$	Bottom side of hand in frame of reference of top of block (z direction)
$z_{\mathrm{F}}, \dot{z}_{\mathrm{F}}$	Distance from the bottom of the block to the floor
$T_{\mathrm{L}}, \dot{T}_{\mathrm{L}}$	Location of nearest edge of block in x direction in coordinate frame defined by left edge of table
$T_{\mathrm{R}}, \dot{T}_{\mathrm{R}}$	Location of nearest edge of block in x direction in coordinate frame defined by right edge of table
$T_{\mathrm{T}}, \dot{T}_{\mathrm{T}}$	Location of nearest edge of block in y direction in coordinate frame defined by top edge of table
$c_x, \dot{c}_x$	Location of hand in x direction relative to center of block
$c_y, \dot{c}_y$	Location of hand in y direction relative to center of block
T	Block is grasped, true or false. Becomes true when the hand is touching the block and the 2D distance between the center of the hand and the center of the block is less than 1.5.
bang	Is true when block hits the floor

Table 2 Variables added to the core environment to make up the floating extension environment

Variable	Meaning
f_x^1, f_y^1, f_z^1	Location of first floating object in x, y, and z directions
$\dot{f}_x^1, \dot{f}_y^1, \dot{f}_z^1$	Derivative of f_x^1, f_y^1, f_z^1
f_x^2, f_y^2, f_z^2	Location of second floating object in x, y, and z directions
$\dot{f}_x^2, \dot{f}_y^2, \dot{f}_z^2$	Derivative of f_x^2, f_y^2, f_z^2

In addition, the floating extension environment addresses scaling by showing that most of QLAP is entirely insensitive to the number of variables that are not causally connected (i.e., connected by some chain of contingencies) to events of interest. In the floating extension environment, the objects float around in an invisible box. The variables added to the core environment to make the floating extension environment are shown in Table 2.

6.2 Experimental Conditions

We compare the performance of QLAP with the performance of a supervised learner on a set of tasks. The supervised learner is supervised in the sense that it is trained only on the evaluation tasks. Not knowing the tasks on which it will be evaluated puts the QLAP learner at a disadvantage on the evaluation tasks because QLAP explores broadly and learns more than the evaluation tasks. The evaluation will determine if QLAP can demonstrate developmental learning by getting better at the tasks over time, and if QLAP can do as well as the supervised learner.

Each agent is evaluated on three tasks. These are referred to as the *core tasks*.

Move the block. The evaluator picks a goal to move the block *left* ($\dot{T}_{\mathrm{L}} = [+]$), *right* ($\dot{T}_{\mathrm{R}} = [-]$), or *forward* ($\dot{T}_{\mathrm{T}} = [+]$). The goal is chosen randomly based on the relative position of the hand and the block. A trial is terminated early if the agent hits the block in the wrong direction.

Hit the block to the floor. The goal is to make *bang* = true.

Pick up the block. The goal is to get the hand in just the right place so the robot can grasp the block and make T = true. A trial is terminated early if the agent hits the block out of reach.

The supervised learner is trained using linear, gradient-descent Sarsa(λ) with binary features (Sutton and Barto 1998) where the binary features come from tile coding. Tile coding is a way to discretize continuous input for reinforcement learning. Both the QLAP and the tile-coding agents are evaluated on the core tasks in both the core environment and the floating extension environment under three experimental conditions.

QLAP The QLAP algorithm.
Tile-1 Tile coding choosing an action every time step.
Tile-10 Tile coding choosing an action every 10 time steps.

Tile-1 and Tile-10 are both used because Tile-1 has difficulty learning the core tasks due to high task diameter (the number of time steps needed to complete the task).

QLAP learns autonomously for 250,000 time steps (about 3.5 h of physical experience) as described in Sect. 5. The tile coding agents repeatedly perform trials of a particular core task for 250,000 time steps. At the beginning of each trial, the core task that the tile coding agent will practice is chosen randomly. The state of the agent is saved every 10,000 time steps (about every 8 min of physical experience). The agent is then evaluated on how well it can do the specified task using the representations from each stored state.

At the beginning of each trial, a block is placed in a random location within reach of the agent and the hand is moved to a random location. Then, the goal is given to the agent. The agent makes and executes plans to achieve the goal. If the QLAP agent cannot make a plan to achieve the goal, it moves randomly. The trial is terminated after 300 time steps or if the goal is achieved. The agent receives a penalty of -0.01 for each time step it does not achieve the goal and a reward of 9.99 on the time step it achieves the goal. (Tile-10 gets a penalty of -0.1 every 10th time

step it does not reach the goal and a reward of 9.99 on the time step it reaches the goal.)

Each evaluation consists of 100 trials. The rewards over the 100 trials are averaged, and the average reward is taken as a measure of ability. For each experiment, 20 QLAP agents and 20 tile-coding agents are trained.

6.3 Results

The results are shown in Figs. 3 and 4. Figure 3 compares QLAP and tile coding on the task of moving the block in the specified direction. As can be seen in Fig. 3a, Tile-1 was not able to do the task well compared to QLAP due to the high task diameter. Having the tile-coding agents choose an action every 10 time steps improved their performance, as can be seen in Fig. 3c. But as can be seen by visually inspecting Fig. 3e, the performance of tile coding degrades much more than the performance of QLAP degrades when the distractor objects are added.

Figure 4 shows the performance of QLAP and tile coding on the tasks of hitting the block off the table and picking up the block. For brevity, this figure only contains the final comparison of QLAP and Tile-10 with floating objects on those tasks. We see that QLAP does better than Tile-10 on these more difficult tasks in the environment with floating objects. For all three tasks, QLAP displays developmental learning and gets better over time. It is worth noting that before much learning has taken place, the developmental learning agents running QLAP may not do as well as the reinforcement learning agents. For example, in Fig. 4c, Tile-10 starts out higher than QLAP. This is because a Tile-10 agent can do minimally well on the task by picking a motor value and moving in that direction for 10 time steps. Similarly, in Fig. 4 we see that it takes about 50,000 time steps before the QLAP developmental learning agents gain any proficiency. This result is interesting given the oft-cited observation that humans are born more helpless than other animals (Rosenberg and Trevathan 2002).

6.4 Description of Learned Representations

One of the first contingencies (models) that QLAP learns is that if it gives a positive force to the hand, the hand will move to the right. This contingency is not very reliable because in the simulator it takes a force of at least 300 units to move the hand. The agent learns a landmark at 300 on that motor variable and modifies the model to state that if the force is at least 300, then the hand will move to the right. However, this model still is not completely reliable, because if the hand is already all the way to the right, then it cannot move any farther. But from this model, the agent can note the location of its hand each time it applies a force of 300, and it can learn a

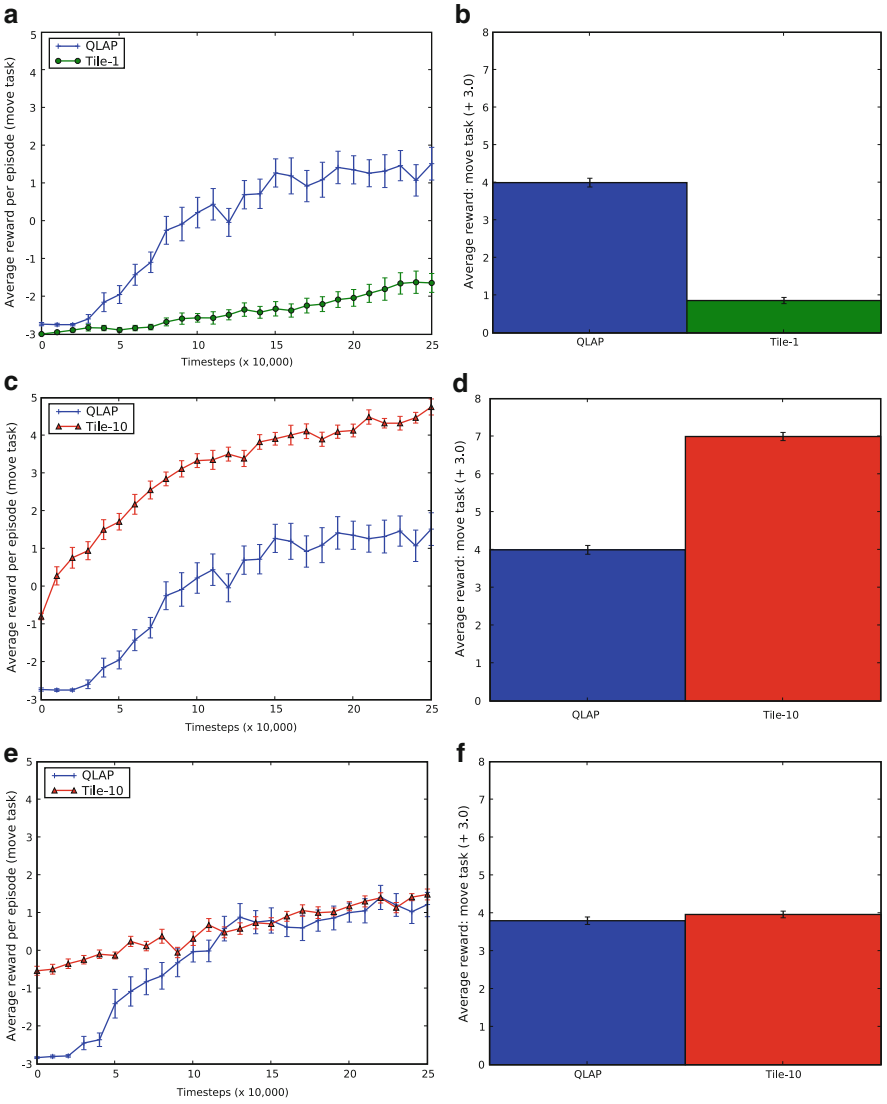

Fig. 3 Moving the block. (**a, b**) QLAP does better than Tile-1 because of the high task diameter. (**c, d**) Tile-10 does better than QLAP. (**e, f**) When the floating objects are added, the performance of Tile-10 degrades much more than the performance of QLAP degrades. (Bar graphs show the average values after 100,000 time steps.)

landmark to indicate when the hand is all the way to the right. The completed model states that if the hand is not all the way to the right and a force of at least 300 is given, then the hand will usually move to the right. Even this model is not completely

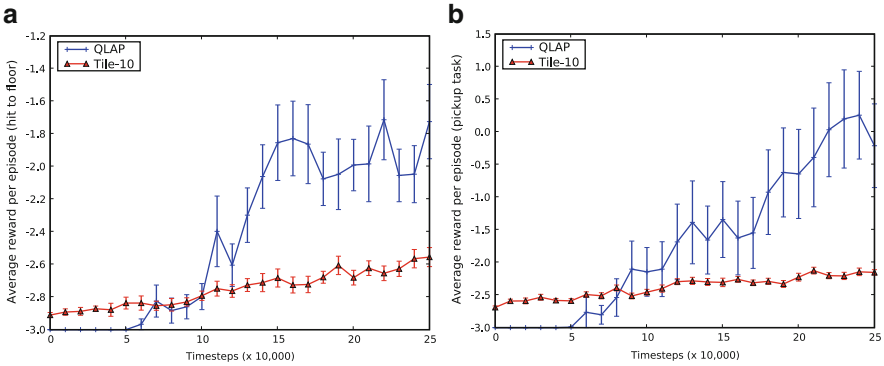

Fig. 4 QLAP outperforms reinforcement learning using tile coding on the more difficult tasks in the floating extension environment. (**a**) Knock the block off the table. (**b**) Pick up the block

reliable, because there are unusual situations where, for example, the hand is stuck on the block. Since the model is probabilistic, it can handle nondeterministic cases.

The agent also learns that the location of the right side of the hand in the frame of reference of the left side of the block has a special value at 0. It learns this because it notices that the block begins to move to the right when that value is achieved. It then creates a landmark to indicate that value and an action to reach that value. Based on this landmark, QLAP can learn a contingency (model) that says if the value goes to 0, then the block will move to the right. It can then learn other landmarks that indicate in which situations this will be successful. In a similar way, it learns to pick up the block and knock the block off the table.

7 Open Issue: Learning Representations Consisting of Combinations of Variables

QLAP can only learn actions to achieve values on existing variables. For example, QLAP cannot learn an action to stack blocks if there is no input variable to indicate when the blocks are stacked. QLAP can, however, generate sub-actions that are novel combinations of variables. It learns, for example, that picking up the block requires the agent to first center its hand over the block. Centering its hand over the block requires learning a value among a combination of variables and learning how to achieve that value. Nevertheless, all end-result actions must be specified with a single variable. QLAP could conceivably learn to stack blocks if having stacked blocks were a precondition for some higher-level action, but it is an open issue how agents can robustly and efficiently generate representations that involve multiple input variables.

One approach to tackling this problem is the standard AI approach of generate and test (Winston 1992). This approach could also be used to generate useful intermediate variables that are not given in the input (Stober and Kuipers 2008). These intermediate variables could be combinations of existing variables. For example, if for two existing variables a and b, there was a model that was reliable when $a - b = 90$, then it might make sense to create a new variable $v = a - b$ with a landmark at 90. The challenge with a generate-and-test approach is that the space of possible representations is too large to try them all. We need a method to only generate representations that have a high potential for being useful, and an efficient way to test if indeed they are useful for the agent.

A related issue is how developmental learning can take place in domains with tens of thousands of variables. One approach would be to define a type system on the variables and to restrict representation learning to combinations of particular types. It would be interesting to investigate if this type system could itself be learned, possibly through clustering.

8 Conclusion

This chapter presents four principles for autonomously creating representations for a developing agent. It also presents the QLAP algorithm that embodies these principles. To summarize, the principles are as follows. (1) *Exploit the synergy between the created representations and the development of the agent.* Landmarks lead to new models, and each model can add new landmarks to make it more reliable. This means that in QLAP, the drive to create new representations comes from the goal of better predicting the environment based on the current representation. (2) *Generate representations to stand for phenomena in the environment.* QLAP creates landmarks, events, models, plans, and actions. Each of these representations helps QLAP to focus its attention in the continuous flood of input and to track statistics to learn new knowledge. (3) *Break the environment into many small parts.* QLAP learns models by searching for correlations among pairs of events. This results in many small models and many small plans. These models and plans partition the world into small, tractable parts. (4) *Create representations so learning is easy.* QLAP learns landmarks that allow the agent to generalize. And these landmarks lead to events, which makes the learning of models easy. These models, in turn, identify the relevant variables in the environment and lead to plans that use those variables. These models are small and that allows simple planning algorithms to be used. By employing these principles to create representations, QLAP can progress from low-level motor primitives to high-level hierarchical actions.

Future work for QLAP is to apply it to other domains. Since a QLAP agent is motivated to understand and control its environment, QLAP could be useful in any domain where the goal of learning is not known in advance. One such domain is cyber security. In the cyber domain, QLAP could learn to understand normal

patterns of network traffic and to take basic actions on the network such as blocking a port. With this knowledge and ability, the QLAP agent could identify anomalies and actively work the thwart attacks.

Acknowledgements This work has taken place in the Intelligent Robotics Lab at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Intelligent Robotics lab is supported in part by grants from the National Science Foundation (IIS-0713150).

References

Arbib, M. (1992). Schema theory. *Encyclopedia of Artificial Intelligence, 2*, 1427–1443.

Cohen, P. R., Oates, T., Beal, C. R., Adams, N. (2002). Contentful mental states for robot baby. In *Proceedings of the 18th national conference on artificial intelligence (AAAI-2002)*. San Francisco/Cambridge: AAAI/MIT.

Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence, 5*(2), 142–150.

DeCasper, A. J., & Carstens, A. (1981). Contingencies of stimulation: effects of learning and emotions in neonates. *Infant Behavior and Development, 4*, 19–35.

Drescher, G. L. (1991). *Made-up minds: a constructivist approach to artificial intelligence*. Cambridge: MIT.

Fayyad, U., & Irani, K. (1992). On the handling of continuous-valued attributes in decision tree generation. *Machine Learning, 8*(1), 87–102.

Friedman, N., & Goldszmidt, M. (1996). Discretizing continuous attributes while learning bayesian networks. In *Proceedings of the thirteenth international conference on machine learning (ICML'96)* (pp. 157–165). Los Altos: Morgan Kaufmann.

Gergely, G., & Watson, J. (1999). Early socio-emotional development: contingency perception and the social-biofeedback model. In P. Rochat (Ed.), *Early social cognition: understanding others in the first months of life* (pp. 101–136). Mahwah: Lawrence Erlbaum Associates.

Goodman, N., Mansinghka, V., Tenenbaum, J. (2007). Learning grounded causal models. In *Proceedings of the twenty-ninth annual conference of the cognitive science society* (pp. 305–310). Hillsdale: Erlbaum.

Gunderson, J., & Gunderson, L. (2008). *Robots, reasoning, and reification*. New York: Springer.

Hester, T., & Stone, P. (2009). Generalized model learning for reinforcement learning in factored domains. In *Proceedings of the 8th international conference on autonomous agents and multiagent systems* (vol. 2, pp. 717–724). Richland: International Foundation for Autonomous Agents and Multiagent Systems.

Johnson, M. (1987). *The body in the mind: The bodily basis of meaning, imagination, and reason*. Chicago: University of Chicago Press.

Klein, J. (2003). Breve: a 3d environment for the simulation of decentralized systems and artificial life. In *Proceedings of the eighth international conference on artificial life* (pp. 329–334). Cambridge: MIT Press.

Kuipers, B. (1994). *Qualitative reasoning*. Cambridge: MIT.

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by*. Chicago: University of Chicago Press.

Mandler, J. (2004a). *The foundations of mind, origins of conceptual thought*. New York: Oxford University Press.

Mandler, J. (2004b). A synopsis of the foundations of mind: origins of conceptual thought. *Developmental Science, 7*(5), 499–505.

Modayil, J., & Kuipers, B. (2007). Autonomous development of a grounded object ontology by a learning robot. In *Proceedings of the national conference on artificial intelligence (AAAI 2007)* (vol. 22, p. 1095).

Mugan, J. (2010). *Autonomous qualitative learning of distinctions and actions in a developing agent*. PhD thesis, University of Texas at Austin.

Mugan, J., & Kuipers, B. (2007). Learning to predict the effects of actions: synergy between rules and landmarks. In *IEEE 6th international conference on development and learning, ICDL 2007* (pp. 253–258). New York: Institute of Electrical and Electronics Engineers.

Mugan, J., & Kuipers, B. (2012). Autonomous learning of high-level states and actions in continuous environments. *IEEE Transactions in Autonomous Mental Development, 4*(1), 70–86.

Needham, A., Barrett, T., Peterman, K. (2002). A pick-me-up for infants' exploratory skills: early simulated experiences reaching for objects using 'sticky mittens' enhances young infants' object exploration skills. *Infant Behavior and Development, 25*(3), 279–295.

Oudeyer, P., Kaplan, F., Hafner, V. (2007). Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation, 11*(2), 265–286.

Payne, V. G., & Isaacs, L. D. (2007). *Human motor development: a lifespan approach*. New York: McGraw-Hill Humanities/Social Sciences/Languages.

Piaget, J. (1952). *The origins of intelligence in children*. New York: Norton.

Pierce, D., & Kuipers, B. (1997). Map learning with uninterpreted sensors and effectors. *Artificial Intelligence, 92*(1–2), 169–227.

Puterman, M. (1994). *Markov decision problems*. New York: Wiley.

Rosenberg, K., & Trevathan, W. (2002). Birth, obstetrics and human evolution. *BJOG: An International Journal of Obstetrics & Gynaecology, 109*(11), 1199–1206.

Schmidhuber, J. (1991). Curious model-building control systems. In *IEEE International Joint Conference on Neural Networks* (vol. 2, pp. 1458–1463). New York: Institute of Electrical and Electronics Engineers.

Skinner, B. (1961). *Cumulative record*. New York: Appleton-Century-Crofts.

Smith, R. (2004). *Open dynamics engine v 0.5 user guide*. http://www.ode.org/ode-latest-userguide.html. Accessed 15 April 2012.

Stober, J., & Kuipers, B. (2008). From pixels to policies: a bootstrapping agent. In *IEEE 7th international conference on development and learning, ICDL 2008* (pp. 103–108). New York: Institute of Electrical and Electronics Engineers.

Strehl, A., Diuk, C., Littman, M. (2007). Efficient structure learning in factored-state MDPs. In *Proceedings of the national conference on artificial intelligence (AAAI 2007)* (vol. 22, p. 645). Menlo Park, CA; Cambridge, MA; London: AAAI; MIT; 1999.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning*. Cambridge: MIT.

Sutton, R. S., Precup, D., Singh, S. (1999). Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence, 112*(1–2), 181–211.

Vigorito, C. M., & Barto, A. G. (2010). Intrinsically motivated hierarchical skill learning in structured environments. *IEEE Transactions on Autonomous Mental Development (TAMD), 2*(2), 132–143.

Winston, P. (1992). *Artificial intelligence*, 3rd edn. Reading: Addison-Wesley.

Hierarchies for Embodied Action Perception

Dimitri Ognibene, Yan Wu, Kyuhwa Lee, and Yiannis Demiris

Abstract During social interactions, humans are capable of initiating and responding to rich and complex social actions despite having incomplete world knowledge, and physical, perceptual and computational constraints. This capability relies on action perception mechanisms that exploit regularities in observed goal-oriented behaviours to generate robust predictions and reduce the workload of sensing systems. To achieve this essential capability, we argue that the following three factors are fundamental. First, human knowledge is frequently hierarchically structured, both in the perceptual and execution domains. Second, human perception is an active process driven by current task requirements and context; this is particularly important when the perceptual input is complex (e.g. human motion) and the agent has to operate under embodiment constraints. Third, learning is at the heart of action perception mechanisms, underlying the agent's ability to add new behaviours to its repertoire. Based on these factors, we review multiple instantiations of a hierarchically-organised biologically-inspired framework for embodied action perception, demonstrating its flexibility in addressing the rich computational contexts of action perception and learning in robotic platforms.

1 Introduction

When a boxer is facing an adversary, its action perception system operates under hard embodiment constraints. It should not only recognise the adversary's movements, but also select appropriate response actions based on its prediction of the opponent's goals. To react in time, predicting only immediate movements is insufficient. The boxer needs to infer longer sequences of adversarial actions and

D. Ognibene (✉) · Y. Wu · K. Lee · Y. Demiris
Department of Electrical and Electronic Engineering, Imperial College London, UK
e-mail: d.ognibene@imperial.ac.uk; yan.wu08@imperial.ac.uk; k.lee09@imperial.ac.uk; y.demiris@imperial.ac.uk

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_5,

the underlying intentions (e.g. moving the fight to a corner) and perform strategic movements to unveil the opponent's intentions while hiding its own. This example illustrates the human capabilities to actively perceive others' actions, to predict their intentions at different levels of abstraction and to learn from the observation of others' activities. Our research is focused on equipping robots with robust action perception capabilities to allow them to participate in rich social interactions.

This chapter reports on several experiments on robotic platforms investigating the essential factors to achieve robust action perception performance. We argue that these factors include (a) the use of hierarchical knowledge representation and processing architectures; (b) the use of active perceptual systems, where sensors actively seek the required data to process; (c) the prediction of the sensory consequences of the most probable actions; and (d) the reuse of action execution knowledge for action perception.

The remaining of this section reports on the computational principles underlying these factors along with relevant neuroscience research that supports their role in the human action perception system.

Hierarchical action representations have long been adopted in AI and robotics (Tate 1977) both at the planning and execution stages for coping with large search spaces and long-term decisions that characterise real-world conditions. Different hierarchical frameworks for planning have been proposed such as options or angelic semantics (Sutton et al. 1999). Such frameworks share the presence of a relationship connecting each element in a higher or more abstract level to many elements of the lower levels. However, in each framework the semantics of the relationship can be different, for example the execution of one abstract element can represent a selective, parallel, sequential or order-independent execution of the connected lower level elements.

Hierarchical representations also present advantages for learning and adaptation (Hinton 2010). They may allow for more efficient inference and learning with fewer samples by exploiting partial reuse (Theocharous et al. 2004). At the sensory level, hierarchical processing is extremely helpful in integrating cues from several levels of abstraction while avoiding an expensive centralised computation (the "local administration advantage" Dawkins 1976). Such systems have compact representations and exhibit good generalisation between objects with similar parts (Epshtein and Ullman 2007). In biology, the hierarchical structure of the nervous system has been seen as a general principle that enables animals to behave efficiently in complex environments (Simon 1962). Evidence of a hierarchical nervous architecture is present in many vertebrates (Hess 1957; Honeycutt and Nichols 2010) and invertebrates (Liske 1999). An extensive review on the evidence of a hierarchical organisation of action representation in the human brain, including goals (short-term) and intentions (long-term) is available in (Grafton et al. 2007).

Active perception directs sensors and selects data to process using additional sources of information, such as task knowledge, and predictions based on previous inputs (Aloimonos et al. 1988; Bajcsy 1988; Ballard 1991). By selecting only the relevant input for the current task, active perception permits parsimonious use of the sensory and computational resources (Kato and Floreano 2001). Active

perception can also facilitate more efficient exploration of the environment (Rao and Ballard 1995) and enhance the system robustness to conditions in which relevant information is hidden when observed with passive sensors (Ognibene et al. 2011; Suzuki and Floreano 2006). Apart from reducing the computational costs, active perception can, in some cases, facilitate learning (Ognibene et al. 2010). The active and top-down components of human perception have been confirmed by behavioural (Malcolm and Henderson 2010; O'Regan and Noé 2001; Tatler et al. 2011), imaging (Bar et al. 2006) and recording (Buschman and Miller 2007) evidence.

Prediction of the sensory consequences of actions enables a system to recognise actions in unseen contexts by utilising learned causal relationships between actions and their sensory consequences. Predictive approaches have been extensively studied in machine learning to exploit the capability of generative models to use both unlabelled and labelled data (Bishop and Lasserre 2007). Generative models enable learning of hierarchical representations of the task structure using fewer samples since each layer of abstraction captures domain structure information that is exploited by the other levels (Hinton and Ghahramani 1997). Internal generation of expected results enhances perception with a more robust management of noise and missing observations (sensory substitution). By enabling incremental and anticipative recognition of actions, it extends the decision time available to produce effective behavioural responses. The existence of internal representations, prediction mechanisms and simulation machinery in the brain has been one of the most discussed subjects in the cognitive science literature. Recent studies (Grush 2004; Hesslow 2002; Jeannerod 1994; Pezzulo et al. 2011) collectively show that the presence of low level representations and simulation mechanisms strongly coupled with perception and motor control is the foundation for building higher level processes and representations. This hypothesis is supported by modern imaging and recording evidences (Bar 2007; Bar and Biederman 1999; Reddy and Kanwisher 2006).

The reuse of action execution knowledge for action perception allows the observing agent to recognise others' actions based on the agent's own experience of embodied task execution and vice versa. Evidence of shared mechanisms between execution and recognition has been demonstrated in recent neuroscience studies, such as the different activation of brain areas due to variable level of motor proficiency in the observed action (Calvo-Merino et al. 2005), and improved performance in recognition shown by subjects with higher motor proficiency (Pezzulo et al. 2010). The integration of shared mechanisms and internal generation of expected results leads to what is known as the "simulation theory of action perception" (Gallese and Goldman 1998). According to the simulation theory, the observer recognises an action by comparing it with its internal simulations. Simulations are generated from the perspective of the performer and produced through the motor systems of the observer. By using its own motor system, the observer can directly have access to goal-based representations detached from raw observations. In neuroscience, the discovery of mirror-neurons—motor neurons active during both execution and perception—in primates (Gallese et al. 1996) and humans (Fadiga et al. 1995; Gazzola and Keysers 2009; Iacoboni et al. 1999; Keysers and Gazzola

2010) provides support to the simulation theory of action understanding (Cuijpers et al. 2006; Demiris 2007; Shanton and Goldman 2010).

In the next section, we introduce Hierarchical Attentive Multiple Models for Execution and Recognition (HAMMER) (Demiris and Johnson 2003; Johnson and Demiris 2004) as a prototype of hierarchical action recognition architectures which possess the aforementioned characteristics.

2 Hierarchical Attentive Multiple Models for Execution and Recognition

The HAMMER architecture is a framework based on simulation theory, designed to empower robots with capabilities to understand and imitate human actions based on the four factors described in the previous section. This framework has been implemented in real-dynamics robot simulators (Demiris and Hayes 2002) and real robotic platforms (Demiris and Johnson 2003; Demiris and Khadhouri 2008; Johnson and Demiris 2004). Open source versions of the architecture have been freely released (Sarabia et al. 2011) with support for the NAO and iCub humanoids.

Figure 1 shows the schematics of the HAMMER architecture. The basic building block of HAMMER consists of an inverse–forward model pair. The inverse model generates action commands from a set of input states aiming to advance the robotic agent towards a goal. This goal can be implicitly or explicitly specified in the model. The forward model provides an estimate of the upcoming states given the action commands and current state. Predictions of upcoming states in execution mode can be used to overcome delays, to handle input noise and as sensory substitution. When it is used to recognise actions, such predictions from each model are compared against the demonstrator's actual states.

For each inverse/forward model pair, the prediction of the demonstrator's next state is evaluated against the ground truth to provide an error signal. The error signal accumulated over time is used to compute the confidence value of the model pair, which is an indicator of how closely the demonstrated action matches the model. During execution, the confidence value is used to detect the actual context and/or hidden states. This enables the switching from one model to another according to the confidence indicator of the fittest model. The confidence signal can also be used as credit assignment for module training (Haruno et al. 2001, 2003). The architecture recognises actions of others by comparing the observed movements with the different expected results produced by running its own motor models ("putting the observer in the shoes of the demonstrator") in parallel while inhibiting the models from sending their generated commands to motor systems.

The HAMMER architecture incorporates a top-down allocation of sensory and computational resources. For action-execution, HAMMER can rely on the simple principle of "attention for action" and seek the information required by the current task. On the other hand, action-recognition poses new problems for the attention

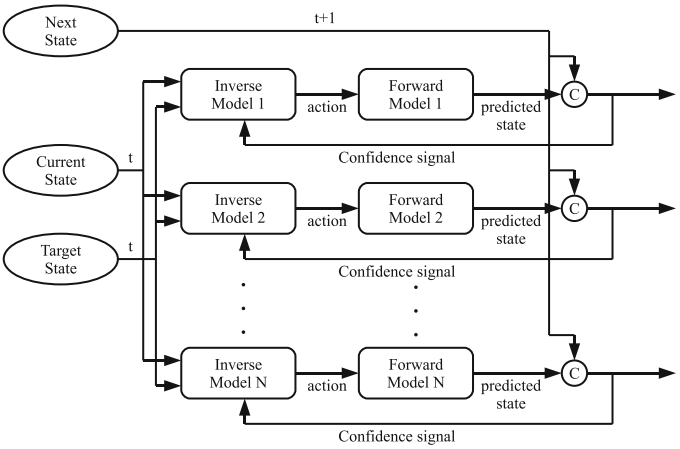

Fig. 1 The core of the HAMMER architecture consists of a distributed network of inverse and forward models that compete to predictively explain the ongoing demonstration. Comparators (C) compute the confidence signals from current and predicted states of the models

system since the observer does not know in advance what the observed task is. HAMMER maps simulations to attentional needs using the following principle: during the demonstration of actions, the information requested by the attention system of the observer are those needed to generate the internal simulated actions (Demiris and Khadhouri 2006). For example, the inverse model for executing an arm movement will request the state of the corresponding arm of the performer when it is used in perception mode. This principle is compatible with the pre-motor theory of attention in humans which states that the preparation or simulation of action enhances the perception of related stimuli (Fagioli et al. 2007).

Action imitation can be achieved by integrating recognition and execution in HAMMER. The system starts in recognition mode, and when a model with the highest confidence reaches a certain threshold, the command inhibition is deactivated to allow the model to reproduce the observed action if required. If no confidence value reaches the threshold within a time limit, a new motor model is learned to represent the postural and posture-object configurations of the observed action.

The HAMMER models can be connected in arbitrarily complex configurations. Their overt execution does not need to be mutually exclusive, i.e. models managing different joints (Demiris and Hayes 2002) can be executed overtly in parallel. This arrangement has been extended to hierarchical structure as shown in Fig. 2 (Demiris and Johnson 2003): primitive models are combined to form higher, more complex sequences, with the eventual goal of achieving increasingly more abstract inverse models (Johnson and Demiris 2004).

Using the underlying principles in HAMMER, Demiris (Demiris and Hayes 2002) derived a set of testable predictions for the behaviour of biological systems. A key prediction states that mirror neurons in monkeys would not fire (or fire less)

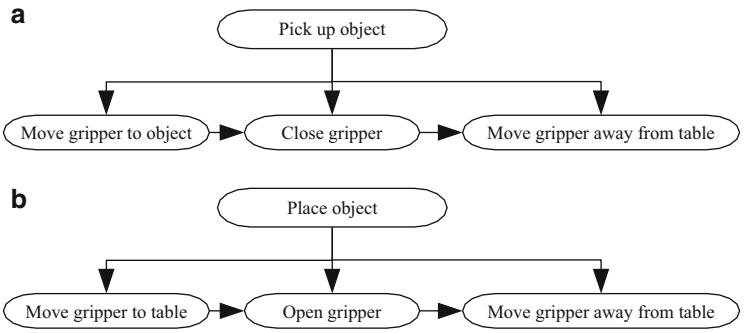

Fig. 2 Example arrangements of primitive inverse models into more complex inverse models: (**a**) pick up object (**b**) place object

when the demonstrated movement was performed at speeds unattainable by the observer. Experiments subsequently reported in (Gangitano et al. 2001) showed that the amplitude of the motor evoked potentials (MEP) induced by transcranial magnetic stimulation (TMS) in humans observing a reaching-grasping action was modulated by the kinematics of the observed finger aperture.

Modelling human grasping action and its perception with HAMMER reproduced several interesting neuroscience observations: (a) the computational grasping model reproduced some of the characteristics of human grasping including (Jeannerod 1981) an overshoot in the grip aperture at approximately 70 % of the movement time (Simmons and Demiris 2006); (b) the TMS-based results on the response of mirror neurons to different action timings and coordination properties reported in (Gangitano et al. 2004).

In the next section, we will present experiments demonstrating how the hierarchical generative approach to action perception may be used to cope with embodiment constraints in action perception.

3 Hierarchical Action Perception and Abstraction

This section describes a HAMMER implementation (Johnson and Demiris 2004) on an ActivMedia Peoplebot, and how its hierarchical representation is used to cope with the "correspondence problem" (Nehaniv and Dautenhahn 2002), the problem incurred by an imitator during imitation of actions produced by a performer with different embodiment.

In this implementation, two kinds of models were used: (a) Primitive models constructed as a simple motor program (the inverse model) tightly coupled with a hand-coded forward model; and (b) Higher level models implemented as graphs: to create a hierarchy, graphs are handled recursively with a graph node

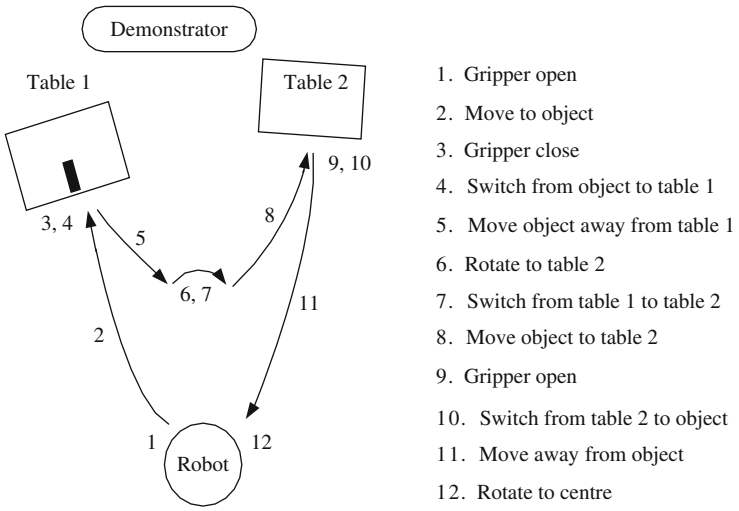

Fig. 3 The sequence of primitive inverse models that constitute the abstract inverse model for moving an object from one table to another

being either a graph itself or a primitive (e.g. Fig. 2). Models connected serially are executed in serial manner, and those connected in parallel are executed in parallel. A goal state is associated with each inverse model. During execution, a graph will execute each of its constituent nodes in turn until completion. At this point, the node will reset its confidence and the graph will continue execution of the subsequent node. For recognition, the sequential execution constraint is relaxed. Inverse models at all levels of the hierarchy are executed in parallel regardless of the recognition stage. At each step, every model signals its performance to all parent-models by propagating its confidence value. Each high-level model computes its confidence based on (and normalised against) the first model in the child model sequence whose confidence value has reached a certain threshold.

In these experiments, the Peoplebot had to learn to recognise and transport objects between two tables following a human demonstration of this task, with the two agents, human and Peoplebot having very different embodiments. State information was extracted using visual markers. The states consisted of the positions of the hand, tables and objects, the relative distances among them, their derivatives, and a boolean flag indicating "object in gripper". The high-level abstract inverse model was constructed by learning primitive inverse models from human demonstration. Twenty three primitive inverse models were available to the architecture, while 20 repeated demonstrations performed at natural speeds and trajectories were conducted in the experiment (Johnson and Demiris 2004). Note that not all actions in Fig. 3 are absolutely necessary for the human demonstrator to achieve the final goal. Moreover for the robot, it is not easy to perceive some of them. The confidence

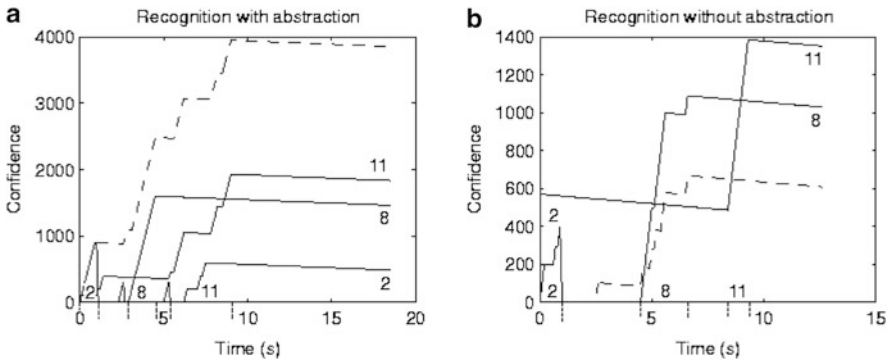

Fig. 4 Graphs of confidence over time for four inverse models in recognition mode. The *dashed* series is the high-level inverse model for moving an object between two tables. The other three inverse models are numbered as in Fig. 3. Graph (A) shows the confidences of these inverse models recognised using the abstraction mechanism. The high-level inverse model, represented by series 1, incorporates the other inverse models as salient features of the demonstration, achieves the highest overall confidence and thus is successfully recognised. Graph (B) shows the confidences of the inverse models when recognition is performed without the abstraction mechanism. The high-level inverse model fails to incorporate the other inverse models, achieves a low overall confidence, and thus is not recognised

evolution plot of recognition with the abstraction mechanism (Fig. 4a) shows how the high-level inverse model recognises the other inverse models as being salient and incorporates them to achieve the highest overall confidence. Figure 4b shows the high-level inverse model failing to achieve high confidence; without the abstraction mechanism, the motor pattern of the high-level inverse model is so fundamentally different to that performed by the demonstrator, that it fails to match. The reported experiments (full details in Johnson and Demiris 2004) show that an agent endowed with hierarchical action representations with abstraction capabilities can understand and imitate a composite action and its final goal even if it cannot directly execute it or does not know each single composing action. Moreover the actions which are recognised and can be executed may not contribute by themselves to the achievement of the goal, or may also miss some precondition to be executable. With a hierarchical representation the agent can use the recognised actions as clues for the rehearsal of a higher level action which will put the recognised lower level actions in the proper context for execution.

4 Acquiring Hierarchical Representations for Integrated Social and Autonomous Learning

The previous section demonstrated how hierarchically organised inverse and forward models were able to observe and imitate a sequence of actions. A fundamental question underlying this research is where do these models come

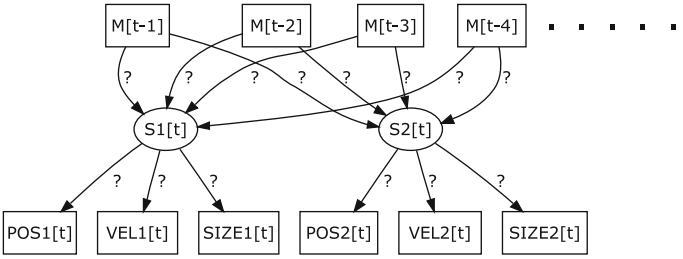

Fig. 5 The Bayesian network for the ActivMedia Peoplebot's gripper forward model. The robot has to learn the mappings (indicated by question marks) between sequences of motor commands (top row M-nodes) and resulting states of the gripper (perceived through a camera, bottom row), despite the inherent sensory delays of real-robotic systems

from? In the following sections we will describe how these hierarchies can be learned, starting from learning primitive forward and inverse models, to learning action descriptions using stochastic context free grammars (SCFG).

4.1 Learning Primitive Models Through Motor Babbling

First, we are faced with the problem of how to learn the models at the lowest part of a hierarchy, i.e. primitive inverse and forward models. We have developed a system that learns primitive forward and inverse models through motor babbling (Dearden and Demiris 2005), a learning method that associates randomly executed motor commands and their effects on environment (Gopnik and Meltzoff 1997). The system learns a forward model implemented with a Bayesian network (Pearl 2000) as shown in Fig. 5 without prior knowledge of its motor system or the external environment. The forward model represents a probability distribution of the states of the robot and other objects in the environment after d time-steps from the previous motor commands. The network structure is learned online by performing a search through the set of possible structures (with different delays in motor commands and different observation nodes for each possible object) and choosing the one which maximises the log-likelihood of the observed experiment results. Expectation maximisation (EM) is used with the inference stage performed with the junction-tree algorithm (Pearl 2000). Moving objects in the scene are automatically detected and tracked by clustering the low-level image features of the visual input (Lucas and Kanade 1981). An inverse model is derived by the forward model by exploiting the Bayesian representation (see Fig. 6).

The learned HAMMER inverse–forward model enables the robot to imitate simple human hand movements by replacing the robot's observations of its own movements by those of a human demonstrator shown in Fig. 7.

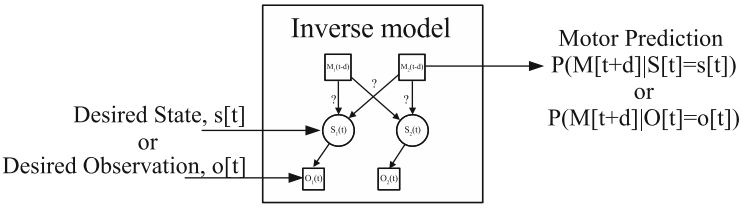

Fig. 6 Using the learnt Bayesian network as an inverse model. Evidence is supplied to the observations or the state, and the task is to infer the probability distribution of motor commands

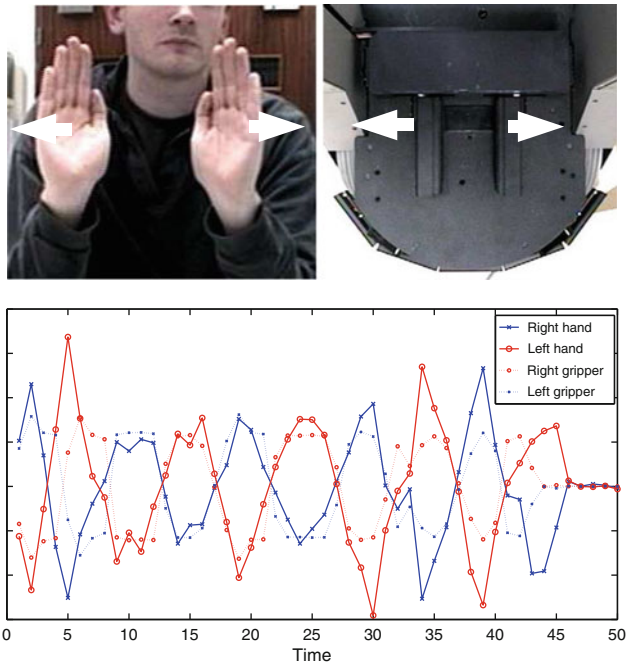

Fig. 7 Imitation using a single inverse model. The top images are corresponding frames from the demonstration (*left*) and imitating (*right*) sequences. The graphs show the trajectory of the demonstrating hands, and the corresponding imitating trajectory of the grippers

4.2 Learning Action Sequences by Demonstration

Having learned the primitive inverse and forward models, imitation can be used to learn sequences of these models in order to complete more complex tasks. An early study (Demiris and Johnson 2003) used two ActiveMedia Peoplebots facing each other. One robot executed a sequence of actions while the other observed and learned this sequence of actions. The observer initially is equipped with basic action primitives to control its gripper (open, close, rise and lower) but does not possess any

high level action model. A typical experiment consisted of the robot demonstrator executing a random sequence of basic action primitives, with the imitator robot observing, storing the observed sequence of inverse models in working memory and subsequently replicating. These early experiments demonstrated how sequence learning can occur, but attempted no further processing in the models in working memory other than simplying storing. For the purposes of this chapter, an interesting aspect is how low level sequences can be generalised to new situations and how we can infer action hierarchies from these observations, in order to utilise their benefits as advocated in this chapter. In order to do so, we first turn our attention to how observations can lead to generalisable primitives; the benefit of the next algorithm (OSILA) is that it can generalise from a single demonstration, but it lacks certain types of expressiveness (for example, it cannot readily learn to represent recursion). The final section will describe an algorithm that enables the learning of more expressive abstract representations involving probabilistic grammars.

4.3 Learning Generalisable Action Templates from Single Observation

Humans can learn new tasks from a single demonstration; an one-shot imitation learning algorithm (OSILA) was proposed in Wu and Demiris (2010) to tackle the problem of learning (through a single observation) action primitives that can be adapted to new contexts. It stores observed actions as human-readable movement templates and re-adapts them according to the constraints of the new contexts. OSILA (Fig. 8) conjectures a spatial relationship between the template and the applied environment to locate relevant invariant landmarks or invariant control points (ICPs) in both contexts. Subsequently, it uses a minimum distortion function based on Thin Plate Splines (TPS) warping to define a mapping between the two spaces. This allows the definition of a set of candidate waypoints in the applied space extracted from the observed action. The adaptation mechanism is based on the use of a warping energy measure that reduces the deformation of the performed action with different environmental structures.

An inverse model in OSILA reproduces the action using the visual state information of the new context and adapting the previously-observed action template, setting a threshold for tolerable warping energy when matching available hypotheses (templates). In this way the algorithm has a principled way for selecting when to learn new primitives or use combinations of the already learnt primitives. Experiments conducted in Wu and Demiris (2010) show that the trajectories stably generated by OSILA resemble the paths produced by humans under similar circumstances. Experimental scenarios using the iCub humanoid robot included the game of tic-tac-toe (Fig. 9), where the robot learned (through one-shot demonstration) different templates of movements (for example, making an O in one position of the

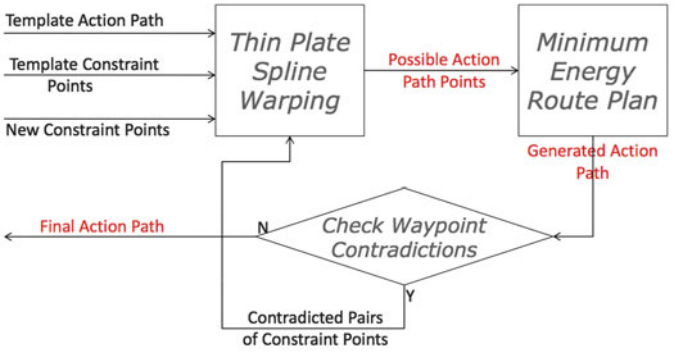

Fig. 8 Adapting a OSILA-learned template of an inverse model to a new context

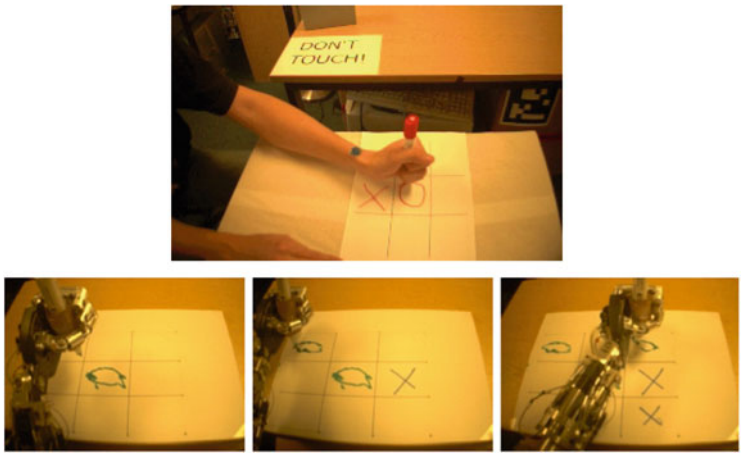

Fig. 9 Using OSILA to adapt learned templates to new board positions using the iCub humanoid in a tic-tac-toe game

board, that it could subsequently generalise to other positions of a different-sized board at another location).

4.4 Learning Action Hierarchies Using Probabilistic Grammars

The previously described learning mechanisms do not explicitly tackle the problem of learning a hierarchical structure which is particularly important both for generalisation and for keeping resource requirements bounded. They also do not explicitly

focus on the advantages of hierarchical representations to boost noise robustness when perceiving complex and long action sequences.

Lee et al. (2012) studied these issues using a generative approach to learn task representations in the form of SCFG from demonstrations. These representations allow the expression of complex hierarchies of actions in a compact and efficient manner. During recognition, they take into account the uncertainties of actions common in real-world settings in a probabilistic manner which makes this framework highly scalable. SCFGs are also capable of recognising arbitrary lengths of action sequences composed of a finite set of action symbols using recursive expressions. SCFGs essentially extend the Context-Free Grammars (CFG) framework (Ryoo and Aggarwal 2006) by associating a probability with each production rule, which enables all parse trees to be assigned with probability values based on the production rules used. In Lee et al. (2012), the terminal symbols of the grammar are generated by primitive action detectors, while non-terminal symbols can be thought of as sequences of primitive actions. In an example scenario, "take out all the objects in a bag and give them to a human", the robot must repeatedly perform high-level actions such as "take out objects" and "give them to a human", which are composed of lower-level actions such as "locate", "approach", "grasp", "move" and "release". Researchers have argued that understanding everyday human behaviour requires such hierarchical and recursive representation (Ivanov and Bobick 2000; Ryoo and Aggarwal 2006) and that the recognition using a direct pattern-matching approach against all possible behaviours is not a computationally efficient approach. As described below, the HAMMER-like SCFG restrains the set of candidate behaviours through the use of higher level grammatical production rules and predicts future observations using the available information online.

Using SCFGs to recognise an action involves selecting a parse tree (that is a hierarchical structure using the connections represented in the grammar) that best explains the observation, i.e. the parsed action sequence with the highest probability. In Lee and Demiris (2011), it was applied in a real-world scenario where an iCub humanoid robot uses task-independent action templates in the form of SCFGs to recognise human behaviours.

The algorithm proposed in Lee et al. (2012) exploits the confidence values computed by the primitive action detectors during both learning and recognition to deal with ambiguities inherent in perception. During parsing, the algorithm computes probability distributions of the possible parse trees based on (noisy) symbols observed so far and updates the distribution after each new input. The probability distribution over the parse tree permits to derive the expectation of the future inputs in a compact way.

During learning, the algorithm starts with a naive grammar containing all input sequences. It then builds grammar hypotheses using "Substitute" and "Merge" operators to find the grammar with the minimum description length (Langley and Stromsten 2000) that maximises the posterior probability. The algorithm actively searches for frequently occurring sub-sequences of actions to infer the hierarchical structure which allows more compact and generalised representations while offering robustness to observations containing errors. Thus, erroneous sequences

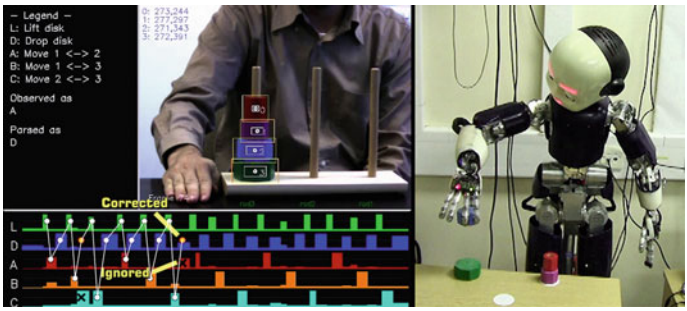

Fig. 10 Observing and learning new hierarchical behaviours in the Towers of Hanoi game, using stochastic context free grammars

are assigned lower probability values than frequently occurring ones. Furthermore, the confidence values of the primitive action detectors are considered to emphasise symbols with less ambiguity. The Substitute operator replaces a partial sequence of symbols in the right-hand side of the rule and groups them into a new symbol, thus building a structural hierarchy. The Merge operator is applied on two symbols in such a way that both symbols are replaced with a single symbol. This process turns the current representation into a more generalised, compact representation. Both operators are applied until the grammar with the minimum description length is found.

The algorithm (Lee et al. 2012) was tested on artificial data sets and on real videos of humans solving the tower of Hanoi game (Fig. 10). In the artificial data set, the tested data was generated by a grammar model ($a^n c b^n$) with various levels of noise manifested by substituting and inserting terminal symbols in the input strings with a random symbol. Each symbol was also assigned with varying confidence values. As compared to other state-of-the-art algorithms, the algorithm proved to be able to produce grammars that were more compact and more robust to noise and execution errors. In the real-world experiments, videos of humans solving the Towers of Hanoi puzzle were used as input (Fig. 10). The primitive action detectors were designed using HMMs that can recognise different actions of moving a disk between two poles. In this experiment, the algorithm was able to acquire accurate representations of the task despite the fact that the observations contained several errors.

These results show an interesting aspect of hierarchical action representations that is advantageous for learning by observation: learned hierarchical structures can effectively deal with observation ambiguities. Moreover, the experiments demonstrate that the chosen hierarchical approach is able to learn a generalised task representation that is able to recognise unforeseen, more complex actions with the same task type, e.g. playing the Towers of Hanoi puzzle with a larger number of disks than those demonstrated.

5 Conclusions

In this chapter, we argued for the computational benefits of hierarchies for embodied action perception. We presented the HAMMER architecture that we use as a framework to empower our robots with capabilities to understand, learn from and imitate human actions.

The experiments reported in this chapter described the essential roles played by hierarchical representations implemented in/with HAMMER in action perception and social learning. We argued that:

- hierarchical representations allow internal representations of a task to match external demonstrations of it when performed by others, even when the embodiment characteristics of the demonstrator and the imitator are different. In general, hierarchical representations allow the demonstrated and predicted information to be compared at the appropriate of level of abstraction, providing flexibility and robustness to execution variability.
- hierarchical representations can be learned through a combination of low-level inverse/forward model learning using techniques such as motor babbling and low-level imitation, while more abstract hierarchical representations can be constructed using tools such as SCFG which exploit the symbolic representations provided by the lower level models.

While the reported experiments demonstrated several successful applications of HAMMER in human robot interactions, natural social interactions pose far more complex challenges. Action perception and imitation capabilities are crucial for enabling robots to behave in unstructured social environments and for natural interactions in dynamic contexts. In these contexts, the information is far richer and more complex, and the set of behaviours that the robot will need to recognise and discriminate will be more wide-ranging. The role played by hierarchical representations will grow in importance as our robots increasingly tackle more challenging social interaction scenarios.

Acknowledgments This research has received funding from the European Union Seventh Framework Programme FP7/2007-2013, under grant agreement no. [270490]- [EFAA].

References

Aloimonos, J., Weiss, I., Bandyopadhyay, A. (1988). Active vision. *International Journal of Computer Vision, 1*(4), 333–356.

Bajcsy, R. (1988). Active perception. *Proceedings of the IEEE, 76*(8), 966–1005.

Ballard, D. (1991). Animate vision. *Artificial Intelligence, 48*, 57–86.

Bar, M. (2007). The proactive brain: using analogies and associations to generate predictions. *Trends in Cognitive Science, 11*(7), 280–289.

Bar, M., & Biederman, I. (1999). Localizing the cortical region mediating visual awareness of object identity. *Proceedings of the National Academy of Sciences USA, 96*(4), 1790–1793.

Bar, M., Kassam, K. S., Ghuman, A. S., Boshyan, J., Schmid, A. M., Schmidt, A. M., Dale, A. M., Hamalainen, M. S., Marinkovic, K., Schacter, D. L., Rosen, B. R., Halgren, E. (2006). Top-down facilitation of visual recognition. *Proceedings of the National Academy of Sciences USA, 103*(2), 449–454.

Bishop, C. M., & Lasserre, J. (2007). Generative or discriminative? getting the best of both worlds. *Bayesian Statistics, 8*, 3–24.

Buschman, T. J., & Miller, E. K. (2007). Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices. *Science, 315*(5820), 1860–1862.

Calvo-Merino, B., Glaser, D., Grèzes, J., Passingham, R., Haggard, P. (2005). Action observation and acquired motor skills: an fmri study with expert dancers. *Cerebral Cortex, 15*(8), 1243–1249.

Cuijpers, R. H., van Schie, H. T., Koppen, M., Erlhagen, W., Bekkering, H. (2006). Goals and means in action observation: a computational approach. *Neural Networks, 19*(3), 311–322.

Dawkins, R., Bateson, P. P. G., & Hinde, R. A. (1976). *Growing points in ethology* (pp. 7–54). London: Cambridge University Press.

Dearden, A. M., & Demiris, Y. (2005). Learning forward models for robots. In *IJCAI-05, Proceedings of the nineteenth international joint conference on artificial intelligence, Edinburgh, Scotland, UK, July 30–August 5, 2005* (pp. 1440–1445).

Demiris, Y. (2007). Prediction of intent in robotics and multi-agent systems. *Cognitive Processing, 8*(3), 151–158.

Demiris, Y., & Hayes, G. M. (2002). Imitation as a dual-route process featuring predictive and learning components: a biologically-plausible computational model. In *Imitation in animals and artifacts*. Cambridge: MIT.

Demiris, Y., & Johnson, M. (2003). Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning. *Connection Science, 15*(4), 231–243.

Demiris, Y., & Khadhouri, B. (2006). Hierarchical attentive multiple models for execution and recognition of actions. *Robotics and Autonomous Systems, 54*(5), 361–369.

Demiris, Y., & Khadhouri, B. (2008). Content-based control of goal-directed attention during human action perception. *Interaction Studies, 9*(2), 353–376.

Epshtein, B., & Ullman, S. (2007). Semantic hierarchies for recognizing objects and parts. In *IEEE conference on computer vision and pattern recognition, 2007. CVPR'07* (pp. 1–8). New York: IEEE.

Fadiga, L., Fogassi, L., Pavesi, G., Rizzolatti, G. (1995). Motor facilitation during action observation: a magnetic stimulation study. *Journal of Neurophysiology, 73*(6), 2608–2611.

Fagioli, S., Hommel, B., Schubotz, R. (2007). Intentional control of attention: action planning primes action-related stimulus dimensions. *Psychological Research, 71*(1), 22–29.

Gallese, V., Fadiga, L., Fogassi, L., Rizzolatti, G. (1996). Action recognition in the premotor cortex. *Brain, 119*(2), 593.

Gallese, V., & Goldman, A. (1998). Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences, 2*(12), 493–501.

Gangitano, M., Mottaghy, F., Pascual-Leone, A. (2001). Phase-specific modulation of cortical motor output during movement observation. *Neuroreport, 12*(7), 1489.

Gangitano, M., Mottaghy, F., Pascual-Leone, A. (2004). Modulation of premotor mirror neuron activity during observation of unpredictable grasping movements. *European Journal of Neuroscience, 20*(8), 2193–2202.

Gazzola, V., & Keysers, C. (2009). The observation and execution of actions share motor and somatosensory voxels in all tested subjects: single-subject analyses of unsmoothed fmri data. *Cerebral Cortex, 19*(6), 1239–1255.

Gopnik, A., & Meltzoff, A. (1997). *Words, Thoughts, and Theories.* Cambridge: MIT.

Grafton, S., et al. (2007). Evidence for a distributed hierarchy of action representation in the brain. *Human Movement Science, 26*(4), 590–616.

Grush, R. (2004). The emulation theory of representation: motor control, imagery, and perception. *Behavioral and Brain Sciences, 27*(3), 377–96.

Haruno, M., Wolpert, D., Kawato, M. (2001). Mosaic model for sensorimotor learning and control. *Neural Computation, 13*(10), 2201–2220.

Haruno, M., Wolpert, D., Kawato, M. (2003). Hierarchical mosaic for movement generation. *Excepta Medica International Coungress Series, 1250*, 575–590.

Hess, W. R. (1957). *The functional organization of the diencephalon*. New York: Grune & Stratton.

Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception. *Trends in Cognitive Sciences, 6*(6), 242–247.

Hinton, G. (2010). Learning to represent visual input. *Philosophical Transactions of the Royal Society B: Biological Sciences, 365*(1537), 177.

Hinton, G. E., & Ghahramani, Z. (1997). Generative models for discovering sparse distributed representations. *Philosophical Transactions of the Royal Society of London B, 352*, 1177–1190.

Honeycutt, C., & Nichols, T. (2010). The decerebrate cat generates the essential features of the force constraint strategy. *Journal of Neurophysiology, 103*(6), 3266.

Iacoboni, M., Woods, R. P., Brass, M., Bekkering, H., Mazziotta, J. C., Rizzolatti, G. (1999). Cortical mechanisms of human imitation. *Science, 286*(5449), 2526–2528.

Ivanov, Y., & Bobick, A. (2000). Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 852–872.

Jeannerod, M. (1981). *Intersegmental coordination during reaching at natural visual objects* (vol. 9, pp. 153–168). Hillsdale: Lawrence Erlbaum Associates, Inc.

Jeannerod, M. (1994). The representing brain: neural correlates of motor intention and imagery. *Behavioral and Brain Sciences, 17*(02), 187–202.

Johnson, M., & Demiris, Y. (2004). *Towards Autonomous Robotic Systems: Proceedings of TAROS 2004*; University of Essex, 6.-8.9.2004. Technical report series/Department of Computer Science, University of Essex. http://books.google.co.uk/books?id=XIzhjwEACAAJ

Kato, T., & Floreano, D. (2001). An evolutionary active-vision system. In *Proceedings of the 2001 congress on evolutionary computation* (vol. 1, pp. 107–114). New York: IEEE. doi:10.1109/CEC.2001.934378.

Keysers, C., & Gazzola, V. (2010). Social neuroscience: mirror neurons recorded in humans. *Current Biology, 20*, 353–354.

Langley, P., & Stromsten, S. (2000). Learning context-free grammars with a simplicity bias. In *Proceedings of the 11th European conference on machine learning* (pp. 321–338). Berlin: Springer.

Lee, K., & Demiris, Y. (2011). Towards incremental learning of task-dependent action sequences using probabilistic parsing. In *IEEE first joint international conference on development and learning and on epigenetic robotics (ICDL-EPIROB 2011)*. Germany: Frankfurt am Main

Lee, K., Kim, T. K., Demiris, Y. (2012). Learning reusable task representations using hierarchical activity grammars with uncertainties. In *IEEE international conference on robotics and automation (IEEE ICRA 2012)*. Minnesota: St. Paul.

Liske, E. (1999). The hierarchical organiztion of mantid behaviours. In F. R. Prete, H. Wells, P. H. Wells, L. E. Hurd (Eds.), *The praying mantids*. Baltimore: Johns Hopkins University Press.

Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of imaging understanding workshop* (pp. 121–130). Darpa.

Malcolm, G. L., & Henderson, J. M. (2010). Combining top-down processes to guide eye movements during real-world scene search. *Journal of Vision, 10*, 1–11.

Nehaniv, C., & Dautenhahn, K. (2002). *The correspondence problems*, Chap. 2 (pp. 41–61). Cambridge: MIT.

Ognibene, D., Catenacci Volpi, N., Pezzulo, G. (2011). Learning to grasp information with your own hands. In *Proceedings of 12th conference towards autonomous robotics systems (TAROS 2011)*. Berlin: Springer. http://link.springer.com/book/10.1007/978-3-642-23232-9/page/1

Ognibene, D., Pezzulo, G., Baldassarre, G. (2010). How can bottom-up information shape learning of top-down attention control skills? In *Proceedings of 9th international conference on development and learning*. New York: IEEE.

O'Regan, J. K., & Noé, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral Brain Science, 24*(5), 939–973.

Pearl, J. (2000). *Causality: models, reasoning and inference*. Cambridge: Cambridge University Press.

Pezzulo, G., Barca, L., Bocconi, A., Borghi, A. (2010). When affordances climb into your mind: advantages of motor simulation in a memory task performed by novice and expert rock climbers. *Brain and Cognition, 73*(1), 68–73.

Pezzulo, G., Barsalou, L. W., Cangelosi, A., Fischer, M. H., Spivey, M., McRae, K. (2011). The mechanics of embodiment: a dialogue on embodiment and computational modeling. *Frontiers in Psychology, 2*(00005).

Rao, R. P., & Ballard, D. (1995). An active vision architecture based on iconic representations. *Artificial Intelligence, 78*(1–2), 461–505.

Reddy, L., & Kanwisher, N. (2006). Coding of visual objects in the ventral stream. *Current Opinion in Neurobiology, 16*(4), 408–414.

Ryoo, M., & Aggarwal, J. (2006). Recognition of composite human activities through context-free grammar based representation. In *IEEE computer society conference on computer vision and pattern recognition, 2006* (vol. 2, pp. 1709–1718). New York: IEEE.

Sarabia, M., Ros, R., Demiris, Y. (2011). Towards an open-source social middleware for humanoid robots. In *Proceedings of the IEEE/RAS international conference on humanoid robotics*. New York: IEEE.

Shanton, K., & Goldman, A. (2010). Simulation theory. *Wiley Interdisciplinary Reviews: Cognitive Science, 1*(4), 527–538.

Simmons, G., & Demiris, Y. (2006). Object grasping using the minimum variance model. *Biological Cybernetics, 94*(5), 393–407.

Simon, H. A. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society, 106*(6), 467–482.

Sutton, R. S., Precup, D., Singh, S. (1999). Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence, 211*, 112–181.

Suzuki, M., & Floreano, D. (2006). Evolutionary active vision toward three dimensional landmark-navigation. In *From animals to animats 9*. Berlin: Springer. http://link.springer.com/book/10.1007/11840541/page/1

Tate, A. (1977). Generating project networks. In *Proceedings of the international joint conference on artificial intelligence (IJCAI-77)* (pp. 888–893). Cambridge: Morgan Kaufmann.

Tatler, B. W., Hayhoe, M. M., Land, M. F., Ballard, D. (2011). Eye guidance in natural vision: reinterpreting salience. *Journal of Vision, 11*(5), 1–23.

Theocharous, G., Murphy, K., Kaelbling, L. (2004). Representing hierarchical pomdps as dbns for multi-scale robot localization. In *2004 IEEE international conference on robotics and automation (ICRA)* (vol. 1, pp. 1045–1051). New York: IEEE.

Wu, Y., & Demiris, Y. (2010). Towards one shot learning by imitation for humanoid robots. In *2010 IEEE international conference on robotics and automation (ICRA)* (pp. 2889–2894). New york: IEEE.

Learning and Coordinating Repertoires of Behaviors with Common Reward: Credit Assignment and Module Activation

Constantin A. Rothkopf and Dana H. Ballard

Abstract Understanding extended natural behavior will require a theoretical understanding of the entire system as it is engaged in perception and action involving multiple concurrent goals such as foraging for different foods while avoiding different predators and looking for a mate. A promising way to do so is reinforcement learning (RL) as it considers in a very general way the problem of choosing actions in order to maximize a measure of cumulative benefits through some form of learning, and many connections between RL and animal learning have been established. Within this framework, we consider the problem faced by a single agent comprising multiple separate elemental task learners that we call modules, which jointly learn to solve tasks that arise as different combinations of concurrent individual tasks across episodes. While sometimes the goal may be to collect different types of food, at other times avoidance of several predators may be required. The individual modules have separate state representations, i.e. they obtain different inputs but have to carry out actions jointly in the common action space of the agent. Only a single measure of success is observed, which is the sum of the reward contributions from all component tasks. We provide a computational solution for learning elemental task solutions as they contribute to composite goals and a solution for how to learn to schedule these modules for different composite tasks across episodes. The algorithm learns to choose the appropriate modules for a particular task and solves the problem of calculating each module's contribution to the total reward. The latter calculation works by combining current reward estimates with an error signal resulting from the difference between the global reward and the

C.A. Rothkopf (✉)
Frankfurt Institute for Advanced Studies, Goethe University, Frankfurt am Main, Germany
e-mail: rothkopf@fias.uni-frankfurt.de

D.H. Ballard
Department of Computer Science, University of Texas at Austin, Austin, TX, USA
e-mail: dana@cs.utexas.edu

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_6,

sum of reward estimates of other co-active modules. As the modules interact through their action value estimates, action selection is based on their composite contribution to individual task combinations. The algorithm learns good action value functions for component tasks and task combinations which is demonstrated on small classical problems and a more complex visuomotor navigation task.

1 Introduction

Making progress in understanding natural human visuomotor behavior requires considering the entire system as it is engaged in solving tasks involving natural perception and action sequences. A wealth of previous research has demonstrated that active perceptual strategies and perceptual states are highly dependent on the ongoing tasks (e.g., Ballard et al. 1995; Land and McLeod 2000; Yarbus 1967). It is therefore important to consider vision as it is performed within its natural setting of an embodied agent who is engaged in goal directed behavior. This requires developing models that inherently represent these behavioral goals. A particularly promising theoretical framework for addressing these questions is therefore reinforcement learning (RL) (Sutton and Barto 1998).

The promise of RL is to have an agent learn how to solve a task based on the experience they accumulate while interacting with an environment. The generality of RL leads to the hope that it may be applied to a large variety of problems in sequential perception and action. This hope has partially been nourished by successes in relating psychophysical and neuronal data obtained under laboratory settings in animals and humans to specific quantities in RL algorithms (Daw and Doya 2006; Schultz et al. 1997). Nevertheless, many open problems in relating everyday human visuomotor behavior in complex environments to RL remain, as the studied examples both at the theoretical and at the experimental level often are limited to worlds, which are simple with respect to how they evolve over time, how the observations of the agent are related to the states of the world, how the actions play out in the world, and how the feedback about the success in achieving the goal of actions is obtained.

One fundamental problem of RL algorithms is that they do not scale up well to large tasks since the state spaces, i.e. the representations of the states of the world, grow exponentially in the number of state variables per dimension: the so-called curse of dimensionality. This problem has made it difficult to apply RL to naturalistic settings, with the result that the state spaces considered are generally small in the number of dimensions and small in the number of states per dimension. Another problem is that RL has classically considered agents that learn to solve only a single task but animals and humans are faced with multiple concurrent and in part highly unrelated tasks. Further complicating the solutions is that new tasks may become relevant over time. A related issue is how to address the availability of different types of reward such as rewards related to different tasks, or related to different types of learners, or related to intrinsic versus extrinsic goals.

Many of these issues with RL have been attacked in the past by trying to use additional structure present in the respective domain so as to somehow factor the problem. This idea has been proposed by several authors early on and has reappeared in many different settings (see, e.g., Dayan and Hinton 1992; Guestrin et al. 2003; Humphrys 1996; Kaelbling 1993; Kok and Vlassis 2004; Russell and Zimdars 2003; Sallans and Hinton 2004; Schneider et al. 1999; Singh and Cohn 1998; Sprague and Ballard 2003). One specific approach, that we pursue here too, is to start with learners that represent separate non-overlapping parts of the state space. The main idea is that there will be a large number of situations that can be handled by different sets of independent modules used concurrently (e.g., Humphrys 1996; Singh and Cohn 1998; Sprague and Ballard 2003). This requires the agent to have access to a collection of independent variables describing the state of the world. In the context of human visuomotor behavior, one navigation module requires a learner to represent the position of obstacles independently from a second module that represents the position of goal positions. Such individual modules can be obtained by solving individual problems separately (Singh and Cohn 1998), or in our case, by having the modules learn their action values when activated concurrently.

The idea of a modular organization of cognitive processes has appeared in the literature in many different variations (see e.g. Barrett and Kurzban 2006; Brooks 1986; Fodor 1983; Minsky 1988; Pinker 1999). Concrete and direct evidence for a modular organization of task solutions in the human brain conform with the modular RL framework used in the present chapter comes from experiments by Gershman et al. (2009). Human subjects made simultaneous decisions with their left and right hands and received separate rewards for each hand movement. Not only was the choice behavior better described by a modular learning model that decomposed the values of bimanual movements into separate values for each component but also blood oxygen level-dependent activity in reward learning related areas of the subjects' brains reflected specific values of modular RL models. This suggests that the human brain can use modular decompositions to solve RL problems allowing for factorization.

Having a collection of individual modular solvers for separate concurrent tasks requires to specify how the rewards from different sources are handled by the agent. In multi-criterion RL a learner tries to maximize the return not only for a single reward source but also for multiple separate rewards (Gábor et al. 1998; Mannor and Shimkin 2004; Natarajan and Tadepalli 2005). Here instead we treat extrinsic and intrinsic rewards as well as rewards from different sources together by assuming that a common currency exist for the comparison of many different rewards. Such a setting has been considered previously within RL by several authors (e.g., Russell and Zimdars 2003; Singh and Cohn 1998; Sprague and Ballard 2003). Empirical evidence for a single currency in biological systems is abundant (see Kable and Glimcher 2009 for a discussion).

But observing only a single composite reward also introduces a computational problem, which is a variant of the classical credit assignment problem. Individual learners have their respective state representations and all observe the global reward,

which we model here as the sum of the individual rewards. The goal is to find such sets of learners across different task combinations so that the global reward obtained is divided up correctly among the learners, given their respective state representations. This means that individual learners do not have access to the full state representation of all other learners. Again, in the context of human visuomotor behavior, the navigation learner avoiding obstacles may not represent how other learners approach goals or carry out hand movements within reaches. Thus, different active reinforcement learning modules have the problem of dividing reward up between them. A solution to this credit assignment problem for the modular case considered here has been previously given in Rothkopf (2008) and Rothkopf and Ballard (2010).

A second problem introduced by the use of individual modules stems from the fact that at different times, different sets of modules may be active within their respective tasks. One therefore needs to specify a module activation policy that attempts to pick a good set of modules for any particular task combination and an action selection process for active modules. In the past, Doya et al. (2002) proposed to use modules that consist each of a state predictor and a controller in an actor-critic-type architecture (Jacobs et al. 1991). After learning, the modules all contribute to the action of the system on the basis of how well each module is able to predict the next state of the world using each of their explicitly learnt models of the world transitions. Thus, the better an individual module is predicting the dynamics of the world the more it contributes to the composite action, irrespective of value or overall outcome. A different approach was used by Daw et al. (2005), who considered selecting a single action from different types of learners. In their system, the learner with the smallest uncertainty associated with its value prediction is selected. Thus, a single controller selects the next action alone based solely on the uncertainty of the current value estimates of a state so that each module needs to represent the same, full set of state variables. Here we propose to learn the selection of module combinations by the respective value of an action that these are predicting. Note that this is fundamentally different, as it allows arbitration of actions at the level of their expected returns, thereby avoiding conflicts that result from arbitration at the level of the actions. As learning of the task solutions by individual modules progresses, better and better estimates of the total discounted reward that each module expects will be available and therefore those modules that promise to give larger returns will be picked probabilistically more often whereas those modules that promise high returns that are not attainable will make progress in learning so. This finally leads to the scheduling of module combinations that obtain the highest combined reward and select individual actions jointly, as their value estimates are combined, and not their actions.

The modular framework that we propose naturally accommodates several desirable properties. First, it allows for continual learning (see, e.g., Ring 1994) as individual modules will learn to solve specific task components when they appear, and solutions learnt by other modules in previous tasks can be reused. This is valid under the assumption that the independent state representations are available to

the learners. Secondly, by learning which modules should participate in individual task combinations the system learns to coordinate the elemental task solutions learnt by the individual modules in order to achieve more complex goals in the composite tasks. By combining all rewards from different sources into a single internal currency, the respective contributions can be weighted by their expected contribution to the current goals and action selection can be carried out on the basis of the sum of the expected reward. Finally, the system is able to adjust to new external and internal reward structures, as it learns and combines the values of individual actions from the individual modules. As rewards change, the corresponding value functions are just rescaled (Von Neumann et al. 1947) instead of learning new task solutions or a new action arbitration.

We describe a solution to the credit assignment problem and the selection of appropriate learners that avoids previous restrictions by introducing additional structure in the form of a module activation protocol and by making the additional assumption that each module has access to the estimated sum of the reward estimates of other active modules. We derive formulas for estimates of reward that, assuming properties specified in the activation protocol, converge rapidly to their true values. Furthermore, it is shown that such learning can be implemented both in the setting where individual modules have independent action spaces as in Chang et al. (2004) and also in the single-agent case where individual modules' policies are combined to specify an action (Russell and Zimdars 2003; Sprague and Ballard 2003). We demonstrate that the algorithm can solve the credit assignment problem in variants of classical problems in the literature (Singh and Cohn 1998; Sutton and Barto 1998) as well as a more complex case of an avatar learning to navigate in a naturalistic virtual environment (Sprague et al. 2007).

2 Background

The problem setting is that of a Markov Decision Processes. An individual MDP consists of a 4-tuple (S, A, T, R) with S being the set of possible states, A the set of possible actions, T the transition model describing the probabilities $P(s_{t+1}|s_t, a_t)$ of reaching a state s_{t+1} when being in state s_t at time t and executing action a_t, and R is a reward model that describes the expected value of the reward r_t, which is distributed according to $P(r_t|s_t, a_t)$ and is associated with the transition from state s_t to some state s_{t+1} when executing action a_t.

Reinforcement learning attempts to find a policy π that maps from the set of states S to actions A so as to maximize the expected total discounted future reward through some form of learning. The dynamics of the environment T and the reward function R may not be known in advance. In the present case an explicit reward function R is learned from experience so that individual modules can learn from experience about their contributions to the global reward. The central goal in model free RL is to assign a value $V^{\pi}(s)$ to each state (s), which represents this

expected total discounted reward obtainable when starting from the particular state s and following the policy π thereafter:

$$V^{\pi}(s) = E^{\pi}\Big(\sum_{t=0}^{\infty} \gamma^t r_t\Big) \tag{1}$$

Alternatively, the values can be parametrized by state and action pairs, leading to the Q-values $Q^{\pi}(s,a)$. Where Q^* denotes the Q-value associated with the optimal policy π^*, the optimal achievable reward from a state s can be expressed as $V^*(s) = \max_a Q^*(s,a)$ and the Bellman optimality equations for the quality values can be formulated as:

$$Q^*(s,a) = \sum_r rP(r|s,a) + \gamma \sum_{s' \in S} P(s'|s,a) \max_{a'} Q^*(s',a') \tag{2}$$

Temporal difference learning (Sutton and Barto 1998) uses the error between the current estimated values of states and the observed reward to drive learning. Evidence for temporal difference learning in animals comes from a multitude of studies (e.g., Schultz et al. 1997). The values of state-action pairs can be updated by this error δ_Q using a learning rate α:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\delta_Q \tag{3}$$

Evidence for the representation of action values has also been found (e.g., Samejima et al. 2005). Two classical learning rules for the Q-values $Q(s,a)$ are (1) the original Q-learning rule (Watkins 1989) and (2) SARSA (Rummery and Niranjan 1994). While the Q-learning rule is an off-policy rule, i.e. it uses errors between current observations and estimates of the values for following an optimal policy, while actually following a potentially suboptimal policy during learning, SARSA is an on-policy learning rule, i.e. the updates of the state and action values reflect the current policy derived from these value estimates. The temporal difference for SARSA is accordingly:

$$\delta_Q = r_t + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t). \tag{4}$$

Empirical evidence for learning in animals consistent with the SARSA learning rule has similarly been found (Morris et al. 2006).

3 Multiple Modules as Individual Task Solutions

Within the composite state space both the relationship between the optimal value functions for each of the individual component tasks and the global task in which multiple objectives are pursued depend on the overall structure of the problem and can be very complex when considering the most general case in which

arbitrary dependencies between state transitions and between rewards exist. Our main simplifying assumption is to define a restricted venue where the required behavior can be realized with separate RL modules. The primary assumption is that such modules are available by virtue of having many modules with independent state variables so that they do not interfere with each other when activated in subsets (Chang et al. 2004; Russell and Zimdars 2003; Sprague and Ballard 2003). Of course in many interesting problems it may turn out that modules interfere with each other, but our assumption is that for the tasks at hand, independent state representations are available for the considered modules. This assumption shifts the problem of interacting modules to having independent state representations.

We will consider two separate but related cases. In the first setting, a collection of individual modules has separate and independent state spaces as well as separate and independent action spaces. In this case, the individual modules do not interact through state transitions or rewards. As an example, Chang et al. (2004) considered the case of ten individual learners working on ten separate and not interacting tasks in ten separate mazes but observing a single cumulative reward signal. The second case also uses the assumption of separate states with independent state transitions but all modules share the same action space. As an example, Singh and Cohn (1998) considered the case of a single agent moving in a grid world while foraging for different food types and avoiding a predator. While individual modules represented their states with respect to the different food types and the predator independently, the action of moving within the maze had to be coordinated within a single action space of moving in the grid world.

Let's first consider the case of separate action spaces. A module with its own actions can be defined as an MDP, i.e. the i-th module is given by

$$M^{(i)} = \{S^{(i)}, A^{(i)}, T^{(i)}, R^{(i)}\} \tag{5}$$

where the superscripts reflect that the information is referred to the i^{th} MDP. The states of the different modules are assumed to be non-overlapping. In such a case, the optimal value function is readily expressible in terms of the component value functions and the states and actions are fully factored so that there is no overlap and additionally the following two conditions hold:

$$P(s_{t+1}^{(1)}, \ldots, s_{t+1}^{(N)} | s_t^{(1)}, \ldots, s_t^{(N)}, a_t^{(1)}, \ldots, a_t^{(N)}) = \prod_{i=1}^{N} P(s_{t+1}^{(i)} | s_t^{(i)}, a_t^{(i)}) \tag{6}$$

$$P(r_t^{(1)}, \ldots, r_t^{(N)} | s_t^{(1)}, \ldots, s_t^{(N)}, a_t^{(1)}, \ldots, a_t^{(N)}) = \prod_{i=1}^{N} P(r_t^{(i)} | s_t^{(i)}, a_t^{(i)}) \tag{7}$$

Then these two conditions can be used together with Eq. (2) in order to arrive at the result:

$$Q(s_t^{(1)}, \ldots, s_t^{(N)}, a_t^{(1)}, \ldots, a_t^{(N)}) = \sum_{i=1}^{N} Q^{(i)}(s_t^{(i)}, a_t^{(i)}) \tag{8}$$

If Eqs. (6) and (7) hold and all the rewards are known, the action maximizing Eq. (8) can be selected and is guaranteed to be optimal. In this decomposed formulation, each module can follow its own policy $\pi^{(i)}$, mapping from the local states $s^{(i)}$ to the local actions $a^{(i)}$. This case is appropriate for the case of all modules having separate action spaces when each module can be identified with an agent that may be expected to act independently.

The second case, which is our focus, is that of a single agent pursuing multiple goals that are divided up between multiple independent modules that the agent can activate internally (Humphrys 1996; Karlsson 1997; Russell and Zimdars 2003; Singh and Cohn 1998; Sprague and Ballard 2003). The main problem specification that this constraint introduces is that the action space is shared such that $a^{(i)} = a$, for all i, so the i-th module is now:

$$M^{(i)} = \{S^{(i)}, A, T^{(i)}, R^{(i)}\} \tag{9}$$

An even simpler case arises, when all individual states $s^{(i)}$ correspond to the full state s, as, e.g., in Russell and Zimdars (2003). Although this obviously does not give any advantage in terms of reducing the dimensionality of the state space for individual modules, this case is interesting, because it is straightforward to prove the optimality of individual modules' values when using an on-policy learning algorithm such as SARSA.

How are actions selected when all modules share a single common action space? This case requires some form of action selection in order to mediate the competition between actions proposed by individual modules. While other modular RL solutions commonly employ action selection at the level of the actions selected by the modules (e.g., Daw et al. 2005; Doya et al. 2002) we have the modules interact at the level of their values. This has the distinct advantage that conflicts of modules proposing different actions can be resolved at the level of the expected total future rewards. We use the probabilistic softmax action selection with temperature τ:

$$P(a_t|s_t^{(1)}, \ldots, s_t^{(N)}) = \frac{\exp\left(\sum_{i=1}^{N} Q^{(i)}(s_t^{(i)}, a_t)/\tau\right)}{\sum_b \exp\left(\sum_{i=1}^{N} Q^{(i)}(s_t^{(i)}, b)/\tau\right)} \tag{10}$$

Once the action has been selected it is used for all modules. Note how the above equation combines the Q-values from all N active modules to probabilistically select an action that jointly promises high reward.

Finally we introduce the idea of a common single global reward that is obtained as result of the actions relative to the individual component state spaces of the contributing tasks. As an example, Sprague et al. (2007) considered the task of navigating along a walkway while avoiding obstacles and approaching targets. This task can be modeled as a composition of three individual component tasks, requiring one module for navigation, a second module for obstacle avoidance, and a third module for target approach. In this case, the global reward is the sum of the individual rewards obtained for each of the three component tasks.

Accordingly, the agent now needs to compute the credit for each module. That is, initially the individual rewards due to each module are not known, but only the global reward is observed by the agents at each time step. Here we assume for simplicity that the total, global reward G_t at some time t is the linear sum of all component MDPs' rewards:

$$G_t = \sum_{i \in \mathcal{M}} r_t^{(i)}. \tag{11}$$

Thus we can write:

$$M^{(i)} = \{S^{(i)}, A, T^{(i)}, G_{\mathcal{M}}\} \tag{12}$$

where the subscript $\mathcal{M}$ denotes that at each time step G is a function of the modules that are active.

4 Learning Module Activation

Our central assumption is that an overall complex problem can be factored into a small set of MDPs. Given a large set of modules with separate state representations it is now necessary to learn which modules to select in order to solve a specific task. In the following we assume that tasks are encapsulated within *episodes* of fixed length parameter Δ (See Fig. 1). During each episode, only a subset of the total module set is active. The guiding hypothesis is that in the timecourse of behavior, a certain set of goals is pursued and therefore the corresponding set of modules that are needed to achieve these goals become active and those that correspond to tasks that are not pursued become inactive (Sprague et al. 2007). During an episode the composition of a particular module set is assumed to not change. Given this constraint, the pivotal idea is that, within each episode, each active module can refine its own reward estimates by having access to the sum of the reward estimates of the other active modules. Thus, the set of active modules may change between episodes so that over time the actions taken direct the agent to different parts of the composite state space. The simplifying assumption of episodes allows reducing the problem of selecting modules to the beginning of an episode and also allows for a formal proof that the rewards of individual modules can be estimated correctly with the proposed credit assignment algorithm. As example, consider the previously mentioned problem of collecting food and avoiding a predator. Within an episode, only two food source may be present, requiring only the two associated foraging modules to be active. But within a later episode, the predator and the remaining food source may also be present, requiring all three food modules and the predator module to be active.

To model the factorization, let us suppose that in a substantial repertoire of N' modules, at the beginning of an episode it can be determined for one reason or

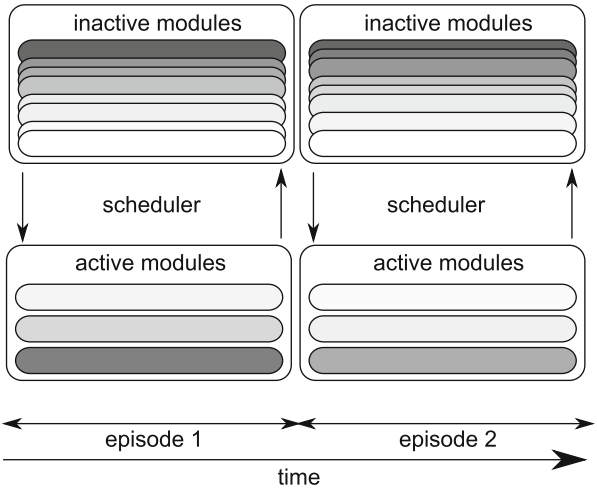

Fig. 1 The credit assignment algorithm exploits the constraint that in any period during behavior there is only a subset of the total module set that is active. We term these periods episodes. In the timecourse of behavior, modules that are needed become active and those that are no longer needed become inactive. The diagram depicts two sequential episodes of three modules each. The different modules are denoted with different shading. The *vertical arrows* denote the scheduler's action in activating and deactivating modules. On the top is the large library of possible modules. Our formal results only depend on each module being chosen sufficiently often and not on the details of the selection strategy. The same module may be selected in sequential episodes

another that only a smaller set N are sufficient for solving the task at hand where typically $N < N'$. So what is the best way to choose N modules from a possible N'? We select modules probabilistically according to the value of the initial state at the beginning of each episode using the softmax function:

$$P(M^{(i)}) = \frac{\mathrm{e}^{\frac{V^{(i)}(s^{(i)})}{\upsilon}}}{Z} \tag{13}$$

where $Z = \sum_{j=1}^{N'} \mathrm{e}^{\frac{V^{(j)}(s^{(j)})}{\upsilon}}$ is a normalizing term over the applicable modules, $V^{(i)}(s^{(i)})$, the value of the sate $s^{(i)}$ for module i is given by $V^{(i)}(s^{(i)}) = \arg\max_a \{Q^{(i)}(s^{(i)}, a)\}$, and υ is the common temperature parameter of the Boltzmann distribution that governs the stochasticity of module selection given the difference in values $V^{(i)}(s^{(i)})$. The idea is that although initially the value estimates of different modules will be unreliable, as the modules learn their respective rewards, the estimates will improve and direct the effort towards modules that can achieve the promised rewards. Note that this sampling strategy does not affect whether or not individual modules' value functions will converge. That is, a given learner of a component MDP $M^{(i)}$ working alone will converge towards its optimal value function as long as the space of states and actions $S^{(i)} \times A^{(i)}$ is sampled

infinitely often in the limit. The same criterion applies when it is working with other modules, although whether or not a module follows its optimal policy is a separate question. Independent agents may follow their own policy recommendations, but for modules activated with a single agent, a separate function that must be supplied to adjudicate among the possibly different action recommendations, as in Sprague et al. (2007), where the action selected was a maximum of reward weighted Q values.

In summary,

1. The overall behavior of modules (or agents in the multi-agent venue) is such that they work together in different subsets for a set time Δ so that the reward estimates can be learned;
2. The sum of the current estimates of the reward across all subset is accessible to each individual module in the subset at each moment by assumption;
3. The sampled subsets must span the module space because the reward calculations demand this.

The consequences of a module being activated are that:

1. It has used an associated procedure, such as a visual routine (Ballard et al. 1997; Ullman 1984), to compute the initial state the module is in. In our examples the state is computed at the beginning of an episode for all modules.
2. Its Q-values are included in the sum indicated in Eq. (10) used to select an action.
3. It influences the global reward that is received at every time step.

5 Credit Assignment with Modular Behaviors

Each active module represents some portion of the entire state space and contributes to the composite actions, but without some additional constraint they only have access to a global performance measure, defined according to Eq. (11) as the sum of the individual rewards collected by all of the $\mathcal{M}$ active modules at each time step to a global reward, i.e. $G_t = \sum_{i \in \mathcal{M}} r_t^{(i)}$. The problem that we tackle is how to learn the composite quality values $Q^{(i)}(s^{(i)}, a^{(i)})$ when only global rewards G_t are directly observed, but not the individual values $\{r_t^{(i)}\}$ (See Fig. 2), across many task combinations. The additional constraint that we introduced is that the system can use the sum of reward estimates from the modules that are co-active at any instant. This knowledge leads to the idea to use the different sets to estimate the difference between the total observed reward G_t and the sum of the current estimates of the individual rewards of the concurrently running behaviors. Credit assignment is achieved by bootstrapping these estimates over multiple task combinations, during which different subsets of behaviors are active. Dropping the temporal subscript for convenience, this reasoning can be formalized as requiring the individual behaviors to learn independent reward models $r^{(i)}(s^{(i)}, a^{(i)})$. The current reward estimate $\hat{r}^{(i)}$ for one particular behavior i, is obtained as

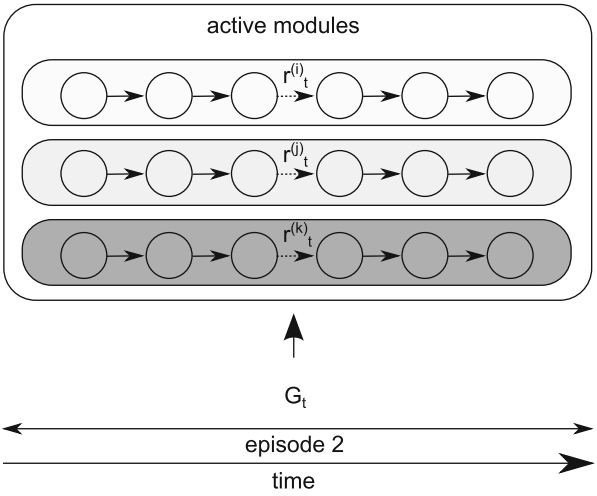

Fig. 2 The fundamental credit assignment problem for a biological agent using a modular architecture. When individual component learners transition between states represented by circles they carry out actions represented by arrows. At a particular instant shown with *dotted arrows*, when multiple modules are active and only a global reward signal G is available, the modules each has to be able to calculate how much of the rewards is due to their activation. Our setting simplifies the problem by assuming that individual reinforcement learning modules are independent and communicate only their estimates of their reward values. The modules can be activated and deactivated asynchronously across episodes

$$\hat{r}^{(i)} \leftarrow \hat{r}^{(i)} + \beta \delta_{r^{(i)}} \tag{14}$$

where the error on the reward estimates δ_r is calculated as the difference between the global reward and the sum of the component estimates:

$$\delta_{r^{(i)}} = G - \sum_{j \in \mathcal{M}} \hat{r}^{(j)} \tag{15}$$

so that Eq. (14) becomes:

$$\begin{aligned} \hat{r}^{(i)} &\leftarrow \hat{r}^{(i)} + \beta \Big(G - \sum_{j \in \mathcal{M}} \hat{r}^{(j)} \Big) \\ &= (1-\beta)\hat{r}^{(i)} + \beta \Big(G - \sum_{j \in \mathcal{M}, j \neq i} \hat{r}^{(j)} \Big) \end{aligned} \tag{16}$$

Together with the module activation protocol, described by Eq. (13) and Δ, Eq. (16) represents the core of our solution to the credit assignment problem. When one particular subset of tasks is pursued, each active behavior adjusts the current reward estimates $\hat{r}_i$ in the individual reward functions according to Eq. (16) at each time step. Over time, the set of tasks that have to be solved will change, resulting

in a different set of behaviors being active, so that a new adjustment is applied to the reward functions according to Eq. (16). This bootstrapping process therefore relies on the assertion that subsets of active behaviors visit all component behaviors. Intuitively, the global reward, which is the sum of all obtained rewards, should equal the sum of the reward estimates of all active modules. Each individual module therefore adjusts its reward estimate by combining its current reward estimate, i.e. what it believes it is contributing, and what the other active modules believe it is contributing, i.e. the difference between the total global reward and the sum of all other modules' reward estimates.

The component Q values for the state-action pairs of the individual behaviors are learned using the above estimates of the individual reward functions. Given the current reward estimates obtained through repeated application of Eq. (16), the SARSA algorithm is used to learn the component Q-functions:

$$Q_i(s_t^{(i)}, a_t^{(i)}) \leftarrow Q_i(s_t^{(i)}, a_t^{(i)}) + \alpha \delta_{Qi} \tag{17}$$

where δ_{Qi} now contains the estimates $\hat{r}_t^{(i)}$ and is given by:

$$\delta_{Q_i} = \hat{r}_t^{(i)} + \gamma Q_i(s_{t+1}^{(i)}, a_{t+1}^{(i)}) - Q_i(s_t^{(i)}, a_t^{(i)}) \tag{18}$$

Note that the usage of an on-policy learning rule such as SARSA is essential as noted in Sprague and Ballard (2003), because the arbitration process specified by Eq. (10) may select actions that are suboptimal for one or more of the modules. This has to be kept in mind when considering the optimality of the learnt solutions (cf. Russell and Zimdars 2003). A feature of the SARSA algorithm is that it makes use of suboptimal policy decisions during learning.

To investigate the dynamics of Eq. (16) we ask: in such a setting, given only global reward together with an arbitrary initial estimate of the rewards, will our algorithm for computing rewards converge?

5.1 Proof of Convergence

First of all, note that the equations for $\hat{r}$ are independent of the Q values and only depend on the sampling strategy. Next, note that, once the rewards have been determined, the MDPs will converge to their correct policies (Russell and Zimdars 2003) starting from any initial condition. While the rewards are being determined, the MDPs may or may not be converging but that is of no consequence to the proof. So it suffices to show that the reward calculation converges. To do this note that Eq. (16) can be rewritten as:

$$\hat{r}^{(i)} = (1 - \beta)\hat{r}^{(i)} - \beta \sum_{j \neq i} \hat{r}^{(j)} + \beta G. \tag{19}$$

This in turn can be written in vector form for a Jacobi iteration k by rewriting the reward estimates $\hat{r}^{(i)}$ at iteration k for all states and actions as a column vector $\mathbf{r}_k$, defining the column vector $\mathbf{G}$ with all entries being equal to the global reward G, and defining the error matrix B according to Eq. (19):

$$\mathbf{r}_{k+1} = B\mathbf{r}_k + \beta\mathbf{G}$$

Now write $\mathbf{r}_{k+1}$ as $\mathbf{r}_o + \mathbf{e}_{k+1}$ where $\mathbf{r}_o$ represents the correct values. Then

$$\mathbf{r}_o + \mathbf{e}_{k+1} = B(\mathbf{r}_o + \mathbf{e}_k) + \beta\mathbf{G}$$

but since $\mathbf{r}_o = B\mathbf{r}_o + \beta\mathbf{G}$ then

$$\mathbf{e}_{k+1} = B\mathbf{e}_k$$

This system will converge to 0 if the spectral radius of the matrix B forming any spanning set of these equations has a value $\rho < 1$. This is satisfied because its diagonal terms are all $1 - \beta$ and the off diagonal terms are either zero or $-\beta$. Thus the determinant is dominated by the trace for small β as the other factors all contain products of at least β^2. Convergence of such a system is therefore geometric provided (a) all the potential variables are included in the equations (b) any two variables do not always appear together, and (c) the diagonal terms in the error matrix B dominate the rest. The first two conditions are satisfied by the assumption in the module activation policy. The last requirement can be assured provided β is chosen appropriately.

5.2 *Incorporating Uncertainty in Reward Model*

One way to handle uncertainty in reward estimates is to model the sum of the other modules' rewards as noise as in Chang et al. (2004). This approach learns correct policies as long as the set of agents (or modules in the single-agent setting) remains constant, but it can introduce severe biases into the Q values. This makes it unusable for our setting where modules are used in different subsets in different episodes must have correct Q values for the equations to work.

Our module activation strategy of using modules in different combinations overcomes the Q value bias problem. When one particular subset of goals is pursued in any particular episode, the corresponding behaviors are active and the estimates of the respective rewards is updated according to Eq. (16) for all component behaviors. The sampling strategy assures that the equation set for the rewards has full rank.

However one can do better. During the computation, the modules' MDPs are typically in different states of completion and consequently have different levels of uncertainty in their reward estimates. This means that on a particular task combination, all component behaviors weight reward estimates in the same way,

independent of how well component behaviors have already estimated their share. Thus a drawback of any updating scheme that uses a fixed β value is that it is possible for a behavior to unlearn good reward estimates if it is combined with other behaviors whose reward estimates are far from their true values. Learning can be made much more efficient by considering the respective uncertainties in the estimates of the respective rewards separately. Thus one can have individual β_i values for each module reflect their corresponding reward estimates of uncertainty values.

Assuming that the between-module fluctuations are uncorrelated and follow a Gaussian distribution one can express the gain for each reward estimate in terms of the individual uncertainties in the respective reward estimates $(\sigma^{(i)})^2$:

$$\begin{aligned} \beta_i &= \frac{(\sigma^{(i)})^2}{\sum_{j=1}^{N} (\sigma^{(j)})^2} \\ &= \frac{(\sigma^{(i)})^2}{\sum_{j \neq i}^{N} (\sigma^{(j)})^2 + (\sigma^{(i)})^2} \end{aligned} \tag{20}$$

where the last term in the denominator is the variance in the observation noise. This can be seen as a straightforward approximation of the respective measurement uncertainties as in a cue integration setting. Since the factor for weighting the current estimate of a module's reward (See Eq. (16)) is $1 - \beta_i$, the effect is that relatively high-variance reward estimates will be discounted with respect to those of co-active modules. Thus, reward estimates for states that have been visited often and modules that have been used will tend to have lower uncertainties than reward estimates of states that have been visited rarely or modules that have not yet been used often.

6 Simulation Results

We demonstrate the algorithm on three separate problems that are chosen to illustrate different aspects of the proposed solution. The first problem uses ten individual modules that live in separate, non-interacting state spaces and carry out independent, non-interfering actions but observe a single common reward, which is the sum of all individual learners' contributions. The example is from Chang et al. (2004) and utilizes component tasks originally introduced by Sutton and Barto (1998). The second example is a predator–prey type problem and considers the case in which all component modules are within the same agent, i.e. where there is only a single action space which is shared among all individual learners. This problem was considered by Singh and Cohn (1998) and has been expanded here from its original version to include 15 different food sources and 5 predators to illustrate how to learn module selection over episodes. The third problem is a multi-tasking problem of an agent walking on a sidewalk while avoiding obstacles and picking up litter (Sprague and Ballard 2003). This problem illustrates the calculation of reward

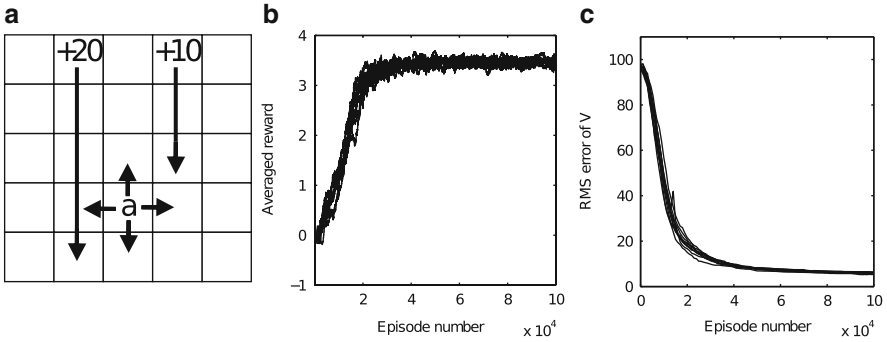

Fig. 3 Learning progress for the separate action spaces case following Chang et al. (2004). (**a**) Each agent is able to move in the four cardinal directions. Each transition results in a reward of 0 with the exception of the two states from where the *arrows* start. (**b**) The plot shows that each of the ten modules obtains the same average reward per iteration after learning of all tasks. (**c**) The plot shows that the RMS error between the optimal value functions and learned value functions decreases with learning

in a simulated three-dimensional world. For all these simulations, the RL learning parameter α was 0.1. The first set of experiments uses both constant β values from the set $\{0.01, 0.1, 0.2, 0.4, 0.8\}$ and the variance weighted β computed according to Eq. (20), i.e. it weights the reward estimates by their respective uncertainties. The remaining experiments use constant β values between 0.01 and 0.5 as indicated on the respective figures.

6.1 *The Computation of Accurate Q-Values with Separate Actions*

The first problem, drawn from Sutton and Barto (1998), allows testing the basic credit computation algorithm by considering multiple modules with independent action spaces, a case that is closely related to multi-agent problems. This results in a setting without the complicating factor of action selection that is required when modeling multiple active modules forming a single agent. With multiple modules, each learner is placed on a 5×5 grid and is able to move in four directions, i.e. North, East, South, and West (See Fig. 3).

A transition between positions results in a reward of 0. If a movement is made toward the walls of the grid, the agent obtains a reward of -1 and stays at the same location. There are two locations on the grid, which result in a transition to a new state for all selected actions. On one of these a reward of 20 units is obtained while in the second case a reward of 10 units is obtained. The problem is set up in such a way that the optimal policy successively collects the reward of 20 units, given a discount factor of 0.9. For the simulation, after each Δ of 30 iterations, a new subset of agents

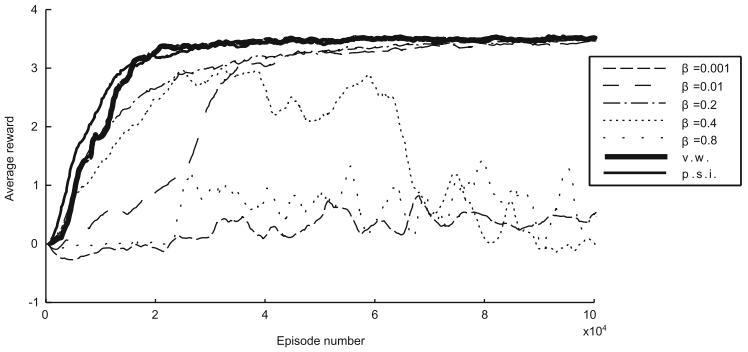

Fig. 4 Comparison of learning rate settings. Effect of different learning rates and the variance-weighted (v.w.) learning on the accumulated reward averaged over all modules in the Sutton and Barto problem in comparison with a Kalman Filter that has perfect state information (p.s.i.) about all component modules

of sizes between 3 and 7 were chosen randomly. The scheduler chose these subsets according to a uniform distribution over all modules. These modules were then run on the grid world and learned their respective Q-values with reward estimates that were updated according to Eq. (16). Part (c) of Fig. 3 shows that the agents learn to divide up the global reward and are therefore able to learn their respective policies. These results show that the basic algorithm for computing Q values works in this multi-agent setting.

While one can choose the value for β guided by the convergence theorem, a more informed possibility is to use Eq. (20) to weight the module's reward estimates by their inverse variances. To test this formula, we ran experiments to compare the weighting of reward estimates by their variance estimates with weightings using hand chosen β values. Figure 4 shows different learning rates compared to the variance-weighted result as well as a Kalman filter simulation with perfect information about the entire state space. The graph clearly shows the superiority of the variance-weighted estimation process compared to the fixed β values as well as the small loss incurred compared to the filtering agent in the considered problem.

6.2 Module Selection for Single Agent with Multiple Reward Sources

This problem is described in Singh and Cohn (1998) where the authors explore the use of multiple modules for a single task in a grid-world problem. These single-agent problems come closer to representing a problem that would have to be addressed by a biological agent since, unlike in the multi-agent problem of the previous section, the action space is shared by the modules.

In the original formulation, an agent moves on a 5×5 grid. Possible actions move the agent in the eight compass directions. Moves at the edge of the grid-world which would result in the agent leaving the grid result in the agent staying in the current position. The grid is populated by three food items and one predator. The picking up of a food item results in a reward of one unit and the repositioning of the food item to a new and empty location. The world is also populated by a predator, which moves every other time unit towards the current position of the agent. The agent obtains a reward of 0.5 units for every time step during which it does not collide with the predator. Each learner represents the position of the respective food item or predator, i.e. there are 625 states for each of the three food modules and for the predator module, where in the original problem a total of four learners were always active in order to solve a single task with four component goals.

Previously Singh and Cohn (1998) and Sprague and Ballard (2003) used this task in multi-goal learning but both studies used individual rewards that were delivered for each task as separate reward signals. In the current study the problem was made harder by making the reward that each behavior sees be the global sum of the individual rewards G. Furthermore, instead of using three food sources and one predator there are a total of 15 types of food sources and 5 types of predators. At the beginning of each episode, three food sources are selected randomly according to a uniform distribution over the total of 15 different food sources. Similarly, one predator is selected randomly from the pool of 5 different predators according to a uniform distribution, so that during every episode a total of three food sources and one predator are present, as in the original problem. But now the set of necessary modules may change for each episode, as each food source and each predator requires the corresponding module to be active. So now the agent has to learn which modules to select for an episode and each module has to learn its respective reward model by solving the credit assignment problem. The probabilistic module activation according to Eq. (13) was used to select four different modules at the beginning of each episode. The behaviors therefore also have to learn when they are best activated.

Simulations were run for different values of β and compared to a learner that instead of choosing the set of modules according to Eq. (13) selected learners at random with equal probability at the beginning of each episode. The learning rate for the reward model of this learner was set to an intermediate value of $\beta = 0.2$. The temperature T in Eq. (13) was changed from 3 to 0.01 over the course of learning. The rewards for all foods and predators were set to the values of the original problem by Singh and Cohn (1998). Figure 5 shows the average reward earned at each time step and the root mean square error between the true and learnt value functions. For intermediate learning rates of β between 0.01 and 0.2, the obtained reward approaches the maximum obtainable reward and the corresponding value functions approach the correct ones.

The learners are able to learn when they are best scheduled to run and the reward estimates become more accurate over trials according to Fig. 6, which demonstrates that for intermediate learning rates for β between 0.01 and 0.2, the reward estimates approach the true reward values as shown by the RMS error between all reward

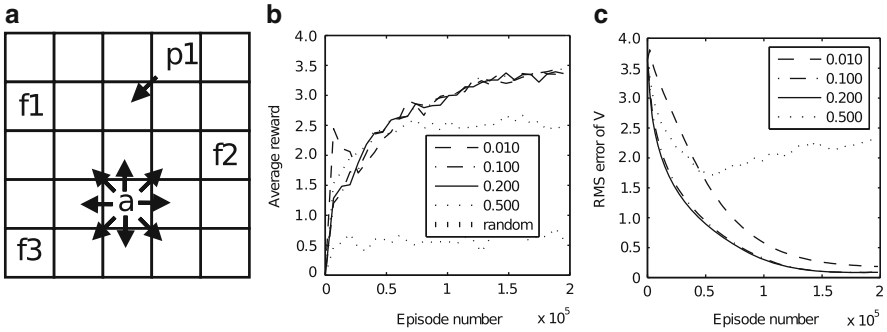

Fig. 5 Learning progress for the common action space case following Singh and Cohn (1998). (**a**) An agent is located on a 5 × 5 grid and searches to find three different food sources f1 to f3 and tries to avoid the predator p, which moves every other time step towards the agent. Simulations were run in which three food sources from a set of 15 and one predator from a set of 5 were selected. (**b**) Average reward collected using consumable rewards with different learning rates β in Eq. 16 and a randomly scheduled agent. (**c**) Root mean squared error between the true value functions and the learned value functions of all behaviors over trials

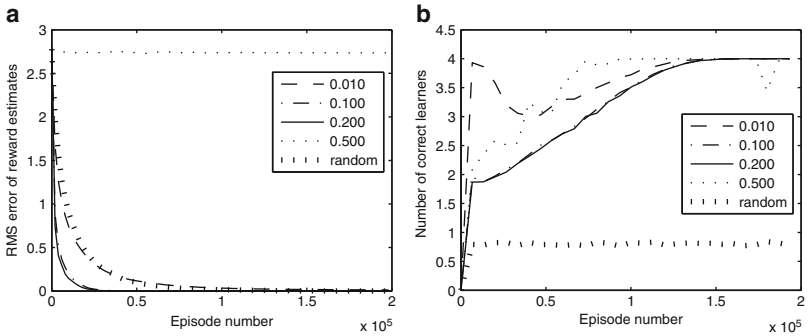

Fig. 6 Learning progress for the foraging task. (**a**) RMS error between the true rewards and the reward estimates of all behaviors over trials. The three curves correspond to different learning rates β in Eq. 16. (**b**) Number of correctly chosen learners on each episode for different learning rates β. As they are chosen in groups of four, choosing the right four modules is the best possible result

estimates and the true rewards. By contrast, a learner with a learning rate of $\beta = 0.5$ does not converge on the correct reward model over the course of the simulations. Accordingly, this learner does neither approach the correct value function nor approach the average reward collected by the other learners as shown in Fig. 5. The random learner's reward estimates improve over time as shown in Fig. 6, because the reward model can be learned using samples from the randomly selected states. But the average reward obtained per iteration cannot approach the level of the other learners, as inappropriate modules continue to be scheduled. Finally, Fig. 7 is a graphical depiction of the active set of modules over the total number of episodes. Black dots represent modules selected by the agent which are not appropriate for

Fig. 7 Learning progress of module selection. Every 10,000 iterations the four modules selected by the evaluation function are plotted using a key of *Black* = inappropriate, *White* = appropriate, and *Gray* = not activated. As the computations progress, the reward values $V(s)$ become increasingly accurate and cause the correct modules to be always selected

the respective episode's task combination, while white dots represent correctly selected modules and gray dots are inactive modules. The plot shows that the learner improves selection of modules right from the first episode and has learnt to schedule the correct subset of modules for the task at hand after 15,000 episodes.

6.3 Learning Walkway Navigation in a Virtual 3D Environment

This problem uses a humanoid agent navigating in a realistic three-dimensional virtual reality environment. The agent must use vision to extract features from the environment that define each module's state space. Also the agent's discrete state spaces must guide it successfully through the fine-grained environment. The walkway navigation task was first considered by Sprague et al. (2007) where a factorized solution was presented. However, that solution was obtained by delivering each of the individual learners their respective reward; that is, the agent received three separate rewards, one for the walkway following module, one for the obstacle avoidance module, and one for the litter picking up module. This problem was modified here with the additional constraint of only global reward being observed by all modules in each task combination. The global reward was always the sum of the rewards obtained by the individual modules according to Eq. (11).

The parameterization of the statespace is shown in Fig. 8. Each module represents the states with a two-dimensional vector consisting of a distance and an angle. For the picking up and the avoidance behaviors, these are the distance to the closest litter object and obstacle, respectively, and the signed angle between the current heading direction and the direction towards the object. The distance is scaled logarithmically similarly to the original setup by Sprague et al. (2007) and the resulting distance d_i is then discretized into $n = 21$ possible values between 0 and infinite distance. The angles within the field of view, i.e. with a magnitude smaller than 50 degrees are similarly discretized to 21 values. The exponential function was chosen such that the edge of an obstacle or target coincides with the edge between the first and second state and state 21 corresponds to all distances greater or equal to $4m$.

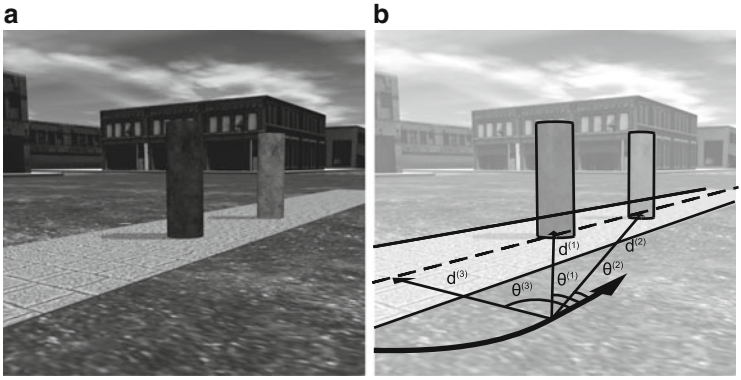

Fig. 8 The walkway navigation tasks. (**a**): typical view from the agent while navigating in the virtual environment. The three possible tasks are following the walkway, avoiding obstacles, which are the *dark cylinders*, and picking up litter, corresponding to the *light cylinders*. (**b**): Schematic representation of the statespace parameterization for the learners. See text for details

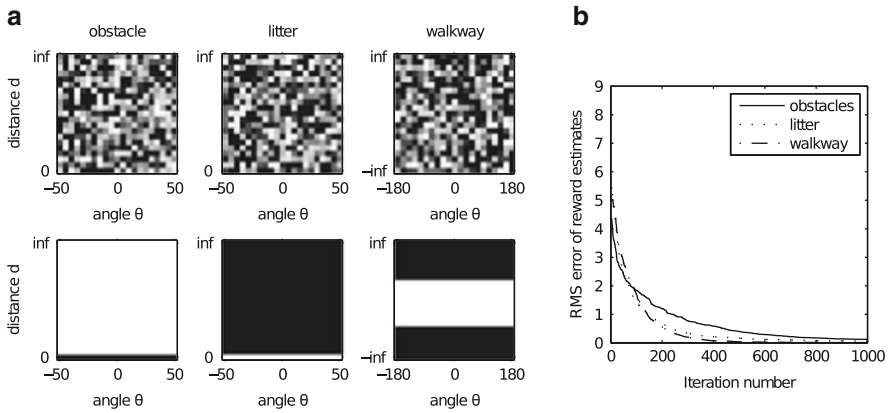

Fig. 9 Modular Reward estimates across learning. (**a**) Reward calculations for the walkway navigation task for the three component behaviors. *Top row*: Initial values. *Bottom row*: Final reward estimates. (**b**) Time course of learning reward for each of the three component behaviors. RMS error between true and calculated reward as a function of iteration number

The walkway statespace is also slightly different from Sprague et al. (2007) in that it represents all positions of the agent relative to the walkway for all possible walking directions. Finally, instead of three possible actions as in Sprague et al. (2007) the current simulations use five actions corresponding to steering at one of the five angles $\{-15, -7.5, 0, 7.5, 15\}$ with additive Gaussian noise of variance $\sigma = 1$. The reward values displayed as a function of the state space locations are shown in Fig. 9. Staring from random values and receiving only global reward at each step, the agent's modules are able to arrive at the true reward values.

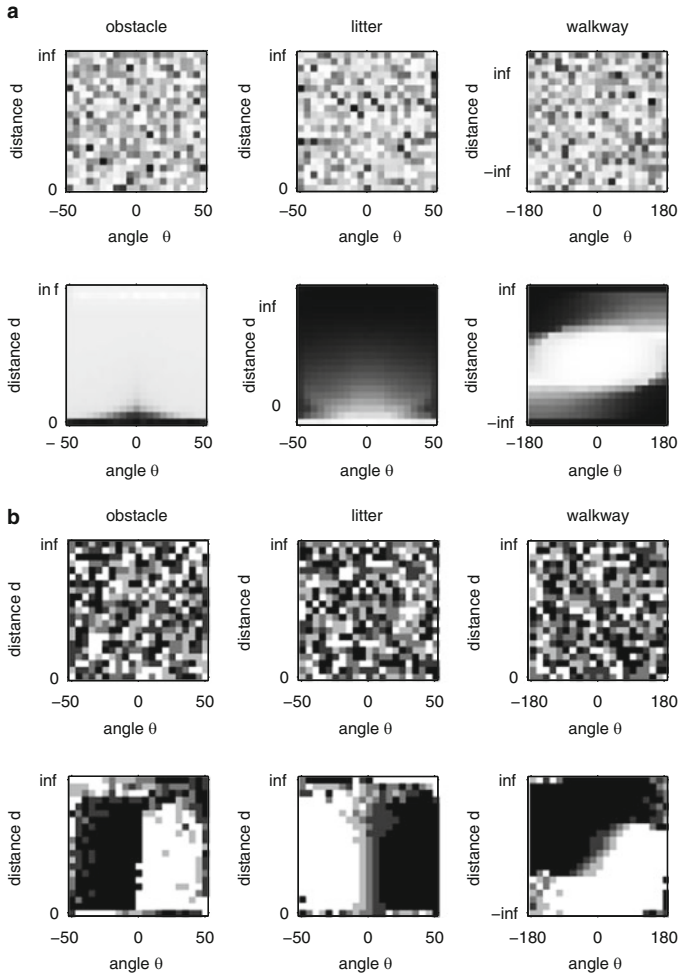

Fig. 10 Values and policies for navigation modules across learning. (**a**) Representations of value functions in the walkway navigation task for the three component behaviors across learning. *Top row*: initial value functions $V^{(i)}(s)$ at iteration 1. *Bottom row*: value functions $V(s)$ after 20,000 episodes. (**b**) Representations of policies in the walkway navigation task for the three component behaviors across learning. *Top row*: policies $\pi^{(i)}$ in episode 1. *Bottom row*: policy estimates. Key: For Value functions, light values are high numbers; for policies light numbers are left turns and darker numbers are right turns

The accuracy of these estimates is shown in Fig. 9. Thus the individual learners are able to learn their correct reward model for their respective tasks.

The value functions and policies of these simulations are shown in Fig. 10, at both the first iteration with random initial values and after learning, when the agent has walked the walkway for 1,000 episodes. As can be seen from the representation of the reward estimates, the individual behaviors have learned the true rewards of

their respective tasks, where not intersecting with an obstacle results in a reward of one unit, intersecting a litter object gives four units of reward, and staying on the walkway results in a reward of 0.8 units.

7 Discussion and Conclusions

Natural behavior consists of extended sequential perception and action with multiple concurrent and changing goals. Solving such tasks can be accomplished by some form of a decomposition such that elemental task solutions from a repertoire of behaviors can be reused and used in combinations. Here a specific modular architecture was considered in which separate independent state representations are assumed to be available for each of a large number of behavioral modules. These modules can be combined to solve tasks and task combinations but they only observe a single global reward value. The goal of the system is to learn about each individual component task's rewards and values and to learn how to schedule combinations of these elemental task solutions in order to achieve good performance on the composite tasks.

The primary contribution of this chapter is to describe a way of learning to activate learners with independent state representations so that they can jointly solve a composite control task. The presented solution is based on a credit assignment computation that enables individual modules to learn their correct reward functions and correct value functions from executing different task combinations while observing only global instead of individual reward. The key constraint, motivated by the need for a system that would potentially scale to a large library of behaviors, is to assume that the overall system can be structured such that it could achieve its goals by using only a subset of its behavioral repertoire at any instant and that the reward gained by this subset is the total of that earned by its component behaviors. The use of modules allows the rewards obtained by reinforcement to be estimated online.

By using an on-policy learning method, it is ensured that the learnt action-value functions are correct estimates of the total discounted future reward of each individual module so that the actions selected by an individual module involved in solving only its respective tasks will be optimal. In task combinations, if the summation of Q-values holds, then the composite tasks will be solvable by some subset of the provided learners and these will learn their respective contributions to the sum of rewards. As action selection is based on combining the estimated action-values and not the action themselves, the selected actions will be optimal with respect to the composite task. If additional learners are available, whose actions, given their state representations, do not contribute to any reward, these will be selected less over the time course of learning. Thus, the system can learn that only a small subset of learners may actually contribute to reward in different task combinations.

The present work is related to other approaches that start out with compositional solutions to individual problems and then devise methods in order to combine a large number of such elemental solutions, e.g. Singh and Cohn (1998) and Meuleau et al. (1998). Both approaches are concerned with learning solutions to large MDPs by utilizing solutions or partial solutions to smaller component MDPs. In Meuleau et al. (1998) the solutions to such components are heuristically combined to find an approximate solution to the composite MDP by exploiting assumptions on the structure of the joint action space. A way of learning a composite MDP from individual component MDPs by merging has been described in Singh and Cohn (1998). First the policies for elemental component MDPs are learned by separate modules. In their formulation, when combined, the different modules share a common action space, i.e. one single action is chosen in the composite problem. This is often used as the criterion to distinguish between single agent modular and multiagent problems. Because of this selection of a single action, each individual action is not necessarily optimal for all the component MDPs. Given that the component value function is not necessarily obtained as a linear sum of the component policies, which are each representing different reward types without a common currency, the authors propose a way of initializing the value iteration process of the composite problem using bounds on the state values derived from the state values of the individual component MDPs. In our venue small numbers of behaviors that are appropriate for the current situation are selected in an online fashion. In this situation it is essential to get the Q-values right. An algorithm that models other modules' contribution as pure noise (Chang et al. 2004) will compute the correct policy when all the behaviors or agents are active but this result will not extend to active subsets of modules and behaviors, as incorrect Q values, when used in subsets, will cause suboptimal behavior.

Many open problems remain to be considered at the computational and algorithmic levels but also in relating the proposed computations to biological systems. It would be desirable to learn the separate state representations that are required by the individual modules, i.e. to learn the underlying factorizations. Which state variables are independent and how can sensory variables be learned for individual modules? Here we considered having a large collection of individual learners with different state representations from the start and presented a way of learning to select the appropriate ones for episodes with the respective task combinations. But the scheduling algorithm may quickly get overwhelmed if the number of available modules is very large. Another central issue relates to the factorization assumption itself. There may be problems that do not allow for such factorizations simply because of the underlying interactions of state variables. Recently, Toutounji et al. (2011) showed how the credit assignment algorithm used in this chapter can be used also for the case of interacting modules. In this case, a hierarchy of modules can learn, adjusting the contributions from low level modules that treat a problem as factorizable although the state transitions interact. By learning to correctly divide up the observed total reward, higher level modules adjust the contributions from the lower level modules to achieve the overall task.

Another central open problem in the context of multiple modules used in the present chapter relates to the balancing of exploration and exploitation. As individual modules have different state representations and may have different interaction histories because they can be scheduled independently, a particular action that may be exploiting for one module may be exploratory for another module. In general, in a model-based RL framework, in which the uncertainties about the rewards and transitions are made explicit, optimal balance between exploration and exploitation can be achieved naturally by acting optimally with respect to the expected total discounted future rewards, i.e. the current belief over the underlying MDPs. Thus, as these modules may be reused during later episodes, the trade-off between exploration and exploitation can be understood as a balance between attaining the objectives in the current episode and the reduction of uncertainties about the component tasks, as they may become valuable in later episodes. Albeit computationally intractable in the general case, this is a principled way of expressing the resulting tendency of an agent to carry out exploratory actions, if these promise a higher reward in the long run, a behavior that can be equated with curiosity.

Acknowledgements The research reported herein was supported by NIH Grants RR009283 and MH060624. C. R. was additionally supported by EC MEXT-project PLICON and by the EU-Project IM-CLeVeR, FP7- ICT-IP-231722.

References

Ballard, D. H., Hayhoe, M. M., Pelz, J. (1995). Memory representations in natural tasks. *Journal of Cognitive Neuroscience, 7*(1), 68–82.

Ballard, D. H., Hayhoe, M. M., Pook, P., Rao, R. P. N. R. (1997). Deictic codes for the embodiment of cognition. *Behavioral and Brain Sciences, 20*, 723–767.

Barrett, H., & Kurzban, R. (2006). Modularity in cognition: framing the debate. *Psychological Review; Psychological Review, 113*(3), 628.

Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation, 2*(1).

Chang, Y.-H., Ho, T., Kaelbling, L. P. (2004). All learning is local: multi-agent learning in global reward games. In S. Thrun, L. Saul, B. Schölkopf (Eds.), *Advances in neural information processing systems 16*. Cambridge: MIT.

Daw, N., & Doya, K. (2006). The computational neurobiology of learning and reward. *Current opinion in Neurobiology, 16*(2), 199–204.

Daw, N. D., Niv, Y., Dayan, P. (2005). Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature Neuroscience, 8*(12), 1704–1711.

Dayan, P., & Hinton, G. E. (1992). Feudal reinforcement learning. In *Advances in neural information processing systems 5* (pp. 271–271). Los Altos: Morgan Kaufmann Publishers, Inc.

Doya, K., Samejima, K., Katagiri, K.-I., Kawato, M. (2002). Multiple model-based reinforcement learning. *Neural Computation, 14*(6), 1347–1369.

Fodor, J. A. (1983). *Modularity of Mind*. Cambridge: MIT.

Gábor, Z., Kalmár, Z., Szepesvári, C. (1998). Multi-criteria reinforcement learning. In *Proceedings of the fifteenth international conference on machine learning* (pp. 197–205). Los Altos: Morgan Kaufmann Publishers Inc.

Gershman, S., Pesaran, B., Daw, N. (2009). Human reinforcement learning subdivides structured action spaces by learning effector-specific values. *The Journal of Neuroscience, 29*(43), 13524–13531.

Guestrin, C., Koller, D., Parr, R., Venkataraman, S. (2003). Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research, 19*, 399–468.

Humphrys, M. (1996). Action selection methods using reinforcement learning. In P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, S. W. Wilson (Eds.), *From animals to animats 4: proceedings of the fourth international conference on simulation of adaptive behavior* (pp. 135–144). Cambridge: MIT, Bradford Books.

Jacobs, R., Jordan, M., Nowlan, S., Hinton, G. (1991). Adaptive mixtures of local experts. *Neural Computation, 3*(1), 79–87.

Kable, J., & Glimcher, P. (2009). The neurobiology of decision: consensus and controversy. *Neuron, 63*(6), 733–745.

Kaelbling, L. P. (1993). Hierarchical learning in stochastic domains: Preliminary results. In *Proceedings of the tenth international conference on machine learning* (vol. 951, pp. 167–173). Los Altos: Morgan Kaufmann.

Karlsson, J. (1997). *Learning to solve multiple goals*. PhD thesis, University of Rochester.

Kok, J. R., & Vlassis, N. (2004). Sparse cooperative q-learning. In *Proceedings of the international conference on machine learning* (pp. 481–488). New York: ACM.

Land, M. F., & McLeod, P. (2000). From eye movements to actions: how batsmen hit the ball. *Nature Neuroscience, 3*, 1340–1345.

Mannor, S., & Shimkin, N. (2004). A geometric approach to multi-criterion reinforcement learning. *The Journal of Machine Learning Research, 5*, 325–360.

Meuleau, N., Hauskrecht, M., Kim, K.-E., Peshkin, L., Kaelbling, L., Dean, T., Boutilier, C. (1998). Solving very large weakly coupled markov decision processes. In *AAAI/IAAI* (pp. 165–172). Menlo Park: AAAI Press.

Minsky, M. (1988). *The society of mind*. New York: Simon and Schuster.

Morris, G., Nevet, A., Arkadir, D., Vaadia, E., Bergman, H. (2006). Midbrain dopamine neurons encode decisions for future action. *Nature Neuroscience, 9*(8), 1057–1063.

Natarajan, S., & Tadepalli, P. (2005). Dynamic preferences in multi-criteria reinforcement learning. In *Proceedings of the 22nd international conference on machine learning* (pp. 601–608). New York: ACM.

Pinker, S. (1999). How the mind works. *Annals of the New York Academy of Sciences, 882*(1), 119–127.

Ring, M. B. (1994). *Continual learning in reinforcement environments*. PhD thesis, University of Texas at Austin.

Rothkopf, C. A. (2008). *Modular models of task based visually guided behavior*. PhD thesis, Department of Brain and Cognitive Sciences, Department of Computer Science, University of Rochester.

Rothkopf, C. A., & Ballard, D. H. (2010). Credit assignment in multiple goal embodied visuomotor behavior. *Frontiers in Psychology, 1*, Special Issue on Embodied Cognition(00173).

Rummery, G. A., & Niranjan, M. (1994). On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University Engineering Department.

Russell, S., & Zimdars, A. L. (2003). Q-decomposition for reinforcement learning agents. In *Proceedings of the international conference on machine learning* (vol. 20, p. 656). Menlo Park: AAAI Press.

Sallans, B., & Hinton, G. E. (2004). Reinforcement learning with factored states and actions. *Journal of Machine Learning Research, 5*, 1063–1088.

Samejima, K., Ueda, Y., Doya, K., Kimura, M. (2005). Representation of action-specific reward values in the striatum. *Science, 310*(5752), 1337.

Schneider, J., Wong, W.-K., Moore, A., Riedmiller, M. (1999). Distributed value functions. In *Proceedings of the 16th international conference on machine learning* (pp. 371–378). San Francisco: Morgan Kaufmann.

Schultz, W., Dayan, P., Montague, P. (1997). A neural substrate of prediction and reward. *Science, 275*, 1593–1599.

Singh, S., & Cohn, D. (1998). How to dynamically merge markov decision processes. In *Neural information processing systems 10* (pp. 1057–1063). Cambridge: The MIT Press.

Sprague, N., & Ballard, D. (2003). Multiple-goal reinforcement learning with modular sarsa(0). In *International joint conference on artificial intelligence* (pp. 1445–1447). Morgan Kaufmann: Acapulco.

Sprague, N., Ballard, D., Robinson, A. (2007). Modeling embodied visual behaviors. *ACM Transactions on Applied Perception, 4*(2), 11.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: an introduction*. Cambridge: MIT.

Toutounji, H., Rothkopf, C. A., Triesch, J. (2011). Scalable reinforcement learning through hierarchical decompositions for weakly-coupled problems. In *2011 IEEE 10th international conference on development and learning (ICDL)* (Vol. 2, pp. 1–7). New York: IEEE.

Ullman, S. (1984). Visual routines. *Cognition, 18*, 97–157.

Von Neumann, J., Morgenstern, O., Rubinstein, A., Kuhn, H. (1947). *Theory of games and economic behavior*. Princeton: Princeton University Press.

Watkins, C. J. (1989). *Learning from delayed rewards*. PhD thesis, University of Cambridge.

Yarbus, A. (1967). *Eye movements and vision*. New York: Plenum Press.

Part II
Hierarchical Organization of Animal Behavior

Modular, Multimodal Arm Control Models

Stephan Ehrenfeld, Oliver Herbort, and Martin V. Butz

Abstract Human and animal behavior can be amazingly flexible and adaptive. Even when only considering the dexterity of human arm movements, a rather complex control architecture appears necessary. This control architecture faces three particular challenges, which we discuss in detail. First, sensory redundancy requires the flexible consideration, combination, and integration of different sources of information about the state of the arm and the surrounding environment. Second, motor redundancy requires the flexible consideration and resolution of behavioral alternatives. Third, the continuous uncertainty about body and environment requires flexible control strategies that take these uncertainties into account. Research in cognitive modeling as well as in psychology and neuroscience suggests that the human control system effectively solves and even partially exploits these challenges to generate the observable dexterity. Besides theoretical considerations from control and cognitive modeling perspectives, we survey the capabilities and current drawbacks of the sensorimotor redundancy resolving architecture (SURE_REACH) of human arm reaching. Moreover, we consider an even more modular model of human motor control, which is currently being developed. Both architectures can yield the dexterous behavioral control observable in humans, but only the latter scales to many degrees of freedom. Thus, the architectures may provide insights on how dexterous motor control is realized in humans and on how more adaptive and flexible robot control systems may be developed in the future.

S. Ehrenfeld · M.V. Butz (✉)
Cognitive Modeling, University of Tübingen, Germany
e-mail: stephan.ehrenfeld@uni-tuebingen.de; martin.butz@uni-tuebingen.de

O. Herbort
Department of Psychology, University of Würzburg, Germany
e-mail: oliver.herbort@psychologie.uni-wuerzburg.de

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_7,

1 Introduction

Humans surpass other species in various aspects, such as planning ahead, communicating, creating art, and constructing and utilizing tools skillfully. To accomplish this, the human cognitive and motor control system needs to be highly flexible and adaptive. Bernstein called this human property *dexterity* (Latash and Turvey 1996), that is, the capability of deciding and executing different goal-directed behaviors in different circumstances, considering different priorities and goals. All the dexterous skills that make humans unique, however, cannot be realized without the seemingly mundane capability of controlling the own body. For example, to communicate an idea—however abstract or philosophical—one has to control the muscles of the vocal cords to speak the idea or one has to move the hand and fingers to type it in a word processor. Even daily activities, such as eating or transporting a glass of water to the mouth, require precise and dexterous control of the human body. Thus, understanding human body control is crucial to understanding human behavior.

In this chapter, we focus on three fundamental challenges that the brain faces when deciding on and executing flexible, skillful, goal-directed movements. First, since the sensory and motor systems encode information about body and environment in various frames of reference, sensory feedback as well as current goals and other priorities are inevitably represented in different modalities. These need to be integrated and exchanged to plan and control dexterous movements. We refer to this problem as the *sensory redundancy problem*.

The second challenge results from the fact that goals can be reached in various ways and with various motor actions. This problem is referred to as the *motor redundancy problem* (Bernstein 1967). Motor redundancy has to be quickly, effectively, and flexibly resolved on the fly to be able to achieve goals in a dexterous way. Sensory and motor redundancy problems are somewhat complementary and are often strongly interdependent.

The third challenge arises from the fact that uncertainty is ubiquitous. The body changes continuously due to muscle fatigue, cloths, growth, injury, etc. Also the environment is continuously in flux and only partially observable. Other agents and physical influences may change the state of the environment at any time and these changes are often overlooked, may be unobservable, or simply stay unnoticed. As a consequence, the sensory and the motor system of any biological system cannot represent the state of body and environment in an absolutely precise manner. Thus, our brain faces an *uncertainty problem*.

Human everyday behavior shows that we do not only solve these problems—mostly without even noticing them—but we are also able to adjust our behavior flexibly to different situations and circumstances. To accomplish this, the brain appears to use an adaptive, highly modular *body schema* (Hoffmann et al. 2010), which integrates and exchanges body information context—dependently across sensory—and motor-grounded frames of reference. The body schema not only enables the flexible exchange of information as well as the combination of goals and

other priorities, but it also constrains these exchanges and combinations by taking the bodily architecture—such as kinematics, dynamics, and their body-dependent interactions—into account. For example, the brain appears to exploit the fact that the location and orientation of "outer" limbs, such as the hand, depends on respective "inner" limbs, such as the shoulder and arm. These constraints are then taken into account for (a) body schema maintenance and for (b) body-schema-dependent decision making and control. Due to the body schema-based constraints, the brain can perceive and avoid obstacles and also somewhat categorize the reachability of regions surrounding the body (Caggiano et al. 2009).

The control system exploits the modular body schema representation for the flexible integration and consideration of various sensory and motor sources of information about body and environment. Depending on the task and the circumstances at hand, one or the other source may dominate current behavioral decision making and control. Moreover, the body schema allows accounting for noise in an effective manner—integrating noisy information, ignoring false, and substituting missing sensory information in an approximately optimal way based on statistical decision theory (Herbort et al. 2007; Trommershäuser et al. 2003).

In this chapter, we first detail the computational problems due to the challenges of sensor redundancy, motor redundancy, and uncertainty. Next, we discuss implications for a control architecture that can deal with or even exploit these challenges. Finally, we detail our efforts in modeling human arm reaching with the SURE_REACH model and a more recent modular modality frame model, which further modularizes SURE_REACH (Butz et al. 2007; Ehrenfeld and Butz 2011; Herbort et al. 2010). A final discussion concludes the chapter.

2 Computational Problems in Motor Control

Although motor acts such as reaching and grasping for objects appear introspectively simple and usually do not require much thought or attention, computationally a number of complex, ill-defined problems have to be solved each time an interaction unfolds. This section describes three facets of motor control that need to be addressed: sensory redundancy, motor redundancy, and uncertainty. Subjectively, these aspects may not be considered "problems," because humans cope with these aspects with ease in many situations. Moreover, sensory and motor redundancy are crucial for the flexible, dexterous, and robust control of one's own body. Nevertheless, to be able to benefit from redundancy and cope with uncertainty, rather complex computational problems have to be solved.

2.1 Sensory Redundancy

Humans use different sensory modalities to gather information about the world and their own body. For example, the arm can be perceived by means of proprioceptive

and visual as well as sometimes by auditory and tactile sensors. Grounded in the different sensory modalities, the arm is thus co-represented in these modalities and the information has to be exchanged and combined across modalities. Since the sensory capabilities in each modality also determine particular frames of reference, information transfer has to be realized by incorporating transfer functions. For example, proprioceptive signals generally allow the determination of the visual perception of arm orientation, given body schema knowledge. To integrate or compare the information body-state dependently, the reliability of each sensor has to be estimated, transfer functions need to be available, and state information as well as reliability estimates have to be transferable.

Computationally, two challenges for motor behavior may be distinguished as resulting from the multimodal, redundant, but highly interactive representations of body and environment. First, the information of different modalities needs to be effectively exchanged and suitably combined to facilitate motor control. Second, goal representations may be embedded in different modalities or may focus on particular modalities, so that goals and constraints in different modalities need to be considered and properly combined to generate maximally dexterous behavior.

2.1.1 Multi-modal Information

For an effective interaction with the environment, the body state and the surrounding environment have to be suitably estimated. For proper state estimations, typical information-theoretic and control-theoretic techniques can be applied, such as Kalman filtering or Bayesian information integration (Doya et al. 2007; Knill and Pouget 2004; Welch and Bishop 1995; Wolpert and Ghahramani 2000). To do so, the variances of different sources of sensory information need to be estimated and the respective sensory information needs to be exchanged with respect to the variance estimates. For example, in darkness the role of visual feedback for the estimation of the body state will be reduced. However, since each sensory modality is encoded in particular frames of reference, information integration and combination requires the transfer of variance estimates across different frames of reference. Whether the brain integrates the available information or only combines it for motor control decisions is still under investigation (Serwe et al. 2009).

During action execution, the ongoing movement has to be monitored and corrected if necessary. To be able to do so, an accurate representation of the state of the body has to be maintained *over time*. In this case, *delayed* sensory information needs to be integrated, thus requiring (a) the generation of expected bodily and consequent sensory states for online control as well as (b) the maintenance of expectations of the delayed sensory feedback for information integration, and, particularly, for movement error monitoring. Transformations from sensory to motor representations for control and vice versa for predicting sensory consequences have to be available (Pouget and Snyder 2000). Since the feedback about body state and body motion is again encoded in various modalities and is exposed to different delay

durations, the most effective sensory information integration can be expected to occur modality-respectively (Körding et al. 2004; Körding and Wolpert 2004).

While the above-mentioned aspects and facts pose challenges to a control system, multi-modal, redundant sensory information can also be viewed as enabling the integration of various information sources in the first place. The redundant information allows the maintenance of a more faithful representation of the current body state. As multiple channels provide comparable information, movements can still be controlled if one modality is not available or if it sends faulty information. Thus, a sensor-grounded, multimodal body schema representation may enable a highly versatile body state encoding and facilitate dexterous goal-directed planning and body control.

2.1.2 Multi-modal Goals

Motor control can be understood as the process of translating a desired perceptual state (or goal) into a sequence of motor commands, which realize this state (Elsner and Hommel 2001; Greenwald 1970; Hoffmann and Möller 2003; Hoffmann 1993). In many cases, a goal may be defined in a single modality. For example, when reaching out to press a light switch, the desired position of the hand is mostly defined by the visually perceived position of the light switch.

However, frequently goals comprise elements that are represented in different modalities. A simple example is pointing to a light switch (e.g., to instruct another person to switch the light off). In this case, the arm has to be directed toward the switch, which is perceived visually. At the same time, it is necessary to align upper arm, forearm, hand, and the index finger in order to enable another person to recognize the movement as a pointing gesture.

Another example is the execution of sequential actions, such as grasping and rotating an object. In such cases, it is frequently observed that humans grasp the object with a posture that enables ending the object rotation with a rather comfortable neutral position (Herbort and Butz 2010, 2012; Rosenbaum et al. 1990, 1996). When the grasping segment of such a movement sequence is planned, two constraints need to be integrated. First, the to-be-rotated object is visually defined in extrinsic space. Second, the desired orientation of the forearm when grasping the object is defined proprioceptively, based on intended future interactions with the object. Likewise, the anticipatory adjustments of the arm postures that are assumed in intermediate targets in two-step pointing sequences can then be understood as the integration of visually and proprioceptively defined constraints (Fischer et al. 1997; Herbort and Butz 2007). In sum, for dexterous interactions with the environment, with people, and with objects, it is necessary to integrate desired body states across modalities. As for the case of multi-modal information, this requires the ability to effectively and accurately match sensory states that are encoded in different modalities.

For decision making, also payoff needs to be considered and integrated. This may be accomplished using Bayesian decision theory, which is approximated

by the brain in various circumstances (Körding and Wolpert 2006; Trommershäuser et al. 2003). Due to the multi-modal nature of information and goal representations, however, decision making may need to be distributed across modalities, or an internal, common modality may serve as the mediator for decision making. When multi-modal goals are strived for, a lot of computational effort may be necessary to exchange the respective priorities effectively for dexterous decision making and motor control. However, again the multi-modal representation may be viewed as a feature that actually facilitates dexterity, as long as the necessary information and priority exchanges are handled properly.

2.2 Motor Redundancy

Many different movements lead to the same sensory change due to *motor redundancy*. For example, a light switch can be pushed with various effectors, different final arm postures, and different movement trajectories. On the one hand, motor redundancy makes motor learning and planning more difficult because one-to-many-mappings have to be acquired and redundancy has to be resolved during motor planning. On the other hand, motor redundancy makes our actions flexible. In the face of changing environments, altered joint characteristics or obstacles, motor redundancy offers many movement options.

In many situations, our goals are encoded in a low-dimensional space whereas the space of possible actions is high-dimensional. An example is moving the hand to a particular location in 3D space. Even if we constrain possible actions by specifying the to-be-used hand and by trying to only use the arm for the movement (while keeping the rest of the body motionless), the postural redundancy due to the seven degrees of freedom of the human arm enables reaching the goal with many different end-postures and on many different trajectories to the respective end-postures. Thus, in almost all cases, a goal does not specify all parameters of an action.

Even though many movements could be executed in many different ways, humans rely on a small subset of movements in unconstrained tasks. This suggests that free parameters of movements are not determined randomly but are selected according to specific criteria, such as movement costs (Engelbrecht 2001; Flash and Hogan 1985) and by habit and bodily-determined motor synergies (Herbort and Butz 2011; Latash et al. 2007). The additional criteria can be derived from integrating additional aspects of the (external) task in the motor plan and by applying internal ("default") planning criteria. For example, redundancy with respect to how an object is grasped can be resolved by integrating forthcoming actions into the movement plan (Herbort and Butz 2012). Intrinsic constraints can be imposed by, for example, selecting a movement velocity profile that reduces end-point variance (Harris and Wolpert 1998). Likewise, the postures assumed at the end of a movement are selected to reduce movement costs (Rosenbaum et al. 1995; Soechting et al. 1995). Moreover, the selection of different effectors for the control of movement can also be interpreted as a reduction of movement costs (Rosenbaum 2008; Rosenbaum et al. 1991).

In sum, again motor redundancy may seem like a burden from a computational perspective, because it requires the selection of one particular movement out of an infinite number of possibilities each time a movement is executed. However, redundancy enables flexible and effective behavior and may be used to facilitate or enable the execution of many actions in the first place. Thus, motor redundancy enables highly dexterous behavior.

2.3 Uncertainty

A final key difficulty of movement planning and control is the uncertainty associated with bodily motion as well as sensory feedback. Many attributes of a movement task can change. On a short timescale, the goal or the external constraints can change quickly. The movement goal can change, for example, when humans interact with moving objects. Likewise, task constraints may change, for example, when moving obstacles have to be avoided or when the weight of the hand changes when temporarily carrying an object. On an intermediate timescale, environmental factors may change. Such factors influence the mapping from motor action to movement outcome, for example, when the nature of the ground on which humans are walking changes, when joints become sore, or when muscles become tired. On an even longer timescale, the body kinematics and dynamics themselves change (Wells et al. 2002), undergo injury, and continuously slowly change due to the amount of bodily exercise, food intake, etc. (Shadmehr and Wise 2005).

These changes make it difficult to plan movements, because many factors that influence how a planned movement unfolds cannot be fully determined before movement onset. Even in a simple task with none of these difficulties, the movement could still not be perfectly executed because movement execution is imprecise. In fact, movement execution noise plays a significant role in motor variability (van Beers et al. 2004). For example, the motor variability in reaching movements increases with movement speed (Fitts 1954; Harris and Wolpert 1998). Thus, even if we were able to perfectly perceive the goal and the external context, noise in the transformation of the movement plan into actual movements results in the association of some uncertainty to the outcome of any action.

The uncertainty generates the need for humans to monitor their own body and their environment in almost any task—goals and obstacles have to be tracked, and deviations due to unexpected external forces, noisy movement execution, or an outdated body schema have to be observed. Besides movement control, environment and body have to be observed and their actual states inferred to properly *prepare* effective actions. The need to monitor body and environment, however, poses another big challenge, as the processing of sensory input takes time and the sensory feedback itself is delayed and noisy. The sensory information can consequently resolve the uncertainty only in part, so that a finite uncertainty remains and internal body and environmental state estimates will never be perfectly accurate. Thus, optimal human motor control, planning, and decision making is a hard problem

not only due to the continuous changes and fluctuations in goals, constraints, the environment, and the body, but also due to uncertainty that is inherent in movement execution, and noisy, often delayed sensory feedback. Humans can cope with these difficulties and generalize or adapt their skills very well in most circumstances. This is apparent, for example, in predictive reaching tasks (von Hofsten 1980), obstacle avoidance (Van Hedel et al. 2002), changing environment tasks (Rieser et al. 1995), and infant growth (Konczak and Dichgans 1997). In sum, uncertainty poses additional difficulties to motor control and inevitably forces dexterous motor control to flexibly account for varying amounts of uncertainty.

3 Modularity and Hierarchy in the Brain (and Beyond)

The previous sections have shown that the control of the human body is a computational challenge. Here, we focus on how the brain as a control system appears to be able to control the human body robustly and with high dexterity. We survey studies on brain structures and anatomy that suggest how the brain is organized hierarchically and modularly to reach the observable levels of robustness and dexterity.

We use the term modularity to refer to structures where different parts (modules) execute different computations largely independently from each other. Modular structures enable robustness due to this partial independence and autonomy of the modules. Additionally, modularity enables flexibility because different modules may be invoked to enable task-dependent planning and adjustments of the control process. Hierarchy refers to structures in which information that is generated at "higher levels" is forwarded to "lower levels," where additional computations may be carried out. Hierarchical structures enable the organization of a broad behavioral repertoire because they enable the division of a complex computation into smaller, tractable parts. These two facets of behavioral organization are reflected in many aspects of motor control in the brain. We now first focus on modular representations for motor control in the brain and then on evidence for distributed, modular planning and control.

3.1 Modular Representations

The brain is generally organized in a modular structure wherein many distinct areas and networks cooperate to generate human movement. For example, body and environment are encoded differently in different brain areas, where different sensory and anticipatory information dominates the respective encodings (Schwartz et al. 2004). Different components for human motor control were identified, separating state estimation from sensorimotor prediction, payoff generation and estimation, and interaction control, attributing these aspects to the parietal cortex, the cerebellum, the basal ganglia, and motor cortical regions, respectively (Doya 1999; Shadmehr

and Krakauer 2008). Graziano (2006) points out that the motor cortex itself is structured highly modularly with different sub-structures being responsible for particular, ethologically-relevant behaviors. Moreover, parietal and motor cortical areas interact sub-modularly in that different interactive behaviors appear to be controlled by respective pairs of sub-modules, such as defensive behaviors, which are dominantly controlled and co-represented by the ventral intraparietal area (VIP) and the polysensory zone in the precentral gyrus (PZ) (Graziano and Cooke 2006). A different study points out that the learned sensorimotor transfer induced by different, novel types of tools is modularly represented in the cerebellum (Imamizu et al. 2003). Transfer studies on uni-manual and bi-manual actions within different force fields suggest that learning and adaptation modularly separate respective skills but also allow skill transfer, when seemingly appropriate (Nozaki et al. 2006; Tong and Flanagan 2003). We believe that this structural modularity provides the neural basis for robust and dexterous behavioral control.

The representation of the body state is also highly modular, because the state of the body is encoded for individual body segments and for different modalities. For example, body segments are separated in our awareness (de Vignemont et al. 2009) and are addressed modularly but synergistically by primary motor cortex output (Gentner and Classen 2006; Latash et al. 2007). These representations of body (and possibly movement targets) are encoded in different modalities that can be, for example, retinal-, eye-, or hand-related, or proprioception-based (Battaglia-Mayer et al. 2003; Bernier et al. 2007). Not only are they represented in different modalities, but they can also be processed independently (Bernier et al. 2007; Sarlegna 2007). Some of these modalities can even be body-relative (Battaglia-Mayer et al. 2003; Buneo et al. 2002). These different representations enable humans to perceive and control their own body with a manifold of different variables, frames of reference, and modalities, thus enabling the pursuance of multi-modal goals.

3.2 Modularity and Hierarchy for Planning and Control

It is generally assumed that motor planning is a hierarchically organized process, in which higher-level intentions are transformed into lower-level motor control signals (Butz et al. 2007; Hoffmann et al. 2007b; Rosenbaum et al. 1995). According to several models, moving the hand to a desired position is realized by first mapping the extrinsically encoded desired position of the hand onto corresponding body postures. These body postures are then further transformed into a sequence of motor commands (Butz et al. 2007; Rosenbaum et al. 1995). This notion is supported by neurophysiological evidence, which shows that motor cortical neurons partly encode final movement postures. Thus, reaching an extrinsically encoded goal requires the re-encoding of that goal in different sensory modalities. Additionally, the memory representation, planning and execution of sequential movements is organized hierarchically (Rosenbaum et al. 1984, 1983; Schack and Mechsner 2006).

Multimodal coding of goals may also be helpful to control movements. As it has become apparent above, the physical properties and the neural circuitry in the spinal cord operate as feedback loops and thus may alleviate the central nervous system from precisely monitoring and regulating the states of individual muscles. This results in a cascade of nested feedback loops with different feedback delays (Hoffmann et al. 2007a; Koechlin and Summerfield 2007), which control the unfolding movement concurrently.

Besides the brain, spinal neurons and the properties of the muscles and the body also facilitate movement control (Loeb et al. 1999; Shadmehr and Wise 2005). The physical properties of muscles make human movements robust to external perturbations and facilitate control (Birbaumer and Schmidt 1996; Hof 2003). For example, it has been proposed that a joint settles in a unique posture, given a specific combination of activation in the attached muscles (Bizzi et al. 1976; Polit and Bizzi 1979). Finally, body geometry and joint function have evolved to enable and facilitate ecologically crucial behavior, such as locomotion and visually guided manual actions. While these geometrical and physical features stabilize human movement, spinal networks may also partially compensate for external perturbations and support the central nervous system computationally (Adamovich et al. 1997; Feldman 1966; Feldman and Levin 1995; Mussa-Ivaldi and Bizzi 2000; Mussa-Ivaldi et al. 1994). Thus, modularity and hierarchy play crucial roles in motor planning and control beyond the borders of the central nervous system (Loeb et al. 1999).

4 Modular and Hierarchical Models

In the previous sections, we identified sensory redundancy, motor redundancy, and uncertainty as factors that deem motor control a computationally daunting problem. We argued that the modular and hierarchical organization of motor control structures and motor behavior in the brain enables humans to deal with this complexity. In this section, we review two computational models for arm reaching that dwell heavily on hierarchical and modular organizations and thereby address these problems. First, the sensorimotor unsupervised redundancy resolving architecture (SURE_REACH) is discussed, which employs a hierarchical structure to generate dexterous arm reaching movements while coping with and exploiting sensory and motor redundancy. Second, an even more modular architecture is introduced, which modularly combines sensory, motor, and body schema knowledge for the generation of accurate state representations and dexterous body control.

4.1 SURE_REACH

SURE_REACH is a hierarchical, modular neural network model of motor learning, planning, and control (Butz et al. 2007; Herbort and Butz 2010). The model

generally accounts for the relationship between behavioral flexibility, motor planning, and unsupervised learning in human arm reaching movements. In this chapter we focus on its hierarchical, modular structure and the resulting benefits. For more details on the involved Hebbian-based learning algorithms and the detailed implementation of its components, we refer the reader to previous descriptions (Butz et al. 2007; Herbort and Butz 2010; Herbort et al. 2008).

In SURE_REACH, two different sensory modalities are processed. The position of the hand is encoded in a body-relative frame of reference system akin to visual input. The posture of the arm is encoded in a proprioceptive frame of reference system. Both modalities are represented by distinct populations of neurons, where each neuron is tuned to a particular hand position or posture (Fig. 1). Both modalities can be used to specify goals and to monitor ongoing movements. To plan and execute movements, higher-level goal representations, such as visually encoded goals, have to be translated into motor commands.

Two learned internal models achieve the necessary transformations to realize dexterous motor control. A *kinematic model* maps hand positions onto postures (in SURE_REACH, this model is called posture memory, Fig. 1). The model connects hand space with posture space, encoding the redundancy of the controlled arm. Second, a *sensorimotor model* encodes how different actions (i.e. activations of muscles) affect the arm state. This model is used to generate a movement plan, which associates different arm postures with those motor commands that move the arm closer to the goal. As this structure ultimately controls the movement of the arm, this part of the model is referred to as *motor controller*. Both internal models are realized by single-layered associative neural networks. They are trained by means of Hebbian learning. As both internal models may encode one-to-many mappings, they can encode motor redundancy, which is resolved explicitly during motor planning or implicitly during action execution.

Consider the execution of an arm movement toward a desired hand position. In this case, the desired hand position is encoded in hand space. Using the posture memory, the hand position is transferred into a representation of all those arm postures in which the end-effector of the arm (the hand) happens to reach the desired position. Thus, the goal has been transformed from an extrinsic code to a proprioceptive representation that conserves motor redundancy. Next, a motor plan is generated by an adaptive planning mechanism (a form of neural dynamic programming, see Butz et al. 2007 for details). Using the posture space representation of possible desired end-postures and the sensorimotor contingencies encoded in the sensorimotor model, the planner generates a mapping which determines an optimal motor command for each possible posture that the arm could assume. Executing the successive motor commands moves the hand to the goal location on an approximately optimal trajectory dependent on additional constraints. By default, a movement is executed that reaches the goal as fast as possible.

To reach the goal, the arm posture is continuously forwarded to the neural structures that support the motor plan (motor controller). These neural structures then, in turn, provide updated motor commands.

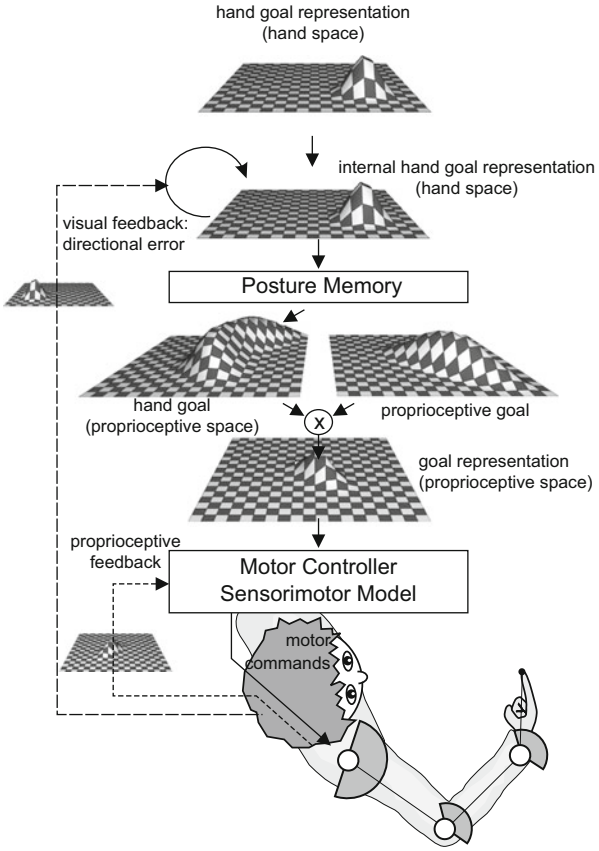

Fig. 1 The sensorimotor unsupervised redundancy resolving architecture SURE_REACH is hierarchically structured. The architecture enables the processing of multimodal goals (e.g., visual and proprioceptive) and constraints (e.g., obstacles, posture constraints, and movement costs) and can thus generate highly dexterous environmental interactions. Visual error feedback continuously adjusts the internal goal state to minimize the final error. Proprioceptive feedback is used for continuously updating the posture state estimate

As mentioned above, the modular structure of the model can also accommodate additional goal constraints. Consider the example of pointing at something. In this case, an extrinsically defined constraint (where to point) and a proprioceptively defined constraint (align forearm and hand) have to be integrated. The SURE_REACH model can integrate multimodal constraints because the goal in hand space is converted into a redundant goal representation in posture space. From these potential goal postures, postures can be selected that match the additional, proprioceptively-defined constraint. Given the resulting more restricted set of potential end postures, further motor planning can proceed. Figure 1 illustrates this process.

Similarly, other constraints can be taken into account, such as the anticipated requirements of subsequent movements (Herbort and Butz 2007). Due to the redundancy encoded in the sensorimotor model, movement plans can be generated that also avoid movements of specific joints (Butz et al. 2007). Additionally, visually or proprioceptively encoded obstacles can be avoided by biasing the generation of the motor plan (Butz et al. 2007). In all these examples, the encoding of redundant sensory channels and the encoding of motor redundancy enable the integration of goal constraints from multiple modalities, projected into the posture space modality. The architecture essentially shows that a hierarchical, modular structure can account for dexterity.

As in any motor controller, the motor commands generated by SURE_REACH may be incorrect, for example, due to insufficient training of the internal models or due to motor noise. Also, goals may change during actions. Thus, mechanisms to monitor the progress of the movement and adjust actions accordingly have to be integrated. On the lowest level, the motor controllers map the continuously updated arm posture onto suitable motor commands, thus causing the arm to move until an acceptable end-state is reached. Additionally, visual feedback can be used to regulate movements, consequently controlling the arm at multiple levels (Herbort et al. 2008). In this case, the postural feedback loop is nested into a visual feedback loop, which may adjust the internal visual hand goal representation when an error is detected. For example, if the hand is to the left of its desired position, the internal hand goal representation is shifted to the right. This altered internal goal representation is then projected via the internal models as before, resulting in an altered proprioceptive goal representation, which causes the adaptation of the movement plan, which ultimately causes corrective movements. As corrections to visual errors are recoded into proprioceptive representation, it is possible to maintain postural constraints while adjusting the position of the arm.

In sum, SURE_REACH is structured in a modular, hierarchical fashion, enabling flexible and robust control. The modular organization of the sensory representations enables the integration of multimodal goal representations for the generation of a coherent motor plan. The hierarchical organization enables the usage of proprioceptive and visual feedback in parallel, without compromising goal constraints defined in either modality. SURE_REACH shows that a modular neural architecture enables dexterous motor planning and control. The system was also successfully applied to control a dynamic robot arm with underlying PD-based dynamics control (Herbort and Butz 2010) and it was combined with a motivational architecture, confirming that the coupling with an independent payoff module is possible (Butz and Pedersen 2009).

Nonetheless, there is a rather significant restriction to SURE_REACH, which is the population-encoded representation. Since posture space and location space are encoded by distributed neural fields with local neurons, the granularity of the neurons is inevitably restricted. Particularly in posture space more degrees of freedom, such as the seven-dimensional human arm, yield huge populations of neurons, even if each sub-space is covered with rather large receptive fields. Computationally, the spatial requirements of a grid-based population code, as used

in the original SURE_REACH architecture, scale exponentially in the number of degrees of freedom covered by the code given a constant number of neurons per dimension. More suitably distributed encodings may alleviate some of the computational burden, but cannot overcome the exponential scalability problem. Thus, further modularizations of SURE_REACH are inevitable. The following subsection sketches-out an approach we are currently pursuing.

4.2 Modular Modality Frame Model

Since SURE_REACH encodes the posture space with a single population of neurons, the number of neurons necessary to cover the whole space with an equal number of neurons per dimension grows exponentially with the number of degrees of freedom. Thus, although this encoding provides a representation that enables dexterous motor control, the representations used by SURE_REACH are computationally very expensive given higher degrees of freedom.

To cope with this scalability problem, we recently introduced the Modular Modality Frame model (MMF)—a model that splits the posture space into interacting modules and thus builds a modular hierarchical representation on several levels. We term each module in the resulting architecture a *modality frame*, denoting a spatial representation that is tied to a particular modality (e.g., vision or proprioception) and a frame of reference (e.g., head-centered or limb-centered). Further modularization applies because modality frames exist for each body part. For example, the seven-dimensional posture space of a humanoid arm is split into three two- and three-dimensional spaces.

This modularization overcomes the scalability problem and has additional features, such as improved noise robustness and automatic detection of sensor errors, as detailed below. Adaptability in movement *planning*, however, becomes more challenging because of the model's local nature of the posture space. For the present, we use an explicit model of the arm and disregard the learning challenge.

4.2.1 Modular Modality Frames

While SURE_REACH distinguishes only between a hand location space and an arm posture space, MMF represents each limb and joint separately in four different frames of reference and modalities. The model incorporates the forward and inverse kinematic mappings of SURE_REACH, but it applies them locally to the kinematic chain of the represented arm. Moreover, the local constraints of the kinematic arm chain are considered. Figure 2 illustrates the different modality frames considered in MMF. Two sets of modality frames use a reference frame that is centered at the shoulder (global location and global orientation), while the two other sets use arm-relative frames of reference (local orientation and local angles). The first frame encodes a "visual" frame by specifying the end-point of a limb segment in

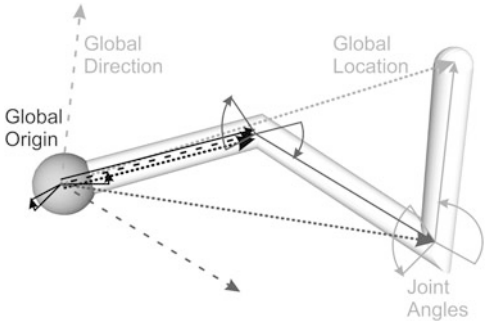

Fig. 2 Different frames of reference are encoded for each limb (Limb 1 in *black*, Limb 2 in *dark gray*, and Limb 3 in *light gray*) in MMF. Joint locations (*dotted arrows*) and limb orientations (*dashed arrows* at origin) are represented shoulder-centered. Moreover, limb orientations are also represented relative to the previous limb (*solid arrows* on arm). Finally, joint angles are encoded (*curved arrows*)

a shoulder-centered $3D$-Cartesian frame of reference. Limb orientation, which is composed of a direction and a vector perpendicular to the direction, is encoded in a shoulder-centered frame of reference. Since only unit vectors are encoded and both vectors of the orientation are perpendicular to each other, the orientation frame is also three dimensional. The two other frames encode limb orientations relative to the next upper limb (e.g., forearm orientation relative to the upper arm orientation) and joint angles. The upper arm (black), forearm (dark gray), and hand (light gray) are modeled as segments.

The body state is represented modularly with respect to both, different body segments and different modality frames. Connections between these modular modality frames form a graph that constitutes the forward and inverse kinematic mappings as well as dependencies along the kinematic chain (cf. Fig. 3). The mappings between the modality frame modules are shown in Fig. 3, where the number of equally-colored incoming nodes determines the complexity of the respective mapping. Note that the state of any arm segment in any frame depends only on frames of the same or the previous segment at the moment. Across all arm segments, the whole architecture implements a hierarchical modular arm state representation, which can be used for both body state maintenance and planning.

4.2.2 Body State Maintenance

Current research in robotics stresses the importance of utilizing statistical methods to deal with noise and to process information effectively (Vijayakumar et al. 2009). In MMF, both mean and variance estimates from every module are transformed to every other connected module. The transformed state and variance estimates from the other modalities form multiple inputs, which are weighted based on their variance estimates. This process is shown on the left-side of Fig. 4. Recently,

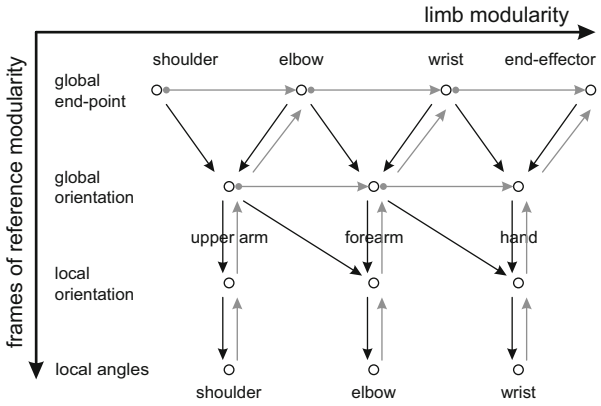

Fig. 3 The system is modular in two forms: with respect to limbs and with respect to different frames of reference. Every *circle* in the figure depicts a maximally three-dimensional spatial representation. *Gray arrows* denote forward kinematic mappings. *Black arrows* the inverse kinematic mappings

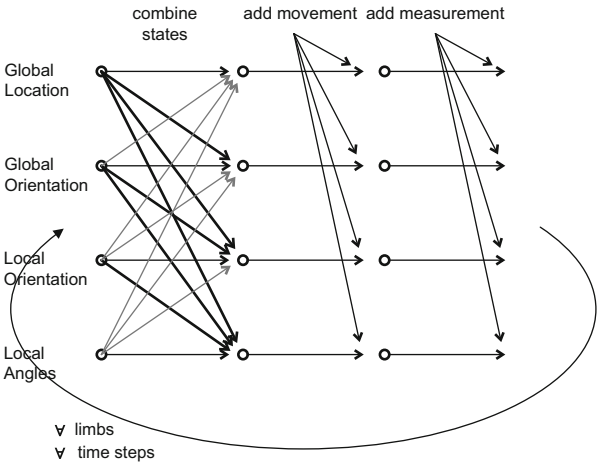

Fig. 4 Data flow in the filter system. For each joint and time, the states and variances are first transformed from each modality frame to all others. The movement is then added and the estimated states and variances are finally combined with the current respective measurements

we have also added an online comparison mechanism, which estimates the plausibilities of each sensory information in comparison with the other available sensors using Bayesian principles (Ehrenfeld and Butz 2013). In result, the system is able to identify unexpectedly noisy sensors as well as faulty sensors, effectively downscaling the very noisy or approximately ignoring the faulty information, respectively.

Once the states of all modules are exchanged, the movement can be planned by comparing the new estimated state to the goal in its respective modality frame. Given a current movement command, a forward model predicts the expected sensory consequences in each module. This prediction of the reafference (von Holst and Mittelstaedt 1950) is then used to filter the consequent sensory information. This process is displayed in the center and on the right side of Fig. 4. All three stages are applied successively, effectively maintaining the body state over time in a highly distributed fashion. At the moment, the filtering stages are realized with local Kalman filters, but any other filtering technique may be applied.

4.2.3 Movement Planning

The presented model generally allows for the definition of goals in arbitrary modality frames and separately for each body segment. In each modality frame, the goal is compared with the estimated body state resulting in a difference vector. This difference is then projected into the corresponding angular modality frame. A simple kinematic, step-wise motor controller converts the desired angular change into step-wise angular adjustments. The inclusion of more distributed planning processes for the flexible avoidance of obstacles or for the incorporation of additional movement or posture constraints is currently being pursued. In the following, we focus on exemplary performance on the arm state estimation while moving the arm to a goal location.

4.2.4 Exemplar Performances

To illustrate the current capabilities of MMF, we show performance in two setups with faulty sensors. We averaged results over 150 runs. The controlled, simulated arm had three limbs with a limb length of 0.5 units each. Each joint had three degrees of freedom with unrestricted rotation options. Goal-directed movements were executed with a maximum angular velocity of $0.05 \cdot \sqrt{3}$rad per iteration per arm limb. The recorded results were gathered from exemplary motion trials where the arm moved from a random start posture to a random goal posture.

Two forms of Gaussian noise were present: velocity-dependent motor noise and sensor noise. Motor noise had a standard deviation equal to the velocity. In Setup A, we set the orientation-sensor's standard deviation to 0.1 and the angular-sensor's standard deviation to 0.1rad per dimension. Moreover, the global end-point sensor was more accurate with a standard deviation of 0.01 in units of limb length. To model sensor failure, an offset equal to 0.5 limb length units was applied to the second dimension of all three global location modality frame sensors from time step 10 until time step 50. In Setup B, we set the sensor noise standard deviations to $\sigma = 0.1$ in all modality frames (in units of limb length, dimensionless, or in rad). While thus the sensor noise in the global location frames was even higher in Setup B, we also increased the sensor failure in Setup B: besides adding an offset of 0.5 limb

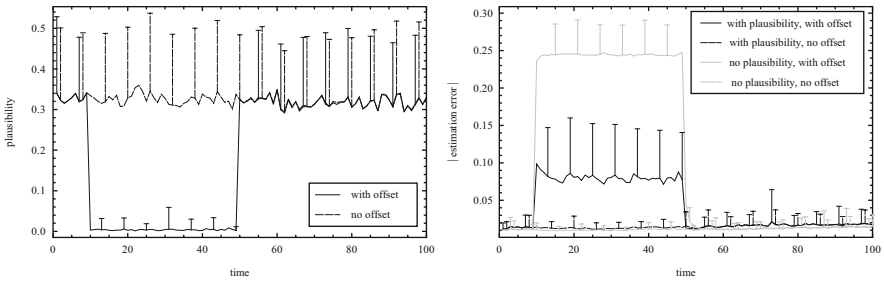

Fig. 5 Setup A: The plausibility estimates of the global location sensor (left-hand side) show that MMF identifies the faulty sensor very well, consequently decreasing its state estimation influence. In consequence, an arm state estimate with significantly lower error than without plausibility estimate is maintained (right-hand side). Without sensor failure, the addition of the plausibility estimate does hardly have any negative impact (right-hand side). Error bars show the estimated standard deviation computed over 150 independent runs

length units to the second dimension of all three global location modality frame sensors, we also added an offset of 0.5rad to the second dimension of all three angular modality frame sensors. Again, these offsets were applied between time step 10 and time step 50. Each modular modality frame of MMF was informed about the noise settings in both setups but not about the offsets.

Figure 5 shows the resulting plausibility estimates and mean absolute errors of the global location modality frame in Setup A. MMF can handle this case very well, effectively ignoring the location sensor during its failure. An accurate arm state estimate is consequently maintained over time. Albeit this estimate is worse than when no sensor failure is encountered, it is still much less error prone than when MMF does not compute sensor plausibilities.

In Setup B, noise was even stronger than in Setup A. Figure 6 shows that MMF is still able to detect the sensor failure, consequently downscaling the plausibility of the faulty location sensor (the plot for the angular sensor looks similar). Still, the state estimate encounters a larger error due to the double sensor failure. This failure is, however, lower than when plausibilities are not estimated. Note that the large standard deviations are partially due to the random start and goal posture selections in the conducted 150 independent experiments. When the sensor offsets are not applied, MMF without plausibility estimates performs slightly better. This was expected, because the plausibility estimates inevitably are prone to noise and thus disrupt the sensory information slightly when it fits perfectly with the prior sensor noise estimates given to the system.

The usage of probabilistic representations together with the combination of multiple modalities both for sensors and state estimates yields many benefits. In the results above, we have shown that sensor failures can be detected effectively. Also highly noisy sensors can be identified with similar techniques (cf. Ehrenfeld and Butz 2013). Moreover, high noise robustness and efficient integration of information has been shown as well as the ability to avoid linearization errors (Ehrenfeld and

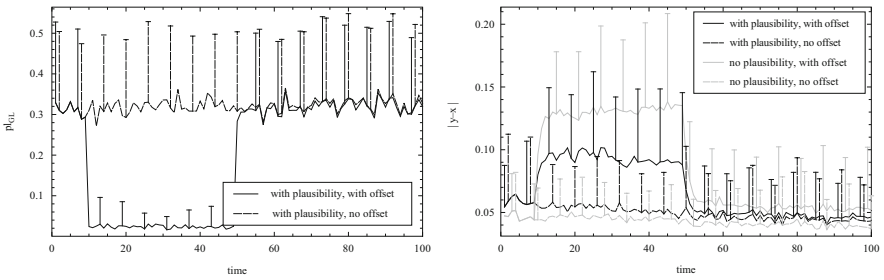

Fig. 6 Setup B: Given the concurrent failure of the two sensors during iteration 10 and 50, the plausibility estimates of the global location sensor (left-hand side) show that MMF identifies the faulty location sensor still very well. Due to the double failure, the error of the end-point location estimate is larger than in Setup A but still lower than when plausibility is not computed (right-hand side). Due to the increase in noise compared to Setup A, the computation of the plausibility estimates decreases the quality of the arm state estimation slightly when the sensor noise matches the prior noise estimate perfectly (right-hand side). Error bars show the estimated standard deviation computed over 150 independent runs

Butz 2011). Knowledge can generally be exchanged interactively across modalities and even across arm limbs. Goals or constraints that are defined partially in any combination of modality frames can be easily implemented. The computational complexity scales linearly with the number of joints. At the moment, we are working on adding distributed population codes in each modality frame to enable more distributed planning, and thus to achieve the dexterous motor control capabilities of SURE_REACH in bodies with even more degrees of freedom.

5 Discussion and Conclusion

Given the complexity of the task to control the human body, and the high accuracy and flexibility that humans achieve, modeling human behavior is a daunting challenge. We reviewed how the brain achieves this performance and conclude that the problems can only be solved by a modular, hierarchical control structure. Several computational models embrace this notion. Based on our work on modeling human arm reaching, we showed that goal-directed movement execution can be effectively controlled by a modular, body schema-based processing and control architecture, which we called the *modular modality frame model* (MMF). In this model different sources of information can be effectively exchanged or integrated, flexible and highly complex planning can be accomplished, and uncertainties can be resolved. While MMF is certainly an interesting achievement, we believe that it has the potential to be combined and enhanced in various ways.

So far the forward and inverse kinematics in MMF are hard-coded and are assumed to be exact. If the body kinematics were not exactly known, uncertainty estimates may be added to the kinematic-based transformations in MMF. The mappings could also be learned, as was done in the less modular SURE_REACH

architecture introduced above. On the other hand, so far the mappings only take dependencies downwards the kinematic chain into account. For example, highly accurate information about the location of the hand does not influence the elbow location estimate at the moment. Such backwards estimations may make the model even more robust and enforce even further consistency of the body state estimates. Moreover, such capabilities seem mandatory to model bodily illusions, such as the rubber-hand illusion (Botvinick and Cohen 1998).

Besides these mapping-related aspects, several other additions to MMF may be considered. Sensory delays may need to be handled in an actual robotics architecture. Also, system dynamics may need to be accounted for, and force control as well as task-oriented compliances deserve further consideration. As sketched out above, the brain clearly implements a hierarchical control architecture starting with low-level, closed-loop muscle control to abstract, high-level planning. SURE_REACH and MMF currently only distinguish location, orientation, and posture spaces for planning and control. Additional control levels may be added on both sides of the spectrum. Further control particularizations are needed for coping with system dynamics, such as forces and unexpected perturbations. On the other hand, further abstractions and goal sequencing may allow the realization of more complex and more dexterous object manipulations.

For the latter case, the encoding of more complex motion sequences and the goal-oriented coordination of such sequences seem to be of vital importance. We believe that MMF provides the framework to apply such motion sequences selectively to task-relevant modality frame modules. Such a combination could, for example, enable the system to use the elbow instead of the hand to push down a handle. Besides, movement is not only our way to manipulate objects in our environment but also our means of social communication. Body language, facial expressions, and speech are all essentially motor actions. It has been proposed that abstract cognitive processes, such as social interaction or object categorization, are grounded in motor processes. For example, motor simulation could be used to understand the intentions of other humans (e.g., Gallese and Goldman 1998), to predict events in the world (e.g., Schubotz 2007), or even to form the basis for the evolution of cognition (Cruse 2003). Even though such simulation accounts are most likely insufficient as a general account for all cognitive capabilities, these approaches give hints as to how abstract cognition may be grounded in basic motor processes (Barsalou et al. 2007; Jacob and Jeannerod 2005; Keysers and Gazzola 2007; Saxe 2005).

A computational model that links motor processes to more abstract processes is the hierarchical MOSAIC (modular selection and identification for control) model (Wolpert et al. 2003). The model assumes that multiple controllers for different tasks exist in parallel in the brain. Each module can be an internal forward model representing a specific context, or a motor action. A predictor is linked to each module, and multiple predictions of the dynamics and behavior of the body are generated and can be compared to sensory measurements. The distance (error) of the predictions from the measurements give the probability that the simulated modules coincide with the actual context or movement. Thus, each predictor is a hypothesis tester for its module. For example, by picking up a glass and comparing

the predicted with the actual sensory feedback, the amount of liquid in the glass may be estimated (cf. also Roy et al. 2006, where an approximate object weight was determined by that principle). Of course, the MOSAIC model can also be used for movement adjustments. In that case, different actions serve as modules and the difference of the predictions from the sensory feedback result in probabilities for action activations.

It is even more interesting to model the state of other people and to understand their actions. First, only a visual input is available in this task. Thus, it has to be converted into a suitable representation, e.g. to joint angles. Second, multiple modules, which may induce different actions, have to be simulated. Their outputs serve as inputs to their respective predictors and the predictions are compared to the converted sensory input. The errors give matching likelihoods back to the respective modules. Modules with high likelihoods then can form a symbolic code of the actor's state. Thus, observing a continuous body trajectory can be converted into a symbolic stream of module activities. The symbolic stream does not store the entire trajectory as it has a lower dimension than the actual movement. But the body models differ between different people, and the movement is represented in the observer's model. Both influences lead to errors in the estimates about the motor commands the actor actually activated. Third, to be able to understand the actor, the model activities may be sufficient to derive the actor's intentions, since each available module is tuned to accomplish a specific task. Thus, the goals of the task may be deduced, depending on the current context and the derived module activities. If the observer wants to not only understand the actor but also to imitate him, he may actually invoke his own corresponding modules and replace or adapt his usual sequence of module activations according to the observed one.

To allow for generalization and the ability to generate a large variety of desired movements even under changing environments and contexts, the MOSAIC model has to be embedded in or extended to a hierarchical structure. Haruno et al. (2003) discuss a hierarchical MOSAIC model consisting of several MOSAIC layers. Between these layers, information can flow bi-directional. Along the bottom-up direction, lower levels provide a posterior probability that encodes which modules are selected given the behavioral situation. This posterior probability serves, together with an abstract symbolic desired trajectory, as an input for higher levels. Along the top-down direction, higher levels can force prior probabilities on lower levels. These prior probabilities are chosen by the motivation to achieve a desired behavior. Higher levels can also learn to predict the posterior probability of lower levels after the next time step. This provides a bi-directional interaction, both during learning and during control.

As a next step towards dexterous environment manipulation and possibly even social communication, we thus propose to combine the strengths of the hierarchical MOSAIC architecture with the MMF detailed here. The MMF has the advantage that multiple representations of interactions coexist and can be selectively recruited for task-specific simulation, reproduction, and possibly even comprehension. The recent work of Calinon and Billard (2009) shows that only a few trials may be necessary to actually identify the frames of reference necessary to realize a

particular imitation. Within the MMF, not only posture or location space may be identified as relevant, but also potentially the selective responsibility of particular limbs or joints, or even the exchangeability of them. Thus, future research promises novel discoveries in cognitive modeling and the development and understanding of abstract cognitive processes, grounded in highly modular and hierarchical dexterous control architectures.

References

Adamovich, S. V., Levin, M. F., Feldman, A. G. (1997). Central modifications of reflex parameters may underlie the fastest arm movements. *Neurophysiology, 77*(3), 1460–1469.

Barsalou, L. W., Breazeal, C., Smith, L. B. (2007). Cognition as coordinated non-cognition. *Cognitive Processing, 8*(1), 79–91. doi:10.1007/s10339-007-0163-1.

Battaglia-Mayer, A., Caminiti, R., Lacquaniti, F., Zago, M. (2003). Multiple levels of representation of reaching in the parieto-frontal network. *Cerebral Cortex, 13*(10), 1009.

Bernier, P. M., Gauthier, G. M., Blouin, J. (2007). Evidence for distinct, differentially adaptable sensorimotor transformations for reaches to visual and proprioceptive targets. *Journal of Neurophysiology, 98*(3), 1815.

Bernstein, N. A. (1967). *The co-ordination and regulation of movements*. Oxford: Pergamon.

Birbaumer, N., & Schmidt, R. (1996). *Biologische Psychologie [Biological Psychology]*, 3rd edn. Berlin: Springer.

Bizzi, E., Polit, A., Morasso, P. (1976). Mechanisms underlying achievement of final head position. *Journal of Neurophysiology, 39*(2), 435–444.

Botvinick, M., & Cohen, J. (1998). Rubber hands 'feel' touch that eyes see. *Nature, 391*, 756.

Buneo, C., Jarvis, M., Batista, A., Andersen, R. (2002). Direct visuomotor transformations for reaching. *Nature, 416*(6881), 632–636.

Butz, M. V., Herbort, O., Hoffmann, J. (2007). Exploiting redundancy for flexible behavior: unsupervised learning in a modular sensorimotor control architecture. *Psychological Review, 114*, 1015–1046.

Butz, M. V., & Pedersen, G. K. M. (2009). The scared robot: motivations in a simulated robot arm. *32nd Annual Conference on Artificial Intelligence, KI 2009*, 460–467.

Caggiano, V., Fogassi, L., Rizzolatti, G., Thier, P., Casile, A. (2009). Mirror neurons differentially encode the peripersonal and extrapersonal space of monkeys. *Science, 324*, 403–406.

Calinon, S., & Billard, A. (2009). Statistical learning by imitation of competing constraints in joint space and task space. *Advanced Robotics, 23*(15), 2059–2076.

Cruse, H. (2003). The evolution of cognition—a hypothesis. *Cognitve Science, 27*, 135–155.

de Vignemont, F., Majid, A., Jola, C., Haggard, P. (2009). Segmenting the body into parts: evidence from biases in tactile perception. *The Quarterly Journal of Experimental Psychology, 62*(3), 500–512.

Doya, K. (1999). What are the computations of the cerebellum, the basal ganglia and the cerebral cortex? *Neural Networks, 12*(7–8), 961–974.

Doya, K., Ishii, S., Pouget, A., Rao, R. P. N. (2007). *Bayesian brain: probabilistic approaches to neural coding*. Cambridge: MIT.

Ehrenfeld, S., & Butz, M. V. (2011). A modular, redundant, multi-frame of reference representation for kinematic chains. In *IEEE International Conference on Robotics and Automation* (pp. 141–147).

Ehrenfeld, S., & Butz, M. V. (2013). The modular modality frame model: continuous body state estimation and plausibility-weighted information fusion. *Biological Cybernetics, 107*, 61–82. doi:10.1007/s00422-012-0526-2.

Elsner, B., & Hommel, B. (2001). Effect anticipation and action control. *Journal of Experimental Psychology: Human Perception and Performance, 27*, 229–240.

Engelbrecht, S. E. (2001). Minimum principles in motor control. *Journal of Mathematical Psychology, 45*, 497–542.

Feldman, A. G. (1966). Functional tuning of nervous system with control of movement or maintenance of a steady posture. II. Controlable parameters of the muscle. *Biophysics, 11*, 565–578.

Feldman, A. G., & Levin, M. F. (1995). Positional frames of reference in motor control: origin and use. *Behavioral and Brain Sciences, 18*, 723–806.

Fischer, M. H., Rosenbau, D. A., Vaughan, J. (1997). Speed and sequential effects in reaching. *Journal of Experimental Psychology: Human Perception and Performance, 23*(2), 404–428.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental PSychology, 74*, 381–391.

Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *The Journal of Neuroscience, 5*(7), 1688–1703.

Gallese, V., & Goldman, A. (1998). Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences, 2*, 493–501.

Gentner, R., & Classen, J. (2006). Modular organization of finger movements by the human central nervous system. *Neuron, 52*, 731–42.

Graziano, M. S. A. (2006). The organization of behavioral repertoire in motor cortex. *Annual Review of Neuroscience, 29*, 105–134.

Graziano, M. S. A., & Cooke, D. F. (2006). Parieto-frontal interactions, personal space, and defensive behavior. *Neuropsychologia, 44*, 845–859.

Greenwald, A. (1970). Sensory feedback mechanisms in performance control: with special reference to the ideo-motor mechanism. *Psychological Review, 77*, 73–99.

Harris, C. M., & Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature, 394*, 780–784.

Haruno, M., Wolpert, D., Kawato, M. (2003). Hierarchical mosaic for movement generation. In *International Congress Series* (vol. 1250, pp. 575–590). Amsterdam: Elsevier.

Herbort, O., & Butz, M. V. (2007). Encoding complete body models enables task dependent optimal behavior. In *Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, 12–17 August 2007* (pp. 1424–1429).

Herbort, O., & Butz, M. V. (2010). Planning and control of hand orientation in grasping movements. *Experimental Brain Research, 202*, 867–878.

Herbort, O., & Butz, M. V. (2012). The continuous end-state comfort effect: weighted integration of multiple biases. *Psychological research, 76*, 345–363. doi:10.1007/s00426-011-0334-7.

Herbort, O., & Butz, M. V. (2011). Habitual and goal-directed factors in (everyday) object handling. *Experimental Brain Research, 213*, 371–382. doi:10.1007/s00221-011-2787-8.

Herbort, O., Butz, M. V., Hoffmann, J. (2008). Multimodal goal representations and feedback in hierarchical motor control. In *Proceedings of the International Conference on Cognitive Systems 2008.*

Herbort, O., Butz, M. V., Pedersen, G. K. M. (2010). The SURE_REACH model for motor learning and control of a redundant arm: from modeling human behavior to applications in robotics. In O. Sigaud & J. Peters (Eds.), *From motor learning to interaction learning in robots* (pp. 85–106). Berlin: Springer.

Herbort, O., Ognibene, D., Butz, M. V., Baldassarre, G. (2007). Learning to select targets within targets in reaching tasks. In *6th IEEE international conference on development and learning, ICDL 2007* (pp. 7–12).

Hof, A. L. (2003). Muscle mechanics and neuromuscular control. *Journal of Biomechanics, 36*, 1031–1038.

Hoffmann, H., & Möller, R. (2003). Unsupervised learning of a kinematic arm model. In O. Kaynak, E. Alpaydin, E. Oja, L. Xu (Eds.), *Artificial neural networks and neural information processing—ICANN/ICONIP 2003*. LNCS (vol. 2714, pp. 463–470). Berlin: Springer.

Hoffmann, J. (1993). *Vorhersage und Erkenntnis: Die Funktion von Antizipationen in der menschlichen Verhaltenssteuerung und Wahrnehmung. [Anticipation and cognition: The function of anticipations in human behavioral control and perception.]*. Göttingen: Hogrefe.

Hoffmann, J., Berner, M., Butz, M. V., Herbort, O., Kiesel, A., Kunde, W., Lenhard, A. (2007a). Explorations of anticipatory behavioral control (ABC): a report from the cognitive psychology unit of the University of Würzburg. *Cognitive Processing, 8*, 133–142.

Hoffmann, J., Butz, M., Herbort, O., Kiesel, A., Lenhard, A. (2007b). Spekulationen zur strukturideo-motorischer beziehungen. *Zeitschrift für Sportpsychologie, 14*(3), 95–103.

Hoffmann, M., Marques, H., Arieta, A., Sumioka, H., Lungarella, M., Pfeifer, R. (2010). Body schema in robotics: a review. *IEEE Transactions on Autonomous Mental Development, 2*, 304 – 324.

Imamizu, H., Kuroda, T., Miyauchi, S., Yoshioka, T., Kawato, M. (2003). Modular organization of internal models of tools in the human cerebellum. *PNAS, 100*(9), 5461–5466.

Jacob, P., & Jeannerod, M. (2005). The motor theory of social cognition: a critique. *Trends in Cognitive Sciences, 9*(1), 21–25.

Keysers, C., & Gazzola, V. (2007). Integrating simulation and theory of mind: from self to social cogniton. *Trends in Cognitive Sciences, 11*(5), 194–196.

Knill, D. C., & Pouget, A. (2004). The bayesian brain: the role of uncertainty in neural coding and computation. *Trends in Neurosciences, 27*(12), 712–719.

Koechlin, E., & Summerfield, C. (2007). An information theoretical approach to prefrontal executive function. *Trends in cognitive sciences, 11*, 229–235. doi:http://dx.doi.org/10.1016/j.tics.2007.04.005.

Konczak, J., & Dichgans, J. (1997). The development toward stereotypic arm kinematics during reaching in the first 3 years of life. *Experimental Brain Research, 117*, 346–354.

Körding, K. P., pi Ku, S., Wolpert, D. M. (2004). Bayesian integration in force estimation. *Journal of Neurophysiology, 92*, 3161–3165.

Körding, K. P., & Wolpert, D. M. (2004). Bayesian integration in sensorimotor learning. *Nature, 427*, 244–247.

Körding, K. P., & Wolpert, D. M. (2006). Bayesian decision theory in sensorimotor control. *Trends in Cognitive Sciences, 10*(7), 319–326. Special issue: Probabilistic models of cognition.

Latash, M. L., Scholz, J. P., Schöner, G. (2007). Toward a new theory of motor synergies. *Motor Control, 11*, 276–308.

Latash, M. L., & Turvey, M. T. (Eds.), (1996). *Dexterity and its development*. Hove: Psychology.

Loeb, G. E., Brown, I. E., Cheng, E. J. (1999). A hierarchical foundation for models of sensorimotor control. *Experimental Brain Research, 126*, 1–18.

Mussa-Ivaldi, F. A., & Bizzi, E. (2000). Motor learning through the combination of primitives. *Philosophical Transactions of the Royal Society: Biological Sciences, 355*, 1755–1769.

Mussa-Ivaldi, F. A., Giszter, S. F., Bizzi, E. (1994). Linear combinations of primitives in vertebrate motor control. *Proceedings of the National Academy of Sciences, 91*, 7534–7538.

Nozaki, D., Kurtzer, I., Scott, S. H. (2006). Limited transfer of learning between unimanual and bimanual skills within the same limb. *Nature Neuroscience, 9*(10), 1–3. Retrived October 11, 2006 from http://www.nature.com/neuro/journal/vaop/ncurrent/pdf/nn1785.pdf.

Polit, A., & Bizzi, E. (1979). Characteristics of motor programs underlying arm movements in monkeys. *Journal of Neurophysiol, 42*, 183–194.

Pouget, A., & Snyder, L. H. (2000). Computational approaches to sensorimotor transformations. *Nature Neuroscience, 3*, 1192–1198.

Rieser, J., Pick Jr, H., Ashmead, D., Garing, A. (1995). Calibration of human locomotion and models of perceptual-motor organization. *Journal of Experimental Psychology, 21*(3), 480–497.

Rosenbaum, D. A. (2008). Reaching while walking: reaching distance costs more than walking distance. *Psychonomic Bulletin and Review, 15*(6), 1100–1104.

Rosenbaum, D. A., Inhoff, A. W., Gordon, A. M. (1984). Choosing between movement sequences: a hierarchical editor models. *Journal of Experimental Psychology: General, 113*(3), 372–393.

Rosenbaum, D. A., Kenny, S. B., Derr, M. A. (1983). Hierarchical control of rapid movement sequences. *Journal of Experimental Psychology: Human Perception and Performance, 9*(1), 86–102.

Rosenbaum, D. A., Loukopoulos, L. D., Meulenbroek, R. G. J., Vaughan, J., Engelbrecht, S. E. (1995). Planning reaches by evaluating stored postures. *Psychological Review, 102*(1), 28–67.

Rosenbaum, D. A., Marchak, F., Barnes, H. J., than Vaughan, J., Siotta, J. D., and Jorgensen, M. J. (1990). Constraints for action selection: overhand versus underhand grips. In M. Jeannerod (Ed.), *Attention and performance* (vol. XIII, pp. 321–345). Hillsdale, New Jersey, Hove and London: Lawrence Erlbaum Associates.

Rosenbaum, D. A., Slotta, J. D., Vaughan, J., Plamondon, R. (1991). Optimal movement selection. *Psychological Science, 2*, 86–91.

Rosenbaum, D. A., van Heugten, C. M., Caldwell, G. E. (1996). From cognition to biomechanics and back: The end-state comfort effect and the middle-is-faster effect. *Acta Psychologica, 94*, 59–85.

Roy, D., yuh Hsiao, K., Mavridis, N., Gorniak, P. (2006). Ripley, hand me the cup: sensorimotor representations for grounding word meaning. In *International Conference of Automatic Speech Recognition and Understanding*.

Sarlegna, F. (2007). Influence of feedback modality on sensorimotor adaptation: Contribution of visual, kinesthetic, and verbal cues. *Journal of Motor Behavior, 39*(4), 247–258.

Saxe, R. (2005). Against simulation: the argument from error. *Trends in Cognitive Sciences, 9*(4), 174–179.

Schack, T., & Mechsner, F. (2006). Representation of motor skills in human long-term memory. *Neuroscience Letters, 391*, 77–81.

Schubotz, R. I. (2007). Prediction of external events with our motor system: towards a new framework. *Trends in Cognitive Sciences, 11*, 211–218.

Schwartz, A. B., Moran, D. W., Reina, G. A. (2004). Differential representation of perception and action in the frontal cortex. *Science, 303*, 380–383.

Serwe, S., Drewing, K., Trommershuser, J. (2009). Combination of noisy directional visual and proprioceptive information. *Journal of Vision, 9*, 1–14.

Shadmehr, R., & Krakauer, J. W. (2008). A computational neuroanatomy for motor control. *Experimental Brain Research, 185*(3), 359–381.

Shadmehr, R., & Wise, S. P. (2005). *The Computational Neurobiology of Reaching and Pointing: A foundation for motor learning*. Cambridge: MIT.

Soechting, J. F., Buneo, C. A., Herrmann, U., Flanders, M. (1995). Moving effortlessly in three dimensions: does Donders' law apply to arm movement? *Journal of Neuroscience, 15*, 6271–6280.

Tong, C., & Flanagan, J. R. (2003). Task-specific internal models for kinematic transformations. *Journal of Neurophysiology, 90*, 578–585.

Trommershäuser, J., Maloney, L. T., Landy, M. S. (2003). Statistical decision theory and the selection of rapid, goal-directed movements. *Journal of the Optical Society of America A, 20*, 1419–1433.

van Beers, R., Haggard, P., Wolpert, D. (2004). The role of execution noise in movement variability. *Journal of Neurophysiology, 91*(2), 1050.

Van Hedel, H. J. A., Biedermann, M., Erni, T., Dietz, V. (2002). Obstacle avoidance during human walking: transfer of motor skill from one leg to the other. *The Journal of Physiology, 543*(2), 709.

Vijayakumar, S., Toussaint, M., Petkos, G., Howard, M. (2009). Planning and moving in dynamic environments. In B. Sendhoff, E. Körner, O. Sporns, H. Ritter, & K. Doya, *Creating Brain-Like Intelligence. Lecture Notes in Computer Science* (Vol. 5436, pp. 151–191). Berlin: Springer. doi:10.1007/978-3-642-00616-6_9.

von Hofsten, C. (1980). Predictive reaching for moving objects by human infants* 1. *Journal of Experimental Child Psychology, 30*(3), 369–382.

von Holst, E., & Mittelstaedt, H. (1950). Das Reafferenzprinzip. *Naturwissenschaften, 37*, 464–476.

Welch, G., & Bishop, G. (1995). An introduction to the Kalman filter. Technical Report TR 95-041, University of North Carolina at Chapel Hill, Department of Computer Science.

Wells, J., Hyler-Both, D., Danley, T., Wallace, G. (2002). Biomechanics of growth and development in the healthy human infant: a pilot study. *JAOA: Journal of the American Osteopathic Association, 102*(6), 313.

Wolpert, D. M., Doya, K., Kawato, M. (2003). A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society of London, 358*, 593–602.

Wolpert, D. M., & Ghahramani, Z. (2000). Computational principles of movement neuroscience. *Nature Neuroscience, 3*, 1212–1217.

Generalization and Interference in Human Motor Control

Luca Lonini, Christos Dimitrakakis, Constantin A. Rothkopf, and Jochen Triesch

Abstract Humans (and robots) interacting with the environment have to deal with a continuous stream of sensory inputs in an incremental fashion. Such systems face two fundamental issues: (1) they must acquire new skills in a *cumulative fashion*, that is exploiting previous knowledge to learn new behaviors, and (2) they must avoid the so-called *catastrophic interference*, where learning new knowledge destroys existing memories. Here, we analyze the problem from the perspective of biological motor control. We first review experimental results of consolidation of procedural memories and the factors affecting it. Then the problem of generalization and interference is examined together with some interpretations in terms of computational models. Finally, we present some possible approaches to the issue of learning multiple tasks while avoiding catastrophic interference in bio-inspired learning architectures.

L. Lonini (✉)
Frankfurt Institute for Advanced Studies, Goethe University Frankfurt, Ruth-Moufang Str. 1, 60438 Frankfurt am Main, Germany
e-mail: lonini@fias.uni-frankfurt.de

C. Dimitrakakis
Chalmers University of Technology Sweden, SE-412 96 Gothenburg Sweden
e-mail: dimitrakakis@fias.uni-frankfurt.de

C. Rothkopf
Institute of Psychology Organization: Technical University Darmstadt, Karolinenplatz 5 64289 Darmstadt, Germany
e-mail: rothkopf@fias.uni-frankfurt.de

J. Triesch
Frankfurt Institute for Advanced Studies, Goethe University Frankfurt, Germany
e-mail: triesch@fias.uni-frankfurt.de

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_8,

1 Introduction

The aim of this chapter is to provide an overview of some insights on the issue of consolidation, generalization, and interference of procedural memories provided by recent studies in motor learning. Understanding the brain mechanisms involved in consolidation of a newly acquired skill and how this is integrated with previous knowledge represents a major challenge in neuroscience. Can we draw inspiration from the consolidation processes implemented in biological brains for learning and generalization to design computationally efficient learning algorithms? Multiple research areas could benefit from investigating this issue: as an example, neuro-rehabilitation could design more efficient training to maximize the effect of a therapy on a patient; robotics can draw inspiration for the development of algorithms with greater generalization capabilities to successfully cope with unstructured environments, as is desirable for robots working in close contact with humans (e.g., humanoid robots). The results of these studies, together with the computational models developed so far in the field of motor control, suggest some directions for future work in the design of bio-inspired controllers that need to cope with the problem of acquiring multiple motor skills, during interaction with a variable environment. To this purpose, robotic platforms are going to provide a useful benchmark to test neuroscientific theories (Rucci et al. 1999; Cheng et al. 2007).

The chapter is organized as follows: Sect. 2 reviews findings about consolidation of procedural memories and the factors affecting it; Sect. 3 contains an overview of motor learning studies addressing the issue of generalization and interference; Sect. 4 concludes the chapter by describing the computational approaches proposed in the computational neuroscience and robotics literature to face the problem of learning multiple motor tasks and suggests some future research directions.

2 Consolidation of Procedural Memories

How the brain acquires and retains multiple motor skills is a major topic of research in neuroscience that has been studied from a number of different viewpoints. It is common evidence that we never forget how to drive a car or how to ride a bicycle. Also, new skills can be acquired apparently without interfering with memories of previously learned skills. This form of memory is defined as *procedural memory*, in contrast to the memory for facts or events which is broadly defined as *declarative*. While a complete description of the memory systems is beyond the scope of this chapter, it is worth mentioning that the two forms of memory are thought to rely on different neural structures. Indeed, patients with lesions of the medial temporal lobe (e.g., the hippocampus) are able to acquire novel motor skills, despite being unable to form any semantic or declarative memory. The case of patient HM is paradigmatic for this result (Corkin 1968; Scoville and Milner 1957). Procedural memories are instead thought to rely on the striatum and the cerebellum.

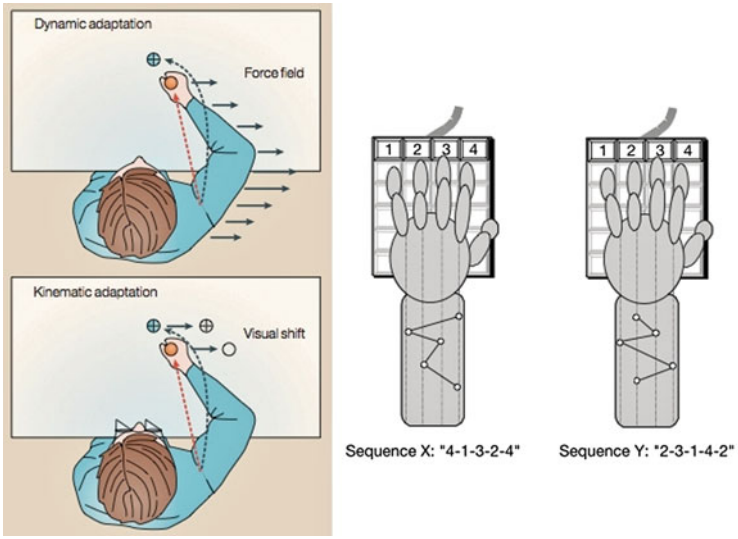

Fig. 1 Experimental paradigms used to assay motor learning and consolidation of motor memories: (*Left, top*) In dynamic adaptation, reaching movements to a target (*blue circle* with a *cross*) are perturbed by a force field generated by a robotic manipulandum (not shown), so that trajectories are initially curved (*blue dotted arrow*). At the end of learning, subjects are able to compensate for the force field imposed by the robot so that trajectories become straight as in unperturbed reaching (*red dotted arrow*); (*Left, bottom*) in kinematic adaptation a visual shift between the position of the hand and that of a displayed cursor is produced (for example through prism goggles), so that subjects have to form a new map between motor commands and hand positions. (*Right*) Finger tapping task used to assess consolidation of sequences. Subjects are required to press the keys according to a specific sequence as quickly and accurately as possible. Reprinted by permission from Macmillan Publishers Ltd: Nature Reviews Neuroscience (Robertson et al. 2004a), copyright 2004 and (Walker et al. 2003), copyright 2003

Despite the apparent ease in the acquisition of new abilities, studies on consolidation of motor tasks have produced a wide range of results with sometimes controversial interpretation, such that the neurobiological and computational mechanisms underlying it are still far from being completely understood. Motor learning is typically assayed through two experimental setups: one involves adaption to a dynamic or kinematic perturbation produced by a robotic device (or by prism goggles) during reaching movements to targets. Subjects will produce curved movements in the first trials, but through practice they will form a new mapping (i.e., an internal model) between sensory and motor commands so that they will finally produce straight trajectories as in the unperturbed environment (Fig. 1, left). The other paradigm used is the finger tapping task, where subjects learn to type a sequence on a keypad as fast and accurately as possible (Fig. 1, right). Sometimes this task is also referred to as the Serial Reaction Time (SRT) task, that includes the association between a visual cue presented on a screen with each of the buttons of the keypad (Robertson 2007).

Consolidation is assessed by measuring *savings* for the newly acquired motor task as well as its robustness to interference with another task. Savings represents a reduction in fragility of a memory (*memory stabilization*) and can be measured as an improved performance compared to naive subjects at the beginning as well as at the end of a new practice session (Fig. 2, right). Also, the rate of relearning can be higher compared to the rate of original learning. Savings are demonstrated also for simple reflexes such as eyelid conditioning (EC) in which a conditioned stimulus (CS, e.g. tone) is paired with an unconditioned stimulus (US, e.g. air puff) such that after learning, the CS will produce a conditioned response (CR) before appearance of the US. Extinction of the CR requires undergoing a few trials of the CS without the US, but relearning the association is faster than the first time and thus shows savings (Medina et al. 2001).

There is also ample evidence that sleep and time play an important role in consolidation of some tasks. Performance on short sequences of finger movements (finger tapping task) proved to increase, after a night of sleep, with respect to the previous performance at the end of learning. In contrast, these improvements were blocked when subjects learned a different sequence before sleep (Walker et al. 2003). This phenomenon is also known as *off-line learning*, that is the improvement of a skill attained when the skill is not practiced (Fig. 2, left). Off-line learning is not always sleep-dependent. In a rotary pursuit task, where subjects were required to hold a stylus on a rotating target for as long as possible, improvements occurred after a 15 min rest between two consecutive sessions. Such improvements are blocked if subjects learned another version of the task before resting, which rules out effects of fatigue as a possible cause of improvements (Eysenck 1965).

Consolidation seems to be sleep dependent in two cases: one is awareness of the task, for example by introducing a cue to signal a specific sequence of finger movements (Robertson et al. 2004b); another is integration of multidimensional cues (Spencer et al. 2006). In contrast, studies on kinematic and dynamic adaptation show no off-line learning at all. It has to be noted that one main difference with respect to a finger tapping task is that subjects will not be exposed to the perturbation anymore at the end of the experimental session, that is they will move in a non-perturbed environment. Indeed, as detailed in the following section, these tasks have been designed so far to assess stability of motor memories as well as generalization of a novel visuomotor transformation.

Finally, stabilization of a memory trace is also measured in terms of resistance to interference to a novel task or a variation of the original task. Typically subjects are trained on a task A and then on a novel task B. After a specific time interval they are re-tested on A to assess savings of initial learning. Performance on re-test on task A can be impaired by two possible causes: one is retroactive interference, that means learning of B has disrupted the initial learning of A; the other is a proactive source that is learning of task B interferes with relearning of task A (Fig. 3). In other words, proactive interference impairs the *retrieval* of memory of A because of learning B and can be prevented by washout of the learning of B (i.e., executing trials in a non-perturbed environment to cancel aftereffects of learning B) (Krakauer et al. 2005).

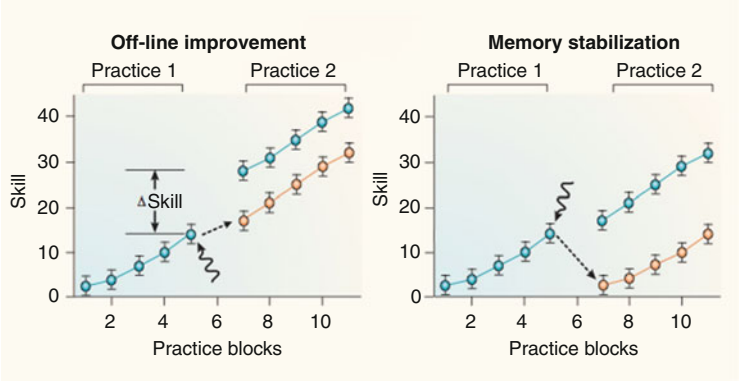

Fig. 2 Procedural consolidation can be measured either as an improvement over time when the skill is not practiced (Off-line improvement, *left, blue line*) or as a stabilization of the memory for that skill, such that performance continues to improve in the second practice session (memory stabilization, *right, blue line*). If memory for the task is disrupted, for example by applying transcranial magnetic stimulation (TMS) at the end of a training session (*curved arrow*), then the improvements are no longer observed (*orange lines*). Reprinted by permission from Macmillan Publishers Ltd: Nature Reviews Neuroscience Robertson et al. 2004a, copyright 2004

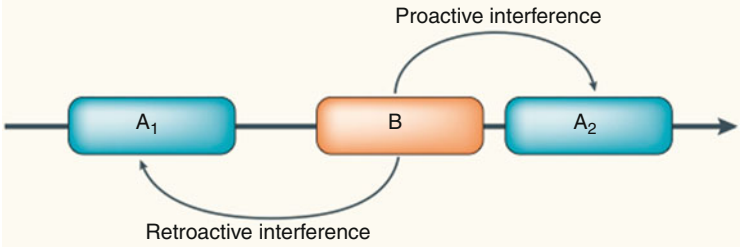

Fig. 3 Retroactive and Proactive Interference. Retroactive interference causes disruption of initial memory of task A because of learning of task B. Proactive interference causes impairment on the retrieval of memory of A induced by learning of B. Reprinted by permission from Macmillan Publishers Ltd: Nature Reviews Neuroscience, Robertson et al. 2004a, copyright 2004

3 Interference and Generalization in Biological Systems

In the last two decades, studies on reaching tasks with robotic devices have offered insights into the possible mechanisms underlying learning, generalization, and consolidation of motor memories. Two main theories were proposed to explain how the Central Nervous System (CNS) might generate fast reaching movements. The *equilibrium point* hypothesis suggests that the CNS relies on the spring-like properties of muscles and peripheral reflex loops that pull the joints back to an equilibrium posture. Such posture is thus determined by the muscles co-contraction level as well as by the reflex gain. By changing these parameters, a movement

could be generated through a sequence of equilibrium positions (Bizzi et al. 1984; Feldman 1966; Hogan 1984). In order to generate fast reaching movements as the ones observed experimentally however, the mechanical stiffness of the arm should remain quite high, which is not the case for the execution of well-practiced movements (Gomi and Kawato 1996). The *internal model* hypothesis, on the other hand, suggests that learning occurs via formation of an internal representation of the kinematics and/or the dynamics of the body and/or the world. Internal models allow the prediction of sensory consequences of motor commands (forward models) as well as the computation of the required motor commands to execute a planned trajectory (inverse models) (Kawato 1999). Internal models can be exploited by the CNS in a number of ways, including compensation for delayed sensory feedback, integration of noisy sensorimotor information and sensory cancellation mechanisms (Shadmehr et al. 2010). Importantly, stiffness of the arm could remain low during the execution of the movement, which resembles the experimental observations.

Such studies also highlighted how catastrophic interference can stem from practice of multiple motor or visuomotor adaptation tasks. A summary of key findings is given in Table 1. For instance, after an internal model of a force-field (A) is learned, it can be recalled in the future. However, savings are prevented if, after learning A, an opposite force-field B is presented to the subject, no matter what the time interval between A and B (Caithness et al. 2004). However other studies reported different results, showing that the presence of "catch trials," i.e. trials in which the force-field is unexpectedly turned off, produced savings of A (Overduin et al. 2006). Interestingly, a unifying view has been recently proposed that motor learning can be the result of multiple systems learning in parallel at different timescales. Here, a fast process yields not only rapid learning but also rapid decay; a slower process requires more trials for a given amount of adaption but it will decay slowly (Huang and Shadmehr 2009; Smith et al. 2006). A possible motivation is that the brain estimates how likely it is that the environment will change over time. Such an estimate will influence memory decay for that context. Thus, the goal of the learner is to estimate the most likely timescale which caused the observed error. For example, a fast timescale disturbance can be related to muscle fatigue, while a slow timescale disturbance can be due to illness or age-related growth (Kording et al. 2007). Criscimagna-Hemminger and Shadmehr (2008) assayed decay of motor memories when subjects learned a single force-field A, or when they learned A followed by a brief exposure to an opposite force-field B. At the end of training, memory of the task is measured through error-clamp trials, that is the robot constrains the subject to move into a virtual channel while recording the lateral forces she exerts. This prevents relearning of the force field and at the same time allows to measure the amount of remaining memory, when subjects are re-exposed to the task (reactivated memory). Results showed that memory of A and B both exist and can be decomposed into a fast and a slow state: the fast state is highly sensitive to errors but has poor retention, while the slow state has poor sensitivity to errors but shows large retention. Additionally, fast states are gradually transformed into more stable states and depending on whether the subjects are exposed (active state) or not (inactive state) to the task, the memory will display different decay

Table 1 A summary of the main findings about the role of generalization and interference in motor learning experiments

Task	Paradigm	Main results	Refs.
Force field learning (reaching)	A-B-A (reaching in A, followed by B and then A); A and B opposite force fields	Fast and slow memory systems; Interference stems from the interaction between fast and slow memory	Caithness et al. 2004; Smith et al. 2006; Criscimagna-Hemminger and Shadmehr 2008
Force Field learning (reaching)	A-B randomly interleaved with contextual cues	Reduced interference between A and B	Osu et al. 2004
Visuo-motor learning (reaching)	Training on randomly changing rotations (e.g. uniformly distributed between $-90°$ and $+90°$)	Reduced interference between opposite rotations (e.g. $-60°$ and $+60°$) because of structure learning; Generalization to transformations of same structure (e.g. horizontal rotations)	Braun et al. 2009
Visuo-motor learning	Training arm and wrist on opposite rotations (A and B)	No interference between A and B when learned by separate body parts	Krakauer et al. 2005; Krakauer et al. 2006
Motor sequence (finger tapping task)	X-Y (X, Y two sequences)	Overnight improvements in accuracy are blocked when training of X is followed by Y	Walker et al. 2003
Arm movement patterns	Training different arm movements under a blocked or random schedule	Random training favors retention and generalization compared to a blocked training schedule	Shea and Morgan 1979; Kantak et al. 2010

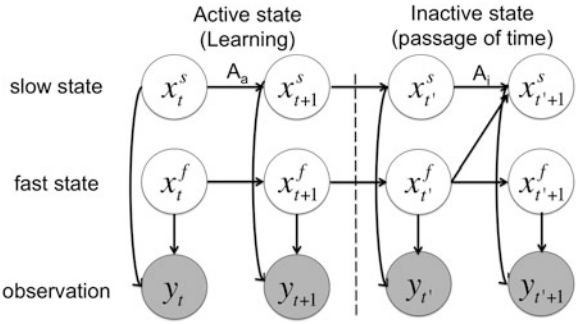

Fig. 4 Generative model of decay of motor memories made up of 2 states (fast and slow) and two working conditions (Active and Inactive memory) (Criscimagna Hemminger and Shadmehr 2008). A_a is the transition matrix for the active state (memory decay during reactivation, i.e. error-clamp trial); A_i is the one for the inactive state (memory decay with time, i.e. subjects not engaged in the task). A_i is non-diagonal to account for the result that an initially fast memory gains stability with passage of time. The observation is the position of the hand in the perturbed environment

rates. They used a two state (fast and slow timescale) linear generative model for each condition (active and inactive state) to fit decay of motor memories obtained from experimental data (Fig. 4).

We must always expect some interaction between previous learning experiences and learning a novel task. Generalization refers to transfer of knowledge from previously acquired skills into a new context: sometimes learning one task makes it easier to learn another, such as the ability to drive different cars after having learned to drive one; on the other hand learning to play badminton might produce unwanted generalization if we want to learn tennis. Bayesian models constitute a principled way to update previous knowledge when new evidence is observed and can be used to compute optimal decisions in the face of uncertainty. It has been proposed that the brain performs Bayesian inference to estimate the sources of errors and thus update internal models (Berniker and Kording 2008; Wei and Kording 2009). According to the same framework, Krakauer and colleagues (Krakauer et al. 2006) tested whether learning of a visuomotor rotation task with the arm will transfer to the wrist and vice versa. They found that training with the arm will benefit subsequent training on the wrist, but not the other way around. Secondly, opposite transformations (i.e., clockwise and counterclockwise) can be learned without interfering with each other as far as they are learned by separate body parts (i.e., the arm and the wrist). They conjecture that the limb segment is a contextual cue for the task and the learner is affected by the history of how she has used different body parts. According to these hypotheses, they propose a Bayesian model to explain experimental data.

Additionally, several psychology studies have focused on the issue of practice scheduling and how it affects motor retention: it is well known that, when training, presenting tasks in random order leads to a higher retention than presenting one task at a time (contextual interference effect). In their pioneering study, Shea and Morgan (1979) showed that subjects who learned three rapid arm movement patterns

under a random presentation performed better at a retention test (at 10 min and 10 days) than subjects who learned each of the three tasks in a separate training block. The improved performance of the random group at retention occurred even if the blocked group improved faster than the random group during training. Also, the random group performed better both when tested in the same training condition (random) and in the opposite training condition (blocked). The benefits of interleaved versus blocked practice on retention were observed on multiple learning domains, including perceptual learning (Mitchell et al. 2008) and verbal learning (Battig 1972). Of notice, Osu et al. (2004) found that people are able to acquire two opposing force fields when they are presented in a random order together with contextual cues. Despite the large number of studies addressing these results, the learning mechanisms leading to it are still a matter of debate.

Recently, the idea of structural learning has been proposed as a possible strategy exploited by the CNS to reduce the number of control parameters that have to be learned when experiencing new environments that share some structure with previously encountered ones (Braun et al. 2009). Specifically, two groups of participants were tested on reaching movements toward four targets in a three-dimensional virtual reality environment. Each group experienced either vertical or horizontal rotations, where the rotation angle changed randomly every four reaches between $-60°$ and $+60°$. After learning, the participants experienced four trials in a non-perturbed environment (to cancel aftereffects from previous trials) and finally they learned a $45°$ rotation both in the vertical and horizontal direction. Had subjects acquired the structure of the rotation (i.e., horizontal or vertical) interference would have been reduced only for trials of the same structure. Indeed the results showed this effect as well as that the variance of the movements for iso-structural trials (with same structure as the one previously learned) is enhanced, that is each group explored preferentially in the direction of the learned structure. Thus, an interesting suggestion is that random training reduces interference only when the practiced tasks belong to the same structure. Hierarchical Bayesian models can provide an interesting computational tool to develop structure learning (Braun et al. 2010) and can also be tested with robotic platforms.

4 Multi-task Learning

In this section we want to address the general issue of learning multiple tasks in an artificial system while minimizing negative interference between them. We will focus on two specific aspects: first we formalize the general problem in the context of machine learning; second, we will review how some of these issues are addressed within the reinforcement learning literature and by models of human motor control. The aim is to provide a reference for the development of novel strategies for cumulative learning based on biological findings.

4.1 Decision Theory View

The problem of learning multiple tasks poses several interesting computational challenges. The most fundamental of these is how to detect the existence of a new task. The second is how to share knowledge between tasks. In both cases, there are some prior beliefs that we must necessarily rely on in order to make an appropriate decision with respect to task learning. It is easy to see that determining the existence of a new task must necessarily depend on how likely we think the appearance of a new task is. Also, we must somehow quantify how useful knowledge of one task is when attempting to learn another task. As we shall see, there are intrinsic connections between those two aspects of our belief which are due to the structure of the complete set of tasks. While these concepts are further explored in Dimitrakakis (2012), this section gives a brief overview.

We begin by describing a general setting, where the agent is given a sequence of tasks. At the beginning of each stage in the sequence, the decision maker is given a task $c \in C$ and must select a policy π from a set of policies P. Execution of the policy results in an outcome $x \in X$.[1] The decision maker may have different preferences for different outcomes and policies. These preferences will depend on the task. We can formalize this through the definition of a function $U : C \times X \times P \to \mathbb{R}$, such that we prefer policy π to some other policy π', given that the task is c and we observe x, if and only if $U(c, x, \pi) \geq U(c, x, \pi')$. In the general setting, x may be selected by another agent, or it maybe sampled stochastically from a distribution depending on π and c. Assuming the latter setting, we will be interested in the *expected utility*:

$$\mathbb{E}(U \mid c, \pi) = \sum_{x \in X} U(c, x, \pi)\, \mathbb{P}(x \mid c, \pi). \tag{1}$$

In particular, we define π_c^* as the policy that maximizes the expected utility for task c, so that $\mathbb{E}(U \mid c, \pi_c^*) \geq \mathbb{E}(U \mid c, \pi)$, $\forall \pi \in P$.

In general, there is no single policy that is optimal for all tasks. However, it could be that a single policy is nearly-optimal for more than just one task, depending on the structure of C and U. The first important consideration is the size of C. One may assume that the number of tasks is finite, countably or uncountably infinite.[2] Secondly, we must consider how the given tasks change from stage to stage. This is necessary if we wish to differentiate between observing a subtly different task from a previously known task, or a slight change in the previous task. In addition, while

[1] The policies and the outcomes may be complex. For example, the outcomes may actually be sequences of elementary observations, while the policies may be made up of sequences of elementary actions.

[2] A set is *countable* if every element of the set can be associated with a natural number. An example of countable set is the set of natural numbers $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$. The set $\mathbb{R}$ of real numbers is an example of *uncountable* set.

we are still learning a task, we shall need to differentiate between performing badly because we have not yet learnt the task adequately, and performing badly because the task is novel. The above considerations are closely linked to classical problems in statistical learning, novelty detection and outlier detection. In fact, all algorithms that attempt to learn new tasks on the fly are implicitly making some of the above assumptions and performing some form of novelty detection.

The next consideration is whether there is some additional structure in the space of tasks. Ideally, we would like the set of tasks C to be equipped with some metric ρ, such that the distance between two tasks $c, c' \in C$ is $\rho(c, c')$. If tasks that are close to each other also have close optimal decision functions, then it is much easier to learn new tasks. In fact, one may make suitable assumptions such that, if two tasks c, c' are close to each other, then the utility of employing the same policy π to either task is also close. Thus, a policy which is optimal for some task c will be nearly optimal for another task c'. One way to express this mathematically is via a *Lipschitz* property on the space of tasks. If there is no structure in the problem, for example if $\rho(c, c') = 1$ for all $c \neq c'$, then the amount of knowledge that can be transferred is limited.

For each possible class of task structures there is a different class of learning mechanisms which is optimal. An important question is whether it is possible to find learning mechanisms that are robust across a wide variety of multi-task learning problems. Any such mechanisms will be required to include a computation for representing the number of tasks present, as well as its uncertainty about the number of tasks; structures that represent the relations between tasks and how they are selected and finally a way to construct a policy given the current knowledge about tasks. One would expect that in different settings, different approaches for gathering information would be required.

The connection between the value of information gathering and multi-task problems has been recognized at least as far as the seminal work of Lindley (1956) in the context of experimental design. Therein, the problem of obtaining as much as information as possible, when having no particular task to perform, is stated in statistical terms. One simple idea that explicitly connects multi-task problems and curiosity is given in (Dimitrakakis 2012), which introduces the setting of *sparse reward processes*. There, the agent is playing a multi-stage game against an unknown opponent. The agent is always in the same unknown environment, at each stage of the game, he is given a different task to solve by the opponent. The paper shows that in order for the agent to perform well when the opponent is adversarial, he must try and obtain more information about its current environment, rather than immediately trying to obtain the available reward. Then, the knowledge gained while performing the current task will be potentially useful in performing further tasks. However, it is also shown that when the sequence of tasks is very random, then no additional gain is obtained by explicit curiosity, as the agent is always forced to explore in order to achieve the current goal.

Similar ideas are directly applicable to tasks that can be hierarchically decomposed. There, the task space C obtains structure directly through the existence of the hierarchy. If tasks can be decomposed into sub-tasks, then learning a given

sub-task is potentially useful for future tasks, even performing a particular sub-task is currently of no value to the agent.

4.2 Multiple Tasks in Reinforcement Learning

The problem of a learning agent solving multiple tasks has been explored in a large number of studies within the area of reinforcement learning (RL) (see, for example, Bertsekas and Tsitsiklis 1996; Sutton and Barto 1998). In the most general form, RL describes the problem of an agent interacting with a dynamic environment by performing actions according to the sensed information (the state). The goal of the agent is to maximize the rewards (or minimize the punishments) delivered by the environment as a result of its actions. Importantly, the environment does not provide the agent with the correct actions to be performed, which distinguish RL from supervised learning approaches. In fact, the reinforcement learning problem can be cast in the setting discussed in the previous section. Specifically, in fully-observable RL problems, the observations correspond to sequences of the true state of the world, i.e. $x = s_1, \ldots, s_t, \ldots$, where each state s_t belongs to some state space S. A policy π maps sequences of states to elementary actions $a_t \in A$. After an action a_t is taken at time t, the agent observes a new state s_{t+1} and obtains a reward $r_{t+1} \in \mathbb{R}$.

Formally, the RL problem is closely related to a class of statistical models called Markov Decision Processes (MDPs) (see, for example, Puterman 1994). The constituents of an MDP M are the set of possible states S, and the set of possible actions A, as well as a transition and a reward distribution. The first specifies the transition probabilities $P_M(s'|s,a)$ of going to a new state s' when being in state s and carrying out action a, i.e. $s_{t+1} \mid s_t = s, a_t = a \sim P_M(s'|s,a)$. The second is the distribution of rewards $R_M(r|s,a,s')$ that are obtained for being in state s, carrying out action a, and going to the new state s', i.e. $r_{t+1} \mid s_{t+1} = s', s_t = s, a_t = a \sim R_M(r|s,a,s')$. Finally, the utility is defined as the discounted sum of obtained rewards $\sum_{t=0}^{\infty} \gamma^t r_t$, where $\gamma \in [0,1]$ is a discount factor. Consequently, the expected utility is:

$$\mathbb{E}(U \mid M, \pi) = \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, M, \pi \right). \tag{2}$$

In reinforcement learning problems, typically P_M and R_M are unknown and must be discovered by interaction with the environment. However, the general solution methods are largely the same. In fact, in a Bayesian setting, reinforcement learning can be reduced to an MDP (DeGroot 1970).

By looking at each MDP M as a different task, one can see (2) as a function that gives us the value of a particular policy π for each task. Thus, the MDP formalism is very well suited for capturing task inter-dependencies. For that reason, solving multiple tasks has been considered in various RL scenarios. Some authors

have investigated problems in which an agent is exposed to a succession of tasks with different reward functions between tasks, while the transition function remains constant. Similarly, other authors have considered the case in which the tasks differ only in the underlying transition function or cases in which both the reward and transition function change (see Taylor and Stone 2009 for a comprehensive review). In terms of motor control experiments, different types of perturbations to which subjects are exposed can be expressed as different transition functions. While in one task the applied perturbation may implement a rotation $+60°$ a second task may apply a rotation of $-60°$ degrees. Similarly, changing the target of a reach in such experiment can be modeled as having different reward functions where reward is given out at the goal state of the reach. In the following we will give two examples that exemplify approaches from this literature.

The case in which the reward function changed across different tasks was considered by Atkeson and Santamaria (1997). The transition function was constant across all task instances and the learning agent learned a locally weighted regression of this transition function. Whenever the agent encountered a new task, it used new observations of the rewards together with the learned transition function to plan an optimal policy in order to maximize its reward. Accordingly, the agent could take advantage of the learned system transition function and then performed better than a learner that started without prior knowledge. This advantage in performance was seen in the initial reward, the total cumulative reward, and also in the asymptotic performance. The specific advantage in this case was obtained because the agent learned an explicit model of the environment, in this case the dynamics, which could be transferred across tasks.

A different approach to building up knowledge about the structure of the transition dynamics was developed by Ferguson and Mahadevan (2006). Here the idea is to use the so-called proto-value functions (Mahadevan and Maggioni 2007), which are task independent basis functions for all possible value functions on a particular state space and the dynamics governing the state transitions within that state space. These proto-value functions are obtained by spectral decomposition of random walks on the state space graph. As they encapsulate the geometrical structure of the state transition graph, they lead to the ability to transfer knowledge and representations between related tasks. Accordingly, learning of approximately optimal policies for new tasks can take advantage of the knowledge about the state space that is contained in these proto-value functions.

Many more problems can be formulated, including multiple tasks resulting from differences in the start state, the goal state, the set of available actions, or the set of state variables. A significant example is the *option* framework (Sutton et al. 1999) which is based on the use of macro-actions, i.e. sequence of actions that can be invoked in a certain state. Concisely, an option is defined by: the option policy π_0, an initiation set I_0 which is the set of states under which the option can be executed, and a termination condition β_0, giving the probability of terminating the action in a given state. Options allow to simplify the control problem and constitute a natural way of creating hierarchies across tasks (see also Barto and Mahadevan 2003).

Furthermore, an important distinction between different solutions to multiple task RL is the evaluation criterion. While it may be of interest to evaluate the success of a multiple task method by the initial offset in performance when starting a new task, it may also be the final performance, or the total reward accumulated while carrying out a task, or the time required to reach a performance criterion.

Finally, it is worth mentioning the link that has been proposed between intrinsically motivated behavior and the acquisition of cumulative skills. Although the meaning of intrinsic vs. extrinsic rewards is not uniquely defined, psychologists consider a behavior being intrinsic when it is not directly aimed at an explicit external goal, but is inherently enjoyable (e.g., exploration, play). Importantly, such behaviors can be later used as building blocks to acquire new skills that produce an extrinsic reward.

Barto et al. (2004) showed that the introduction of intrinsically motivated actions in an artificial agent can lead to the development of hierarchical skills. They build the notion of intrinsic rewards on top of the option framework. These intrinsic rewards are generated by unexpected "salient" events that allow the agent to learn basic skills (e.g., create options); some of these skills are needed to attain the overall task (e.g., the extrinsic reward). An intrinsic reward is proportional to the error in predicting the corresponding event, according to the model learned for that option (see also Schmidhuber 1991 for a similar idea). As a consequence, when the agent is "bored" it will be less likely that it will perform the action. Indeed, this idea is inspired by the role of the neuromodulator Dopamine in biological systems, which is released when an unpredicted reward is delivered. As the event becomes more and more familiar, the dopamine burst is extinguished (Dayan and Balleine 2002). They show that an artificial agent endowed with such intrinsic rewards is likely to acquire simpler skills first. Subsequently, the agent can exploit some of these skills to attain the extrinsic reward (i.e., the overall task).

More recently, Singh et al. (2010) proposed the emergence of intrinsically motivated behaviors as a result of an *evolved* reward function. In their framework, they perform an evolutionary search across reward functions that maximize the cumulative fitness of the agent across a distribution of environments. The optimal (evolved) reward function can lead to the emergence of behaviors that are not directly related to the fitness function. They demonstrate the idea in a simple grid world: here an agent that is initially rewarded for opening boxes (intrinsically motivated behavior) could exploit such ability later, if such boxes will contain food, which increase the fitness of the agent. Thus, although the manipulation skill is not directly related to the fitness initially, it will later allow the agent to increase its fitness.

4.3 Motor Control Models

From a neuro-computational perspective, the problem of motor control has typically been formulated as the learning of a mapping from some sensory space $x \in X$ to a motor command $y \in Y$, with $y = \sum_i w_i g_i(x)$, where $g_i : X \to Y$ are fixed basis

functions and $w_i \in \mathbb{R}$ are adjustable parameters of the mapping (Poggio and Bizzi 2004).

Such a scheme finds a biological foundation in the idea of *motor synergies*, that are a combination of a small number of muscle activations to generate the variety of motor behaviors (Thoroughman and Shadmehr 2000). One of the biological findings that inspired such a theory has been provided by Mussa-Ivaldi et al. (1994) and colleagues (d'Avella et al. 2003). They showed that simultaneous stimulation of distinct sites of the frog's spinal cord leads to a vectorial summation of the forces generated at the frog's ankle by stimulation of each site separately. The idea is appealing both for the neurobiology and the robotics domain as a way of simplifying the control of a system with a high number of degrees of freedom. However the existence of such motor synergies is still a matter of debate, as other studies did not find evidence for their existence (Valero-Cuevas et al. 2009).

As a matter of fact, many models of motor adaptation assume that learning modifies the weights w_i of the basis functions g_i through gradient descent. However, training such a model over multiple patterns (tasks) will produce dramatically different results from biological evidence. For example, as seen for the structural learning framework, training a visuomotor mapping with random rotations facilitates acquisition of a novel rotation of the same type (Braun et al. 2009). In contrast, adaptation of the weights w_i of the model with gradient descent will produce the average mapping to be learned (i.e., the average over all rotation angles experienced). This is a common issue of connectionist models, known as *Catastrophic Forgetting*, that is learning of new information will destroy previously acquired knowledge. Different solutions have been proposed to the problem of catastrophic interference (for a review, see French 1999). Some propose the use of dual-network models: one for new information and the other for long-term storage. The exchange of information between the two networks is used to rehearse patterns of already acquired information using the idea of pseudo-pattern transfer (French 1997). Other authors proposed the idea of using different learning rates for the two networks (Westermann and Mareschal 2008) or weights that are updated at two different timescales (Hinton and Plaut 1987). These approaches are thought to resemble the neurobiology of the memory systems inside the brain as in the separation between the hippocampus and the neocortex (Norman and O'Reilly 2003). It is worth noting that these solutions share again the idea of multiple timescales as proposed for motor learning processes. While they have been mostly proposed in the context of declarative memory or cognition models, an extension of them to the domain of motor learning is still lacking and could provide an interesting area of future research for learning multiple skills without interference.

Modular architectures based on *mixture of experts* (MEX) (Gomi and Kawato 1993; Jacobs et al. 1991) or multiple internal models (Wolpert and Kawato 1998) have been proposed in the context of motor control. In the MEX there are multiple function approximators (experts) whose outputs are linearly combined by a classifier (a gating module) that has the role of a centralized switch. Each expert fits the data over a local region of the domain and the gating module determines the degree

of contribution of each expert. The selection is centralized in the gating module and thus is independent of the activity of the expert modules. The Modular Selection and Identification for Control (MOSAIC) model (Haruno et al. 2001) has been proposed on the idea that the brain prepares several modules each containing a forward and an inverse model. Each module specializes in a given task (i.e., controlling a specific object) and the individual outputs of the inverse models are mixed to generate the motor command. The prediction error of each forward model is used to compute the responsibility signal according to a soft-max function. The forward model with the smallest error indicates that its paired inverse model is appropriate for controlling the current context (Fig. 5). MOSAIC is an extension of the mixture of experts model. Each expert contains a one-step predictor of the next system state and module switching is related to the predictive performance of each module. Thus, the switching function depends on each single expert prediction instead of being centralized into a single gating module that splits the input space independently of the single experts. Variations of MOSAIC have been proposed also in the context of Reinforcement Learning (Doya et al. 2002), and hierarchical control (Emadi Andani et al. 2009; Haruno et al. 2003), but most of them have been validated only in simulated environments. Thus, an important benchmark for these approaches is their implementation on robotic platforms for learning of multiple motor skills. To do so, efficient methods that deal with high-dimensional data and online learning must be addressed. Also, it has to be noted that an issue for these architectures is that the number of modules has to be specified a priori: this poses the nontrivial problem on how to choose these modules to cover all possible tasks and adequately generalize over them. This issue is of utmost importance in an architecture that needs to learn from a continuous stream of sensory data in an incremental fashion.

Locally Weighted Learning (LWL) methods based on local linear regressions and receptive fields, such as Locally Weighted Projection Regression (LWPR), proved to be a feasible way to implement incremental learning of sensorimotor mappings on robots (Vijayakumar et al. 2005). The core idea in this approach is a piecewise approximation of a function with local regressions, each valid in a region of the input domain defined by a Gaussian kernel (i.e., a receptive field). New models are added whenever an input data point does not activate any of the existing models above a given threshold, thus allowing for incremental learning without pre-specifying the number of local models. Stochastic gradient descent is used to adjust the weighting kernels by minimizing a cross-validation error independently for each local model. Additionally input data is reduced through partial least squares to reduce the computational burden of the algorithm so that it can be efficiently used in real time robot learning. In contrast to global learning methods, one interesting feature of LWPR is the use of a local learning strategy that is used to minimize negative interference with previously acquired data. Such an architecture proved to be a viable alternative to learn multiple motor tasks in an incremental way (Lonini et al. 2009) without pre-specifying the number of internal models or the structure of the learning system. Of notice, LWPR is one of the few architectures that have

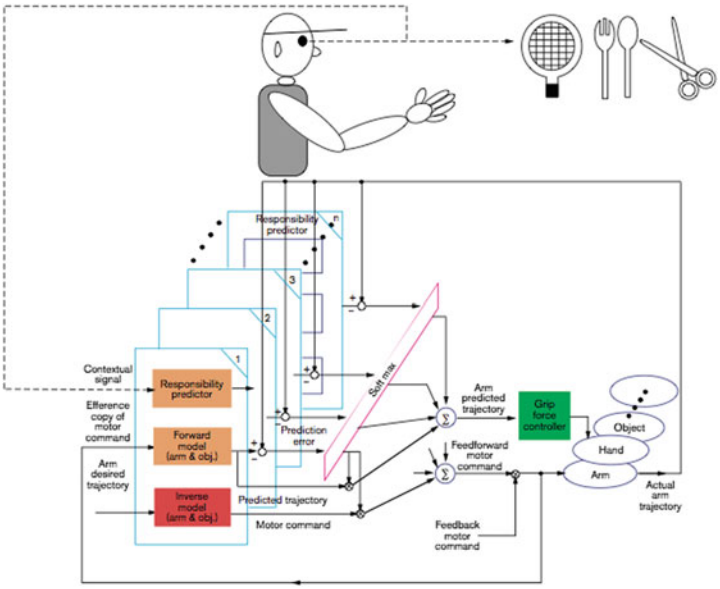

Fig. 5 MOSAIC model general architecture. Each module of the architecture is specialized on a motor task. The control action is made up of a feedback and a feedforward component. The inverse model of each module provides the feedforward component. The output of each module is weighted using a soft-max function computed on the predictions of the forward models and the responsibility predictors. Reprinted with permission from (Kawato 1999)

been tested on robotic platforms in contrast to other architectures which have been validated only in simulated settings.

Finally, a notable work that integrates the notions of motor synergies with the idea of timescales for learning multiple behaviors is the work done by Yamashita and Tani (2008). Instead of representing explicitly a hierarchical structure with local modules and a gating module, they proposed a fully continuous time recurrent neural network (CTRNN) with two distinct types of neurons having two different timescales (fast and slow). The network is trained to predict the proprioceptive and visual input of a humanoid robot interacting with an object, using a supervised learning algorithm (i.e., back-propagation through time). Training on different sensorimotor patterns produces the interesting result that fast neurons encode sensorimotor patterns (primitives) that are reused for different tasks, while slow context neurons encode switching between these patterns. The network is ultimately able to control the humanoid robot executing five different types of behavior, such as moving an object into different positions and clapping its hands. Thus, encoding of primitives and their switching emerges as a self-organization of the network, instead of being pre-defined through a discrete set and a switching module, as in previous approaches (e.g., MOSAIC) (Fig. 6). This idea is appealing since it allows a dynamical creation and modification of the primitives during learning of tasks.

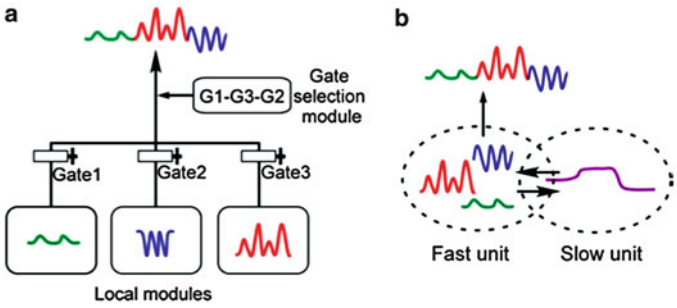

Fig. 6 (**a**) Local (modular) representation for multiple motor behaviors (motor synergies). The combination of multiple synergies is used to produce a complex motor behavior (**b**) Representation of the same synergies through two different timescales encoded in distributed architectures (i.e. neural networks) that interacts and generates the same complex behavior. Reprinted from Yamashita and Tani (2008)

5 Discussion

Motor learning is a cumulative process that requires efficient exploitation of previously acquired skills (i.e., generalization) in face of the ability to minimize negative interference between new and old skills. A number of studies in psychology and neuroscience have addressed the issue of consolidation of motor memories revealing that multiple processes at different timescales as well as sleep or passage of time play a role in storing of new information. Artificial neural networks with multiple timescales or novelty detection mechanisms have been used to solve the problem of catastrophic interference when learning new patterns on top of old ones. However most of them have been confined to the domain of cognitive science. Regarding motor control, generation of movements through a combination of a fixed number of primitives has been proposed both in neuroscience and robotics as a biologically principled way to produce different motor behaviors and generalize over them. According to this view, modular hierarchical architectures were implemented in several models of biological motor control. Most of them define a crisp segmentation of the primitives through a fixed number of modules and a gating mechanism that learns to split data for each module according to an explicitly defined rule. In contrast, architectures exploiting the idea of multiple timescales without pre-specifying the number of primitives can be a promising solution to be explored both in robotic systems and computational neuroscience. Furthermore, Bayesian models are likely to play an important role as a way to explain how the brain updates previous beliefs in light of new evidence.

Finally, we mention some open challenges, in the area of computational motor control, that are relevant for the topics discussed in this chapter.

– *Neurobiological correlates*. Although many studies are elucidating the computational mechanisms that could be used by the nervous system for motor

learning, knowledge of the neurobiological underpinnings for such mechanisms is still lacking. As an example, we mentioned the findings about the fast and slow component of the motor memory. While some recent experimental studies showed the existence of different timescales in the change of activity of neural populations of neurons (Mandelblat-Cerf et al. 2011), the neuronal basis of the fast and slow motor memory is still unclear. Future studies should address this question by combining experimental and computational techniques.

– *Relations between different memory systems*. At the beginning of this chapter, we have mentioned the separation between procedural and declarative memory. Remarkably, a recent study showed that a declarative task could interfere with the fast memory, but not with the slow memory, during execution of a motor task (Keisler and Shadmehr 2010). This result points out the importance of understanding the interactions between different memory systems, to shed light on the mechanisms of generalization and interference.
– *Learning of multiple motor tasks*. Some experiments showed how generalization across multiple tasks can be interpreted using the idea of structural learning. Although an interesting hypothesis, the computational as well as the neural underpinnings are still missing. Revealing the existence of motor synergies that reflect structural learning could be an interesting benchmark for the hypothesis. On the other hand, these studies could be linked to the curse of dimensionality problem that arise in reinforcement learning applied to robot control. As a matter of fact, hierarchical models appear as a natural solution to the problem of structural learning.
– *How rewards contribute to motor memory*. Rewards are indeed a fundamental component of learning in biological systems. While the effect of rewards on learning has been extensively studied in classical conditioning or decision-making tasks, there are only a few studies linking the effect of rewards on motor learning and generalization. One of these studies (Pekny et al. 2011) described the effect of withholding rewards in the retrieval of competing motor memories. Thus, the role of reinforcers in the acquisition and retention of motor skills is a topic of major importance.

References

Atkeson, C., & Santamaria, J. (1997). A comparison of direct and model-based reinforcement learning. In *Proceedings., 1997 IEEE international conference on robotics and automation, 1997* (vol. 4, pp. 3557–3564). New York: IEEE.

Barto, A., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems, 13*(4), 341–379.

Barto, A., Singh, S., Chentanez, N. (2004). Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd international conference on development and learning*. ICDL. (pp. 112–119).

Battig, W. (1972). Intratask interference as a source of facilitation in transfer and retention. In R. F. Thompson & J. F. Voss (Eds.), *Topics on Learning and Performance*. New York: Academic Press. (pp. 131–159).

Berniker, M., & Kording, K. (2008). Estimating the sources of motor errors for adaptation and generalization. *Nature Neuroscience, 11*(12), 1454–1461.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*. Belmont: Athena Scientific.

Bizzi, E., Accornero, N., Chapple, W., Hogan, N. (1984). Posture control and trajectory formation during arm movement. *The Journal of Neuroscience, 4*(11), 2738–2744.

Braun, D., Aertsen, A., Wolpert, D., Mehring, C. (2009). Motor task variation induces structural learning. *Current Biology, 19*(4), 352–357.

Braun, D., Waldert, S., Aertsen, A., Wolpert, D., Mehring, C., Gribble, P. (2010). Structure learning in a sensorimotor association task. *PLoS ONE, 5*(1), e8973.

Caithness, G., Osu, R., Bays, P., Chase, H., Klassen, J., Kawato, M., Wolpert, D., Flanagan, J. (2004). Failure to consolidate the consolidation theory of learning for sensorimotor adaptation tasks. *Journal of Neuroscience, 24*(40), 8662–8671.

Cheng, G., Hyon, S., Morimoto, J., Ude, A., Hale, J., Colvin, G., Scroggin, W., and Jacobsen, S. (2007). CB: a humanoid research platform for exploring neuroscience. *Advanced Robotics, 21*(10), 1097–1114.

Corkin, S. (1968). Acquisition of motor skill after bilateral medial temporal-lobe excision. *Neuropsychologia, 6*(3), 255–265.

Criscimagna-Hemminger, S., & Shadmehr, R. (2008). Consolidation patterns of human motor memory. *Journal of Neuroscience, 28*(39), 9610–9618.

d'Avella, A., Saltiel, P., Bizzi, E. (2003). Combinations of muscle synergies in the construction of a natural motor behavior. *Nature Neuroscience, 6*(3), 300.

Dayan, P., & Balleine, B. (2002). Reward, motivation, and reinforcement learning. *Neuron, 36*(2), 285–298.

DeGroot, M. H. (1970). *Optimal statistical decisions*. New York: Wiley.

Dimitrakakis, C. (2012). Sparse reward processes. Technical Report arXiv:1201.2555v1, EPFL.

Doya, K., Samejima, K., Katagiri, K., Kawato, M. (2002). Multiple model-based reinforcement learning. *Neural computation, 14*(6), 1347–1369.

Emadi Andani, M., Bahrami, F., Jabehdar Maralani, P., Ijspeert, A. (2009). MODEM: a multi-agent hierarchical structure to model the human motor control system. *Biological Cybernetics, 101*(5), 361–377.

Eysenck, H. (1965). A three-factor theory of reminiscence. *British Journal of Psychology, 56*(163–181), 50–130.

Feldman, A. (1966). Functional tuning of the nervous system with control of movement or maintenance of a steady posture. ii. controllable parameters of the muscle. *Biophysics, 11*(3), 565–578.

Ferguson, K., & Mahadevan, S. (2006). Proto-transfer learning in markov decision processes using spectral methods. In *Computer Science Department Faculty Publication Series* (p. 151). Amherst: University of Massachusetts.

French, R. (1997). Pseudo-recurrent Connectionist Networks: An Approach to the "Sensitivity-Stability" Dilemma. *Connection Science, 9*(4), 353–380.

French, R. (1999). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences, 3*(4), 128–135.

Gomi, H., & Kawato, M. (1993). Recognition of manipulated objects by motor learning with modular architecture networks. *Neural Networks, 6*(4), 485–497.

Gomi, H., & Kawato, M. (1996). Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement. *Science, 272*, 117–120.

Haruno, M., Wolpert, D., Kawato, M. (2001). Mosaic model for sensorimotor learning and control. *Neural Computation, 13*(10), 2201–2220.

Haruno, M., Wolpert, D., Kawato, M. (2003). Hierarchical mosaic for movement generation. In *International Congress Series* (vol. 1250, pp. 575–590). Amsterdam: Elsevier.

Hinton, G., & Plaut, D. (1987). Using fast weights to deblur old memories. In *Program of the ninth annual conference of the cognitive science society* (pp. 177–186). London: Lawrence Erlbaum.

Hogan, N. (1984). An organizing principle for a class of voluntary movements. *The Journal of Neuroscience, 4*(11), 2745–2754.

Huang, V., & Shadmehr, R. (2009). Persistence of motor memories reflects statistics of the learning event. *Journal of Neurophysiology, 102*(2), 931–940.

Jacobs, R., Jordan, M., Nowlan, S., Hinton, G. (1991). Adaptive mixtures of local experts. *Neural Computation, 3*(1), 79–87.

Kantak, S., Sullivan, K., Fisher, B., Knowlton, B., Winstein, C. (2010). Neural substrates of motor memory consolidation depend on practice structure. *Nature Neuroscience, 13*(8), 923–925.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology, 9*(6), 718–727.

Keisler, A., & Shadmehr, R. (2010). A shared resource between declarative memory and motor memory. *The Journal of Neuroscience, 30*(44), 14817–14823.

Kording, K., Tenenbaum, J., Shadmehr, R. (2007). The dynamics of memory as a consequence of optimal adaptation to a changing body. *Nature Neuroscience, 10*(6), 779–786.

Krakauer, J., Ghez, C., Ghilardi, M. (2005). Adaptation to visuomotor transformations: consolidation, interference, and forgetting. *Journal of Neuroscience, 25*(2), 473–478.

Krakauer, J., Mazzoni, P., Ghazizadeh, A., Ravindran, R., Shadmehr, R. (2006). Generalization of motor learning depends on the history of prior action. *PLoS Biology, 4*(10), e316.

Lindley, D. V. (1956). On a measure of the information provided by an experiment. *Annals of Mathematical Statistics, 27*(4), 986–105.

Lonini, L., Dipietro, L., Zollo, L., Guglielmelli, E., Krebs, H. (2009). An internal model for acquisition and retention of motor learning during arm reaching. *Neural Computation, 21*(7), 2009–2027.

Mahadevan, S., & Maggioni, M. (2007). Proto-value functions: a Laplacian framework for learning representation and control in Markov decision processes. *Journal of Machine Learning Research, 8*, 2169–2231.

Mandelblat-Cerf, Y., Novick, I., Paz, R., Link, Y., Freeman, S., Vaadia, E. (2011). The neuronal basis of long-term sensorimotor learning. *The Journal of Neuroscience, 31*(1), 300–313.

Medina, J., Garcia, K., Mauk, M. (2001). A mechanism for savings in the cerebellum. *Journal of Neuroscience, 21*(11), 4081–4089.

Mitchell, C., Nash, S., Hall, G. (2008). The intermixed-blocked effect in human perceptual learning is not the consequence of trial spacing. *Journal Of Experimental Psychology, Learning, Memory and Cognition, 34*(1), 237.

Mussa-Ivaldi, F., Giszter, S., Bizzi, E. (1994). Linear combinations of primitives in vertebrate motor control. *Proceedings of the National Academy of Sciences, 91*(16), 7534.

Norman, K., & O'Reilly, R. (2003). Modeling hippocampal and neocortical contributions to recognition memory: A complementary-learning-systems approach. *Psychological Review, 110*(4), 611–646.

Osu, R., Hirai, S., Yoshioka, T., Kawato, M. (2004). Random presentation enables subjects to adapt to two opposing forces on the hand. *Nature Neuroscience, 7*(2), 111–112.

Overduin, S., Richardson, A., Lane, C., Bizzi, E., Press, D. (2006). Intermittent practice facilitates stable motor memories. *Journal of Neuroscience, 26*(46), 1188–1892.

Pekny, S., Criscimagna-Hemminger, S., Shadmehr, R. (2011). Protection and expression of human motor memories. *The Journal of Neuroscience, 31*(39), 13829–13839.

Poggio, T., & Bizzi, E. (2004). Generalization in vision and motor control. *Nature, 431*(7010), 768–774.

Puterman, M. (1994). *Markov decision processes: discrete stochastic dynamic programming*. New York: Wiley.

Robertson, E. (2007). The serial reaction time task: implicit motor skill learning? *Journal of Neuroscience, 27*(38), 10073.

Robertson, E., Pascual-Leone, A., Miall, R. (2004a). Current concepts in procedural consolidation. *Nature Reviews Neuroscience, 5*(7), 576–582.

Robertson, E., Pascual-Leone, A., Press, D. (2004b). Awareness modifies the skill-learning benefits of sleep. *Current Biology, 14*(3), 208–212.

Rucci, M., Edelman, G., Wray, J. (1999). Adaptation of orienting behavior: from the barn owl to a robotic system. *IEEE Transactions on Robotics and Automation, 15*(1), 96–110.

Schmidhuber, J. (1991). A possibility for implementing curiosity and boredom in model-building neural controllers. In *From animals to animats: proceedings of the first international conference on simulation of adaptive behavior (SAB90)*. Citeseer.

Scoville, W., & Milner, B. (1957). Loss of recent memory after bilateral hippocampal lesions. *Journal of Neurology, Neurosurgery & Psychiatry, 20*(1), 11–21.

Shadmehr, R., Smith, M., Krakauer, J. (2010). Error correction, sensory prediction, and adaptation in motor control. *Annual Review of Neuroscience, 33*, 89–108.

Shea, J., & Morgan, R. (1979). Contextual interference effects on the acquisition, retention, and transfer of a motor skill. *Journal of Experimental Psychology: Human Learning and Memory, 5*(2), 179–187.

Singh, S., Lewis, R., Barto, A., Sorg, J. (2010). Intrinsically motivated reinforcement learning: an evolutionary perspective. *IEEE Transactions on Autonomous Mental Development, 2*(2), 70–82.

Smith, M., Ghazizadeh, A., Shadmehr, R. (2006). Interacting adaptive processes with different timescales underlie short-term motor learning. *PLoS Biology, 4*(6), e179.

Spencer, R., Sunm, M., Ivry, R. (2006). Sleep-dependent consolidation of contextual learning. *Current Biology, 16*(10), 1001–1005.

Sutton, R., Precup, D., Singh, S. (1999). Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence, 112*(1), 181–211.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: an introduction*. Cambridge: Cambridge University Press.

Taylor, M., & Stone, P. (2009). Transfer learning for reinforcement learning domains: a survey. *The Journal of Machine Learning Research, 10*, 1633–1685.

Thoroughman, K., & Shadmehr, R. (2000). Learning of action through adaptive combination of motor primitives. *Nature, 407*(6805), 742.

Valero-Cuevas, F., Venkadesan, M., Todorov, E. (2009). Structured variability of muscle activations supports the minimal intervention principle of motor control. *Journal of Neurophysiology, 102*(1), 59.

Vijayakumar, S., D'souza, A., Schaal, S. (2005). Incremental online learning in high dimensions. *Neural Computation, 17*(12), 2602–2634.

Walker, M., Brakefield, T., Hobson, J., Stickgold, R. (2003). Dissociable stages of human memory consolidation and reconsolidation. *Nature, 425*(6958), 616–620.

Wei, K., & Kording, K. (2009). Relevance of error: what drives motor adaptation? *Journal of Neurophysiology, 101*(2), 655.

Westermann, G., & Mareschal, D. (2008). A dual-memory model of categorization in infancy. In *From associations to rules: connectionist models of behavior and cognition: proceedings of the tenth Neural Computation and Psychology Workshop, Dijon, France, 12–14 April 2007* (vol. 17, p. 127). Singapore: World Scientific Pub Co Inc.

Wolpert, D., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks, 11*(7–8), 1317–1329.

Yamashita, Y., & Tani, J. (2008). Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. *PLoS Computational Biology, 4*(11), e1000220.

A Developmental Framework for Cumulative Learning Robots

Mark Lee, James Law, and Martin Hülse

Abstract Developmental psychology is the study of human cognitive growth. However there exists a huge gap between the psychologist's theories and knowledge of behaviour and our ability to implement developmental processes in autonomous agents. In this chapter we describe an approach towards developmental growth for robotics that utilises natural constraints in a general learning mechanism. The method, summarised as Lift-Constraint, Act, Saturate (LCAS), is described and illustrated with results from experiments. We discuss how this approach is grounded in the topics of sensory-motor abstraction, intrinsic motivation (as novelty), and staged learning, and our belief that robotics can learn much from infant psychology.

1 Introduction

Research in Artificial Intelligence has produced some very impressive results over recent years and many of these advances have been implemented in robotics projects. But despite this success the realisation of truly autonomous robots has proved elusive and remains a difficult and significant challenge.

Any really intelligent robot should be autonomous in the sense that it can operate in the real world and can *cope* with new experiences and events by drawing on its previous experiences and building up cognitive competences and skills similar to those seen in humans and animals. This requires qualities such as adaptation, learning and versatility. But robots, like humans and unlike other computational systems, are embedded in the real world and have to experience noisy, chaotic, unstructured environments. Hence, completely novel experiences are unavoidable

M. Lee (✉) · J. Law · M. Hülse
Department of Computer Science, Aberystwyth University, UK
e-mail: mhl@aber.ac.uk; jxl@aber.ac.uk

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_9,

and this demands more than simple adaptation or the learning of stable events; indeed, new learning processes must emerge as conditions change and new events, environments and relationships are experienced.

It seems that such cognitive flexibility, as readily seen in humans and other altricial animal species, is dependent on a prolonged period of parental care, during which occurs some remarkable processes of structured growth generally known as "development". Given the long history of research into learning, in both animals and machines, it is surprising that only recently has any real attention been given to the concept of development as an important factor for artificial autonomous systems. This is remarkable because psychologists and other scientists have studied development in great detail and the idea is not new or original. Indeed, the great computer pioneer Alan Turing actually suggested this in 1950:

> Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. . . . We have thus divided our problem into two parts. The child programme and the education process. These two remain very closely connected. (Turing 1950)

Turing did not give a starting age for the child program, (he said "Opinions may vary as to the complexity which is suitable in the child machine") and the "course of education" can be taken in its broadest sense to include experience in general, but it is clear that he is talking about cognitive development:

> In the process of trying to imitate an adult human mind we are bound to think a good deal about the process which has brought it to the state that it is in. (Turing 1950)

Most work in bio-inspired robotics has drawn on the burgeoning growth of research in neuroscience and brain science generally. However, most brain science is concerned with understanding structure and function, with less emphasis on how those structures arise and how they are shaped by experience. But fortunately, developmental aspects are now gaining attention and the field of developmental robotics has now become established (Lungarella et al. 2003). This approach assumes a developmental framework that allows the gradual consolidation of coordination and competence, and emphasises the role of environmental and internal factors in shaping adaptation and behaviour (Prince et al. 2005).

The challenge from this viewpoint is in finding effective algorithms for processes that support the development of learning and adaptation. There exists a very significant lacuna between our psychological theories of development and our ability to implement working developmental algorithms in autonomous agents. In this chapter we explore this area by first examining the role of staged growth, constraints and infant learning (Sects. 2 and 3). In Sects. 4 and 5 we describe our approach for building developmental algorithms, and in Sect. 6 illustrate them with results from experiments. In Sects. 7 and 8 the role of novelty for motivating an active learning architecture is examined. Finally, in Sects. 9–11, we discuss the broader context and some key research challenges that emerge from this approach.

2 Developmental Stages

The scientific study of human cognitive growth is known as developmental psychology. Experiments on children and adults produce data which inform the production of theories that could explain the growth of skill and competence over time. Unlike neuroscience, psychology has no direct access to brain processes and instead attempts to derive inferences about possible internal mechanisms and events through observations of behaviour. Hence, patterns of behaviour and behavioural dynamics become the currency of experimental investigations. This level of indirectness allows much scope for interpretation and variety in theories of development in psychology.

A key characteristic of animal development is the centrality of behavioural sequences: no matter how individuals vary, all infants pass through sequences of development where some competencies always precede others. This is seen most strongly in early infancy as one pattern of behaviour appears before another: looking; reaching; standing; walking, etc. These regularities are the basis of the concept of behavioural stages—periods of growth and consolidation—followed by transitions—phases where new behaviour patterns emerge. The most influential theories of staged growth have been those of Jean Piaget who emphasised the importance of sensory-motor interaction, staged competence learning and a constructivist approach (Piaget 1973). Very briefly, Piaget defined four periods in an individual's life: beginning with the sensory-motor period (up to 2 years) during which infants are not fully capable of symbolic representation; then the Preoperational period (2–6 years) is characterised by egocentric behaviour; then follows the Concrete Operational period (6–12 years) in which abilities in classification and linear ordering are seen; and finally, the Formal Operation period (from 12 years onwards) displays capabilities in formal, deductive, and logical reasoning. Other psychologists, such as Jerome Bruner, have further studied the plasticity seen in infant studies and developed Piaget's ideas further, suggesting mechanisms that could explain the relation of symbols to motor acts, especially concerning the manipulation of objects and interpretation of observations (Bruner 1990; Kalnins and Bruner 1973).

All developmental stages have vague boundaries, and overlap and merge with other stages. They also show considerable temporal variation between individuals. Consequently, there is a great deal of debate about the origin and drivers for developmental change. At one extreme, Nativism argues that the machinery for development is genetically determined and the processes of growth are largely preprogrammed, with any apparently acquired cognitive competence being either ignored or refuted. In stark opposition, Empiricism takes the view that experience is a major factor in shaping the course of development and that any structures acquired to support development are shaped by experience (Spelke 1998). This debate, also known as the Constructivism versus Evolutionism argument, still continues after many decades and has various implications for psychological theory. Nevertheless, all viewpoints recognise the existence of stages as manifestations of development and their role in the growth of cognition appears to be very significant.

2.1 Early Infancy

We believe that research into developmental algorithms for robotics should be firmly grounded in the sensory-motor period. This is for several reasons: (1) it is logical and methodologically sound to begin at the earliest stages because early experiences and structures are highly likely to determine the path and form of subsequent growth in ways that may be crucial; (2) according to Piaget, the sensory-motor period consists of six stages that include concepts such as motor effects, object permanence, causality, imitation, and play—these are all issues of much relevance to robotics; (3) sensory-motor adaptation is a vital process for autonomous robots; and (4) it seems likely that sensory-motor coordination is a significant general principle of cognition (Pfeifer and Scheier 1997).

Furthermore, although the sensory-motor period covers 2 years, we believe it is essential to focus on the very beginnings of this period, before speech, before locomotion, and before other competences have become established. We are inspired by the first three months after birth, when control of the eyes, head and limbs is just emerging and growing. To the casual observer the newborn human infant may seem helpless and slow to change but, in fact, this is a period of the most rapid and profound growth and adaptation. From spontaneous, uncoordinated, apparently random movements of the limbs the infant cumulatively gains control of the parameters and coordinates sensory and motor signals to produce purposive acts in egocentric space (Gallahue 1982). We believe there is much for autonomous robotics to learn from this scenario.

For further support for this view, that we must start developmental learning at the very earliest stage possible, see Smith and Gasser (2005) where developmental psychologists argue from "six lessons from babies" that initial prematurity is crucial for the growth of embodied intelligence, in both babies and other agents.

3 The Importance of Constraints

All processes of adaptation and learning must have some form of underlying bias or *a priori* assumptions. This is because choices have to be made in representations, learning approach, possible actions, etc., even before learning begins. These biases are treated in developmental psychology in terms of constraints and there are many theories as to the origins, role and effects of such constraints on development. See Keil (1990) for a review of constraints under different theories.

Any restriction on sensing, action or cognition effectively reduces the complexity of the inputs and/or possible actions, thus reducing the task space and providing a constraining framework which shapes learning (Bruner 1990; Rutkowska 1994). In robotics, such constraints might be seen as sensory bandwidth reduction or the restriction of some degrees of freedom in motor actuation. The reduced task space can then be explored through active learning and its structure captured in the

learned representation. When a high level of competence at the current task has been reached then a new level of task or difficulty may be exposed by the lifting of a constraint (Rutkowska 1994). The next stage then discovers the properties of the newly scoped task and learns further competence by building on the accumulated experience of the levels before.

Various examples of internal sensory and motor constraints are seen in the newborn, for example the neonate has a very restricted visual system, with a kind of tunnel vision (Hainline 1998) where the width of view grows from around 30 degrees at 2 weeks of age to 60 degrees at 10 weeks (Tronick 1972). Although this may seem restricted, these initial constraints on focus and visual range are "tuned" to just that region of space where the mother has the maximum chance of being seen by the newborn. When "mother detection" has been established then the constraint can be lifted and attention allowed to find other visual stimuli.

Many forms of constraint have been observed or postulated (Hendriks-Jensen 1996) and we can consider a range of different types:

Anatomical/Hardware: These are the physical limitations imposed by the morphology of an embodied system. These include kinematic restrictions from structural (e.g. skeletal) joints and spatial configurations. Also mechanical constraints may be relevant, such as motor limitations preventing freedom of movement. See Pfeiffer and Bongard (2006) for expansion of this key topic. We notice that the anatomy of an infant changes markedly from birth, when we observe narrow shoulders and hips, large head as well as relatively short arms and legs.

Sensory-Motor: All sensors have their own limitations; usually these are specified in terms of accuracy, resolution, response rates and bandwidth. Motor systems also have similar characteristics with additional features for dynamic performance. Most changes in such sensory-motor characteristics can be linked to maturational growth.

Cognitive/Computational: Constraints on cognition take many forms, not only relating to speed but also relating to information content and structure. Many constraints that affect artificial neural systems are now known, and similar effects are being found in the brain (Casey et al. 2005). Sensory constraints can also affect or limit the input to cognitive processes.

Maturational: These are the most difficult to enumerate but it is certain that internal biological growth processes influence and maybe facilitate cognitive growth (Johnson 1990). Both neural and endocrinal support systems will have effects as they mature, for example neurogenesis is still under way for much of early infancy.

External/Environmental: External constraints are those that restrict behaviour or sensory input in some way but originate from the environment not from the individual or agent. This is a very powerful source of constraint as they can be applied at any time and are not related to the individual's stage of growth. If the constraints are carefully structured, especially by another agent to assist learning or adaptation, this is known as scaffolding (Bodrova and Leong 2006). Examples of

scaffolding are seen in parental care, education, and many other social situations, where interactions or tools generate patterns in the environment in order to direct attention or action towards a goal.

Much of the developmental literature concerns the role of constraints in higher order cognitive tasks such as number, language and reasoning (Campbell and Bickhard 1992). These internal, cognitive constraints deal with issues like representation and could be termed "soft" constraints. Because we are interested in the earlier processes of sensory-motor adaptation we must also be concerned with the "hard" constraints that emerge from the actual physical properties and features of the system. These are known as Type 4 constraints by Campbell and Bickhard (1992) and strongly influence the construction of the adaptive processes.

4 The LCAS Approach

It is important to state that we strive for general rather than task-specific mechanisms of development. Consequently we emphasise explicit, abstract models and try to avoid prestructured internal representations or assumptions about internal belief states or internal causal knowledge. In particular we avoid the early adoption of neural network models and other connectionist methods as these can be difficult to analyse and interpret (Gasser and Smith 1998; Lee et al. 2006). Such methods also often entail extensive training schedules which is counter to most of the empirical evidence that indicates learning and adaptation can be very fast, and in some cases require only one trial of experience (Angulo-Kinzler et al. 2002; Rochat and Striano 1999). Accordingly, we appreciate the "content-neutral" methodology of Thelen and Whitmyer (2005) and we try to follow a similar approach.

We view "constraint lifting" as a key mechanism for progression towards increasing competence. Transitions between developmental stages are related to the lifting of constraints, although the nature of such transitions is not fully understood. It seems that the enabling conditions for transitions must be related to internal global states, not local events, because local activity cannot capture cumulative experience. For example, assume novelty is a motivating driver, then if a novel event occurs (a new stimulus or behaviour) this will raise some local attention or excitation levels for that particular event. However, successive similar stimuli will be less excitatory, both in time and space, and eventually similar stimuli may be experienced for all spatiotemporal possibilities. At this point, such events are no longer novel and do not raise excitation levels significantly. A global state indicator (being the spatiotemporal sum of all local excitation) will thus reach a stable plateau when no novel events have been seen for a long time. Thus, high competence at a level is equivalent to all incoming experience matching expectations, with no novel changes or unexpected events.

Thus, global states, such as global excitation, can act as indicators that can detect qualitative aspects of behaviour, e.g. when growth changes have effectively ceased or when experience has become saturated. They can then signal the need to enter a

new level of learning by lifting a constraint (such as accessing a new sensory input). In this way, further exploration may begin for another skill level, thus approximating a form of Piagetian learning.

Our approach then consists of implementing the cycle: Lift-Constraint, Act, Saturate (LCAS), at a suitable level of behaviour. First, the possible or available constraints must be identified and a schedule or ordering for their removal decided. Next, a range of primitive actions must be determined together with their sensory associations. Also, mechanisms for sensory-motor learning or adaptation are incorporated at this point. A set of global measures need to be established to monitor internal activity and some kind of intrinsic motivation must be provided to initiate action. We use a simple novelty function as the motivational driver.

When this is implemented the initial behaviour may seem very primitive, but this is because all, or nearly all, constraints have been applied and there is little room for complex activity. The "Act" stage generates spontaneous and varying patterns of action so that the scope for experience is thoroughly explored and all new experiences are learned and consolidated. Eventually there are no new experiences possible, or they are extremely rare, and this level becomes saturated. The global indicators then reach a critical level and the next constraint in the schedule is lifted and the cycle begins again.

This general description of LCAS above is not meant to be prescriptive in detail. Constraint lifting should not be triggered by explicit mechanisms but should be the emergent effect of the computational processes involved. Similarly, we do not advocate the programming of the stages by following an explicit constraint lifting schedule; such a method would defeat the very purpose of investigating development. But it is necessary to study the role of constraints and their relation to staged behaviour, and so constraint schedules need to be investigated in order to further our understanding of their relationships and interaction, and also to support the design of experiments.

Our approach to developmental robotics assumes that the robot's task is to learn as much as possible and we try to avoid pre-programmed competencies. This is an experimental stance and does not represent a position on the empiricist/nativist spectrum in developmental psychology (Spelke 1998). We note that nativists need to explain the origins of any innate structures that they propose and any experiments on learning may shed light on this, just as much as they may support an empiricist stance.

4.1 Representations

In early infancy changes in behaviour can be discerned where initially spontaneous, apparently random, limb movements gradually become coordinated and organised. During this process the proprioceptive and kineasthetic space of the limbs becomes calibrated and correlated with the motor space of the actuation system of the muscles. The spontaneous movements would be primitive examples of active

learning, observed as "motor babbling" which is seen at many levels of behaviour. The very earliest examples of such proprioceptive behaviour probably occur in the womb (Ververs et al. 1998) and could provide the first sensory-motor correlations. We believe early correlation between proprioceptive space and motor space should be the foundation for building internal models of local egocentric space and thus form a substrate for future cross-modal skilled behaviours.

Interestingly, the same issue is found in robotics research where it is necessary to coordinate the differing spatial frameworks of the various sensory and motor systems. For example, coordinating the spatial frame of an eye system with the spatial structure of a hand/arm system requires cross-modal relations to be established and understood; in this case, image-based information needs to be related to the spatial coordinates accessible by a multi-degree-of-freedom mechanism. Even within a single modality there are often various correlations that have to be established, usually between sensors that relate to a particular motor system. For example, when moving an eye or a camera to a new fixation target it is necessary to relate the desired displacement on the image to the associated motor action that will achieve the target.

These concerns reflect on the problem of representation; how should these coordinations be implemented in a computer model so as to best capture the correlation relationships, while allowing that these must be learned and not programmed? In order to follow the methodology mentioned above, we have adopted a simple and general model that permits wide variation without being committed to a particular learning mechanism. Our model also has many neuromorphic features as it attempts to be compatible with knowledge from neuroscience.

4.2 A Sensory-Motor Mapping Model

We now describe the general idea behind our topological mapping method for the representation of sensory-motor events and coordinations. In particular, we show how such structures can form the key substrate for a developmental style of learning and can be fast, accurate and flexible.

In several previously reported experiments (Lee et al. 2006, 2007b), we have used two-dimensional maps with explicit links between corresponding sensory or motor values for the representation of sensory-motor spaces. Although three dimensions might seem appropriate for representing spatial events, we take inspiration from neuroscience, which shows that most areas of the brain are organised in topographical two-dimensional layers (Braitenberg and Schüz 1991; Mallot et al. 1990). This remarkable structural consistency has inspired our explorations of the potential and efficacy of two-dimensional structures. Also there is evidence to believe that the human spatial system separates depth (distance from the body) from lateral displacement (up-down, left-right). For example, when depth is an object feature being detected (along with colour, intensity, orientation, etc.) then this can be processed in parallel with a second feature; an unusual effect that does not

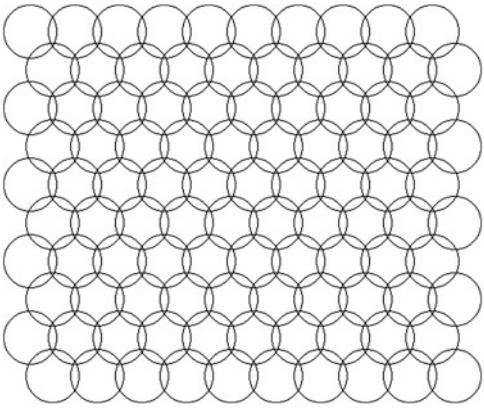

Fig. 1 Field overlap on regular *triangular* mesh, 10×10 fields, $r = 0.6$

apply to other features (Nakayama and Silverman 1986). This suggests that a 2.5D architecture is appropriate, with lateral and vertical locations as the axes of a 2D map, and depth values recorded at specific locations. A 2.5D structure is different from a full 3D form in that only one depth value can exist for each location; but this makes sense in an egocentric space where the nearest object occludes all those behind it.

A typical mapping will consist of a 2D array representing two sensory or motor variables, known as a *map* or *surface*, connected to another 2D array by a set of *links* that join points or small regions, known as *fields*, in each array. The links are bidirectional enabling access in both directions and are collectively known as a *mapping*. Fields are local regions of equivalence and are defined by a boundary function; we generally use simple circular fields. With circular fields it is not possible to completely cover a surface without some overlap between fields, unlike with pixels. For a uniform distribution of fields on a 2D grid, the most efficient surface covering is given for an equilateral triangular grid, where the minimum radii to ensure complete covering is 0.577 (with unity grid spacing), see Fig. 1. For this case the area of overlap is only 21 %, (i.e. for any field 79 % of its area is not shared with another field).

4.2.1 Field Distributions and Overlap

Overlapping structures may seem inefficient or inappropriate for computational implementation but it is well known that overlap occurs in many neuronal and sensing mechanisms in the brain (Carreira-Perpinan et al. 2005). For example, in the eye the sensing receptors do not physically overlap, but they are connected to ganglion cells that provide functional overlap (Sterling 1999). We believe this property is responsible for some interesting and very useful effects. We note that very large overlaps can have little purpose, because, in any given locality, many of the fields will have almost identical coverage. Thus a stimulus would give the same

effect over a wide area. Two possible functions for highly overlapped fields might be redundancy and low pass filtering. However, lesser overlap, when the field radii are below the grid spacing distance, appears much more promising.

Our mapping method consists of two processes: the creation of new fields on a map surface; and the generation of explicit links between two fields on different maps. A field captures a local region on a map such that all stimuli within the field can be represented and transmitted to other surfaces through a single field reference point. For circular fields the reference is the centre point. Links between maps are established through correspondence; if a field in one map can be reliably observed to become excited in temporal correlation with another field on another map, then an explicit link is created between the fields' centres. This is equivalent to finding the strongest weights between elements in a fully connected 2-layer network after training for temporal correspondence. Using this criteria gives very strong links for just a few fields and no coordination at all for all the others.

This simple Hebbian learning system (Martinetz 1993) has proved very effective in building maps in systems with strong and stable correlations. A generalisation of this approach is to use probability density functions to represent the unknown map linkages during learning. Variations of this include mechanisms based on radial basis functions (Pouget and Snyder 2000). This mapping method, although simple in concept, has several valuable properties. The number of links needed to effectively map between two surfaces can be quite low and increases with desired accuracy. This means that learning such maps can be very fast; we have grown mappings on laboratory hand/eye robot systems and achieved complete mapping coverage of a reach space of 3944 cm^2 in real time in 5 h (Hülse et al. 2010b). This scheme can also be adjusted so that links may be removed or changed when errors are detected thus providing adaptation and placticity.

5 Building Maps from Experience

Questions now arise as to when fields should be created and how they should be organised on a mapping surface. This concerns both the initial placement of the fields on the map and methods for exploring the sensory-motor spaces so that fields are populated with data from experience. In addition, we also need to consider the possibilities for adaptation of developed maps to take account of external changes.

5.1 Field Generation

There are essentially two main parameters for configuring the fields that comprise a map: they could either be of varying or fixed size, and they could be generated

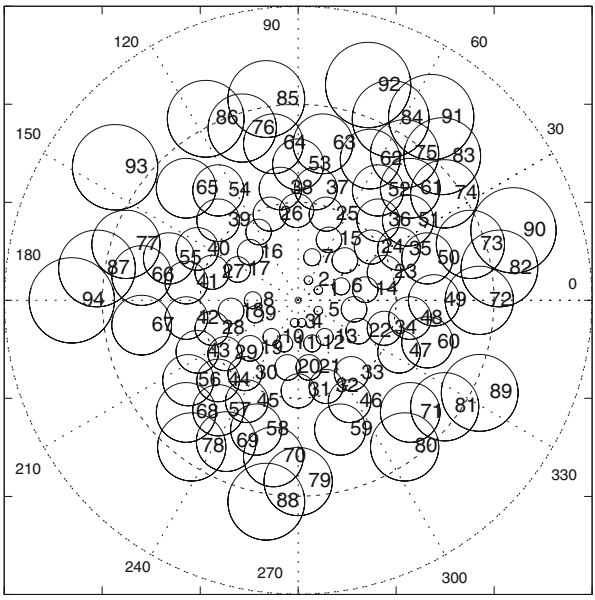

Fig. 2 Fields being generated from a polar grid

onto a grid (prior structure) or they could be placed at any stimulus location (free format). This gives four possibilities in all, each of which we now consider in turn.

Assume that a learning process generates stimulus points, $p_i = (x_i, y_i)$, on a two-dimensional surface, S, which is initially empty. If a stimulus point is already covered by a field on S, i.e. is within the radius of some existing field, then no action is required. But if p_i is not covered, then a new field must be generated for this location.

Uniform size fields on a regular grid For this case, over the map S we arrange a grid of fields with uniform spacing of centre points. Initially all fields are unassigned but an uncovered stimulus point will cause the nearest field to be assigned to the map. Eventually all points will be covered by one or more fields. If the grid of fields has low levels of overlap ($r <$ the grid spacing), for example as in Fig. 1, then there will always be places covered by only one field and so eventually all fields in the grid will be used. Conversely, with large overlap many of the possible fields will not need to be generated, as each point on the surface only needs to be covered once. Figure 2 shows a retina design with considerable overlap but, during learning, fields are only taken from the grid as needed and the covering process stops when every possible stimulus point has been covered. It may seem unprincipled to use a pre-structured grid but topographic maps are widespread in the brain. There is evidence that this topographic structure is determined during neurogenesis by many influences and *both* genetic and experiential inputs have strong effects (Goodhill and Xu 2005). It is possible that genetically encoded neural growth patterns provide

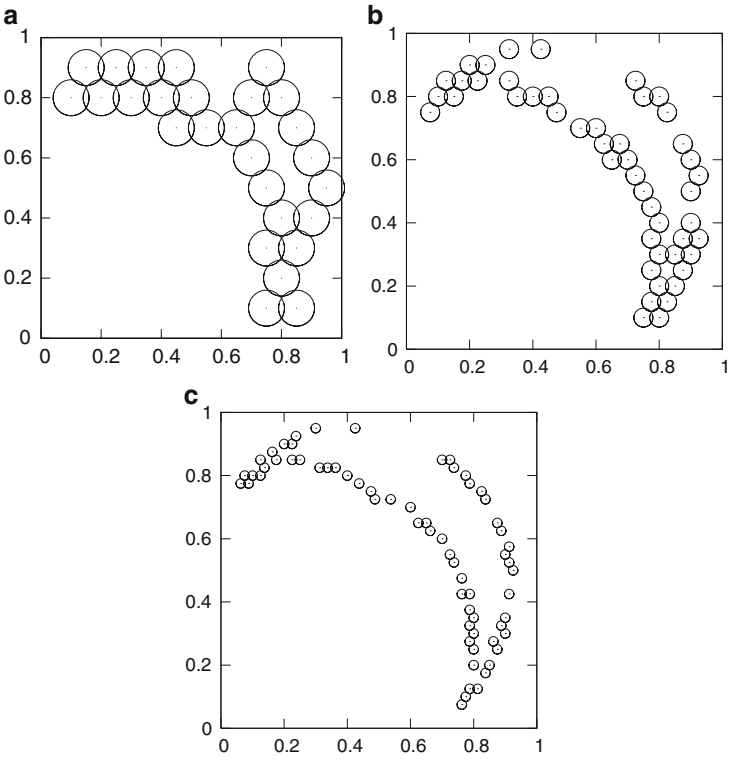

Fig. 3 A single map composed of three layers, each of different field sizes. (a) Large fields. (b) Medium fields. (c) Small fields

regular arrays of neural sheets and then the interconnections are established by a separate process of coordination. There is also evidence that neurons can expand their receptive fields in order to adapt to a damaged area caused by a lesion (Einarsdottir et al. 2007) and we note that such plasticity is better served by a uniform grid structure rather than an irregular covering of fields.

Variable sized fields on a regular grid This case is the same as above except that the field sizes are individually set according to the space available in their local neighbourhood (Meng and Lee 2007). For experimental purposes we found it more insightful to maintain several uniform maps, each with a different size of field. Figure 3 shows three such surfaces from our experiments. When a new stimulus point requires a new field we processed all the maps in parallel and so the three maps are built up together. When later using the maps to select a field, the large fields can be used for crude but fast coverage of the space, while the smaller fields require more learning but give finer articulation. This provides options for accuracy/speed trade-offs as and when needed. This effect was also reported by Gomez et al. (2004).

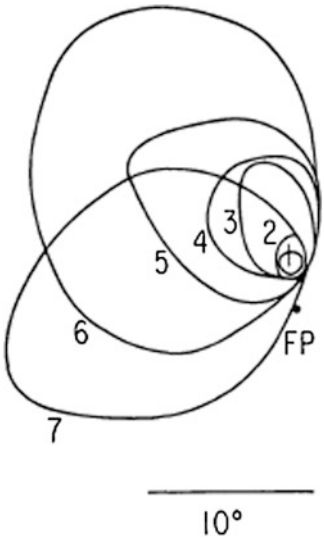

Fig. 4 Aligned neural fields of different sizes in the superior colliculus, from Wurtz and Goldberg (1972)

It is interesting that similar layers of fields of different sizes all responding to the same stimuli location have been found in the superior colliculus and elsewhere in the brain (Wurtz and Goldberg 1972). Figure 4 is taken from Wurtz and Goldberg (1972) and reveals increasingly larger field sizes at deeper layers of the superior colliculus. Note how a stimulus point in field 1 is covered by all fields, with each successive field usually overlapping all of the preceding field.

Uniform size fields with irregular locations In this case fields are simply created with their centres located at the incoming stimulus points. This is generation-upon-demand and so the shape of the final coverage will not only be irregular but will also vary considerably depending upon the order in which the stimuli arrive. It may not be appropriate to be driven by event order, as very idiosyncratic mappings can develop which might not easily generalise over different tasks. However, this objection can be overcome by allowing fields to be replaced by some form of decay mechanism that removes infrequently used fields.

We have found this to be a simple and effective generation method for situations where there are no topographical requirements or when the nature of the spaces is unknown. For example, in correlating a robot arm with a motorised camera system we found that a mapping of arm end-point with camera gaze point gave a very effective coordination scheme (Hülse et al. 2010b). Figure 5 shows such a mapping. This system was driven by an exploratory algorithm that looked for unmapped areas and attempted to find new points equidistant from the nearest existing fields.

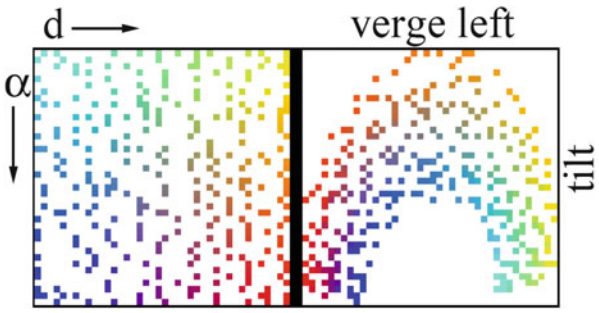

Fig. 5 Mapping between robot arm (*left*) and eye system (*right*). Fixed field size, free location. Similar colours indicate linked fields

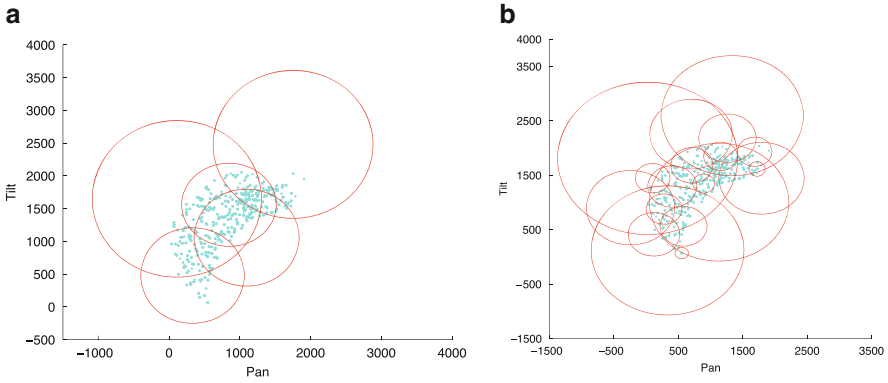

Fig. 6 Variable field sizes and locations. (a) Coarse coverage. (b) Finer coverage

Variable sized fields with irregular locations Completely irregular field growth is also possible. We have experimented with mechanisms where any uncovered stimulus causes a new field to be created that is precisely centred on the stimulus, with the size of the field determined to maximally fill the gap between the nearby fields. Figure 6 shows some results from radial basis function experiments (Meng and Lee 2007). The left diagram shows how only five fields initially developed to cover a set of points; crudely but quickly. The right-hand diagram is after further learning where more, and smaller, fields give a finer coverage with more resolution. A similar reduction in cortical field areas relating to infant growth in object recognition ability has been reported (Westermann and Mareschal 2004).

We notice that grid placement is useful for situations where some external structure must be taken into account. This is seen in sensory maps that must reflect the structure of the sensing system. For example, peripheral vision has a much lower resolution than that of the foveal region and so we use increasingly larger sized fields towards the periphery in retinal maps. By designing a grid before use, we can impose desired constraints on field size and placement.

5.2 Populating Fields Through Exploration

The next issue is how map building processes are to be driven and organised in a learning or growth scenario. Consider the original problem at the point where no known structure or patterns have been discovered. Hence, no links yet exist. Assume we have two 2D surfaces, S and M, and we wish to establish how they are related. The surfaces both have two variables: (x_i, y_i) for S and (k_j, l_j) for M; and we can perturb these in order to detect any correlations. However, it is generally not possible to vary sensory inputs at will (at least not directly) and so only M is available for exploration. This means we can vary M and observe any effect on S; we cannot operate in the other direction. Thus, starting from a condition of no prior knowledge, some initial motor action must be performed, and the least specific action is simply to start moving in some arbitrary direction. Such action will eventually terminate when the physical or anatomical limits are reached for that particular system. If this is followed by similar movements in other directions, then the effect is produced of exercising M over its range of variables. This behaviour will explore the maximum and minimum extent of M over its range limits. We note that this appears very similar to the behaviour known as motor babbling in human infants (Piek 2002; Piek and Carman 1994). If the two variables for M are independent, then a rectangular plot will emerge for the boundary, showing the ranges of k_j and l_j along the axes. Now, the values of x_i and y_i may vary in a complex way with M but if the motor values are constrained to their extrema, then the S values will similarly describe the limits of their range. The plots thus produced will display the boundaries of the mapping for the operating regions of S and M. This assumption rests on the surfaces being smooth and continuous and having planar topologies. Thus, no points can exist *external* to the S boundary: otherwise, some regions of S would map into 2 or more separate places in M; which could not provide a mapping. Figure 7 shows the (kinesthetic) sensory space produced for a two-limb jointed arm as the motors drive the angles of the joints through their extremities. Arrows are shown on the figure to illustrate the effect of sweeping one motor variable at a time. It is important to notice that such a structured strategy is not necessary for discovery of the boundaries. It is significant that *any* motor action that stops at an extrema for one of the variables will find a boundary point and the corresponding point on S will be revealed.

It is noticeable that while the motor surface covers the full range of the variables, as would be expected if the system is exercised over its extent, the sensory fields do not cover all possible locations on their surface. This is a common effect as sensory systems will often cover larger spaces than those of associated motor systems. For example, a camera on a pan-and-tilt head may have a wider field of view than the pan limits; and the human eye can see more than the eyeball can turn to. This means that the off-boundary (*internal*) fields in the motor surface will be mapped across to the *internal* region of the sensory map and the sensory fields must be effectively distorted to fit *within* the boundary shape. Figure 8 shows three different sensory examples for the motor pattern in the lower left. The motor system of the arm has two independent motors each with limits on their extent. This gives a rectangular

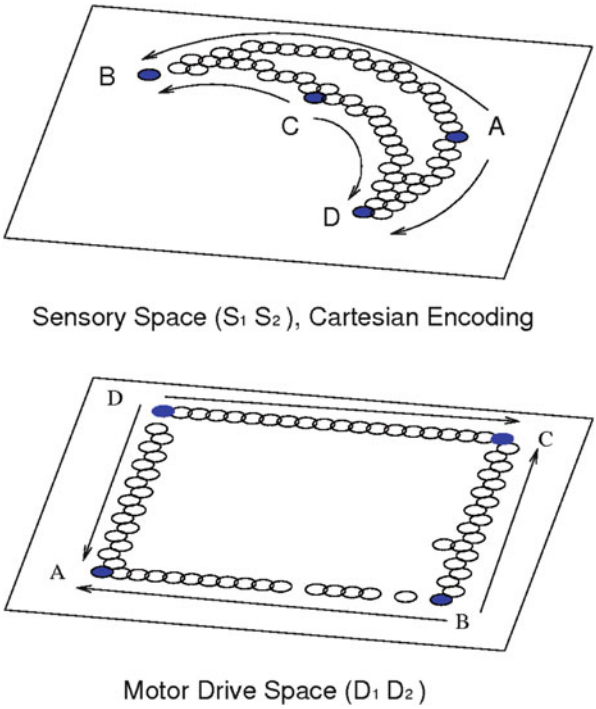

Fig. 7 Boundary fields can be established first

shape for the fields experienced at those limits (lower left in Fig. 8). The other three plots show the corresponding fields for three different sensory configurations (using different geometric configurations of the proprioceptive sensors).

A consequence of this kind of motor babbling behaviour is that the extremities of the space will be explored before the fine detail of the internal regions. If we consider using a regular grid for the fields, as in Sect. 5.1, then the boundary distortion can be seen as a kind of warping of M to S. Consequently, we could warp the regular grid to match the boundary and thus obtain a good first estimate of the locations of the internal fields *even before any such data points have been experienced*. Standard image warping methods are not useful here as they use parametric transformations that do not handle local distortions well. Various kinds of local distortion methods do exist but a far simpler computational approach is provided by the elastic membrane concept. An elastic sheet can be pulled and stretched in various directions to fit both local distortions and global warpings. The method uses the idea that elasticity is a simple relation between distance and force whereby each molecule of material moves to a position that minimises the total of the forces from its neighbours. So an effective relaxation algorithm is easily implemented by minimising the sum of the distances between each node in the grid and its immediate neighbours. This elastic neighbourhood consists of the six nearest fields in a triangular array (or four or

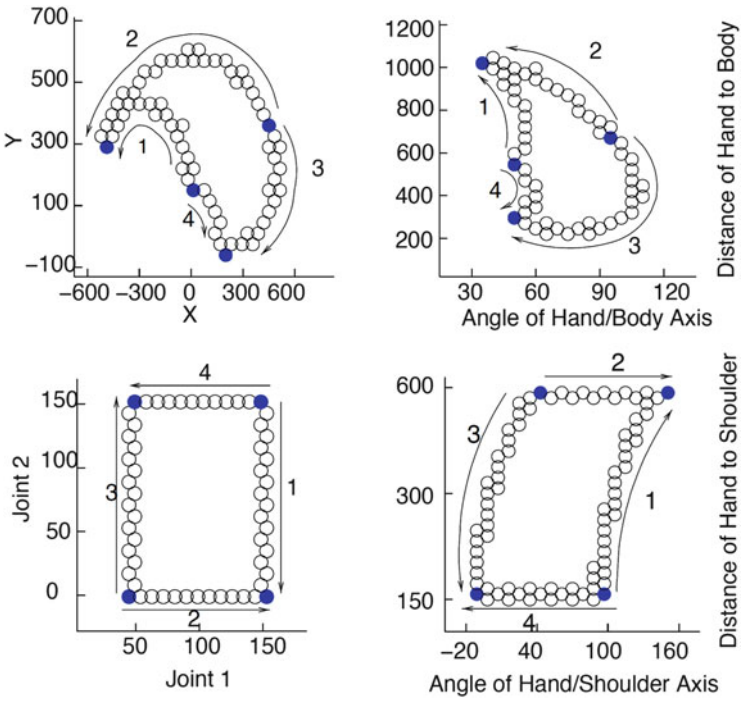

Fig. 8 Different boundaries for different arrangements of (muscle-based) sensor systems. *Bottom left*: motor map, *Top left*: Cartesian space (ideal) as would be seen by overhead camera, *Bottom right*: space of arm vector from hand to shoulder, *Top right*: space of vector from hand to body-centre

eight for a rectangular system) and as a node is moved to a new position so the local neighbours are pushed and pulled in the same direction but to lesser amounts the further away they are. Figure 9 gives an example of a grid with both boundary distortion and some internal node displacement. The algorithm calculates the error in each node and relaxes over all nodes until the total error in the system falls below a threshold. If we use the elastic sheet method to fit all the (originally rectangular) edge nodes in S to the new boundary (matching corresponding fields), then we will obtain a warp of the regular grid that suggests reasonable estimates for the locations of the internal fields.

5.3 Adaptation and Plasticity

Although we expect mappings at the lower levels of sensory-motor experience to be relatively stable, the possibility of error is always present and any mapping might be

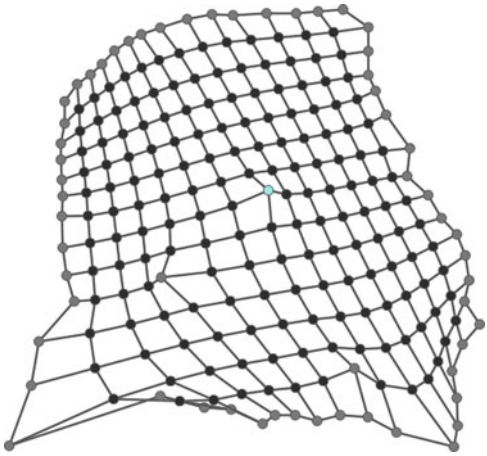

Fig. 9 Boundary and internal distortions of elastic grid

required to adjust its structure in some way. This could be a local or a global effect: either a relatively small number of fields may be found to be located in error and need adjustment or the whole map might be in error and then a full scale remapping is needed.

Local Adaptation

Errors in field placement may occur due to noise and tolerance effects in sensing and/or motor structures. For example, an action may not be exactly repeatable due to low muscle control or environmental disturbance. These conditions can give rise to local error and the need for local adjustment.

The elastic sheet method, described in the last section, allows any point to be adjusted at any time, with the neighbours also making compensatory adjustments. This is idealy suited for the correction of local errors, where a field centre is found to be incorrect and is to be moved. Thus, if a local variation is created by some sensory or bodily change, then the discovered effects can be used to adapt the mapping by either small movements of the fields concerned or reassignment of a few links between the relevant maps.

Re-calibration and Realignment

In cases of major reconfiguration of sensory or motor systems the adaptation required may be so severe that complete realignment of all of a mapping may be required. This can happen when, for example, a camera in a hand/eye system is shifted to a new position relative to the rest of the system. In humans, similar disruptions are experienced in prism adaptation experiments where prisms applied to the eyes cause major and total redistribution of the visual image (Redding and

Wallace 2006). Such global changes will essentially involve all the links in a mapping being reassigned to quite different (i.e. non-local) fields. This is a serious disruption to any agent and presents a major re-learning challenge. Fortunately, such situations are generally rare and dealing with these cases efficiently may not be so important as ensuring good local adaptation.

The ability to change or remove links in error provides a degree of plasticity in the mapping that covers these problems. We have experimented with a mechanism for adaptation that allows links to be removed if they become unused (Hülse et al. 2010b). We gave every link an "age" value. When a link is first established and whenever it is subsequently used successfully then its age is set to zero. Otherwise the age of every link is increased at each application of the mapping. Any links that are not found useful are deleted from the mapping and this allows space for new links to be established. Thus, small local adaptations may occur continuously. See Hülse et al. (2010b) for further details. We have found that a threshold for the minimum age of links effectively controls the total number of links in a mapping and is a very useful parameter in experimental explorations of map growth and development.

6 Examples of Emergent Behavioural Stages from LCAS Experiments

In this section we present a few examples of our method to illustrate the concepts and mechanisms involved. As an experimental design we explored various sensory-motor subsystems individually. This approach also fits well with the idea of early constraints being quite severe and apparently limiting activity to one or two modalities at a time. Consequently, we started with an empty system and considered what sensory or motor components should or could be developed first. The eye is particularly active after birth, followed by limb movements, and then hand/eye interaction. Accordingly, we carried out separate implementations on eye saccading (Chao et al. 2010) and arm reaching (Lee and Meng 2005), and then combined these in hand/eye experiments (Chao 2009).

Consideration of the motor system shows that the kinaesthetic sense plays an important role in motor control and in spatial cognition. For example, when a visual stimulus is to be brought to the foveal region of the retina it is not just the sensed image and the eyeball muscles that are involved: the proprioceptive sensors in the muscle spindles provide very important data on eye position, or gaze. We found that proprioception was very significant in all the systems we investigated.

Our first study on the eye implemented a simple mechanism for saccade learning. We assumed that no calibration between image and eyeball position could be achieved before images were available (i.e. before birth) and used the methods described above to generate active exploratory behaviour and build mappings between stimuli fields on the retina (image) and motor positions of the eyeball

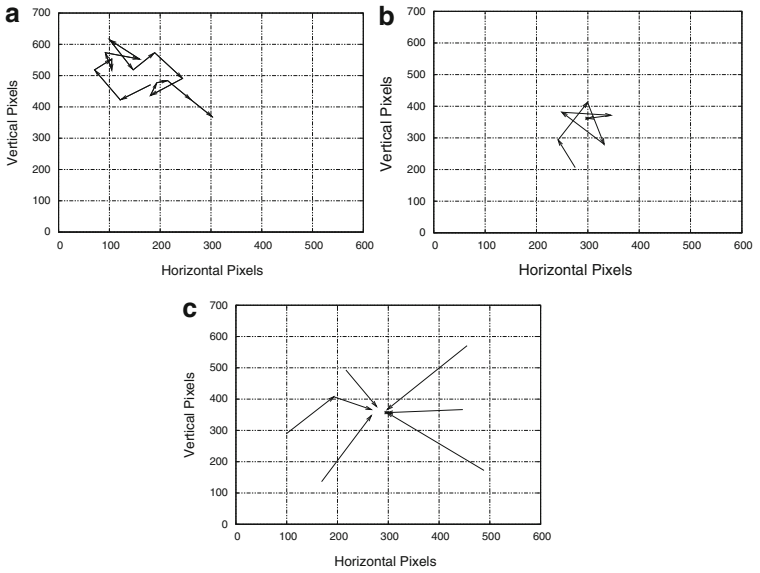

Fig. 10 Qualitatively different behaviours during learning. (a) Early stage trace. (b) Intermediate stage. (c) Final stage trace

(camera gaze direction). The aim was to learn a mapping between image locations and motor values so that a single direct movement (saccade) could be made to bring any image point to the centre of the image (fovea).

At first, when no fields or links had been established, the eye movements appeared as random walks, eventually finding the centre region. After a few fields had been created, the random moves would often find one of these and then move among the fields nearer the centre. Finally, single saccades would appear as the map became fully populated. We were surprised to see that the results showed clear qualitative differences in the behavior as the maps grew.

Figure 10 shows example traces clearly exhibiting different stages. The first trace performs 15 movements before finding the target, the second trace required 6 moves, and the last trace shows 6 different saccades (superimposed), all but one being single motor acts. The results were classified into three types: no local field exists near the stimulus; a neighbouring field is found and used; and a stimulus-covering field exists. These were plotted in Fig. 11, which shows three runs for each type to illustrate the variation. The labels correspond to those in Fig. 10. Type A is seen to start first but reaches a plateau after about 60 moves, while type B starts later and also plateaus later. The single move or fully learned saccades do not appear at all during the first 18 moves but grow fast until they become the only type to exist. Because fields are being generated for every movement the map builds quickly; in fact, Fig. 2, on page 187, shows the retinal map for this experiment after 94 fields have been entered. Already, much of the total space has been covered, after only around 180 learning attempts.

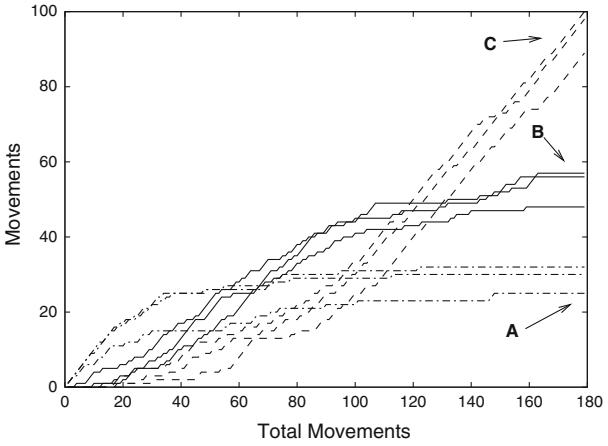

Fig. 11 The growth rates of three types of saccade behaviour

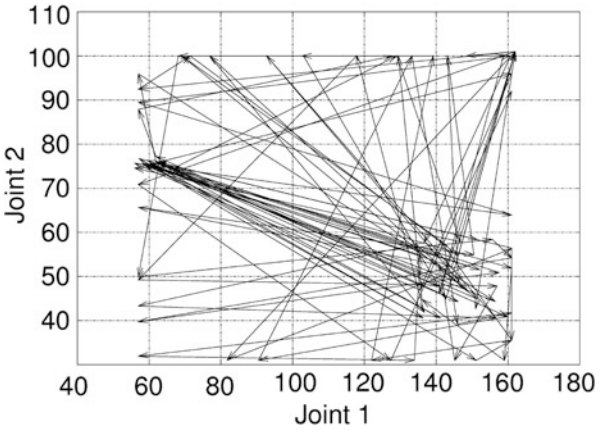

Fig. 12 Exploratory arm motions shown as vectors in arm joint space

Our next experiment imposed the constraint that the vision system would be inactive while the arm was active. This is equivalent to limb movement out of sight of the eye. Again we started with an empty map system. A rest position for the (single) arm was provided, equivalent to a low-energy state, and spontaneous motor values were set to drive the arm through the workspace. Eventually the arm comes to a halt, an unexpected event, and fields are generated. The process is repeated and the fields relating to the boundary of its workspace are soon discovered, as shown and described in Sect. 5.2. Figure 12 shows traces of motor acts; the darker section in the middle are movements from the rest position to the fully extended arm position (these were the initial motor settings) and other traces can be seen as the arm makes spontaneous moves and reaches various locations on its operating boundary.

As the map becomes populated so the rate of field discovery saturates, as does the opportunity to learn. To enable further learning, a constraint can be lifted, and in this case, we activated a tactile sensor in the robot end-effector. This allowed the arm to make contact with objects, interrupt the action, and record a new spatial sensation. As the arm touched objects in different locations so the internal fields in the map were created and eventually the map was completed. After this point any target "felt" location could be reached by a single direct arm movement.

The final constraint to be lifted was to allow both arm and eye to operate together. As both had near fully complete maps from the previous stages, this stage involved the creation of new mappings that relate the visual space of the eye to the reach space of the arm. We found the visual gaze space (i.e. the angles of eye fixation) to be the most appropriate frame for the integration of the two subsystems (Chao 2009).

It is interesting to notice the kinds of behaviour produced from this series of experiments. We observed a progression of qualitatively distinct behavioural patterns:

Visual stimuli fixation - through three stages from eye wandering to direct saccading, during the creation of the image map (constraints: eye only active)

"Blind groping" - actions mainly directed towards the body area (constraints: arm only active)

Extended groping - directed towards the bounding limits of the agent's egocentric space, during the creation of the boundary space map through arm proprioception (constraints: arm only active)

Unaware contact - seen as pushing or ejecting objects out of the local environment, due to a constraint on the tactile sense (constraints: arm only active)

Contact sensitivity - or "sensitive groping" where limb movements are interrupted by tactile sensing events, and the non-boundary space map is constructed (constraints: arm and tactile active)

Repeated cycles of contact - observed as repeated "touching" behaviours directed at detected objects (constraints: arm and tactile active)

Hand regard - or hand fixation, when the eye sees the hand as an object but one whose movement correlates with arm activity. The "object" is marked as a special case (constraints: eye and arm active)

Arm follows eye fixations - when the location of a stimulus, found by the eye, is mapped into the arm space and excites arm action to reach to the same place. This is the basis of reaching and grasping of seen objects (constraints: eye and arm active)

Eye follows tactile contact - the converse case, where the arm has touched a stimulating object and its location is mapped to the eye system which then saccades to fixate on the object (constraints: eye, arm and tactile active)

All these behaviours, including the various forms of motor babbling and the sometimes rather ballistic motor actions, are widely reported in young infants (Piek and Carman 1994).

Our choice of constraining visual development until after a kinaesthetic sense has been established could be controversial but the results show that this is not an

unreasonable developmental sequence. Much of the psychological literature tends to assume that vision is the dominant sense and that visually guided reaching is the earliest accurate reaching behaviour to occur. Infants spend time observing their hands around 12 weeks and "visually guided" reaching begins between 15 and 20 weeks. Reaching after 22 weeks is visually triggered rather than guided. However, Clifton et al. (1993) have performed infant reaching experiments in the dark and shown that infants of around 15 weeks are able to use proprioception alone, without vision, in successful reaching tasks. A form of "hand looking" behaviour can be expected to occur when the hand first enters the visual field as an "unknown" object; but the question is whether this stage is essential to, and therefore must occur before, visually-guided behaviour or whether there could be other schedules. Our study confirms the view of Clifton et al. by showing how proprioceptive learning *can* occur *prior* to visual development, can be used to guide action, and does not necessarily depend upon visual confirmation. A well-developed kinaesthetic sense could be a great advantage in supporting visual-guidance and visual coordination by providing a ready mapping of the local operating space. As Clifton et al. state: "Prior accounts of early reaching have underemphasized the role of proprioception in infants' acquisition of prehension" (Clifton et al. 1993).

The integration of proprioception and tactile senses can produce a powerful haptic system, but it is an open question as to which part should develop first. Our speculation that tactile sensing could be delayed until after significant kinaesthetic growth in the same modality appears to be supported by our results. At least, it is a viable strategy to reduce the complexity of the learning input by discovering some of the structure of local space before the structure of tactile sensing data is explored. Of course, a very complex tactile system such as the hand with many types of receptors sensing heat, vibration, pain and touch may well need a period of familiarisation to establish the various functions, but this is distinct from object detection and could take place in parallel with other activities. From these considerations it is clear that both components of the haptic system could develop together or proprioception could lead tactile and somatic sensing; but tactile cannot lead propreoception. On reflection we see that this is a logical necessity because the tactile system must rely on an existing spatial frame if its experiences are to have any spatial context or meaning.

Regarding environmental constraints, we have only used the idea of scaffolding to the extent that we could place objects in areas that were under-explored and thus direct attention to gain developmental experience in those areas. This was only possible after the tactile constraint had been lifted; before then objects would be ignored and possibly ejected from the agent's personal space. In later work we have examined the effects of known objects being removed, and this leads on to object permanence and the detection of moving objects and external agency.

The size of the fields is a useful constraint that is easily overlooked. If large fields are generated initially, then a rapid but crude mapping of space can be obtained. When this is no longer creating new experiences, then the field size can be reduced thus refining the accuracy with new map entries. It is interesting that the receptive field size of visual neurons in infants is reported to decrease with

age and development, and this leads to more selective responses (Westermann and Mareschal 2004).

7 Novelty as Motivation

Motivation is an essential function to drive autonomous development, with current approaches often using externally driven processes (e.g. Bullock and Grossberg 1988; Caligiore et al. 2008; Gasser and Smith 1998). However, internal drives are necessary for true autonomy and we employ novelty-based functions as intrinsic motivators to drive autonomous development through increasing complexity. Examples of similar approaches and effective results are seen in Kaplan and Oudeyer (2003, 2007) and Schmidhuber (1990).

In our work we have used a simple novelty indicator to motivate learning and trigger the removal of constraints. When the robot encounters a novel stimulus it will repeat the action that caused that stimulus to occur. As the action is repeated, the novelty dissipates, and the robot has an increasing tendency to perform alternative actions. Over time the robot finds and performs cumulatively more of the actions available to it, and so the number of remaining novel actions diminishes (Lee and Meng 2005). As fewer novel actions are found, the rate of learning in the robot saturates. In order to enable learning to continue, a constraint can be lifted, as described in Sect. 6. This opens up a new range of actions for the robot and provides a new source of novel stimuli to investigate.

To implement this idea we allow the fields to store several variables; this is equivalent to maintaining an interlayer for each variable in a map surface. These variables can include stimulus-type quantities for the field location (e.g. depth, colour, intensity), and also excitation, activity, and any other indicators that prove useful as a substrate to support learning (e.g. object markers for short term memory). We use an excitation value to record the current salience or importance of a field's contents and an activity value for the usage the field has received. Novel events at a given spatial location bestow high excitation on the relevant field. Excitation levels gradually decay with time and are also reduced by habituation following repeated stimulations. Thus a few constants are required for habituation rate, recovery time, excitatory decay and possibly other influences. This mechanism allows a simple selection function; the field with the highest excitation provides the motivation for the next action.

Notice that we do not record novelty as a value—rather novelty increases excitation and in this way the meaning of novelty can change. Thus the effect of different events on excitation will change with experience of those events. Hence this approach covers a range of new events: new fields, new coordinations (links), new action changes, new sensing stimuli, new cross-modal events, etc. Global excitation (the normalised sum of all the individual excitation levels above a threshold) is an indicator that can signal low attention and can trigger a return to spontaneous motor action. Global activity (the normalised and inverted summation

of the field activity levels) decreases with familiarity and can indicate saturation. We have previously suggested that the degree of motor noise in an action is related to muscle tone and that tone should increase with excitation. If this is implemented, then highly focussed action (on targets of high interest or novelty) will have less noise and better accuracy, while low excitation will accompany low tone and higher motor noise resulting in more exploratory action.

8 Developmental Action Formation

In Sect. 2 we argued for the importance of grounding research in developmental robotics in the sensory-motor period of human infant development. However, developmental psychology supports the theory that early motor skills are related to perceptive and cognitive development (Thelen 1995), which varies from child to child. Although there is no single development sequence for all infants—they develop in their own ways and at their own speeds with some variation—they do generally conform to a common set of developmental stages. Through our research, we have identified some of these common sequences, which provide the foundations for our work on the developmental formation of actions.

From the literature on child development we have constructed a large, general time-line chronicling the observable development in infant sensor and motor behaviour over the first 12 months. The information is too extensive to cover here, but brief summaries can be found in Law et al. (2011) and Hülse et al. (2010a).

Based on this time-line we have generated some general sequences of development appropriate for implementation on humanoid robots. For example, Fig. 13 shows a partial development sequence for the upper body of a humanoid robot. Note that this sequence only focuses on motor development; similar charts cover sensory growth and attentional preference. Development of motor function is indicated by shaded areas, which get progressively darker as control improves (note that the termination of a shaded bar does not indicate the termination of that skill, but that it has been sufficiently developed as to no longer appear in the literature). It is interesting to note that the sequence for motor development begins with the eyes and progressively moves down the body, through the neck and arms, with torso and wrist control being refined last. The exception to this is the hands, which become active in the first month, but continue to develop until the tenth month. This time-line, along with its counterparts, begins to address the problem of how constraints should be ordered in a robot. For other work on time-lines, see Metta et al. (2009).

The framework to support this emergence of skill is provided for in our systems by an arrangement of constraints. Table 1 shows how, following our developmental time-lines, a series of constraints can be assembled to support the emergence of actions. In this case, we suggest a possible sequence for developing gaze-directed reaching beginning with the learning of arm proprioception and eye saccades.

Activated abilities are represented by crosses in Table 1. This does not necessarily mean that the corresponding function is completely unconstrained, but that it is

Table 1 A possible framework for staged development of integrated reaching

	Eye				VM		Arm		Grips				
Development stage	M	P	LR	HR	IOR	Ex	M	P	Pa	Pi	Cu	Saturation criteria	Observed behaviour
1 Eye saccade	x	x	x									Low occurrence of unlearnt saccades	Eye can saccade to fixate on any stimulus on the retina
2 Inhibited visual search	x	x	x		x							High occurrence of new saccades	Ignores recurrent visual stimuli
3 Hand-eye coordination	x	x	x				x	x				Low prediction errors for visually guided reaching	Reaches to fixation point
4 Reach guided eye-saccade	x	x	x			x	x	x				Low prediction errors for reach guided saccades	Fixates on reach point
5 Self-aware reach	x	x	x		x		x	x				Low occurrence of hand fixation	Ignores own hand
6 Retaining grasp							x	x	x		x	Long retention duration	Holds objects with reflex grasp
7 Intentional gross pick-up	x	x	x				x	x	x		x	Saturation of success rate for picking up large objects	Reaches towards objects and picks them up with palmar grasp
8 Intentional precision pick-up	x	x	x				x	x		x	x	Saturation of success rate for picking up small objects	Reaches towards objects and picks them up with pincer grasp
9 Selective grasping (combination of tactile and visual data)	x	x	x	x			x	x	x	x	x	Saturation of success rate for picking up arbitrary objects	Reaches towards objects and picks them up with the appropriate grasp
10 Visual search with identification and memorization	x	x	x	x	x	x						Saturation of the change of the visual selection criteria	Attraction to specific objects and refining of visual selection criteria
11 Tactile search with identification and memorization					x	x	x	x	x	x	x	Saturation of the change of the tactile selection criteria	Attraction to specific objects and refining of tactile selection criteria
12 Integrated object pick-up	x	x	x	x	x	x	x	x	x	x	x		

At each stage the system is constrained so that it only has access to data marked "x", where *M*, motor values; *P*, proprioceptive sensor values; *LR*, low resolution visual layer; *HR*, high resolution visual layer; *VM*, visual memory; *IOR*, inhibition of return; *Ex*, visual excitation; *Pa*, palmar grasp; *Pi*, pincer grasp; *Cu*, cutaneous sense

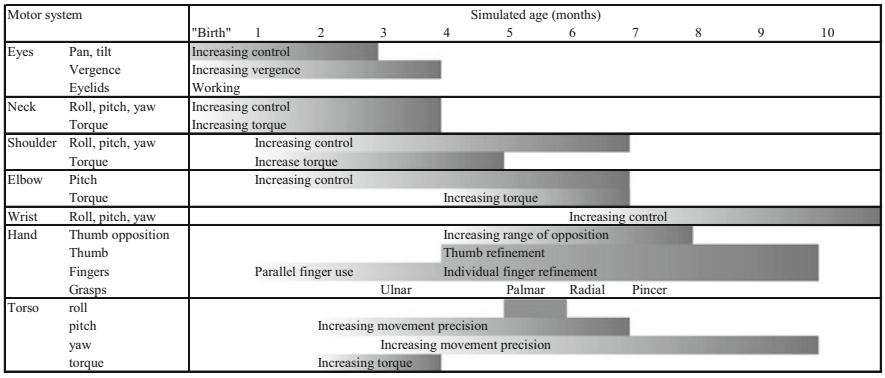

Fig. 13 Partial motor development sequence for a humanoid robot. *Shaded* regions relate to periods of development of each ability as observed in infants. *Darker* shading indicates more advanced ability

available in some form for the system to use. For example, when sensor and motor maps become available for use they may initially have additional constraints on their resolutions. In this example, the robot would start out at the first stage with access only to the eyeball proprioceptive sensors and motors, and low resolution visual feedback. This constrains the robot and focuses its attention on learning to saccade to simple stimuli. When sufficient learning has taken place development moves onto the second stage, where additional functionality is enabled. A new round of learning begins, this time reincorporating skills learnt in the previous stage. At each stage, one or more constraints on functionality are removed, allowing the robot to learn new skills. In this way, the robot progresses from being able only to make uncoordinated motion with an eye or an arm to being able to reach to a seen object in an integrated and goal directed behaviour.

It is important to note that the stages in Table 1 are not fixed in their order, nor are they necessarily triggered in isolation. Some stages may be learnt in parallel, and similar levels of development may appear in different orders. The robot is able to influence the order of constraint removal, so we would expect initially identical robots to develop in individual ways, much as human infants would. Importantly, the constraints do not directly control the developmental stages, but simply release more complexity to the learning processes. Thus stage transitions are emergent; their ordering and timing are not easily predictable. Indeed, the system may regress to earlier stages when an action cannot be successfully learned due to gaps in the system's previous experience.

This raises the question of how different sensory-motor systems can be combined. Consider the example of three important maps: the image map from the retina of the eye; the gaze map which maps the orientation of the eye fixation point; and the reach map of an arm that records the places the hand or end-effector can occupy. These are all *spaces* in the sense that each one models the structure of a particular sensory-motor system and can relate an action to its effects. However they are quite

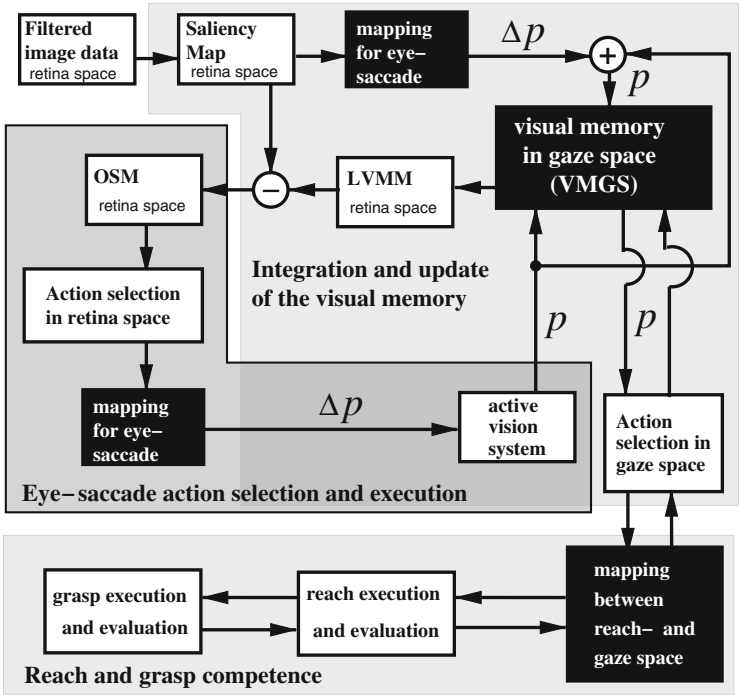

Fig. 14 Combining image, gaze and arm maps

different and distinct, yet must cooperate during behaviour. For example, if the eye notices a stimulus in a peripheral area of the retina, it must rotate the eyeball to bring the gaze to focus on the object; this involves a mapping from image map to gaze space. Next the arm can reach for the object but to do so the gaze location must be mapped into the reach space of the arm.

We have investigated several methods for combining gaze and reach mappings, such as those briefly mentioned in Sect. 6. In Hülse et al. (2009b) we used a robot arm to place objects within a camera's field of view, and the system correlated the position of the object in the gaze space with the known positions of the arm joints. By repeating the process with objects placed at different locations, the system built up a mapping of gaze space to reach space. This system is outlined in Fig. 14.

Visual stimuli on the retina are mapped to eye motor movements in the mapping labelled "mapping for eye-saccade". Here, a motor movement is mapped to a stimulated field in the retina map such that it will perform a saccade (as described in Sect. 6). This saccade results in a specific absolute motor configuration p, and the range of absolute motor configurations define the gaze space.

Points in the gaze space can be associated with physical locations in space by mapping them to arm configurations. In our experiments, the arm placed a coloured object within the field of view. The joint positions of the arm were then mapped to

the visual field stimulated by the object. This "mapping between reach space and gaze space" enables the robot to reach to a point of interest in the scene (Hülse et al. 2010b, 2009b).

Together these two mappings enable the robot to saccade to, and reach to, a target. However, to prevent the robot repeatedly saccading and reaching to the same targets, a visual memory is included, labeled VMGS (Visual Memory in Gaze Space). The VMGS stores the absolute motor configurations, p, of the active vision system resulting from a saccade. For each visual stimuli detected, the system is able to predict the corresponding gaze space configuration by applying the mapping for eye-saccade. By cross referencing these targets with those stored in the VMGS, the system can inhibit visual stimuli that have already been saccaded to (LVMM). The overlaid saliency map (OSM) contains the difference between the stimuli on the retina and the previously examined ones, thus enabling the system to only perform saccades to new stimuli (Hülse et al. 2009a).

9 Research Challenges

The work described above presents us with a number of challenges for the future. Some of these, and hints at solutions, are summarised below:

What field sizes should be used to create the various mappings? We have learned much about different field structures and how accuracy of representation is dependent on field size. But there is much more to be understood about the granularity and structure of human egocentric space. We suspect that further work on this topic might reveal some general principles for developmental robotics.

What are the properties of overlapping arrays? The idea of overlap is rather contrary to intuition and existing mathematical approaches. However, overlapping neural and sensory structures are ubiquitous in the biological world. Although overlap suggests engineering problems like signal crosstalk we believe it has considerable beneficial properties, especially for spatially grounded systems. Further theoretical work on models of topological maps is required, e.g. (Carreira-Perpinan et al. 2005).

How can mappings between multiple sensory-motor systems be combined? The work described here has focussed on creating mappings between one or two sensory or motor systems. Although these provide foundation building blocks, more complex activity requires the combination of multiple sensory-motor systems. Full scale humanoid robots will need many mappings across many different subsystems. This raises important organisational questions involving architectures such as hierarchies and networks.

How should constraints be released? It is too simplistic to design a constraint table and then follow the dictates of this structure. This amounts to simply

programming the observed behaviour of an infant directly. But there are much more subtle ways in which constraints may influence and modulate a learning system. We have experimented with emergent constraints and investigated simultaneous map generation and compared this with sequential generation and other schemes (Hülse and Lee 2010), and we find that explicit triggers for constraint lifting are not necessary (or even desirable). It is possible for constraints to be treated by these systems in an *emergent* way and then behavioural stages emerge as a consequence of the current state of the developing system. Further investigation is needed to understand this phenomena in terms of the literature surrounding developmental studies (Law et al. 2011).

How do the maps fit into a body-centric model of space? In our investigations, mappings have used individual reference frames. Those incorporating arm movements have been centred on the body centreline, whereas those incorporating visual saccades have been grounded in the visual space. To achieve composite tasks, such as visually guided reaching, we built mappings between these different reference frames. This opens up major questions about the structure of egocentric space and how an integrated sense of space can grow and be maintained. Such spaces must include tactile, visual and other representations and so become a kind of "ego-space".

How can neural models of biological systems (e.g. the basal ganglia) be integrated into the LCAS framework? As our system is expanded, we expect situations to arise such as action selection, where a choice between alternative actions must be made. Maintaining the biological inspiration of the project, we see how neural models, such as those by Prescott et al. (2006) for action selection, can be incorporated. Further investigation is required to establish how other such models can be integrated into developmental approaches such as the LCAS architecture.

What is the role of novelty, and how does it relate to intrinsic motivation? We have used a single idea for intrinsic motivation: novelty, and a very elementary implementation of this. Research into novelty is not our main focus and this simple technique was used as a minimum complexity driver to provoke autonomous action. We appreciate that as competency increases more elaborate algorithms for novelty detection will be required. For example, our architecture cannot currently detect novelty in temporal events. We aim to integrate more sophisticated novelty detection algorithms such as those by Neto and Nehmzow (2007) and Oudeyer et al. (2007) in the future, but we also note that the whole question of intrinsic motivation is broader than novelty and may include other drivers.

10 Relation to Other Work

While there is now growing research activity in the area of developmental robotics, most related work deals with specific topics such as motivation, active vision, self-awareness, interaction, and modelling issues. Much of this research has relevance

for our approach, as seen in the citations given throughout this chapter, particularly those that shed light on possible mechanisms and algorithms. But there is still a disproportionate lack of research that takes account of the large body of experimental work in psychology and attempts to extract algorithms that might capture some of the infant's impressive cognitive growth.

One of the most comprehensive efforts at computer-based modelling of early development following a Piagetian approach has been that of Drescher (1991). This used the concept of sensory-motor *schemas* drawn from Piaget's conception of schemas in human activity (Piaget 1973) and had similarities with early schema experiments by Becker (1973). Unfortunately, Drescher's implementation was a simulation with very primitive sensory-motor apparatus and so many issues that concern embodiment were not exposed. Maes showed how Drescher's approach can be improved by using focus of attention mechanisms, specifically using sensory selection and cognitive constraints (Foner and Maes 1994).

A few models of infant grasping have been produced and some recent ones (Oztop et al. 2004) hint that visual guidance may not be central for reaching; however, none cover the growth of proprioception. There are many more models of sensory-motor coordination, and the vast majority of these have been based on connectionist architectures (Kalaska 1995). For example, Baraduc et al. (2001) designed a neural architecture that computes motor commands from arm positions and desired directions. Other models use basis functions (Pouget and Snyder 2000) but all these involve weight training schedules that typically require in the region of 20,000 iterations (Baraduc et al. 2001). They also tend to use very large numbers of neuronal elements. Interestingly, the model of Baraduc et al. is one of the few that apply adaptation to the proprioception signals, and obtain good accuracy from very few input examples. While our models could be cast into a connectionist framework and, we believe, would give identical performance, we wish to formulate general methods for constraint models and so favour more explicit algorithms.

11 Conclusions

The framework described in this chapter builds sensory-motor schemas in terms of topological mappings of sensory-motor events, pays attention to novel or recent stimuli, repeats successful behaviour, and detects when reasonable competence at a level has been achieved.

Support for our approach comes from various data. For example, studies of the order of cell activation in the foetus report that the first cells to be detected as active are the somatosensory cells, then the auditory, then visual, and finally, the multisensory cells become active (Meredith et al. 1987). This suggests what we have found that there are advantages if proprioception leads before vision in sensory development. Other work has also experimented with low resolution in sensors and motor systems and then shown that increasing resolution leads to more effective learning (Gomez et al. 2004). Reduction in degrees of freedom obtained by staged

development is also reported to be an effective strategy (Lungarella and Berthouze 2002; Sporns and Edelman 1993), as is the concept of constraints being beneficial to the emergence of stable patterns and helping to bootstrap later stages (Berthouze and Lungarella 2004).

Regarding our sensory-motor coordination method, we have avoided the long training times of connectionist methods and used a fast, incremental, and constructive mechanism. This is in accord with several researchers who report that infant learning and adaptation can be very fast (Angulo-Kinzler et al. 2002; Rochat and Striano 1999) and in some cases only one trial or experience is needed to alter behaviour.

Our experiments have shown how stages in growth and behaviour may emerge from embodied agents through their exploration of the sensory-motor environment under constraint. It is important that behaviour grows and changes *without* any programming but through the shaping influence of experience on internal mechanisms. As an early researcher stated:

> Gradual removal of constraint could account for qualitative change in behaviour without structural change (Tronick 1972)

We must continue to search for mechanisms and supporting substrates that allow increasingly advanced behaviour to emerge from consolidated prior experience. Studies of constraint relationships are part of this endeavour and the role of constraints must be better understood.

We have argued that it is necessary to begin modelling development at the earliest possible behavioural stages. We agree that "early infant life is . . . systematic exploration" (Rochat 2003) and believe that robotics can learn much from infant psychology. Although most psychological theories are not fully articulated enough to allow testing via implementation, psychologists have built up considerable understanding and insights into cognitive development through experimental studies. We should make more use of this in autonomous systems research so that we might make steps towards the goal of "continuous development". In the longer term, we hope this will lead to new methodologies for building autonomous robot systems and better understanding and insights into human behaviour and growth.

References

Angulo-Kinzler, R., Ulrich, B., Thelen, E. (2002). Three-month-old infants can select specific leg motor solutions. *Motor Control, 6*(1), 52–68.

Baraduc, P., Guigon, E., Burnod, Y. (2001). Recoding arm position to learn visuomotor transformations. *Cerebral Cortex, 11*, 906–917.

Becker, J. D. (1973). A model for the encoding of experiental information. In R. C. Schank & K. M. Colby (Eds.), *Computer models of thought and language* (pp. 396–434). San Francisco: W.H. Freeman and Company.

Berthouze, L., & Lungarella, M. (2004). Motor skill acquisition under environmental perturbations: on the necessity of alternate freezing and freeing of degrees of freedom. *Adaptive Behavior, 12*(1), 47–64.

Bodrova, E., & Leong, D. (2006). *Tools of the mind: the Vygotskian approach to early childhood education*. Columbus: Prentice Hall.

Braitenberg, V., & Schüz, A. (1991). *Anatomy of the cortex: Statistics and geometry*. Berlin: Springer.

Bruner, J. (1990). *Acts of meaning*. Cambridge: Harvard University Press.

Bullock, D., & Grossberg, S. (1988). Neural dynamics of planned arm movements: emergent invariants and speed-accuracy properties during trajectory formation. *Psychology Review, 95*(1), 49–90.

Caligiore, D., Ferrauto, T., Parsi, D., Accornero, N., Capozza, M., Baldassare, G. (2008). Using motor babbling and Hebb rules for modeling the development of reaching with obstacles and grasping. In *Proceedings of the international conference on cognitive systems* (CogSys 2008), University of Karlsruhe, Karlsruhe, Germany, April 2–4, 2008.

Campbell, R., & Bickhard, M. (1992). Types of constraints on development: an interactivist approach. *Developmental Review, 12*(3), 311–338.

Carreira-Perpinan, M., Lister, R., Goodhill, G. (2005). A computational model for the development of multiple maps in primary visual cortex. *Cerebral Cortex, 15*(8), 1222–1233.

Casey, B., Galvan, A., Hare, T. (2005). Changes in cerebral functional organization during cognitive development. *Current Opinion in Neurobiology, 15*(2), 239–244.

Chao, F. (2009). *Constraint lifting and its application in developmental robotics*. PhD thesis, Department of Computer Science, Aberystwyth University, Wales.

Chao, F., Lee, M., Lee, J. (2010). A developmental algorithm for ocular motor coordination. *Robotics and Autonomous Systems, 58*, 239–248.

Clifton, R., Muir, D., Ashmead, D., Clarkson, M. (1993). Is visually guided reaching in early infancy a myth? *Child Development, 64*(4), 1099–1110.

Drescher, G. (1991). *Made up minds: a constructivist approach to artificial intelligence*. Cambridge: MIT.

Einarsdottir, H., Montani, F., Schultz, S. R. (2007). A mathematical model of receptive field reorganization following stroke. In *IEEE 6th international conference on development and learning* (pp. 211–216), Imperial College London, 11–13 July 2007, IEEE Computational Intelligence Society.

Foner, L., & Maes, P. (1994). Paying attention to what's important: Using focus of attention to improve unsupervised learning. In *Proceedings of the 3rd international conference on simulation of adaptive behaviour* (pp. 256–265). Cambridge: MIT.

Gallahue, D. (1982). *Understanding motor development in children*. New York: Wiley.

Gasser, M., & Smith, L. B. (1998). Learning nouns and adjectives: a connectionist account. *Language and cognitive processes, 13*(2–3), 269–306.

Gomez, G., Lungarella, M., Hotz, P. E., Matsushita, K., Pfeifer, R. (2004). Simulating development in a real robot: on the concurrent increase of sensory, motor, and neural complexity. In L. Berthouze, H. Kozima, C. G. Prince, G. Sandini, G. Stojanov, G. Metta, C. Balkenius (Eds.), *Proceedings of the fourth international workshop on epigenetic robotics: modeling cognitive development in robotic systems* (vol. 117, pp. 119–122). Lund University Cognitive Studies. Lund: LUCS.

Goodhill, G., & Xu, J. (2005). The development of retinotectal maps: a review of models based on molecular gradients. *Network: Computation in Neural Systems, 16*(1), 5–34.

Hainline, L. (1998). How the visual system develops. In A. Slater (Ed.), *Perceptual development: visual, auditory, and speech perception in infancy* (pp. 5–50). Hove: Psychology.

Hendriks-Jensen, H. (1996). *Catching ourselves in the act*. Cambridge: MIT.

Hülse, M., & Lee, M. H. (2010). Adaptation of coupled sensorimotor mappings: an investigation towards developmental learning of humanoids. In S. Doncieux, B. Girard, A. Guillot, J. Hallam, J.-A. Meyer, J.-B. Mouret (Eds.), *From animals to animats 11. Lecture notes in computer science* (vol. 6226, pp. 468–477). Berlin: Springer.

Hülse, M., McBride, S., Law, J., Lee, M. (2010a). Integration of active vision and reaching from a developmental robotics perspective. *IEEE Transactions on Autonomous Mental Development, 2*(4), 355–367.

Hülse, M., McBride, S., Lee, M. (2010b). Fast learning mapping schemes for robotic hand-eye coordination. *Cognitive Computation, 2*(1), 1–16.

Hülse, M., McBride, S., Lee, M. H. (2009a). Implementing inhibition of return; embodied visual memory for robotic systems. In L. Cañamero, P.-Y. Oudeyer, C. Balkenius (Eds) *Proceedings of the ninth international conference on epigenetic robotics* (vol. 146, pp. 213–214). Lund University Cognitive Studies. ISBN 978-91-977-380-7-1

Hülse, M., McBride, S., Lee, M. H. (2009b). Robotic hand-eye coordination without global reference: a biologically inspired learning scheme. In *IEEE 8th international conference on development and learning, (ICDL)* (pp. 1–6). New York: IEEE.

Johnson, M. (1990). Cortical maturation and the development of visual attention in early infancy. *Journal of Cognitive Neuroscience, 2*(2), 81–95.

Kalaska, J. (1995). Reaching movements: implications of connectionist models. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 788–793). Cambridge: MIT.

Kalnins, I., & Bruner, J. (1973). The coordination of visual observation and instrumental behavior in early infancy. *Perception, 2*(3), 307–14.

Kaplan, F., & Oudeyer, P.-Y. (2003). Motivational principles for visual know-how development. In C. G. Prince, L. Berthouze, H. Kozima, D. Bullock, G. Stojanov, C. Balkenius (Eds.), *Proceedings of the third international workshop on epigenetic robotics* (vol. 101, pp. 73–80). Lund University Cognitive Studies. ISBN 91-974741-X.

Kaplan, F., & Oudeyer, P.-Y. (2007). In search of the neural circuits of intrinsic motivation. *Frontiers in Neuroscience, 1*(1), 225–236.

Keil, F. (1990). Constraints on constraints: Surveying the epigenetic landscape. *Cognitive Science, 14*(4), 135–168.

Law, J., Lee, M. H., Hülse, M., Tomassetti, A. (2011). The infant development timeline and its application to robot shaping. *Adaptive Behaviour, 19*(5), 335–358.

Lee, M. H., & Meng, Q. (2005). Psychologically inspired sensory-motor development in early robot learning. *International Journal of Advanced Robotic Systems, 2*(4), 325–334.

Lee, M. H., Meng, Q., Chao, F. (2006). A content-neutral approach for sensory-motor learning in developmental robotics. In F. Kaplan, P.-Y Oudeyer, A. Revel, P. Gaussier, J. Nadel, L. Berthouze, H. Kozima, C.G. Prince, C. Balkenius (Eds), *Proceedings of the sixth international workshop on epigenetic robotics* (vol. 128, pp. 55–62), September 20–22, 2006, Paris, France, Lund University Cognitive Studies. ISBN 91-974741-6-9

Lee, M. H., Meng, Q., Chao, F. (2007a). Developmental learning for autonomous robots. *Robotics and Autonomous Systems, 55*(9), 750–759.

Lee, M. H., Meng, Q., Chao, F. (2007b). Staged competence learning in developmental robotics. *Adaptive Behaviour, 15*(3), 241–255.

Lungarella, M., & Berthouze, L. (2002). Adaptivity through physical immaturity. In C.G. Prince, Y. Demiris, Y. Marom, H. Kozima, C. Balkenius (Eds) *Proceedings of the second international workshop on epigenetic robotics* (vol. 94, pp. 79–86). Lund University cognitive studies, August 10–11, 2002, Edinburgh, Scotland. *Studies, 94*. Lund: LUCS

Lungarella, M., Metta, G., Pfeifer, R., Sandini, G. (2003). Developmental robotics: a survey. *Connection Science, 15*(4), 151–190.

Mallot, H., Von Seelen, W., Giannakopoulos, F. (1990). Neural mapping and space-variant image processing. *Neural Networks, 3*(3), 245–263.

Martinetz, T. (1993). Competitive Hebbian learning rule forms perfectly topology preserving maps. In *Proceedings of the ICANN'93, International conference on artificial neural networks* (vol. 93, pp. 427–434). Springer.

Meng, Q., & Lee, M. H. (2007). Automated cross-modal mapping in robotic eye/hand systems using plastic radial basis function networks. *Connection Science, 19*(1), 25–52.

Meredith, M., Nemitz, J., Stein, B. (1987). Determinants of multisensory integration in superior colliculus neurons. I. Temporal factors. *Journal of Neuroscience, 7*(10), 3215–3229.

Metta, G., Craighero, L., Fadiga, L., Ijspeert, A., Rosander, K., Sandini, G., Vernon, D., von Hofsten, C. (2009). A roadmap for the development of cognitive capabilities in humanoid robots. Technical Report D2.1, University of Genoa.

Nakayama, K., & Silverman, G. (1986). Serial and parallel processing of visual feature conjunctions. *Nature, 320*(6059), 264–265.

Neto, V., & Nehmzow, U. (2007). Visual novelty detection with automatic scale selection. *Robotics and Autonomous Systems, 55*(9), 693–701.

Oudeyer, P., Kaplan, F., Hafner, V. (2007). Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation, 11*(2), 265–286.

Oztop, E., Bradley, N., Arbib, M. (2004). Infant grasp learning: a computational model. *Experimental Brain Research, 158*, 480–503.

Pfeifer, R., & Scheier, C. (1997). Sensory-motor coordination: the metaphor and beyond. *Robotics and Autonomous Systems, 20*(2), 157–178.

Pfeiffer, R., & Bongard, J. (2006). *How the body shapes the way we think*. Cambridge: MIT.

Piaget, J. (1973). *The child's conception of the world*. London: Paladin.

Piek, J. P. (2002). The role of variability in early motor development. *Infant Behavior and Development, 25*(4), 452–465.

Piek, J. P., & Carman, R. (1994). Developmental profiles of spontaneous movements in infants. *Early Human Development, 39*(2), 109–126.

Pouget, A., & Snyder, L. (2000). Computational approaches to sensorimotor transformations. *Nature Neuroscience, 3*, 1192–1198.

Prescott, T., Montes González, F., Gurney, K., Humphries, M., Redgrave, P. (2006). A robot model of the basal ganglia: behavior and intrinsic processing. *Neural Networks, 19*(1), 31–61.

Prince, C., Helder, N., Hollich, G. (2005). Ongoing emergence: a core concept in epigenetic robotics. In L. Berthouze, F. Kaplan, H. Kozima, H. Yano, J. Konczak, G. Metta, J. Nadel, G. Sandini, G. Stojanov, C. Balkenius (Eds.) *Proceedings of the fifth international workshop on epigenetic robotics* (vol. 123, pp. 63–70). Lund University cognitive studies. ISBN 91-974741-4-2.

Redding, G., & Wallace, B. (2006). Generalization of prism adaptation. *Journal of Experimental Psychology: Human Perception and Performance, 32*(4), 1006–1022.

Rochat, P. (2003). Five levels of self-awareness as they unfold early in life. *Consciousness and Cognition, 12*(4), 717–731.

Rochat, P., & Striano, T. (1999). Emerging self-exploration by 2-month-old infants. *Developmental Science, 2*(2), 206–218.

Rutkowska, J. (1994). Scaling up sensorimotor systems: constraints from human infancy. *Adaptive Behaviour, 2*, 349–373.

Schmidhuber, J. (1990). Learning algorithms for networks with internal and external feedback. In *Proceedings of the 1990 connectionist models summer school* (pp. 52–61), San Mateo, CA.

Smith, L., & Gasser, M. (2005). The development of embodied cognition: six lessons from babies. *Artificial Life, 11*(1–2), 13–29.

Spelke, E. (1998). Nativism, empiricism, and the origins of knowledge. *Infant Behavior and Development, 21*(2), 181–200.

Sporns, O., & Edelman, G. (1993). Solving Bernstein's problem: a proposal for the development of coordinated movement by selection. *Child Development, 64*(4), 960–981.

Sterling, P. (1999). Deciphering the retina's wiring diagram. *Nature Neuroscience, 2*, 851–852.

Thelen, E. (1995). Motor development. *American Psychologist, 50*, 79–95.

Thelen, E., & Whitmyer, V. (2005). Using dynamic field theory to conceptualize the interface of perception, cognition, and action. In J. Lockman & J. Rieser (Eds.), *Action as an organizer of learning and development, Minnesota symposium on child psychology* (vol. 33, pp. 243–277). New York: Lawrence Erlbaum Associates, Inc.

Tronick, E. (1972). Stimulus control and the growth of the infant's effective visual field. *Perception and Psychophysics, 11*(5), 373–376.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind, 59*(236), 433–460.

Ververs, I., Gelder-Hasker, V., Marja, R., De Vries, J., Hopkins, B., Van Geijn, H. (1998). Prenatal development of arm posture. *Early Human Development, 51*(1), 61–70.
Westermann, G., & Mareschal, D. (2004). From parts to wholes: mechanisms of development in infant visual object processing. *Infancy, 5*(2), 131–151.
Wurtz, R., & Goldberg, M. (1972). The primate superior colliculus and the shift of visual attention. *Investigative Ophthalmology & Visual Science, 11*(6), 441–450.

The Hierarchical Accumulation of Knowledge in the Distributed Adaptive Control Architecture

Encarni Marcos, Milanka Ringwald, Armin Duff, Martí Sánchez-Fibla, and Paul F.M.J. Verschure

Abstract Animals acquire knowledge as they interact with the world. Several authors define this acquisition as a chain of transformations: data is acquired and converted into information that is converted into knowledge. Moreover, theories on cumulative learning suggest that different cognitive layers accumulate this knowledge, building highly complex skills from low complexity ones. The biologically, based Distributed Adaptive Control cognitive architecture (DAC) has been proposed as a cumulative learning system. DAC contains different layers of control: reactive, adaptive and contextual. This hierarchical organization allows for acquisition of knowledge in a bottom-up interaction, i.e. sampled data is transformed into knowledge. DAC has already been used as a framework to investigate fundamental problems encountered in biology. Here we describe the DAC architecture and present some studies focused on its highest cognitive layer where knowledge is constructed and used. We investigate the roles of reactive and contextual control depending on the characteristics and complexity of the tasks. We also show how multi-sensor information could be integrated in order to acquire and use knowledge

E. Marcos (✉) · A. Duff · M. Sánchez-Fibla
SPECS, Technology Department, Universitat Pompeu Fabra, Carrer de Roc Boronat 138, E-08018 Barcelona, Spain
e-mail: encarnacion.marcos@upf.edu; armin.duff@upf.edu; marti.sanchez@upf.edu

M. Ringwald
INI Institute of Neuroinformatics, UNI-ETH Zürich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland
e-mail: mila@ini.phys.ethz.ch

P.F.M.J. Verschure
SPECS, Technology Department, Universitat Pompeu Fabra, Carrer de Roc Boronat 138, E-08018 Barcelona, Spain

ICREA Institució Catalana de Recerca i Estudis Avançats, Passeig Lluís Companys 23, E-08010 Barcelona, Spain
e-mail: paul.verschure@upf.edu

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_10,

optimally. Finally, we discuss the possible problems of cumulative learning and the adopted solutions in the context of DAC.

1 Introduction

Intended behavior is mostly learned and depends on a cognitive system that can acquire the targets and the means necessary to achieve them in a given situation, sculpted by experience (Abbott et al. 1985; Barsalou and Sewell 1985; Norman and Shallice 1986). Animals accumulate knowledge and abilities that serve as building blocks for subsequent plans and behavior making optimal use of the acquired information in the context of their goals, i.e. rational decision making (Newell 1990). Such layered or sequential learning appears to be an essential mechanism, both in acquiring useful abstractions that serve rational behavior and in producing new foundations for further adaptation. The psychological theory of cumulative learning (Gagné 1968) favors this view suggesting that learning is hierarchical: low complexity skills are transformed into more complex ones achieving higher levels of performance and knowledge. In general, humans and animals make optimal use of this acquired knowledge (Gallistel 1990; Tversky et al. 2001). In foraging, for example, animals are able to adapt to the novelty of the environment, acquiring new strategies to optimize rewards while avoiding risks (Davis 1996; MacDonall et al. 2006; Roberts 1992). Foraging tasks are an example of a goal-oriented exploration for resources, e.g. food or shelter. In this case exploration is firstly driven by reactive behaviors that allow the acquisition of the representations of the space. This representation might contain the integration of multi-modal information to acquire knowledge. Hence, foraging tasks provide a suitable test case to study acquisition of knowledge in artificial control systems.

A number of authors consider knowledge as a hierarchy and distinguish between data, information, and knowledge (Alter 1995; Beckman 1997; Tobin 1996; Van Der Spek and Spijkervet 1997). In this hierarchical structure data is defined as facts, sounds, or images. The filtered, formatted and summarized data is called information. Knowledge is defined as the ideas, rules, and procedures that guide actions and decisions. Knowledge is acquired from information, it is generally personal, subjective, and inherently local. Knowledge is internalized by the knower and it is shaped based on previous perceptions and experiences (Hey 2004). Therefore, data, information, and knowledge is seen as a chain of transitions where data is transformed into information and information is transformed into knowledge. The original meaning of the verb inform is "to give form to." Therefore the transformation of data into information is seen as a process of shaping data in a useful way to *make sense out of it*. Useful is in this case defined in terms of generating actions that allow the realization of goals. The transformation of information into knowledge is explained as an accumulation of multiple pieces of information providing structure to it (Hey 2004).

In biological systems the generation of internal representations from external data is achieved through perceptual learning (Gibson and Gibson 1955). The majority of perceptual learning models are based on statistical methods (Becker and Plumbley 1996; Hopfield 1982; Rosenblatt 1958). Generally the sensory input is compressed conserving the most relevant features. However, in most behavioral models, perceptual learning is practically ignored (Duff et al. 2010). Behavioral learning is reduced to associating actions, i.e. acquire policies, to predefined states or sequences of states that describe the task domain. The acquisition of these states and the adaptation of the action association to a changing state space are generally not considered (Barraclough et al. 2004; Klopf 1988; Montague et al. 1995; Sutton and Barto 1981). These models can develop policies for well-defined state spaces but do not explain how the state space they are operating on is constructed or learned in the first place. In a similar way, machine learning techniques based on Markov decision processes (Burago et al. 1996; Kaelbling et al. 1998) require a predefined state space on which behavioral learning can be performed. However, in a system that generates the internal states through its interaction with the environment, perceptual and behavioral learning have to be considered and treated as inter-dependent processes. The robot-based cognitive architecture Distributed Adaptive Control architecture (DAC) (Verschure and Althaus 2003; Verschure et al. 1993, 2003; Verschure 2012) has been explicitly defined to address this problem of priors and it comprises both perceptual and behavioral learning in a unified framework (Verschure and Coolen 1991; Verschure et al. 2003). DAC shows how in an incremental learning process data is transformed into information and knowledge and subsequently integrated into optimal policies for goal-oriented action through the continuous interaction between and agent and its environment.

DAC is a robot-based neuronal model of classical and operant conditioning (Verschure and Althaus 2003; Verschure and Coolen 1991). Classical conditioning is a form of associative learning (Pavlov 1927) where the presentation of a neutral stimulus (conditioned stimulus *CS*) together with a significant stimulus (unconditioned stimulus, *US*) leads to an association of the initially neutral stimulus to a, so-called, conditioned response (*CR*). In one interpretation the CS substitutes the *US* because the behaviorally significant stimulus (*US*) triggers an unconditioned response as an innate automatic response that forms a template for the CR. If *CS* and *US* are repeatedly paired, the two stimuli become associated and the organism begins to behavioral respond to the presence of *CS* alone. One typical paradigm is eyelid conditioning where an air-puff (*US*), after which inevitably the animal reacts with an eyelid closure, is paired with the presentation of a tone (*CS*). After a number of trials the animal begins reacting to the *CS* with a *CR* similar to the *UR*, even if the air-puff (*US*) is not present anymore (Mackintosh 1990). Operant, or instrumental, conditioning is also a form of associative learning. However, the association is not always as direct as in classical conditioning. Series of actions are needed to reach a reward or punishment (Thorndike 1911). These actions are weighted with different values depending on the *US* resulted from an action, i.e. with an appetitive or aversive US, so the ones that leaded to a reward will occur with greater frequency that the ones that were paired with punishment. DAC proposes that Classical and

Operant Conditioning reveal a fundamental scaffolding of learning that advances through three stages. First, sensor statistics based on *perceptual learning* provides a "neutral" representation of the state space. Second, mechanisms underlying classical conditioning provide for a biasing of this state space representation with respect to its immediate survival value (construction of *CS* representations) and the shaping of discrete actions (tuning of the amplitude time course of the *UR* to define the *CR*). Subsequently operant conditioning builds on the representational building blocks provided by the preceding two stages to construct plans for actions and provide a foundation for cognition and problem solving.

DAC has been investigated using formal approaches (Verschure and Althaus 2003; Verschure and Coolen 1991) and robots (Verschure et al. 1993, 2003). In this chapter, we review some of the studies done using the DAC architecture. We focus the review on the ability of DAC to display cumulative learning from low reactive control to high-level cognitive capacities. In particular, we show how reactive and cognitive behaviors might complement each other and how they scale as task complexity increases (Marcos et al. 2010). We also discuss the importance of integrating multi-sensory information in memory and its implications in the acquisition of information and construction of knowledge (Ringwald and Verschure 2007b).

2 Description of DAC

The DAC architecture is based on the assumption that learning consists of the interaction of three different layers of control: reactive, adaptive, and contextual, as illustrated in Fig. 1. The most basic behavior generated at the lowest reactive layer allows interaction with the environment while acquiring information which is converted into knowledge by the higher adaptive and contextual layers. We will briefly describe the characteristics and dynamics of the three layers focusing on how knowledge is obtained and used by the highest layer of the architecture, i.e. the contextual layer.

2.1 Reactive and Adaptive Layers

The reactive layer provides pre-wired responses from stimuli to actions that allows for a simple interaction with the environment ($US \rightarrow UR$), e.g. aversive or appetitive actions. This reactive behavior allows the sampling of data from the environment which are learned and classified in terms of their valence and associated actions by the adaptive layer. The adaptive layer creates internal representations of the data and learns to associate these internal representations with actions ($CS \rightarrow UR$). In this way, data sampled from the environment is converted first into information (internal representations) and then into knowledge (association between *CS* and *UR*). Later on, this knowledge is accumulated in the contextual layer.

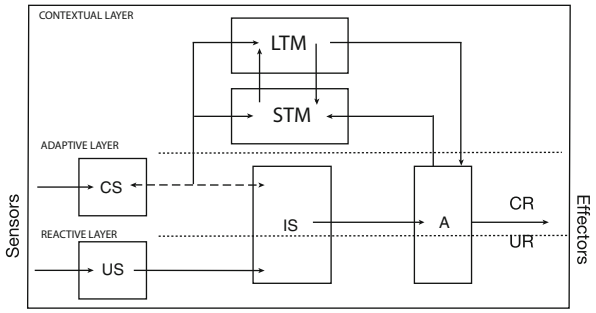

Fig. 1 Schematic representation of the DAC architecture. It is based on the assumption that behavior results from three tightly coupled layers of control: reactive, adaptive, and contextual. *Squared boxes* stand for neuronal groups. *Arrows* stand for static (*solid*) and adaptive (*dashed*) synaptic connections between cell groups. Abbreviations mean: *US*, unconditioned stimulus; *CS*, conditioned stimulus; *IS* , internal states; *A*, action group; *UR*, unconditioned response; *CR*, conditioned response; *STM*, short-term memory; *LTM*, long-term memory

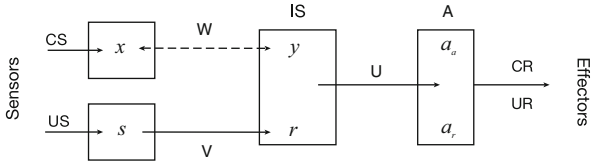

Fig. 2 Reactive and adaptive layer. *Squared boxes* stand for cell (neuronal) groups. *Arrows* stand for static (*solid*) and adaptive (*dashed*) synaptic connections between cell groups. Abbreviations mean: *US*, unconditioned stimulus; *CS*, conditioned stimulus; *IS*, internal states; *A*, action; *UR*, unconditioned response; *CR*, unconditioned response; *W*, *V*, and *U* are connection matrices

Following the abbreviations from Fig. 2, the actions generated by the reactive (a_r) and adaptive (a_a) layers are defined as:

$$a_r = U^{\mathrm{T}} r H(r - \theta^A) \tag{1}$$

$$a_a = U^{\mathrm{T}} y H(y - \theta^A) \tag{2}$$

where r stands for the contribution of the *US* to *IS* $\in \mathbb{R}^K$, y stands for the contribution of the *CS* to *IS* $\in \mathbb{R}^K$, U is the weight matrix from the *IS* to the *A* cell group, and *H(.)* is the Heaviside or step function.[1] Their mathematical expression is: $r = V^{\mathrm{T}} s$ and $y = W^{\mathrm{T}} x$, where V is the weight matrix from *US* to *IS* cell group $\in \mathbb{R}^{NxK}$ and W is the weight matrix from *CS* to *IS* cell group $\in \mathbb{R}^{NxK}$.

[1] $H(x - \theta^A)$ is 1 if $x \geq \theta^A$ and 0 if $x < \theta^A$

The weight matrix W changes following a, so-called, predictive Hebbian learning rule (Verschure and Pfeifer 1992) as:

$$\Delta W = \eta(x - \gamma e)((1 - \zeta)y + (1 + \zeta)r)^{\mathrm{T}} \tag{3}$$

where e will be referred as *CS* prototype and is the backwards projection of y to the *CS* cell group described as $e = Wy$, η is the learning rate, ζ, $\zeta \in [-1, 1]$, balances the influence of the behavioral (xy^{T}) and perceptual (xr^{T}) learning and γ controls the influence of e assuring convergence. This learning rule directly captures the Rescorla and Wagner laws of associative competition that essentially state that animals only learn when events violate their expectations (Rescorla and Wagner 1972).

For further explanation about the dynamics of the reactive and adaptive layers, see Duff et al. (2010); Duff and Verschure (2010).

2.2 Contextual Layer

The contextual layer provides the means for memorizing and recalling behavioral sequences. It consists of two structures: short-term memory (STM) and long-term memory (LTM).

The contextual control is based on the following assumptions:

- Memorize:
  - STM stores sensory-motor events generated by the adaptive layer.
  - When a goal state is reached the content of the STM is stored in the LTM and the STM is initialized.
- Recall:
  - The content of LTM is compared with ongoing sensory events.
  - Matching elements contribute to action selection.

The STM is a ring buffer with fixed size N_S. Every element of this memory is called a *segment*. A series of consecutive segments is called *sequence*. The STM is formed by one sequence of N_S segments. At each moment the generated CS prototype e and the action a executed by the robot are stored in the STM. When a goal state is reached, the sequence stored in the STM is copied into the LTM and the STM is reset. The LTM has an N_L number of sequences. The size of the LTM is, therefore, $N_L x N_S$. The sequences in the LTM are defined by their different goal states, e.g. -1 for an aversive event such as a collision, $+1$ for an appetitive event such as a reward (Fig.3).

Initially only the reactive and adaptive layers are active. The contextual layer is enable when the predicted *CS* prototype (*CS* e) approximates the actual *CS* (*CS* x) prototype. The quality of this matching is defined by an internally generated

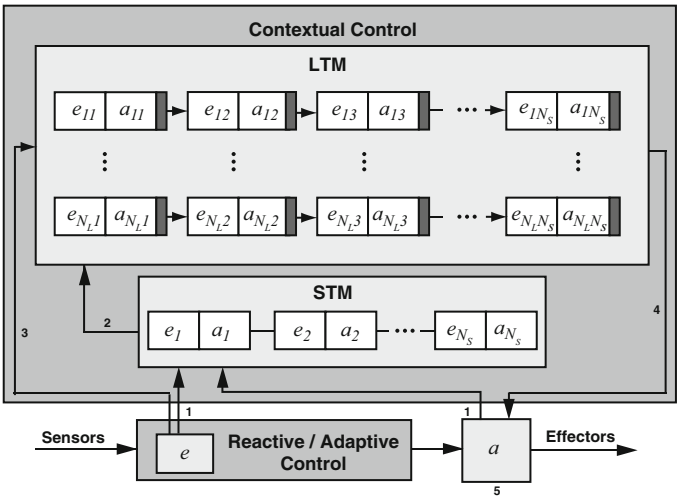

Fig. 3 Contextual layer. (1) The *CS* prototype *e* and the executed action *a* are stored in the STM as a segment. (2) When a goal state is reached the content of the STM is copied in the LTM as a sequence and the STM is reset. (3) The values of the generated *CS* prototype *e* are matched against those stored in the LTM (4) The *A* population receives as an input the action response calculated as a weighted sum over the memory segments. (5) The actions proposed by reactive, adaptive, and contextual layers compete in a priority selection mechanism to control the robot

discrepancy measure (*D*) that is running an average distance between the prototypes *CS x* and *CS e*.

$$D(t+1) = \alpha_D D(t) + (1-\alpha_D) d(x,e) \tag{4}$$

where α_D, $\alpha_D \in [0\ 1]$, is an integration time constant and the distance *d(x,e)* is calculated as:

$$d(x,e) = \frac{1}{N}\sum_{j=1}^{N}\left|\frac{x_j}{\max(x)} - \frac{e_j}{\max(e)}\right| \tag{5}$$

The contextual layer is activated when *D* falls below a certain *confidence threshold*.

During the recall all the *CS e* prototypes stored in the LTM are matched against the generated *CS e* prototype. The degree of matching of segment *l* in sequence *q* determines the input to its, so-called, *collector*:

$$c_{lq} = (1 - d(e, e_{lq}))t_{lq} \tag{6}$$

The collector determines the contribution of the segment to the action selection. Its activity depends on the distance *d(.)* of the generated *CS e* prototype to the *CS e* prototype stored in the segment and on a, so-called, *trigger value t*. The trigger value biases the sensory matching process of the segments and allows chaining through

a sequence. Its default value is 1 and does not bias the collector value. When a segment *l-1* in sequence q is activated the trigger of segment l is set to a value higher than 1. This means that a segment, following a previously effective one, will be given higher priority in future decision making. The trigger decreases its value to 1 asymptotically according to:

$$t_{lq}(t+1) = \alpha_t + (1-\alpha_t)t_{lq}(t) \tag{7}$$

where $\alpha_t \in [0\ 1]$. The trigger of a selected segment resets its value to 1.

The collectors that will contribute to the action proposed are those that satisfy: (1) its activity is above a certain threshold (θ^C) and (2) its activity is inside a predefined percentage range from the maximum collector's activity, i.e. the collectors compete in an E%-Max Winner Take All (%E-Max WTA) mechanism (de Almeida et al. 2009). The actual action proposed from the contextual layer is calculated as:

$$a_c = \sum_{l,q \in LTM} \pm \frac{c_{lq} H(c_{lq} - \theta^C)}{\delta_{lq}} a_{lq} \tag{8}$$

where δ_{lq} is the distance measured in segments between the selected segment l and the last segment in the sequence, i.e. the distance to the goal state. By doing this division the segments closer to the goal have more impact on the contextual action. The sign is plus if the segment belongs to an appetitive sequence and minus if it belongs to an aversive sequence. If a_c results in a negative action, it is filtered out to avoid backwards actions.

The final action executed by the robot is either a_r, a_a or a_c based on their priority: reactive actions have the highest priority then contextual actions and then adaptive actions.

We described above four main changes with respect to the contextual layer of DAC5 (Verschure and Althaus 2003; Verschure and Voegtlin 1998; Verschure et al. 2003). These changes are: (1) the collector unit function (2) the WTA mechanism to select the collectors contributing to the contextual action, (3) the filtering of negative actions, and (4) the priority of actions.

3 Results

We investigate the acquisition of knowledge at the contextual layer of the DAC architecture. First, we investigate the roles of the reactive and the contextual layers and we show how the acquired knowledge at the contextual layer is fundamental for resolving specific tasks (see Sect. 3.1). Then, we propose two methods for spatial integration of information and we show how they improve memory performance as well as influences the formation of knowledge (see Sect. 3.2).

The two studies presented here were tested with a simulated agent. It was implemented in C++ and wSim (Wyss et al. 2006) using the Open Graphics Library approximating a Khepera robot[2] widely used for behavioral modeling. The validity of the simulated robot with respect to a real one has been demonstrated in several studies (Wyss et al. 2006). The robot had a radius of 5.5 cm and it was equipped with eight light sensors and eight proximity sensors. The sensors return a signal based on an exponential decay function that depends on the distance to the light sources or to the obstacles, respectively. The robot was also equipped with a color camera pointing to the floor (with an angle of 45°). The robot moves forward with a speed of 0.1× robot radio and rotates with a speed of 10°. The action group from the architecture was connected to the motor group of the robot. Each cell of the motor group mapped a direction of movement. A winner-take-all (WTA) took place at the motor map level and selected the neuron with highest activity. The default movement of the robot was to move forward.

3.1 The Complementary Roles of Reactive and Contextual Control Systems

Reactive behavior is essential to explore and learn about novel environments. This low cognitive behavior makes possible the interaction with the environment while higher cognitive areas acquire knowledge to exploit it later on. In some situations, reactive behavior is enough to fulfill the internal motivation of the agent, but as the complexity of the environment increases higher cognitive behavior is necessary to optimally exploit the environment. To understand the possible complementary roles of high and low cognitive behavior during foraging tasks we studied the scalability of the contextual and the reactive layers of DAC as task complexity increases. To do so, we equipped the reactive layer with a self-regulatory process that controls the behavior of the robot based on its internal motivation, e.g. food or shelter. The reactive behavior allows the learning of the rules of the environments by the contextual layer. Once this is done, the animal is able to use this knowledge to reach desired positions in an environment.

The self-regulatory process of the robot's internal motivation is based on the concept of *homeostasis* (see Cannon 1929 for a review). It contains a set of homeostatic subsystems (Fig. 4). Each of these subsystems is related to an internal motivation and respond to a gradient placed in the environment. The robot senses the gradient and performs reactive actions in order to place itself in the desired position inside that gradient. Each homeostatic subsystem contains an actual (V_a) and desired (V_d) value in a specific gradient that corresponds to the goal state in that gradient. The homeostatic subsystem tries to get close the actual and the desired value by computing an action that reduces the difference between the two values.

[2]K-Team, Lausanne, Switzerland

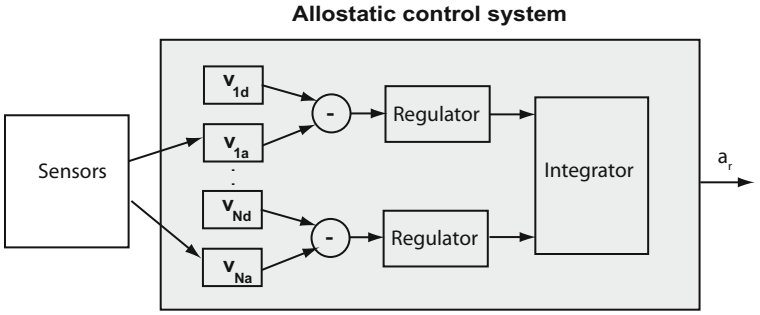

Fig. 4 Self-regulatory control in the reactive layer. Abbreviations mean: V_{xd}, desired value; V_{xa}, actual value, with x ranging from 1 to N subsystems

In the contextual layer, sequences of perception-action are stored and labeled with a number that indicates the internal motivation they relate to. The stored actions are allocentric actions that contain the orientation needed to go from the current position selected from memory to the following one.

The role of contextual and reactive systems was tested in three different foraging tasks, each of them containing visual cues and rewards. The visual cues were patches on the floor whereas the rewards were gradients. The internal motivation of the robot was satisfied when the robot reached the highest point of a gradient. The first task was an open field task with one reward and a direct path (no osbtacles) from any position in the environment to the reward. The gradient of the reward varied its size from 0 to a size that covered the whole space. The second and third tasks were remotely inspired to the Tolman maze (Tolman and Honzik 1930) as in Hartland et al. (2009). The second one had one reward whereas the third one had three different kinds of reward. Every trial finished when the robot reached its goal state and started from the same position but with a random orientation. In these two tasks, obstacles were place in such a way that there was not a direct path between the robot's initial position and the rewards. In the three tasks, we restricted the size of the memory to a maximum of 40 sequences of 120 segments each. We added 5 % of noise to the motors of the robot to simulate real conditions. For every condition, we ran ten experiments with 20,000 cycles each.

In the open field task we obtained a significantly higher performance in the case of the contextual control system for low values of the gradient (Wilcoxon rank sum test $p < 0.001$). However, as the weight of the gradient increased the reactive control system performed better than the contextual control system. This was not surprising since the direct vision from any position in the environment to the reward allows an optimal calculation of the next action by the reactive control system (Wilcoxon rank sum test $p < 0.01$).

The two last tasks showed the importance of the contextual control system since obstacles do not allow direct path to the reward. With only reactive actions the robot collided with the obstacles in many occasions. The contextual control system was

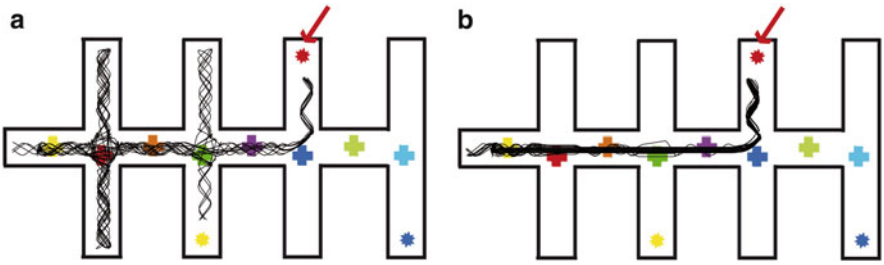

Fig. 5 Trajectory plots when the goal position was indicated by the arrow. *Colored crosses* are visual cues and *colored stars are rewards*. (**a**) Reactive control system. (**b**) Contextual control system

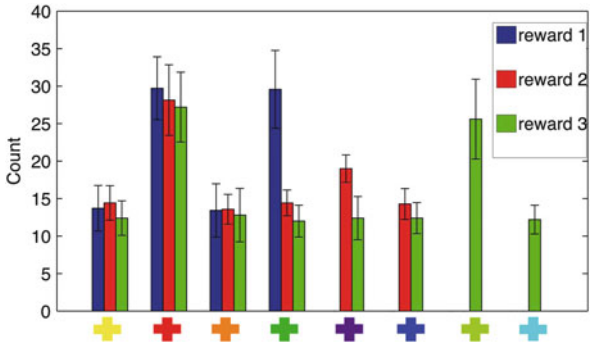

Fig. 6 Count of stored patches. Number of times each patch is stored in memory for each of the three rewards

able to pick up relevant information, convert it into knowledge, and decrease the distance travelled by the robot to reach a goal position (Wilcoxon rank sum test $p < 0.001$). The third task allowed us to generalize the results to more than one reward. Moreover, we also showed that the performance of the robot decreased as the distance to the reward increased. However, we observed that the slope of this decrease in performance was lower in the case of the contextual control system suggesting that the performance of the reactive control system was more affected by the reward's distance. Fig. 5 shows an example of trajectory plots in each layer for this last task.

To better understand how the contextual control system stored information about visual cues and used it to make rules about the different rewards, we looked at the content of the memory in the third task. In Fig. 6 we show the number of times each visual cue was stored in memory for each reward. We observed that the last two patches were used only to reach the third reward whereas the rest of patches were part of the sequences corresponding to all different rewards. Therefore, the memory contained sequences that leaded the robot to the three rewards, but it only used them when they leaded to a reward that fulfilled its internal motivation.

This study allowed us to better understand how reactive and contextual control systems might complement each other and how they might scale with task complexity. Moreover, we proposed that these implications might be extended to animals daily life situations, where same problems might appear in their environments.

3.2 Integration of Spatial Information

Work on hippocampus, the neuronal substrate underlying self-localization and navigation in organisms, suggests that egocentric sensory inputs are stored and linked into behavioral episodes along with allocentric spatial representations (Bower et al. 2005; Dragoi and Buzsaki 2006; Huxter et al. 2003; Lisman 1999; Papp et al. 2007), and that the recall of hippocampal behavioral sequences is accompanied by two phenomena: anticipation of the subsequent location (Dragoi and Buzsaki 2006; Muller and Kubie 1989) and reward (Lansink et al. 2008; Lee et al. 2006; Pennartz et al. 2004). However, it is currently not clear how these concepts are integrated in order to generate structured goal-oriented behavior in the absence of global frame of reference. To explore the local mechanisms of bimodal sensory fusion that could facilitate the transformation of egocentric actions into allocentric ones, we worked in the context of simulated robots performing a foraging task in an open arena. We augmented DAC with spatial cues as a new sensory input, and, designed and evaluated two approaches for bimodal sensor fusion: spatial attention and spatial integration. In the context of the two integration approaches, we designed and investigated the role of the anticipation of the subsequent location and reward on the recall of behavioral sequences.

To integrate the spatial information in DAC, we modified the LTM augmenting its segments to also hold allocentric spatial information, i.e., the position where an event has occurred. We assumed that allocentric spatial information was provided by the place cells of the hippocampus (de Almeida et al. 2009; Guanella and Verschure 2007; Verschure et al. 2006; Wyss et al. 2003; Wyss and Verschure 2004). As an abstract version of the place fields of hippocampal place cells we used perfect 2D-Gaussians, which is consistent with recent work (Wyss and Verschure 2004). Moreover, we proposed two approaches for bimodal sensor fusion: spatial attention and integration. Both methods use spatial information. The spatial attention approach improves local awareness by selecting only those salient cues that are consistent with the spatial information, i.e. segments that match current visual cues go to a second process that compares current spatial information with their stored spatial information. In contrast, the spatial integration approach biases cues, i.e. both, visual cues and spatial information, are used to calculate the likelihood of actions stored in the memory (see Fig. 7). In this case, the collector unit is now given by:

$$c_l{}^k = (1 - d_l{}^k)(1 - m_l{}^k)t_l{}^k \tag{9}$$

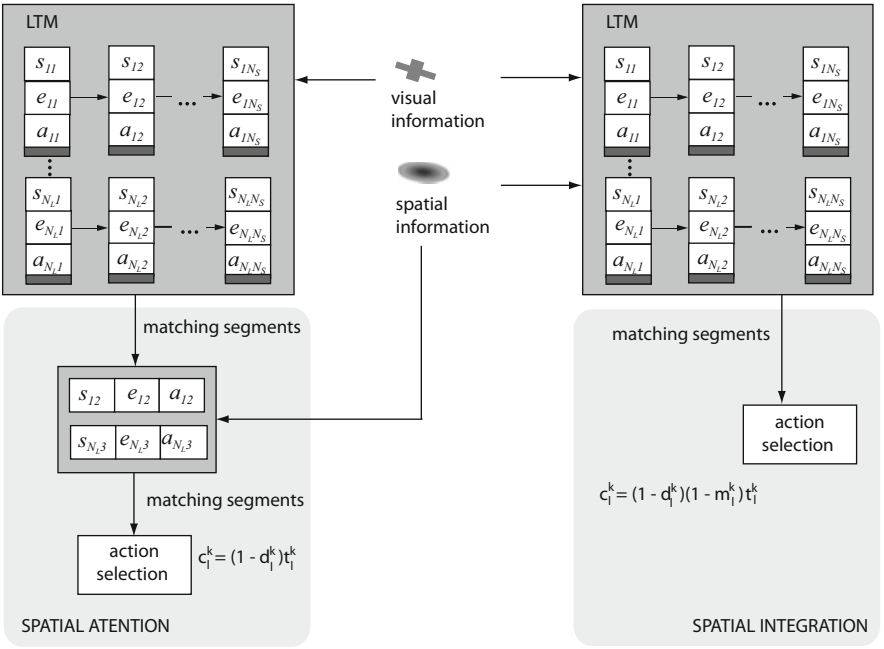

Fig. 7 Spatial attention and spatial integration mechanisms. The content of LTM is the same in both cases: sequences of allocentric spatial information (s_{lq}), prototype (e_{lq}) and action (a_{lq}). *Spatial attention:* The current visual information is compared with the segments stored in LTM. Those matching segments are compared with the current spatial information. Only the visual cue is used to calculate the likelihood of the selected action (c_l^k). *Spatial integration:* Both current visual and spatial information is compared with the stored segments. Visual and spatial information is used to calculate the likelihood of the selected action (c_l^k)

where $d_l{}^k$ is the distance between stored and current prototypes and $m_l{}^k$ is the Euclidean distance between the stored spatial information and the current position of the robot.

As introduced in Sect. 2, DAC implements two mechanisms that bias the memory selection to achieve anticipation of the subsequent location and reward: the trigger unit and the distance to the goal, respectively. The trigger unit implements a feedback mechanism that favors segments following the currently selected one along the corresponding LTM sequence by temporarily increasing their weights. The distance to the goal state is used to assign a higher weight to the segments closer to the goal. We modified these two mechanisms and called them sequence fidelity and goal fidelity, respectively.

We extended the original sequence fidelity (SF) to a mechanism for memory smoothing, defined as follows:

$$SF_l{}^k(t) = 1 - \exp\left(\frac{{d_{\text{segment}_l^k}}^2}{2{\sigma_{\text{smoothing}}}^2}\right) \tag{10}$$

If a segment is selected, its SF value is reset and it falls back over time to its default value according to:

$$SF_l{}^k(t+1) = \alpha_T + (1-\alpha_T)SF_l{}^k(t) \tag{11}$$

where $\alpha_T \in [0;1]$. The contribution of an LTM segment is provided by the value of the collector unit $c_l{}^k$ for segment l of sequence k.

Originally in DAC, a goal fidelity (GF) was introduced by calculating the distance to the goal measured in segments for each selected segment. This value was $+1$ if the sequence leaded to a target, and it is -1 if the sequence leaded to a collision. We introduced two changes in the way the goal fidelity was represented in the system. Firstly, instead of using the distance to the goal state measured in segments, we used the distance measured in number of actual steps that the robot took in order to move between the activated segment and the end of its sequence. In addition, we differentiated how we calculated the goal fidelity value based on the type of the LTM sequence. If the sequence was appetitive, GF was equal to the distance, for an aversive one it was defined as the reciprocal value of it, 1/distance. By this, we biased the memory to follow the shortest way to a target, and the longest way to an obstacle.

In Ringwald and Verschure (2007a) we evaluated the performance of the two integration approaches compared to two control groups in which the recall of the behavioral sequences is based on either the spatial signal or the visual cues stored in LTM. The four set of experiments used solely sequence fidelity to bias the LTM. We used a simulated robot in an open-arena foraging task, where two lights were used as targets. These two targets were detectable by the light they emitted. Only one target was active at a time. As soon as the robot reached the active target, it was deactivated and the other one was activated. The size of the arena was 40 by 25 units, where the unit size equals to the size of a robot. We ran 150,000 cycles for each experiment, measured in the last 50,000 cycles, and averaged over 30 experiments. The LTM memory was constrained to 100 sequences of 50 segments. We showed that the maximum performance for a bimodal approach was seven times better than the maximum performance for an unimodal approach. In addition to this, the memory content of the two control groups showed significantly less information about the environment compared to the bimodal groups.

We also evaluated the performance of the two integration approaches that use both sequence fidelity and goal fidelity to bias the LTM. We performed two sets of experiments in the same open arena as explained above. The first used the "spatial attention" method, while the second assessed the impact of the "spatial integration" method. In each experiment, we let the system run and learn with only the sequence fidelity bias enabled for 100,000 cycles. After this period the learning was stopped, and we let the system run 100,000 additional cycles. In the first half of this test

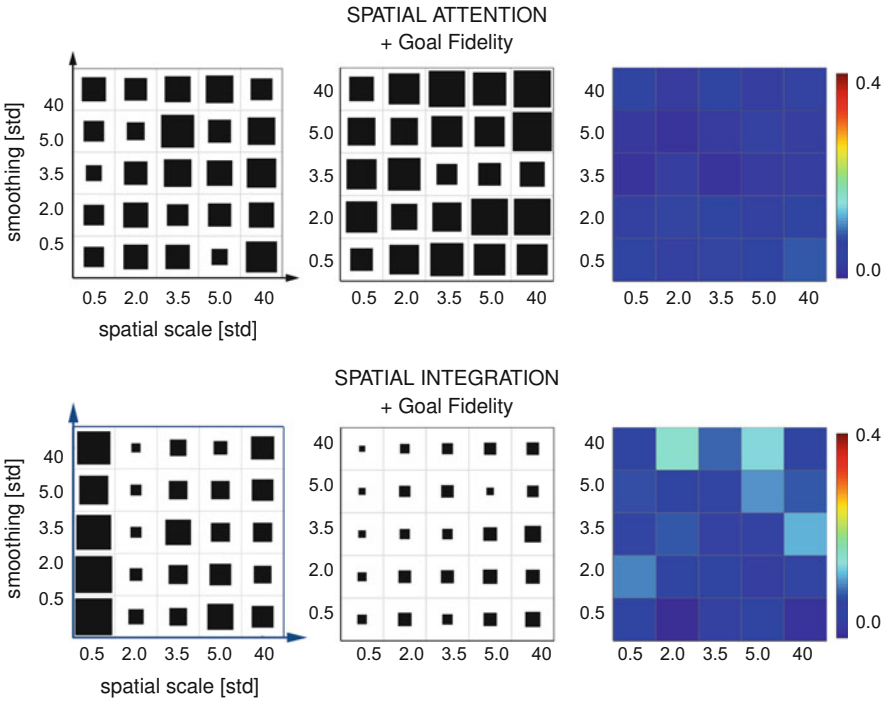

Fig. 8 Performance and memory content. The performance is given in target rate, i. e. the slope of the regression line of the accumulated targets averaged over all experiments. Results are given for both integration scenarios in two cases: first when LTM sequences are biased using only sequence fidelity mechanism, and second with the goal fidelity in addition. The quality of the memory content is shown in the third column plots for both integration scenarios. To asses the quality of the memory content we employed the mutual information method (Mackay 2003) to quantify the amount of missing information in the LTM: 0 means no information was missed.

phase, only the sequence fidelity bias is enabled. In the second half of the test phase, both biases influenced the LTM sequence recall.

In Fig. 8 we show the performance of the two integration approaches with respect to two parameters: scale of spatial information and sequence fidelity. The scale of the spatial information represents the standard deviation of the 2D Gaussian, with the unit size equal to the size of the robot. It directly influences the spatial cue fidelity. The other parameter is the scale of the memory smoothing ($\sigma_{\text{smoothing}}$ in 10). We observed that in the case of sequence fidelity bias mechanism, both factors, scale of the memory smoothing and the sparseness of the spatial information, have an impact on the performance. If the uncertainty of the spatial information and the memory smoothing are in the lower range, the "integration" approach outperforms the "spatial attention." However, a low variance in memory content and performance across the upper range of memory smoothing and spatial scale of the "spatial attention" suggests the feasibility of a combined approach. In the case of the goal

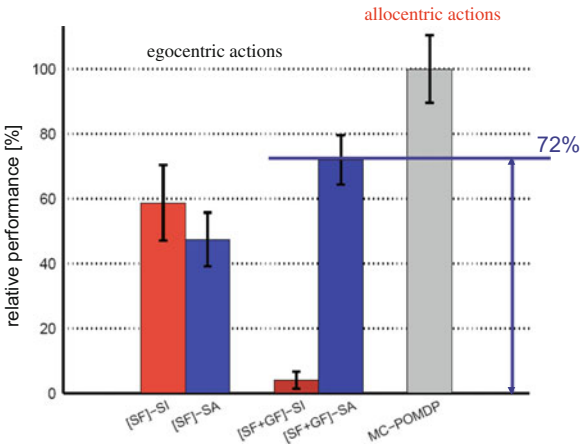

Fig. 9 Comparison of the performance of the DAC-based approaches using egocentric actions and Monte-Carlo Partially Observable Markov Decision Process (MC-POMDP), an optimal Bayesian algorithm using allocentric actions. All approaches use a continuous state space

fidelity mechanism and the "spatial attention" approach, the increase in performance is observed on the whole parameter range. However, in the "integration" approach there is an overall decrease in performance.

The spatial attention method resulted in a low variance in memory content and performance across the given range of memory smoothing and spatial scale values. As a result of the better exploration/exploitation strategy at the higher values, it outperformed the integration method. While the weight of the action stored in the LTM segment was determined solely by the similarity of the disambiguated visual cue, these results further suggest that between the two sources of inputs, spatial and vision driven, the latter had a dominant role. In summary, we showed that the integration of the two sensory inputs not only reduces ambiguities of the egocentric frame reference, however, at the same time, it improves both: an exploratory behavior of the robot and its performance. In addition, we showed that the two integration approaches are complementary, suggesting that spatial cognition is a two-stage process in which the probabilistic integration precedes the spatial attention mechanism.

To evaluate our approach in a broader context, we compare the results of the four DAC-based groups with the performance of the two lower DAC layers and with the MC-POMPD method (Thrun 2000), one well-known method that learns to optimally act in an environment. This method approximates the belief state with a set of particles and employs the value iteration method to learn the optimal policy with allocentrically defined actions. MC-POMDP works over continuous action and state spaces and it employs the value iteration technique for optimal action selection. For the purpose of comparison with DAC, a discrete set of actions (NE, N, NW, W, SW, S, SE, E) is used and distance to the next goal is provided directly as a reward. Each so-called "belief" state is represented by a set of hundred samples, and up to ten similar belief states are stored in a memory. As the performance of the four DAC groups is comparable when spatial uncertainty is set to 0.5 and sequence fidelity is set to 2.0, in all four DAC-based groups we use these two parameter values.

The results in Fig. 9 show that the contextual layer achieves a performance three times better than the one observed in the lower layers of DAC. If we take the performance of the MC-POMDP algorithm as the empirical maximum, we see that the spatial attention approach, which uses combination of sequence and goal fidelity biases, achieves the 72 % of MC-POMDP performance. The performance gap between the DAC-based spatial attention approach and the MC-POMDP algorithm decreases to 10 % when we compare MC-POMDP with the best performing DAC group.

In this study we investigated the influence of a sequence fidelity and goal fidelity bias on the performance of two integration approaches, namely "spatial attention" and "spatial integration," with respect to two parameters: sparseness of sensory input and memory smoothing. We showed that the organization of behavioral sequences is supported by two anticipatory mechanisms: anticipation of the future location and the anticipation of reward. In particular, our results show that the approaches which use both the visual and spatial information perform better than the unimodal ones. At the same time, the bimodal integration approaches improve the robot's exploratory behavior, which we conclude by inspecting the memory contents. In the spatial attention method, the "sequence fidelity" and "goal fidelity" biasing mechanisms facilitate the transformation of egocentrically defined actions into allocentric behavior. In particular, spatial integration performs better when we have precise spatial information and there is no additional goal fidelity bias, whereas spatial attention performs better when we do not have precise spatial information, and we employ both sequence fidelity and goal fidelity biases. These results imply that the spatial integration approach is a good candidate for goal-oriented tasks where spatially congruent stimuli are expected to drive the responses. In contrast to spatial integration, the spatial attention approach is a good candidate for goal-oriented tasks that require choice between alternative strategies. In addition, we compared DAC performance with the performance of the MC-POMDP method. We showed that the DAC of performance was only 10 % lower than the performance of the MC-POMDP method. This result is outstanding since DAC does not require a priori global information to solve a task, i.e. it uses egocentric actions vs. the allocentric ones used by the MC-POMDP method.

4 Discussion

In this chapter we have described the hierarchical accumulation of knowledge in DAC. We have mainly focused on the transformation from sampled data into filtered and formatted data, i.e. information, and from there into rules and procedures that guide actions and decisions, i.e. knowledge. We have specifically emphasized the most recent studies carried out at the contextual level and how the acquisition of knowledge at this level depends on sampled data and extracted information. In concrete, we have also studied the complementary roles of reactive and contextual layers in tasks in which by basic reactive behaviors goal states are reached and

tasks where acquisition of knowledge at the contextual layer is fundamental for obtaining a performance above chance level. In addition, we have also investigated the importance of multi-modal integration in acquiring and recalling knowledge. We have shown that integrating data from visual and spatial cues results in an improvement of the exploratory behavior of the robot.

To study how reactive and contextual layers complement each other we added homeostatic subsystems to the reactive layer to regulate and satisfy the different internal motivations of the robot and extended the memory content of the contextual layer to store and retrieve information related to these internal motivations, i.e. goal states. Furthermore, we integrated both systems converting the egocentric actions from the reactive behavior ones for the contextual control system. We tested this model in a variety of robot foraging tasks. We showed that only reactive behavior is necessary to fulfill the robot's needs when there is a direct path between the robot and the reward. However, when there is not a direct path to reach a reward acquisition of knowledge at the contextual layer is fundamental.

By extending and integrating multiple sources of data to acquire first information and then knowledge we have investigated the local mechanisms of bimodal sensory fusion that could facilitate conversion of egocentric actions into allocentric ones. We have proposed two different integration methods: spatial attention and spatial integration which differ in the way they recall information from memory. We have shown that variations in the scale of memory smoothing and the spatial information have an impact on the performance of both models. A narrow spatial window helps selecting the correct sequences and thus reduces the ambiguity of the information stored in the memory. Widening of the spatial window degrades the performance by introducing additional noise into the system leading to the inclusion of incorrect sequences in the decision-making process. These two integration approaches have shown to be complementary indicating that spatial cognition is a two-stage process where the spatial attention mechanism preceded by the probabilistic integration.

Many different cognitive architectures have been proposed (see Vernon et al. 2007 for a review). DAC differs from them in that it is a self-contained learning system that demonstrates how perceptual and behavioral control can be understood as a bottom-up organization. The reactive layer provides prewired reflexes that allow for an interaction with the environment while sampling sensory data from it. These sensory data are processed and classified by the adaptive layer, which creates representations of sensory stimuli, transforming data into information. In addition to these created representations, the adaptive layer extracts knowledge from this information associating between events and actions. At the level of the contextual layer the information extracted by the adaptive layer is used to express relationships between sensory events and motor actions over time and to extract rules to reach specific targets, creating a representation of knowledge. This learning model is self-contained in the sense that the specifications about an environment are acquired through interaction with the world and are continuously updated in relation to the experience of the agent, by changes in the adaptive layer classification of sensory events or by changes in the formation of new sequences by the contextual layer. A fundamental difference between DAC and traditional cognitive systems is that

DAC knowledge is accumulated in the adaptive and contextual layers through a continuous interaction with the world, as opposed to the traditional cognitive systems where the knowledge is specified a priory. Hence, this difference leads to a change in the boundaries, i.e. traditional models are bounded by their predefined world in contrast to DAC which is bounded by the complexity of the real world making the acquisition of knowledge wider and dynamically adapted to changing environment.

One example of cumulative learning models are the classifier systems (Holland 1986; Holland and Reitman 1978; Wilson 1987) which are specialized applications of genetic algorithms (Holland 1975). They have the ability to acquire knowledge from the outside world and to retain it for future problem solving in dynamic environments. However, one of their limitations is that they assume that changes in the environment are smooth. Therefore, these systems cannot account for dramatic changes. Moreover, the learned rules are acquired and expressed in a short-term memory which leads to the problem of forgetting inactive information over time. This problem is critical in cumulative learning. Therefore, cumulative learning systems have to account for solutions to fundamental problems like the already mentioned influence of intrinsic motivation in formation of knowledge, as well as account for dynamic changes of an environment and forgetfulness of previous acquired information, among others. As presented in this chapter, DAC has allowed us to investigate some of these issues, but some of them, such as the forgetting problem, is still a fundamental issue to be considered. When the memory is full the first stored sequence is deleted so a new sequence can be stored, so newer information automatically replaces old information. This can be a good strategy to follow in static environment, but this might not be the case when the information in the environment changes over time. To study this problem, in Mathews et al. (2010) DAC was extended with an expectation reinforcement method to learn landmark reliability in dynamic environments. The resulted behavior of the model closely resembled real insect navigation.

In summary, we have shown that DAC is an example of a cognitive system where the chaining transformation from data over information to knowledge is fundamental. Different from other cumulative learning models, such as the classifier systems, in DAC the acquisition of knowledge and the behavior are tightly coupled, i.e. actions performed by the robot due to its current knowledge affects the later sampling of data. Therefore, as a biological based architecture, DAC shows that the accumulation of knowledge can be understood in a recurrent process of bottom-up and top-down interaction where executed actions and acquired knowledge influence each other.

References

Abbott, V., Black, J. B., Smith, E. E. (1985). The representation of scripts in memory. *Journal of Memory and Language*, *24*, 179–199.

Alter, S. L. (1995). *Information systems: a management perspective*. Redwood City: Benjamin-Cummings Publishing Co., Inc.

Barraclough, D. J., Conroy, M. L., Lee, D. (2004). Prefrontal cortex and decision making in a mixed-strategy game. *Nature Neuroscience*, *7*, 404–410.

Barsalou, L. W., & Sewell, D. R. (1985). Contrasting the representation of scripts and categories. *Journal of Memory and Language*, *24*, 646–665.

Becker, S., & Plumbley, M. (1996). Unsupervised neural network learning procedures for feature extraction and classification. *Applied Intelligence*, *6*, 185–203.

Beckman, T. (1997). A methodology for knowledge management. In *Proceedings of the IASTED International Conference on AI and Soft Computing*. IASTED.

Bower, M. R., Euston, D. R., McNaughton, B. L. (2005). Sequential-context-dependent hippocampal activity is not necessary to learn sequences with repeated elements. *Journal of Neuroscience*, *25*(6), 1313–1323.

Burago, D., de Rougemont, M., Slissenko, A. (1996). On the complexity of partially observed markov decision processes. *Theoretical Computer Science*, *157*, 161–183.

Cannon, W. B. (1929). Organization for physiological homeostasis. *Physiological Reviews*, *9*(3), 399–443.

Davis, H. (1996). Underestimating the rat's intelligence. *Cognitive Brain Research*, *3*, 291–298.

de Almeida, L., Idiart, M., Lisman, J. E. (2009). A second function of gamma frequency oscillations: an E%-max winner-take-all mechanism selects which cells fire. *Journal of Neuroscience, 29*(23), 7497–7503.

Dragoi, G., & Buzsaki, G. (2006). Temporal encoding of place sequences by hippocampal cell assemblies. *Neuron*, *50*, 145–157.

Duff, A., Rennó Costa, C., Marcos, E., Luvizotto, A., Giovannucci, A., Sánchez Fibla, M., et al. (2010). Distributed adaptive control: a proposal on the neuronal organization of adaptive goal oriented behavior. In O. Sigaud, & J. Peters (Eds.), *From motor learning to interaction learning in robots*. Berlin/Heidelberg: Springer.

Duff, A., & Verschure, P. F. (2010). Unifying perceptual and behavioral learning with a correlative subspace learning rule. *Neurocomputing*, *73*(10–12), 1818–1830.

Gagné, R. (1968). Learning hierarchies. *Educational Psychologist*, *6*(1), 1–6.

Gallistel, C. (1990). *The organization of learning*. Cambridge: MIT.

Gibson, J., & Gibson, E. (1955). Perceptual learning: differentiation or enrichment? *Psychological Review*, *62*(1), 32–41.

Guanella, A., & Verschure, P. F. M. J. (2007). Prediction of the position of an animal based on populations of grid and place cells: a comparative simulation study. *Journal of Integrative Neuroscience*, *6*(3), 433–446.

Hartland, C., Bredechem, N., Sebag, M. (2009). Memory-enhanced evolutionary robotics: the echo state network approach. In *Proceedings of the eleventh conference on congress on evolutionary computation* (pp. 2788–2795). New York: IEEE.

Hey, J. (2004). *The data, information, knowledge, wisdom chain: the metaphorical link*. Intergovernmental Oceanographic Commission (UNESCO).

Holland, J. (1986). *Escaping brittleness: the possibilities of general purpose learning algorithms applied to parallel nile-based systems*, volume 2. San Mateo: Morgan Kaufmann.

Holland, J., & Reitman, J. (1978). *Cognitive systems based on adaptive algorithms, pattern-directed inference systems*. New York: Academic.

Holland, J. H. (1975). *Adaption in natural and artificial systems*. Boston: MIT.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*, 2554–2558.

Huxter, J., Burgess, N., OKeefe, J. (2003). Independent rate and temporal coding in hippocampal pyramidal cells. *Nature*, *425*, 828–832.

Kaelbling, L., Littman, M., Cassandra, A. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*, 99–134.

Klopf, A. H. (1988). A neuronal model of classical conditioning. *Psychobiology*, *16*, 85–125.

Lansink, C. S., Goltstein, P. M., Lankelma, J. V., Joosten, R. N. J. M. A., McNaughton, B. L., Pennartz, C. M. A. (2008). Preferential reactivation of motivationally relevant information in the ventral striatum. *Journal of Neuroscience*, *28*(25), 6372–6382.

Lee, I., Griffin, A. L., Zilli, E. A., Eichenbaum, H., Hasselmo, M. E. (2006). Gradual translocation of spatial correlates of neuronal firing in the hippocampus toward prospective reward locations. *Neuron*, *51*, 639–650.

Lisman, J. E. (1999). Relating hippocampal circuitry to function: recall of memory sequences by reciprocal dentate-CA3 interactions. *Neuron*, *22*, 232–242.

MacDonall, J., Goodell, J., Juliano, A. (2006). Momentary maximizing and optimal foraging theories of performance on concurrent vr schedules. *Behavioural Processes*, *72*, 283–99.

Mackay, D. J. C. (2003). *Information theory, inference and learning algorithms*. Cambridge: Cambridge University Press

Mackintosh, N. J. (1990). *Conditioning and associative learning*, volume 3. Oxford psychology series Reprint. Oxford: Clarendon Press.

Marcos, E., Sánchez-Fibla, M., Verschure, P. F. M. J. (2010). *SAB. Lecture notes in computer science* (Vol. 6226, pp. 370–379). Berlin: Springer.

Mathews, Z., Verschure, P. F. M. J., Bermdez i Badia, S. (2010). An insect-based method for learning landmark reliability using expectation reinforcement in dynamic environments. In *IEEE International Conference on Robotics and Automation, ICRA 2010, Anchorage, Alaska, USA, 3–7 May 2010* (pp. 3805–3812). New York: IEEE.

Montague, P. R., Dayan, P., Person, C., Sejnowski, T. J. (1995). Bee foraging in uncertain environments using predictive hebbian learning. *Nature*, *377*, 725–728.

Muller, R. U., & Kubie, J. L. (1989). The firing of hippocampal place cells predicts the future position of freely moving rats. *Journal of Neuroscience*, *9*(12), 4101–4110.

Newell, A. (1990). *Unified theories of cognition*. Cambridge: Harvard University Press.

Norman, D. A., & Shallice, T. (1986). Attention to action: willed and automatic control of behaviour. In R. J. Davidson, G. E. Schwartz, D. Shapiro (Eds.), *Consciousness and self-regulation: advances in research and theory* (pp. 1–18). New York: Plenum.

Papp, G., Witter, M., Treves, A. (2007). The CA3 network as a memory store for spatial representations. *Journal of Learning and Memory*, *14*, 732–744.

Pavlov, I. P. (1927). *Conditioned reflexes: an investigation of the physiological activity of the cerebral cortex*. Oxford: Oxford University Press.

Pennartz, C. M. A., Lee, E., Verheul, J., Lipa, P., Barnes, C. A., McNaughton, B. L. (2004). The ventral striatum in off-line processing: ensemble reactivation during sleep and modulation by hippocampal ripples. *Journal of Neuroscience*, *24*(29), 6446–6456.

Rescorla, R., & Wagner, A. (1972). A theory of pavlovian conditioning: variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black, & W. F. Prokasy (Eds.), *Classical conditioning II: current research and theory* (p. 6499). New York: Appleton Century Crofts.

Ringwald, M., & Verschure, P. (2007a). The fusion of multiple sources of information in the organization of goal-oriented behavior: spatial attention versus integration. In *ECMR 07: proceedings of the 3rd European Conference on Mobile Robots*, Germany. http://ecmr07.informatik.uni-freiburg.de/accepted_p.html.

Ringwald, M., & Verschure, P. F. M. J. (2007b). The fusion of multiple sources of information in the organization of goal-oriented behavior: spatial attention versus integration. In *ECMR 07: Proceedings of the 3rd European conference on mobile robots* (pp. 1–6) Germany. http://ecmr07.informatik.uni-freiburg.de/accepted_p.html.

Roberts, W. (1992). Foraging by rats on a radial maze: learning, memory, and decision rules. *Learning and memory: The behavioral and biological substrates* (pp. 7–23). Hillsdale: Lawrence Erlbaum Associates, Inc.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–408.

Sutton, R. S., & Barto, A. G. (1981). Toward a modern theory of adaptive networks: expectation and prediction. *Psychological Review*, *88*, 135–170.

Thorndike, E. (1911). *Animal intelligence*. New York: Macmillan.
Thrun, S. (2000). Monte Carlo POMDPs. In S. Solla, T. K. Leen, & K. R. Müller (Eds.), *Advances in neural information processing systems 12* (pp. 1064–1070). Cambridge: MIT.
Tobin, D. R. (1996). *Transformational learning: renewing your company through knowledge and skills*. New York: Wiley.
Tolman, E., & Honzik, C. (1930). Insights in rats. *University of California Publications in Psychology*, *4*(14), 215–232.
Tversky, A., Slovic, B., Kahneman, B. (2001). *Judgment under uncertainty: heuristics and biases*. Cambridge: Cambridge University Press.
Van Der Spek, R., & Spijkervet, A. (1997). *Knowledge management: dealing intelligently with knowledge*. Knowledge Management Network.
Vernon, D., Metta, G., Sandini, G. (2007). A survey of artificial cognitive systems: implications for the autonomous development of mental capabilities in computational agents. *IEEE Transactions on Evolutionary Computation*, *11*, 151–180.
Verschure, P., & Althaus, P. (2003). A real-world rational agent: unifying old and new ai. *Cognitive science*, *27*, 561–590.
Verschure, P., & Coolen, A. (1991). Adaptive fields: distributed representations of classically conditioned associations. *Network*, *17*(2), 189–206.
Verschure, P., Krose, B., Pfeifer, R. (1993). Distributed adaptive control: The self-organization of structured behavior. *Robotics and Autonomous Systems*, *9*, 181–196.
Verschure, P., & Pfeifer, R. (1992). Categorization, representations, and the dynamics of system-environment interaction: a case study in autonomous systems. J. A. Meyer, H. Roitblat, S. Wilson (Eds.), *From Animals to Animals: Proceedings of the Second International Conference on Simulation of Adaptive behavior* (pp. 210–217). Honolulu: Hawaii, Cambridge: MIT.
Verschure, P., & Voegtlin, T. (1998). A bottom-up approach towards the acquisition, retention, and expression of sequential representations: Distributed adaptive control iii. *Neural Networks*, *11*, 1513–1549.
Verschure, P., Voegtlin, T., Douglas, R. J. (2003). Environmentally mediated synergy between perception and behavior robots. *Nature*, *425*, 620–624.
Verschure, P. F. M. J. (2012). Distributed adaptive control: a theory of the mind, brain, body nexus. *Biologically Inspired Cognitive Architectures*. doi: 10.1016/j.bica.2012.04.005.
Verschure, P. F. M. J., Wyss, R., König, P. (2006). A model of the ventral visual system based on temporal stability and local memory. *PLOS Biology*, *4*(5), 1–8.
Wilson, S. (1987). Classifier systems and the animat problem. *Machine Learning*, *2*(3), 199–228.
Wyss, R., König, P., Verschure, P. (2003). Invariant representations of visual patterns in a temporal population code. *Proceedings of the National Academy of Sciences, USA*, *100*, 324–329.
Wyss, R., König, P., Verschure, P. F. M. J. (2006). A model of the ventral visual system based on temporal stability and local memory. *PLoS Biology*, *4*(5), e120. doi:10.1371/journal.pbio.0040120.
Wyss, R., & Verschure, P. F. (2004). *Bounded invariance and the formation of place fields*. Cambridge: MIT.

Part III
Hierarchical Organization of Animal Brain

The Hierarchical Organisation of Cortical and Basal-Ganglia Systems: A Computationally-Informed Review and Integrated Hypothesis

Gianluca Baldassarre, Daniele Caligiore, and Francesco Mannella

Abstract To suitably adapt to the challenges posed by reproduction and survival, animals need to learn to select when to perform different behaviours, to have internal criteria for guiding these learning processes, and to perform behaviours efficiently once selected. To implement these processes, their brains must be organised in a suitable hierarchical fashion. Here we briefly review two types of neural/behavioural/computational literatures, focussed, respectively, on cortex and on sub-cortical areas, and highlight their important limitations. Then we review two computational modelling works of the authors that exemplify the problems, brain areas, experiments, main concepts, and limitations of the two research threads. Finally we propose a theoretical integration of the two views, showing how this allows us to solve most of the problems found by the two accounts if taken in isolation. The overall picture that emerges is that the cortical and the basal ganglia systems form two highly-organised hierarchical systems working in close synergy and jointly solving all the challenges of choice, selection, and implementation needed to acquire and express adaptive behaviour.

1 Introduction

A distinctive feature of animal behaviour is that it supports multiple sensorimotor activities directed to satisfy multiple survival and reproduction needs in variable conditions. The point of departure of the analysis of this chapter is that hierarchical behaviour in animals can be considered as the result of three pivotal classes of

G. Baldassarre (✉) · D. Caligiore · F. Mannella
Laboratory of Computational Embodied Neuroscience, Institute of Cognitive Sciences and Technologies, National Research Council, Via San Martino della Battaglia 44, I-00185 Rome, Italy
e-mail: gianluca.baldassarre@istc.cnr.it; daniele.caligiore@istc.cnr.it; francesco.mannella@istc.cnr.it

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_11,

processes (cf. Alcock 1998; MacFarland 1993): (a) those leading to the acquisition and expression of specific *sensorimotor/cognitive transformations*, or *skills*; (b) those leading to the *selection* of such skills in different circumstances depending on the needs and goals pursued by the animal; (c) and those *guiding* the learning processes underlying the latter selection and the acquisition of skills based on motivations. The implementation of these processes requires a strongly structured brain architecture organised at multiple functional levels where the top ones exert control over the lower ones but at the same time are influenced by them in their functioning (Meunier et al. 2010). This architecture is inherently *softly-modular*: on the one hand, it encodes in distinct neural populations different skills, goals, and other elements that support behaviour so that they do not interfere with each other; on the other hand, if such elements are similar, they are encoded in partially overlapping neural populations so as to exploit generalisation. The brain architecture is also *hierarchical* to allow the implementation of sensorimotor/cognitive transformations, and selection of chunks of behaviour, at multiple levels of abstraction.

The psychological and neuroscientific literature investigating the hierarchical organisation of brain and behaviour is currently basically split into two research threads having strong characterising features in terms of topics, concepts and methods, and limited interactions. The first research thread, mainly involving the sub-fields of *cognitive neuroscience*, *primate neuro-physiology*, and *neuro-psychology*, focusses mainly on the study of cortical systems, runs behavioural/cognitive experiments with human and non-human primates, investigates the non-human primate brain with neurophysiology and the human brain on the basis of brain-imaging techniques, natural brain impairments, or transcranial magnetic stimulation (e.g., Cisek and Kalaska 2010; Gazzaniga 2004; Rizzolatti and Craighero 2004; Walsh and Cowey 2000). The second research thread, mainly involving the sub-fields of *bio-behavioural studies* and *comparative psychology*, focusses mainly on the study of sub-cortical brain structures, runs experiments with rats and sometimes non-human primates, uses behavioural experiments, investigates the brain based on brain lesions and sometimes physiological recordings (e.g., Cardinal et al. 2002; Yin and Knowlton 2006).

The two research threads tend to focus on two critical but distinct classes of phenomena and concepts related to the hierarchical organisation of brain and behaviour. This leads the two threads to produce an incomplete account if taken in isolation. The objective of this work is to show that the integration of the knowledge coming from the two pieces of literature solves most of those problems, gives a better explanation of the system-level organisation of the brain and behaviour hierarchies, and offers a view that uncovers new challenges for empirical and modelling research.

To achieve this objective we will first briefly present the view of the hierarchical organisation of brain and behaviour offered by the two research threads and then we will highlight their limitations and those of the computational models with which they are supported (Sect. 2). Second, we will further characterise the two views on the basis of a rather detailed review of two of our computational models, one focused on the brain hierarchy involving cortical systems (Caligiore et al. 2010;

Sect. 3) and one focussed on the brain hierarchy involving sub-cortical systems (based on Mannella et al. 2010; Sect. 4). This review will allow us to specify at a computational level the typical functions and mechanisms implemented by the two cortical and sub-cortical systems in support of hierarchical behaviour. Third, based on these analyses, and on other knowledge from the neuroscientific literature, we will advance an integrated view of the system-level organisation of the brain underlying hierarchical behaviour (Sect. 5; some ideas and problems expanded here are introduced in Thill et al. 2013). This will allow us to overcome the explanatory limitations of the two accounts and their related models when considered in isolation, and at the same time will furnish the basis to identify new problems that might be investigated empirically or with computational models.

2 Two Research Threads on Hierarchical Brain and Behaviour: Features and Limitations

The cognitive neuroscience literature tends to explain the hierarchical organisation of behaviour in terms of the underlying hierarchical organisation of cortical pathways (Hamilton and Grafton 2007; Kilner 2011; Lestou et al. 2008; Thill et al. 2013). For example, Fuster (2001) proposes that such organisation, sketched in Fig. 1, is formed by cortical pathways implementing sensorimotor mappings at increasing levels of abstraction. Within it, higher levels control lower ones by performing more integrative computations by encompassing an increasing number of information sources at an increasing level of abstraction. The literature on brain system-level organisation has further specified the components of such hierarchy. The first *sensorimotor pathway* directly maps primary sensory cortex (e.g., somatosensory cortex encoding the current state of the musculoskeletal system) to primary motor cortex (encoding motor commands to muscles) (Pavlides et al. 1993; Tokimura et al. 2000). A second *dorsal neural pathway* goes from visual cortex to associative visual/somatosensory parietal areas (encoding affordances; Evangeliou et al. 2009; Fogassi et al. 2005), and then to premotor cortex (encoding action plans; Rizzolatti and Craighero 2004; Rizzolatti et al. 1996) that exerts control on primary motor cortex. This neural pathway is in turn formed by multiple streams (Jeannerod 1999) controlling different actuators, in particular the arm (e.g., for reaching), the hand (e.g., for grasping), and the eye. A third *ventral neural pathway* goes from visual cortex to temporal visual areas (encoding information on relevant aspects of world, such as the identity of objects), and then to prefrontal cortex (integrating various sources of information to form the agent's goals) and, via supplementary cortex, again to premotor/motor cortex (Fuster 2001; Miller and Cohen 2001; Thill et al. 2013).

With respect to behaviour, this literature mainly focuses on the expression of overt/motor behaviour (within the sensorimotor and dorsal pathways) and on the higher-level control exerted on it by the brain executive functions (ventral

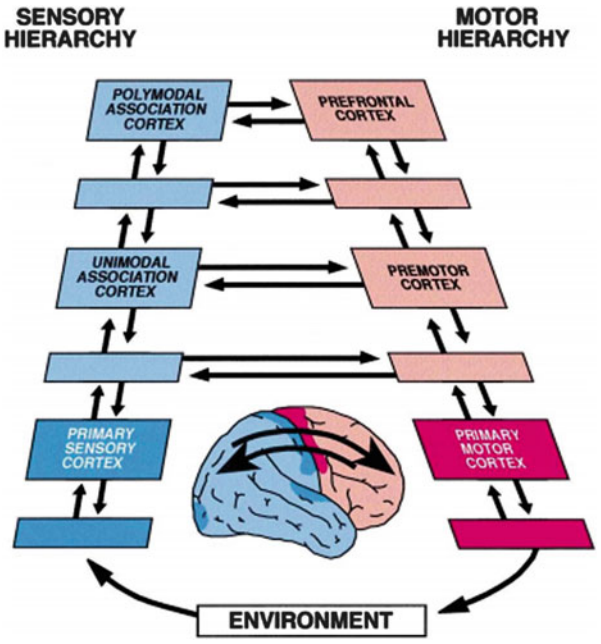

Fig. 1 Sketch of the hierarchical organisation of cortex. Reprinted from Fuster (2001) (Copyright 2001, with permission from Elsevier)

pathway). The specific functions implemented in each pathway can be described by referring to the research addressing the different transformations. So, the direct sensorimotor pathway supports the encoding and expression of dynamic sensorimotor transformations based on a closed loop between the somatosensory and motor cortex (Graziano 2011; Todorov and Jordan 2002). In parallel, the dorsal pathway integrates the sensory patterns *collected from the external world* with the information on the musculoskeletal system to form *affordances* within the parietal cortex (e.g., "the seen object can be grasped with a precision grip") and then contributes to form and control motor plans at the level of premotor cortex (Cisek and Kalaska 2010; Rizzolatti and Craighero 2004). At the highest level, the ventral pathway processes information on the identity/nature of the objects in the environment (adaptively, these are relevant as they represent resources potentially useful for the animal) based on temporal areas. This information reaches the prefrontal cortex that suitably integrates such information with the one on the agent's goals and on this basis contributes to the selection of the possible plans of actions prepared within the dorsal pathway (Fuster 2001; Koechlin and Summerfield 2007; Miller and Cohen 2001; Wallis et al. 2001).

The bio-behavioural literature, instead, tends to focus on the sub-cortical hierarchy of the brain. This involves the basal-ganglia system and its rostro-caudal organisation based on multiple cortical-striatal macro-loops (Fig. 2; the striatum

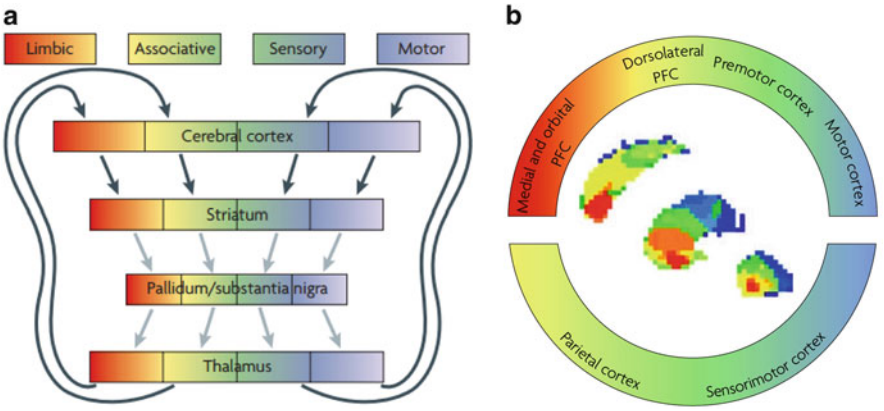

Fig. 2 The parallel basal ganglia-cortical loops forming the subcortical hierarchy. Reprinted by permission from Macmillan Publishers Ltd: *Nature Review Neuroscience*, Redgrave et al. (2010), copyright 2010

is the input stage of basal ganglia) encompassing (Redgrave et al. 2010; Yin and Knowlton 2006): the *limbic loop* (involving the ventral striatum, mainly formed by the accumbens), the *associative loop* (involving the striatum portion called dorsomedial striatum in rats and caudatum in primates), and the *sensorimotor loop* (involving the striatum portion called dorsolateral striatum in rats and putamen in primates). At the highest level, the hierarchy also involves the amygdala complex (Mirolli et al. 2010; Pitkänen et al. 1997) and the hippocampal system (Bast 2007; Lisman and Grace 2005; Voorn et al. 2004) as generators of motivations and as loci of associations between objects/experiences and their subjective "value" (biological relevance, novelty, etc.). Finally, they involve the dopaminergic systems (substantia nigra pars compacta and ventral tegmental area) controlling dopamine, the most important neuromodulator for the guidance of learning and for the energization of behaviour (Berridge and Robinson 1998).

With respect to behaviour, this literature has a strong focus not only on its expression but also on the learning processes that lead to its acquisition, in particular within the context of the numerous classical and instrumental learning paradigms (Cardinal et al. 2002), and on the role that *value and motivations* play in these processes (e.g., supported by dopamine). Overall, the hierarchy contributes to generate behaviour based on the following principles. At the highest levels, some subcortical structures (e.g., the amygdala) interface the brain with the *body homeostatic regulations*, and on this basis allow the assignment of value to environmental stimuli and experiences. This allows the highest levels of the cortico-basal ganglia systems communicating with such structures, the limbic cortico-striatal loop, to suitably direct behaviour based on the activation of specific high-level goals (goal-directed behaviour) (Balleine and Dickinson 1998). The limbic cortico-striatal loop is also an important regulator of dopamine (via its connections to dopaminergic areas), which in turn guides the learning processes leading to the acquisition of behaviour

Table 1 Summary of limitations of the different literatures studying the hierarchical organisation of brain at the cortical and sub-cortical level. "V" and "X," respectively, indicate that the literature, respectively, accounts or not for the classes of issues indicated at the top of the column ("Motivation," "Selection," "Sensorimotor mapping"). "v" indicates a minor account of them

	Literature focussing on cortical hierarchy			Literature focussing on sub-cortical hierarchy		
	Moti-vation	Selection	Sensorimotor mapping	Moti-vation	Selection	Sensorimotor mapping
Empirical research	v	X	V	V	V	X
Computational modelling	X	v	V	v	V	X

(Grace et al. 2007). The goals selected within the limbic loop then influence the selections involving attention, affordances, and sensory processing taking place within the associative loop, on the basis of a number of "cross-loop" mechanisms (e.g., thalamo-cortical connections and "dopamine spirals" (Haber 2003; Haber et al. 2000). Based on similar mechanisms, the associative loop in turn influences the selections of the lower-level processes taking place within the sensorimotor loop.

In part forcing the distinction for the sake of clarity, it appears that the two research threads tend to study the respective brain structures and processes quite in isolation from each other. Indeed, they often present an account of the functioning of hierarchical brain almost as if the studied cortical or sub-cortical systems taken alone were not only necessary but also sufficient for the acquisition and expression of the investigated behaviour. This is problematic as the cortical and sub-cortical brain components play not only partially overlapping but also distinct functions. In this respect, such accounts have important limitations with respect to the three classes of processes supporting hierarchical behaviour illustrated at the beginning of the chapter, namely the implementation of sensorimotor transformations, their selection, and the guidance of the learning processes leading to the acquisition of such transformations and selection capabilities.

These limitations, summarised in Table 1, are now illustrated. In this description we distinguish between the empirical and the computational literatures supporting each view as the two do not always coincide. The empirical literature focussing on the cortical hierarchy (e.g., cognitive neuroscience) investigates to a large extent the sensorimotor-mappings of the dorsal pathways affording and implementing actions, and their control by the ventral-pathway/prefrontal cortex, but rarely tackles the issue of the specific mechanisms that the brain used to select alternative affordances and actions (e.g., Munakata et al. 2011). Moreover, when it investigates the phenomena related to the ventral/prefrontal control of behaviour, it faces the problem of how higher-level goals bias top-down selections (e.g., Miller and Cohen 2001) but it usually neglects the motivational aspects leading to the ultimate formation and selection of goals, and to the guidance of the learning mechanisms

leading to the acquisition of the selection capabilities at all levels (perception, action, goals, etc.).

The computational literature focussed on the cortical hierarchy develops models giving detailed accounts of the dorsal/ventral cortical pathways (e.g., Fagg and Arbib 1998; Oztop and Arbib 2002; see also the example in Sect. 3) but generally neglects the motivational processes at the origin of the overall guidance of behaviour selection and learning processes. On the other side, those same models often incorporate selection mechanisms based on lateral competition (e.g., Cisek 2007; Erlhagen and Schoner 2002), probably because computational implementations make evident the need of having some type of selection mechanism.

The empirical literature (e.g., bio-behavioural neuroscience) focussing on the sub-cortical hierarchy gives a prominent importance to the study of the selection mechanisms, of the learning mechanisms, and of the motivational processes driving them (Cardinal et al. 2002, for a review). However, such literature tends to give accounts of hierarchical behaviour assuming the existence of already acquired whole actions (e.g., "pressing a lever") readily available to be selected or associated with stimuli (e.g., based on S-R associations). This is a limitation as, although several Pavlovian actions are indeed innately encoded in brain, the same is not true for most instrumental actions: in this case the sensorimotor mappings implementing the actions are progressively acquired with learning processes.

The computational literature on the sub-cortical hierarchy relies upon one of the most successful examples of synergies between empirical and computational research, namely the one based on reinforcement learning models (Barto et al. 1983; Houk et al. 1995; Joel et al. 2002; Sutton and Barto 1998). These models are focussed on the learning processes leading to acquire the capacity to select actions by trial-and-error (Table 1) and have been also developed to capture the hierarchical organisation of the brain (Botvinick et al. 2008; Daw et al. 2005; Solway and Botvinick 2012). The models have also been developed to some extent to capture the ultimate motivational sources of learning signals (primary rewards) and behaviour drives (e.g., Barto et al. 2004; Mirolli et al. 2013; Venditti et al. 2009). As for the related empirical literature, however, when used to account for empirical phenomena these models tend to represent "primitive actions" at a rather high level (e.g., "moving from one place to another") that abstracts from the sensorimotor mappings needed to implement them (Table 1). Other times, when they do not assume high-level primitive actions but work on the basis of fine movements, these computational models tend to give a view of striato-cortical loops as learning by trial-and-error to implement fine sensorimotor/cognitive transformations that map sensations to actions. As we shall see in Sect. 5, this view is in contrast to the overall anatomical organisation of striato-cortical loops where cross-loop flows of information tend to be top-down, from goals to actions, rather than bottom-up, from sensations to actions, and to the evidence that fine sensorimotor/cognitive transformations are implemented in cortico-cortical pathways rather than in striato-cortical loops.

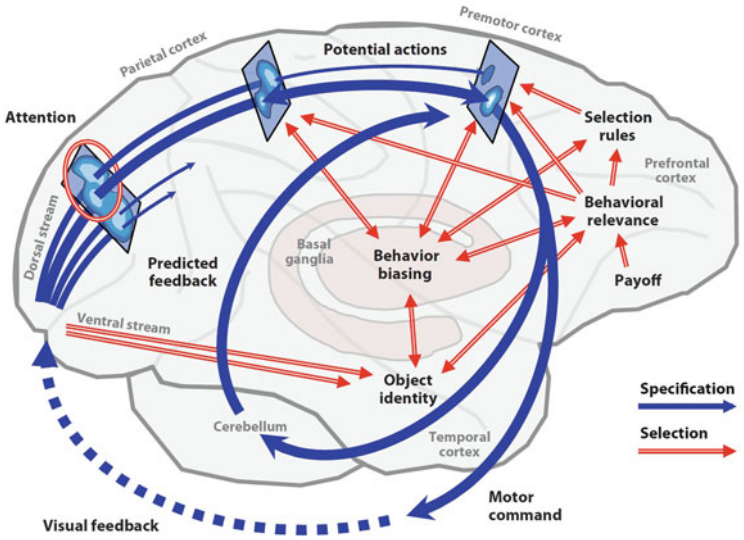

Fig. 3 Illustration of the brain cortical hierarchy addressed by the TRoPICALS model. The processes of action specification (represented by *dark blue arrows*) begin in the visual cortex and proceed rightward across the parietal lobe: these processes transform visual information into representations of potential actions (affordances). Along the dorsal route, sensorimotor transformations leading to produce different actions compete for further processing. This competition is biased by the input from prefrontal cortical regions that collects information for action selection (*red double-line arrows*). The final selected action is released into execution and causes overt feedback through the environment (*dotted blue arrow*). Note that TRoPICALS, as most of the literature it refers to, does not take into consideration the basal ganglia, the cerebellum, and the sources of the "payoff" illustrated in the figure. Reprinted from Cisek and Kalaska (2010) (copyright 2010, permission from Annual Reviews)

3 Cortical Hierarchies

This section will present a more detailed view of the organisation of the cortical hierarchy and its functioning and learning mechanisms by reviewing in depth a specific computational model by the authors (Caligiore et al. 2010, 2012). This will allow us to exemplify more in detail the nature, and also the limitations, of the accounts of the brain hierarchy given by the views focussed on cortex.

The model departs from two principles involving two cortical pathways considered in the previous section (see Fig. 3): (a) the dual route hypothesis (Milner and Goodale 2008; Ungerleider and Mishkin 1982), according to which the cortical brain responsible for visual processing is organized into the dorsal and ventral neural pathways; (b) the role of the prefrontal cortex as a source of top-down biasing that instructs the neural competitions between potential alternative actions at the level of the premotor cortex (Cisek 2007; Cisek and Kalaska 2010; Miller and Cohen 2001). The two principles are now illustrated in detail.

In the original dual route proposal (Ungerleider and Mishkin 1982), the ventral stream runs from early visual cortex areas to inferotemporal cortex and carries information about the identity of the objects ("what" pathway). Instead, the dorsal stream runs from visual cortex areas to the parietal cortex and processes spatial information concerning the location of objects in the visual field ("where" pathway). The scope of this theory was later extended (Milner and Goodale 2008) by proposing that the ventral stream communicates visual information to support higher cognitive processing taking place in prefrontal cortical regions (e.g., not only object recognition but also decision making on actions to be executed and higher-level reasoning). In contrast, the dorsal stream transfers visual information to support on-line performance of actions in downstream motor cortex areas (e.g., not only location of objects but also implementation of the sensorimotor transformations needed to detect affordances and visually guide action).

The sensory system of primates and humans provides detailed information about the external world and on this basis the motor system can perform a large repertoire of actions. This introduces a great potential not only for flexibility but also for interference. To effectively cope with the multitude of possible actions to perform, the brain has acquired mechanisms that coordinate low-level sensory and motor processes on the basis of goals, external context, and internal motivations (Fuster 2001). The prefrontal cortex plays a key role in these processes especially when "top-down" control based on goals (and motivations) is needed (Fuster 2001; Miller and Cohen 2001; Wallis et al. 2001). More in detail, within the ventral pathway the prefrontal cortex can use information from the outer context and the agent's needs to form high-level goals. Based on this information, the prefrontal cortex can act on the dorsal pathway by biasing the selection of affordances and actions. This biasing activity is based on various features of the prefrontal cortex, including its capacity to integrate multiple sources of information, to implement working memory, and to form complex behavioural "rules" (Deco and Rolls 2003).

The computational model TRoPICALS (Fig. 4, Caligiore et al. 2010, 2012) proposed to account for compatibility effects studied in cognitive psychology (Ellis and Tucker 2001),[1] integrates the key features of the cortical hierarchical organization discussed above. More in detail, TRoPICALS incorporates in its architecture the *dorsal/ventral pathways organisation* of cortical areas (Milner and Goodale 2008); the *guidance/biasing* of action selection based on prefrontal cortex "instructions" (Miller and Cohen 2001); and the selection of actions within premotor cortex based on a *competition* between affordances and alternatives actions under a bias from the prefrontal cortex (Cisek 2007; Cisek and Kalaska 2010). The acronym

[1] In a typical compatibility effect experiment participants are asked to produce actions which are either in agreement (compatible) with the actions typically associated with the objects (e.g., a precision grip with a small object) or in contrast (incompatible) to those actions (e.g., a precision grip with a large object) in tasks where the objects size is irrelevant. If the participants exhibit longer reaction times and higher error rates in incompatible trials than in compatible ones, one can infer that seeing objects automatically elicits the representations of their affordances, independently of the performance of the experimental task.

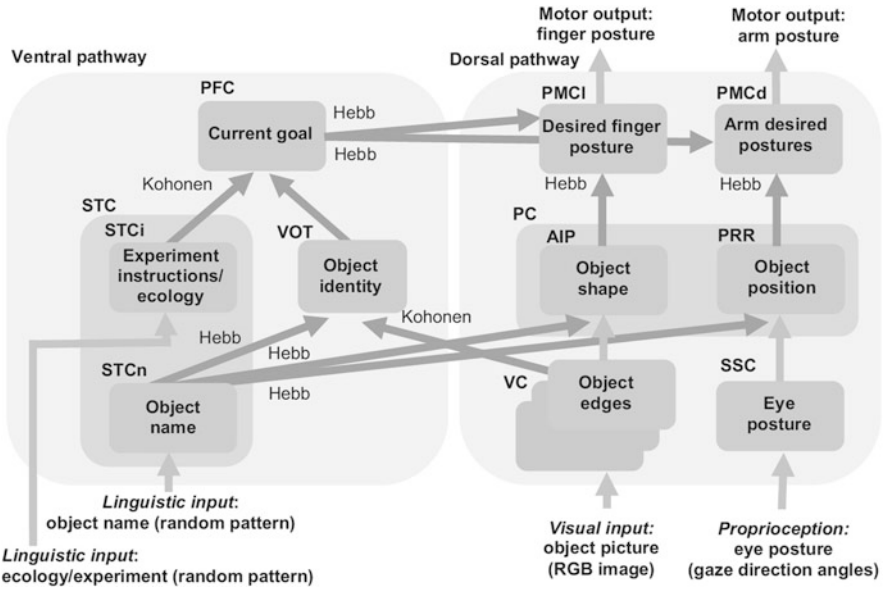

Fig. 4 Architecture of the TRoPICALS model. The *boxes* indicate the components of the model. The *label inside each box* indicates the type of information encoded by the component, whereas the acronym at its *top-left corner* indicates the brain anatomical area putatively corresponding to it. *Light and dark grey arrows* indicate respectively connections which were hardwired and connections which were updated by learning processes based on a Hebb learning rule or a Kohonen learning rule. Acronyms: AIP: the anterior intraparietal sulcus; PFC: the prefrontal cortex; PMCd: premotor cortex dorsal division; PMCl: premotor cortex lateral division; PRR: parietal reach region; SSC: somatosensory cortex; STC: superior temporal cortex; VC: visual cortex; VOT: ventral occipito-temporal cortex. Reprinted with permission from Caligiore et al. (2010) (copyright 2010, APA publisher)

"TRoPICALS" summarises these principles: Two Route, Prefrontal Instruction, Competition of Affordances, Language Simulation (the latter principle, less relevant here, was introduced to account for compatibility effects involving language; cf. Barsalou 2008).

TRoPICALS reproduces the main functions of several dorsal and ventral cortical areas (see Fig. 4 for the acronyms). The model was tested within an embodied system formed by a simulated eye (camera) and a simulated robotic arm/hand (see Fig. 5). The input of the model is formed by three neural maps (VC), encoding an RGB visual input, and a somatosensory map (SSC), encoding the arm angles. The output of the model is the desired posture of the hand encoding different grips (PMCl) or the desired posture of the arm encoding a reaching target (PMCd). Downstream VC and SSC, the model divides into two main neural pathways: the dorsal pathway, which implements suitable sensorimotor transformations needed to perform actions on the basis of perception, and the ventral pathway, which allows flexible control of behaviour, thanks to the biasing effects exerted by PFC on action

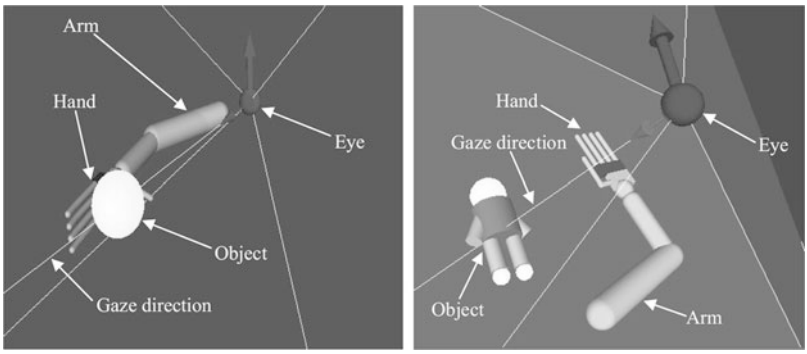

Fig. 5 The robotic set-up used to test TRoPICALS. *Left*: The simulated arm, hand, and eye interacting with a simulated apple. *Right*: The simulated arm, hand, and eye interacting with a simulated doll. In both panels, the line passing through the object indicates the gaze direction (eye control is hardwired), whereas the other four lines indicate the scope of the eye visual field. Reprinted with permission from Caligiore et al. (2010) (copyright 2010, APA publisher)

selection. In turn, the dorsal pathway is formed by a stream controlling grasping and a stream controlling reaching.

With respect to function and learning, the VC performs image edge extraction based on Sobel filters (Sobel and Feldman 1968). Based on this, within the dorsal pathway the AIP extracts the shape of objects, and within the ventral pathway the VOT categorises objects using a self-organising map (SOM; Kohonen (2003)) (see Fig. 6).

Within the dorsal pathway, the AIP-PMCl and PRR-PMCd streams (i.e., the two dorsal neural streams transforming affordances into grasping and reaching actions) are trained on the basis of a Hebbian learning process (Dayan and Abbott 2001; Hebb 1949) that allows the system to learn to associate suitable actions (PMCl, PMCd) to available affordances (AIP, PRR). Importantly for this chapter, the Hebbian process is based on a motor babbling of the hand/arm, and connections are formed only when the hand/accomplish a successful grasp/reach: this means that the *system ultimately uses a trial-and-error mechanism to learn the association that allows it to select the proper actions*.

Within the ventral pathway, PFC uses a second SOM to form representations that combine the seen objects (VOT) and the task to be accomplished (STC) to shape the current high-level goals used to bias action selected within PMCl or PMCd (Fig. 7). The premotor regions (PMCl and PMCd) integrate affordance information from the parietal cortex (PC) (respectively, from AIP and PRR regions) and goal-based information from PFC using a dynamic neural field (Erlhagen and Schoner 2002). The *dynamic field is then used to select actions through neural competition* taking place within premotor areas. The dynamic nature of this competition allows to account for compatibility effects: when the action suggested by affordances are congruent with the PFC command, reaction times for triggering the action are faster than when they are not congruent. Figure 8 illustrates the effect of this neural

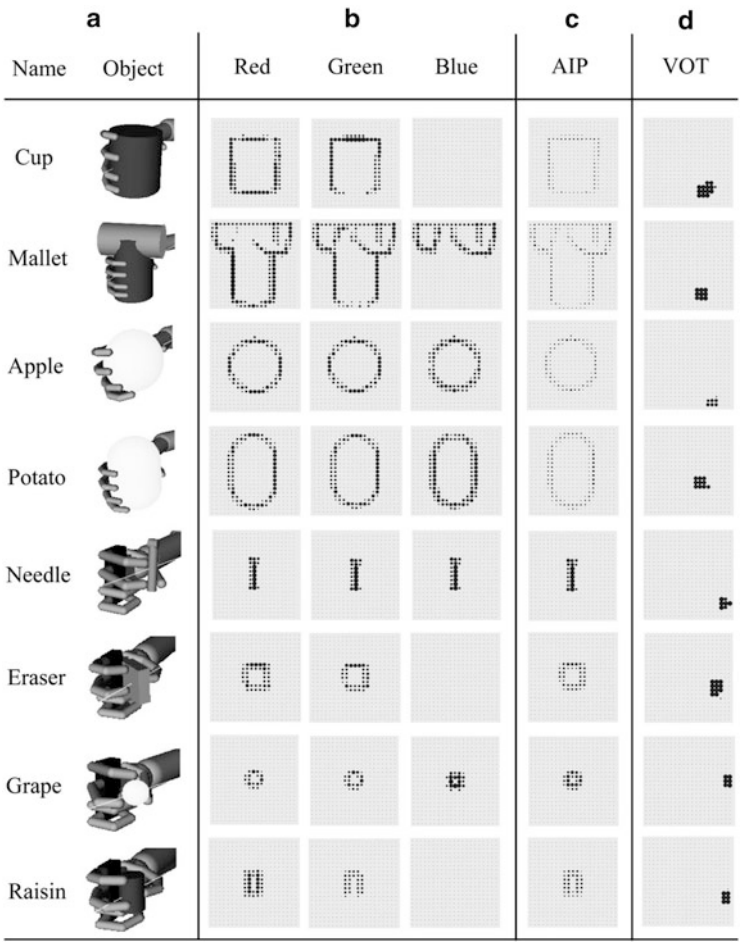

Fig. 6 Activation of the early cortical areas of TRoPICALS. The columns of the figure show: (**a**) the object name, appearance, and handgrip of the objects used in the compatibility effect experiment; (**b**) the activation of the neurons forming the visual cortex (three edge sensitive red-green-blue maps) caused by the objects; (**c**) dorsal pathway: the activation of AIP encoding the shape of objects; (**d**) ventral pathway: the activation of the VOT encoding the identity of objects. AIP: anterior intraparietal cortex; VOT: ventral occipitotemporal cortex. Reprinted with permission from Caligiore et al. (2010) (copyright 2010, APA publisher)

competition on the speed of the selection of actions (actions are encoded using population codes (Pouget et al. 2000) as desired postures of hand, within PMCl, and arm, within PMCd).

The presentation of the model highlighted the typical principles and topics characterising the literature focused on cortical hierarchy and allows us to highlight the two limitations of the approach presented at a theoretical level in Sect. 2. TRoPICALS uses dynamic neural fields (Erlhagen and Schoner 2002) to abstract

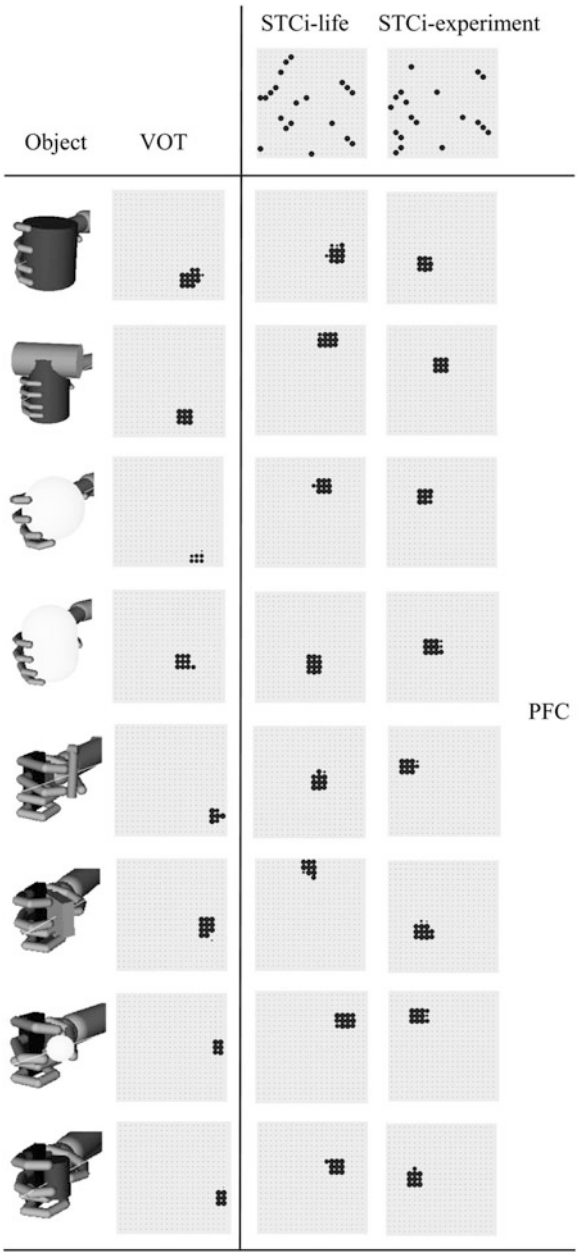

Fig. 7 The activation of the PFC caused by the different representations in VOT (objects) and the two STCi activations representing, respectively, the ecological condition experienced during life, when affordances and motor control capabilities are acquired, and the condition experienced during the psychological experiment. Notice the different representations of the various contexts and objects within the PFC: the richness of such representations is at the basis of the potential of the PFC to flexibly associate any combination of the context elements with any action. PFC: prefrontal cortex; STCi: superior temporal cortex for instructions; VOT: ventral occipitotemporal cortex. Reprinted with permission from Caligiore et al. (2010) (copyright 2010, APA publisher)

affordances and actions selection mechanisms that in brain are mainly implemented by cortical basal ganglia loops (e.g., involving parietal and premotor areas; Alexander and Crutcher 1990; Redgrave et al. 1999; Yin and Knowlton 2006). In a recent extended version of TRoPICALS (Caligiore et al. 2012) developed to account for

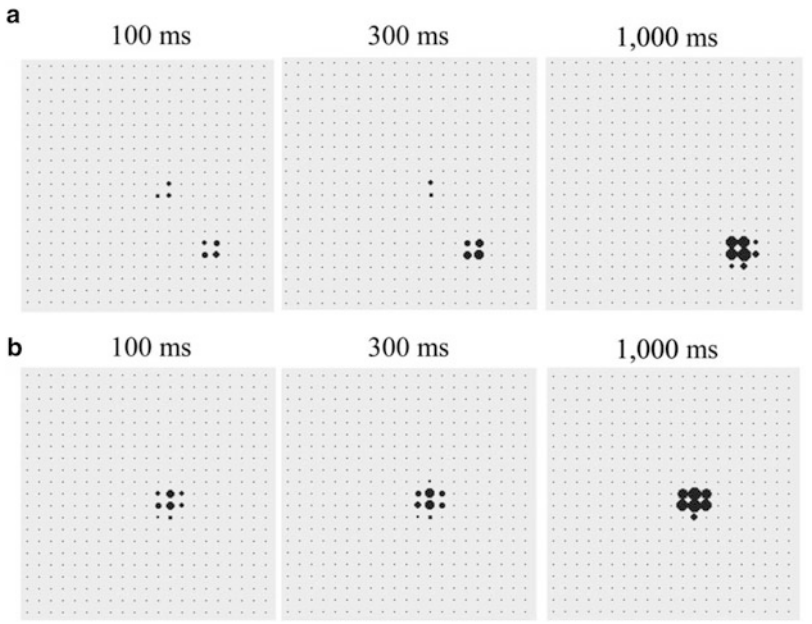

Fig. 8 The activation of the PMCl during the simulation of the compatibility effect experiments. (**a**) Activation of the PMCl in an incongruent trial: the biases from PFC (goals) and PC (affordances) cause two different clusters of neurons, encoding two different grasping actions, to compete until the cluster caused by the PFC suppresses the cluster caused by the PC. (**b**) Activation of the PMCl in a congruent trial: the biases from the PFC and the PC overlap and cause the formation of only one cluster of neurons. The panels depict the activation of the PMC after 100, 300, and 1,000 ms. Notice how in the incongruent condition the stronger top-down bias from the PFC wins. Also notice how in the congruent condition the action cluster forms more rapidly than in the incongruent one so producing faster reaction times (compatibility effect). Reprinted with permission from Caligiore et al. (2010) (copyright 2010, APA publisher)

compatibility effects in the presence of a distractor (Ellis et al. 2007), the biasing effect of the PFC was augmented by adding inhibitory mechanisms which, as explicitly recognised in the paper, abstract the inhibitory effects that PFC can exert on motor cortex via the basal ganglia and the supplementary motor cortex (Nachev et al. 2008). Moreover, the Hebbian-based reinforcement learning mechanism used by TRoPICALS to acquire sensorimotor mappings abstracts the role of the sub-cortical mechanisms producing the reward signals that guide the acquisition of such selection capabilities, and the mechanisms within the basal ganglia implementing such learning processes (Joel et al. 2002; Tang et al. 2007).

These assumptions are viable if one studies phenomena such as compatibility effects, but we think that they give a limited/distorted image if one studies the overall hierarchical organisation of the brain, as they cannot account for a number of interesting phenomena and mechanisms whose study requires an explicit representation of selection and learning guidance processes.

4 Basal-Ganglia Hierarchies

This section reviews another model that focusses on the sub-cortical hierarchy of the brain and that will allow us to highlight the typical features of research focussed on such hierarchy. The review also highlights that this account has the opposite limitations with respect to the approach reviewed in the previous section: it fails to account for sensorimotor/cognitive transformations. The section first introduces the brain features captured by the model and then explains and discusses the model itself.

The basal ganglia are a group of sub-cortical nuclei comprehending the striatum and the sub-thalamic nucleus (STN) as its main input gates, and the internal globus pallidus (GPi) and the substantia nigra pars reticulata (SNpr) as its main output components (Fig. 9). These structures represent fundamental functional processing unit of the vertebrate brain that repeats at multiple functional levels and forms multiple re-entrant loops with various frontal and associative cortical areas. Different loops run in parallel and each loop starts from a cortical area, goes through a subregion of the basal ganglia, and goes back to the cortical area of origin via the thalamus (Alexander et al. 1986; Heimer et al. 1982; Humphries and Prescott 2010; Middleton and Strick 2000; Romanelli et al. 2005). Each loop is involved in the *selection* of the content of the targeted cortical areas, such as a perceptual representation, an action, or a goal (Alexander and Crutcher 1990; Redgrave et al. 1999). This selection takes place through a mechanisms that allows basal ganglia to dishinibit the area of the thalamus, in turn in loop with the cortex, corresponding to the cortical content to be selected (Chevalier and Deniau 1990; Gurney et al. 2001; Mink 1996).

There is now a wide agreement on the fact that three distinct functional domains can be distinguished within the basal ganglia corresponding to the dorsolateral striatum (DLS), dorsomedial striatum (DMS), and ventral striatum (VS), the latter mainly formed by the nucleus accumbens (Fig. 10; Yin and Knowlton 2006). Such domains are identifiable in rats and mice and are homologue to, respectively, the putamen, caudatum, and nucleus accumbens in primate striatum. These domains form distinct loops interacting with distinct portions of cortex. These distinct loops typically play different functional roles depending on the type of information processed within the targeted cortex, and hence are also called limbic, associative, and sensorimotor loops, respectively. The functions of the loops are now explain more in detail.

The cortical areas that reciprocate connections with the VS are various sub-regions of the prefrontal cortex (PFC), in particular the ventro-medial, orbitofrontal, and dorsolateral portions, important for the processing of biologically salient states and outcomes (Humphries and Prescott 2010; Voorn et al. 2004; Zahm 2000). In general, the limbic loop is involved in the selection of final goals (e.g., the achievement of a certain food), and means-to-end goals (e.g., opening a door to access a lever activating a food dispenser), based on motivations. These are important mechanisms underlying goal-directed behaviour (Cardinal et al. 2002; Corbit et al. 2001; Yin et al. 2008). The limbic loop is also important for reward and

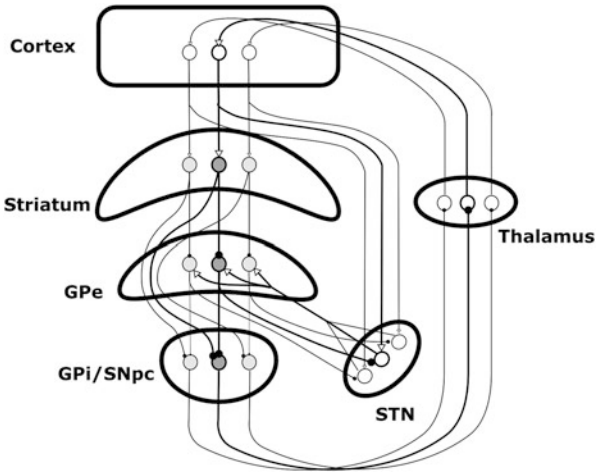

Fig. 9 A schema of the micro architecture of a single cortical basal ganglia loop. *White arrowheads* indicate glutamatergic projections whereas *black arrowheads* indicate GABAergic projections. GPe: globus pallidus, external; GPi: globus pallidus, internal; SNpc: substantia nigra, pars compacta; STN: subthalamic nucleus

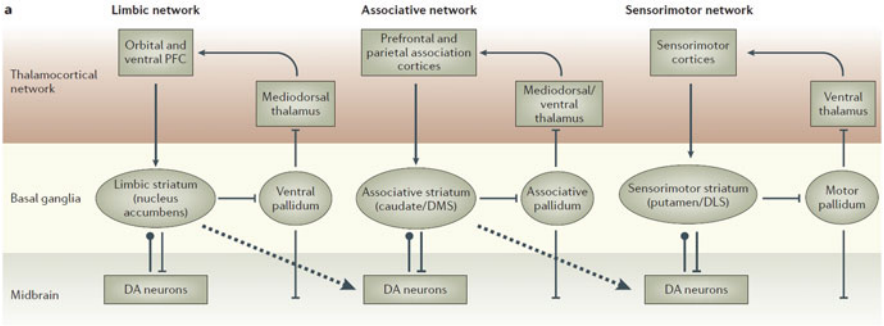

Fig. 10 A scheme of the three main cortico-striatal regions and their interconnections. *Standard arrows* indicate excitatory glutamate connections. *Flat arrowheads* indicate inhibitory GABA connections. *Dot arrowheads* indicate dopaminergic connections (*dashed arrows* indicate the cross-loop ones). Reprinted with permission from Macmillan Publishers Ltd: *Nature Reviews Neuroscience*, Yin and Knowlton (2006), copyright 2006

motivation based on dopamine regulation (Berridge and Robinson 1998; Corbit and Balleine 2011). For example, it plays an important role in the interaction between instrumental and Pavlovian processes, e.g. it allows cues previously associated with reward to energise the performance of instrumental behaviours (Corbit and Balleine 2011; Corbit et al. 2001; Hall et al. 2001).

The cortical areas that reciprocate connections with the DMS are the temporal cortex (TE; Middleton and Strick 1996), the parietal cortex (PC), the frontal eye-fields (FEF), and the dorsal regions of the PFC (Alexander et al. 1986; Voorn

et al. 2004; Yeterian and Pandya 1995). The associative loop is implicated in several high-level cognitive processes (Kimchi and Laubach 2009), in particular it is involved in the formation of high-level visual representations (typically processed in TE; Middleton and Strick 1996), in attention, spatial orientation, and affordance selection (involving FEF and PC, Schrimsher et al. 2002; Volkow et al. 2007), and in working memory tasks (involving various areas of PFC; Levy et al. 1997; Lewis et al. 2004).

Finally, the DLS is in loop with motor cortex (MC), premotor cortex (PMC), and supplementary motor cortex (SMC) (Romanelli et al. 2005; Tang et al. 2007). There is clear evidence that the sensorimotor loop is associated with the control of movement, in particular in the selection of final sensorimotor repertoires based on the current context (Alexander et al. 1986; Haber et al. 2000; Romanelli et al. 2005; Yin and Knowlton 2006).

Within each of the three main loops, several discrete parallel streams run through relatively parallel pathways. For instance, the sensorimotor loop contains a somatotopic motor map that repeats at the level of striatum, globus pallidus, thalamus, and cortex, in particular in these regions separate areas can be found encoding information about arms, legs, and face (Alexander and Crutcher 1990; Romanelli et al. 2005). Moreover, within each one of these streams there is evidence for the existence of *relatively segregated channels* capable of selecting particular cortical restricted targets, for example encoding specific actions (Chevalier and Deniau 1990; Gurney et al. 2001; Mink 1996). This idea is also a key assumption of most models on basal ganglia (e.g., see Joel et al. 2002 for a review). Even if no direct evidence can be given, many researchers assume that the same structure made of separate channels is present also in the associative and limbic loops given the uniformity of the striato-cortical micro-structure over the entire basal ganglia.

The three striato-cortical loops form a functional hierarchy. The decisions about motor actions, supported by the sensorimotor loop, are at a lower level with respect to the decisions about the part of the current context the animal should attend and process, which relies on the associative loop. On their turn, the latter decisions are at a lower level with respect to decisions about the motivationally salient outcomes (the high level goals of the animal) processed by the limbic loop.

Importantly, this functional hierarchy seems to be in line with neural data indicating that the cortico-striatal loops are anatomically organised in a hierarchical manner from ventral to dorsal domains. In this respect, there is evidence that cortices in different cortico-striatal loops are interconnected not only through direct projections (see previous section) but also through the thalamus so to form a cortico-thalamo-cortical pathway from higher more abstract levels to lower sensorimotor levels of the cortico-striatal hierarchy (Haber 2003).

Even more important for the top-down diffusion of "value" (i.e., biological relevance of stimuli) along the hierarchy, Haber (Haber 2003; Haber et al. 2000) discusses anatomical and functional evidence ascribing the control expressed by the ventral cortico-striatal loops to a dopaminergic modulation triggered via the so-called *striato-nigro-striatal spiral pathway*. This pathway involves dopaminergic populations in the ventral tegmental area (VTA) and those in the substantia nigra

pars compacta (SNpc) forming loops that reciprocate various striatal regions with a pattern *moving from the ventral to the dorsomedial and dorsolateral regions*. In particular, projections from ventral compartments of the striatum reach dopaminergic neurons (in particular within the VTA) that target ventral and medial striatal regions, and projections from medial regions contact dopaminergic neurons (in particular within the SNpc) that target medial and lateral striatal regions. Functionally, these projections diffuse the information on value of stimuli and goals from higher levels of cognition (limbic loop) to lower ones (associative and sensorimotor loops).

The model proposed by Mannella et al. (2010) (see also Mannella et al. 2011) captures the main processes illustrated above and specifies them in computational terms. In particular, the model reproduces two of the three cortico-striatal loops, the processing of value within amygdala, and some cortico-cortical connections (which plays the role of carrying sensory and motor information to and from basal ganglia, not of implementing sensorimotor transformations).

The model reproduces *instrumental devaluation effects* and also their absence demonstrated in multiple experiments where different areas of the sub-cortical systems reviewed above are lesioned. Figure 11 shows the simulated robotic rat used to reproduce the devaluation effects. A typical devaluation experiment is formed by three phases (Fig. 12; see Balleine and Dickinson 1998). In the first phase, a rat is first instrumentally trained to work on a manipulandum A (e.g., a lever) to obtain a reward A (e.g., food pellets), and on a manipulandum B (e.g., to pull a chain) to obtain a different reward B (e.g., a sucrose solution). In the second phase, the rat is satiated for one food (e.g., by giving free access to the pellets). In the third phase, the rat is set in front of *both* manipulanda for the first time, and the number of actions performed on them is recorded in extinction (i.e., without reward delivery to avoid re-learning processes). The results of the experiment are that the rat acts more often on the manipulandum that corresponds to the food for which it has not been satiated. This experiment is considered a paradigmatic demonstration that the rat behaviour in the third phase is *goal-directed* as it selects the action that leads to obtain the valued outcome (goal) without the need of re-learning. Indeed, the action is selected on the basis of the value of the two goals (two foods) and not on the basis of the stimuli triggering actions (e.g., the sight of the lever and the chain) as in the case of habitual action.

The architecture of the model is shown in Fig. 13 (Mannella et al. 2010). The model is entirely formed by leaky neurons and uses localistic representations to make its interpretation fully transparent. The model is based on two cortico-basal ganglia loops, namely the limbic and the sensorimotor loops (the associative loop was not represented for simplicity). The selection processes performed in the basal-ganglia cortical loops are represented with a neural competition taking place within premotor cortex. For simplicity, the model represents only the cortical area in loop with dorsolateral striatum (PMC in this case), whereas it abstracts away the NAcc-PFC and the PFC-PMC connections. The model also captures the reinforcement learning processes, guided by phasic dopamine (i.e., dopamine produced in strong, short-lasting bursts), that allow the two loops to acquire their selection capabilities. Finally, and importantly, the VS (nucleus accumbens) communicates with the

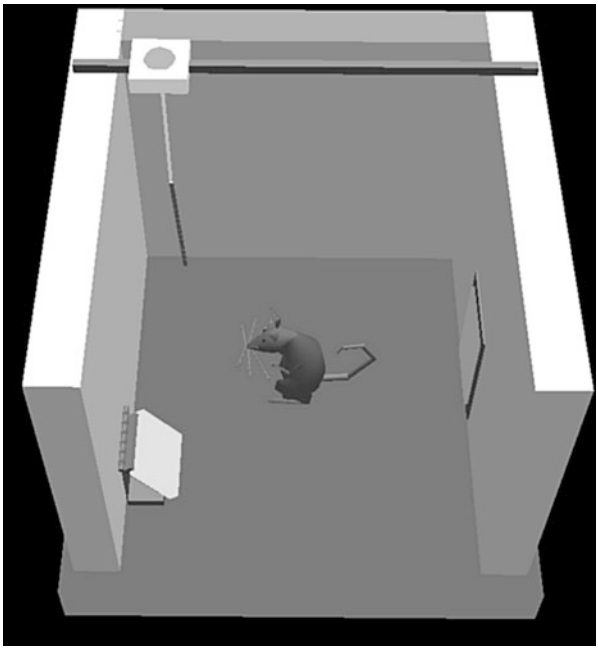

Fig. 11 A snapshot of the simulator used in the study of the devaluation experiment. The simulated rat is at the centre of the experimental chamber, the food dispenser is behind the rat, the lever is at the rat's left-hand side, and the chain is at the rat's right-hand side. Reprinted from Mannella et al. (2010) (copyright 2010, with permission from Cambridge University Press)

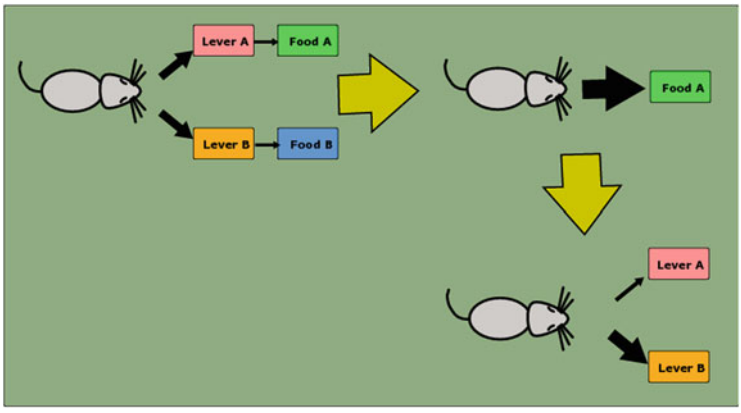

Fig. 12 A simplified schema of the instrumental devaluation paradigm. On the *top-left* the training phase: the animal learns to use two levers to obtain two different rewards. On the *top-right* the devaluation phase: the animal is satiated of one of the two rewards. On the *bottom-right* the test phase: the animal is presented with both manipulanda and actions toward each of them are measured in absence of any reward presentation

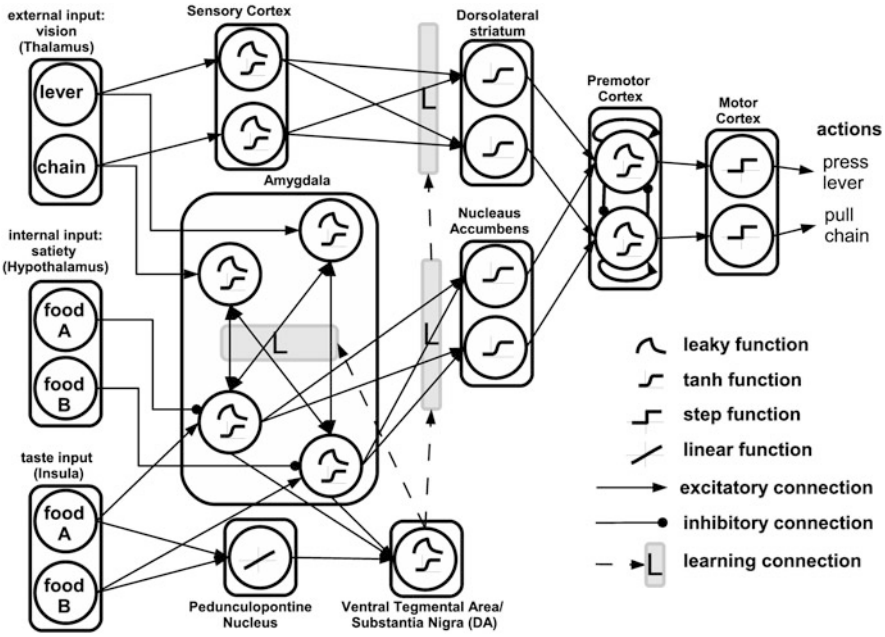

Fig. 13 The architecture of the model used to investigate the devaluation effect. Reprinted from Mannella et al. (2010) (copyright 2010, with permission from Cambridge University Press)

amygdala (Amg) that informs it on the value of stimuli (e.g., the sight of a particular lever is associated with the future appearance of a valuable food). Within Amg, the value of anticipated stimuli (food A, food B) can be regulated by the internal states of the system (e.g., in the model the satiation for one food inhibits the activation of its representation).

In the simulations, during the first phase of the experiment three learning processes take place: one leading to the formation of habits, one leading to the assignment of value to previously neutral stimuli, and one assigning value to outcomes. The first process leads the sensorimotor loop to learn habits based on instrumental (i.e., trial-and-error/reinforcement learning) processes that allow the formation of associations between stimuli (the sight of the lever or of the chain) and responses (lever pressing or chain pulling) on the basis of phasic dopamine (produce by the ventral tegmental area). For example, the sight of a lever is associated with the action of pressing it as this leads to receive food A that in turn causes the production of dopamine.

The second learning process, based on differential Hebbian learning rules, leads the Amg to acquire Pavlovian associations between stimuli (the conditioned stimuli corresponding to the sight of the lever or of the chain) and outcomes (the unconditioned stimuli corresponding to the two foods). For example, within Amg the representation of the sight of a lever gets associated with the representation of food A as these two stimuli are observed one after the other. Once formed,

these associations allow the rat to "assign a value" to previously neutral stimuli, for example to recall the representation of the biologically valuable food A when the lever is seen.

The third process leads the limbic loop to associate the representations of valuable stimuli within Amg to possible outcomes (goals) encoded within VS. For example, the representation of food A in Amg gets associated with the representation of the goal of getting food A in VS. This association takes place within the limbic loop, and the nucleus accumbens is the nexus that links goal representations to their *current* value (i.e., their counterparts in Amg). "Current" because the Amg is capable of changing the value assigned to different stimuli on the basis of the animal current internal states, e.g. if it is hungry or satiated.

Once these associations are formed, the system can exploit them to act adaptively. In particular, the goals encoded and selected within the limbic loop are associated with the actions encoded in the sensorimotor loop. These connections rely on cortico-cortical inter-loop pathways and also on sub-cortical pathways (dopaminergic spirals). In real animals, these associations form with learning but for simplicity in the model they are hand-coded. Once these links have been learned the sole activation of goal representations in VS by the Amg (e.g., because a lever related to a valuable goal is perceived) is sufficient to bias the selection of the action that leads to accomplish the selected goal.

Figure 14 shows how the model reproduces the devaluation effect. The figure reports the number of lever presses in rats with intact Amg (intact, or "sham", rats) and in rats where Amg has been lesioned ("BLA-lesioned" rats: the basolateral amygdala—BLA—is a part of Amg very important for devaluation). The figure shows that sham rats tend to press the lever more frequently when food B (previously instrumentally associated with the chain) is devalued than when food A (previously associated with the lever) is devalued. The causes of this behaviour are as follows. During the training phase, the rat acquires two habits within the sensorimotor loop, one that leads it to press the lever when it sees the lever, and one that leads it to pull the chain when it sees the chain. After the devaluation of one food, the rat is tested by setting it in front of both the lever and the chain at the same time. In this case both habits are triggered and try to recall the action related to them (pressing the lever and, at the same time, pulling the chain). If the Amg is intact, the sight of the lever and the chain recall the food A and food B representations within it. However, only one of these representations can actually manifest as the other is inhibited by the satiation internal state (say for food B). Such active representation (say for food A) can so activate the units of VS corresponding to the food A outcome and then, via the connections to cortex, to bias the selection of one action (e.g., pressing the lever). Instead, when Amg is lesioned then VS cannot preferentially select one outcome and so unbalance the selection for one or the other available actions. As a result the rat will select the two actions with a similar frequency.

This model highlights the typical features, and the limitations, of the research that focusses on the sub-cortical hierarchy of brain. First, the model emphasises the trial-and-error learning processes taking place within each cortico-striatal loop, and its fundamental role for the acquisition of their capacity to select the cortical

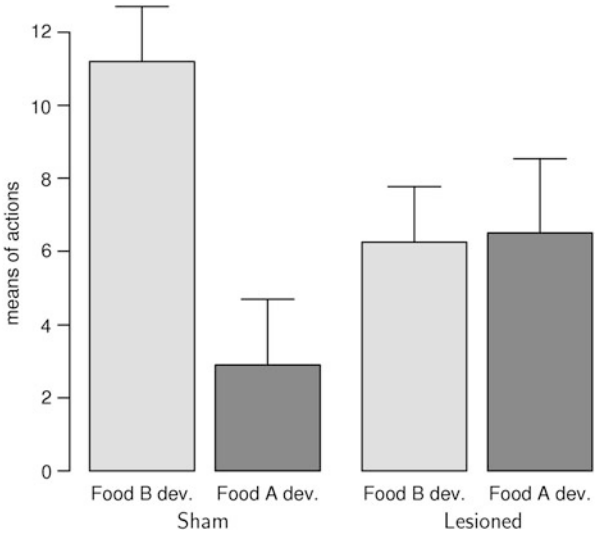

Fig. 14 Behaviour of the simulated rats in the devaluation experiment. The *histogram bars* show the average and standard deviation of actions performed by Sham and BLA-lesioned rats on the lever (previously instrumentally associated with food A) when either food A or food B has been devalued. Reprinted from Mannella et al. (2010) (copyright 2010, with permission from Cambridge University Press)

contents. Second, such learning processes are guided by the value systems, such as the amygdala and the dopaminergic system, that lead the animal to learn to select actions and goals that are valuable for survival and reproduction (e.g., to acquire food). Third, the model captures the hierarchy existing between the different cortico-striatal loops: importantly for this review, the hierarchy captures in particular the flow of control from the higher "limbic" levels, informed on the actual needs of the animal, to the lower sensorimotor levels (notice how this contrasts with reinforcement learning models that use trial-and-error processes to mainly implement sensorimotor transformations linking the sensations from the outer world to the actions to perform).

Aside these strengths, the model has also some important limitations with respect to the explanation of brain hierarchy. The most important ones are that actions (e.g. "pressing a lever" or "pulling a chain") are considered as ready-available wholes that the sensorimotor loop can select in correspondence to stimuli, affordances are abstracted away, and goals are assumed to be already formed and ready to be selected by the limbic loop. This means that the model does not account for the processes involving the sensorimotor/cognitive transformations happening at different levels of abstraction and accounted for by the literature focussing on the cortical hierarchy of brain, e.g. considered in the model reviewed in the previous section.

5 Integrating the Cortical and Sub-cortical Hierarchies

Based on the reviews and the models presented in the previous sections, we can now propose an integrated view of the cortical and sub-cortical hierarchies of brain. This is summarised in Fig. 15. We first present this view by focussing on the role of cortex and basal ganglia in the hierarchy, then we illustrate some detailed aspects of the different functioning mechanisms of the two, and finally we highlight important system-level open problems highlighted by the integrated view.

The general idea is as follows. Cortex implements sensorimotor/cognitive transformations needed to perform and control action at different levels of abstraction. The sensorimotor neural pathway (SSC-MC) implements the dynamic mapping closely linking the proprioception to the control of muscles. The dorsal neural pathway (PPC-PMC-MC) encodes affordances (PPC) and possible motor plans (PMC) to be executed downstream (MC). The ventral neural pathway (ITC-dlPFC) detects the resources available in the environment (ITC) and, based on this information and higher level information from areas encoding value (OFC, vmPFC), biases the selection of motor plans (SMC-PMC; note that there are also important connections from PFC to PC, not reported here for simplicity, that allow PFC to contribute to select affordances within PC). Overall, the various pathways perform different mappings from sensation to action taking place at increasing levels of abstraction: from proprioception to muscles (SSC–MC); from visual information needed to interact with objects to motor plans (VC–PPC–PMC–MC); from visual information on the nature of objects to high level goals (VC–ITC–dlPFC–SMC–PMC); from visceral states to biologically charged goals (Amg/Hip/Hyp–OFC/vmPFC–dlPFC).

Each loop of the basal ganglia collects a rich set of information from various areas of cortex and on this basis selects the contents processed in specific target cortical areas. These selection processes involve the whole cortex with the exception of primary cortical areas and have an increasing importance (e.g., in terms of neural resources involved) going towards the higher levels of the hierarchy. So, at the highest level of the hierarchy the VS, supplied with rich information on value of stimuli by various sub-cortical areas (e.g., Amg, Hip, Hyp), contributes to select biologically relevant goals encoded in OFC/vmPFC (e.g., in relation to the achievement of a particular food). At a lower level, the DMS contributes to selects more abstract goals (e.g., pressing a lever) encoded in dlPFC, affordances encoded in PPC, and object identity encoded in ITC. At the lowest level, the DLS selects motor plans, encoded in PMC, and action implementation processes, encoded in MC. The hierarchy formed by basal ganglia also involve “inter-loop” mechanisms, such as the dopaminergic spirals (VTA, SNpc), that carry information on value, and cortico-thalamo-cortical connnections, not represented in the figure.

The integrated view we just proposed, that assigns to cortex a special role in performing sensorimotor/cognitive transformations and to basal ganglia a special role in performing selection, in particular on the basis of value, is supported by four general system-level features of the two systems. Some aspects of these features

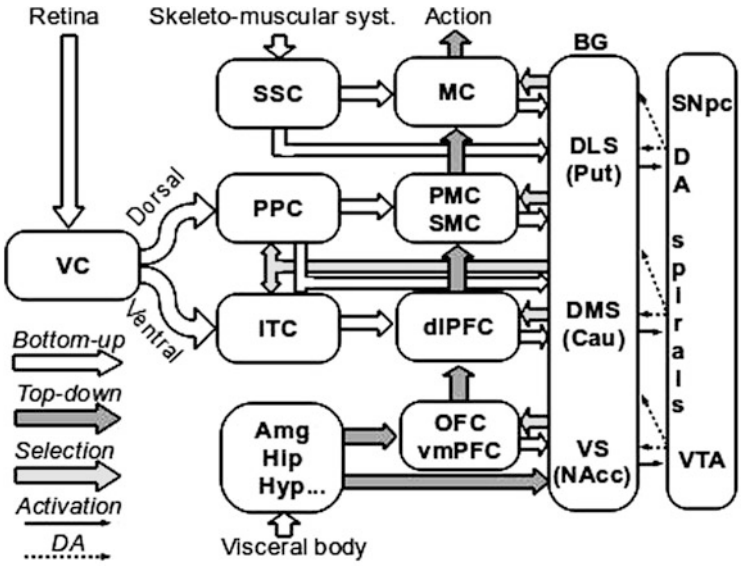

Fig. 15 An integrated view of the cortical and basal ganglia systems. Acronyms: Amg: amygdala; BG: basal ganglia; DA: dopamine; DLS: dorsolateral striatum; DMS: dorsomedial striatum; Hip: hippocampus; Hyp: hypothalamus; dlPFC: dorsolateral prefrontal cortex; ITC: inferotemporal cortex; MC: motor cortex; OFC: orbitofrontal cortex; PMC: premotor cortex; PPC: posterior parietal cortex; SMC: somatosensory cortex; SNpc: substantia nigra, pars compacta; SSC: somatosensory cortex; VC: visual cortex; vmPFC: ventromedial prefrontal cortex; VTA: ventral tegmental area

are represented in Fig. 16. The first feature involves the input/output ratio of each element of the two systems (e.g., PMC or DLS). This ratio is very high in basal ganglia with respect to cortex. Striatal neurons have a very large input from various cortical areas, whereas their output is focussed and concentrated on one specific cortical area with which it forms a loop (Redgrave et al. 1999; Wilson 1995). This makes this funnel-like architecture of basal ganglia ideally suited to perform the selection of whole neural assemblies at the level of the targeted cortex. Instead, cortical areas usually reciprocate a similar amount of connections to the areas from which they receive an input (Felleman and Van Essen 1991). This architecture is ideal to perform complex mappings through which detailed information (e.g., on perception) is transformed into other detailed information (e.g., on motor commands).

A second important feature is the realm of activity of the two systems. Cortex covers all aspect of cognition, from primary sensory processing to primary motor processing, from associative processing to the processing needed to implement executive functions. Instead, basal ganglia play an increasingly important role, measurable in terms of neural resources involved, going from sensory input to motor output, and from lower to higher levels of "cognition" (Alexander and Crutcher 1990). Thus, basal ganglia do not project to primary sensory cortical areas, involved

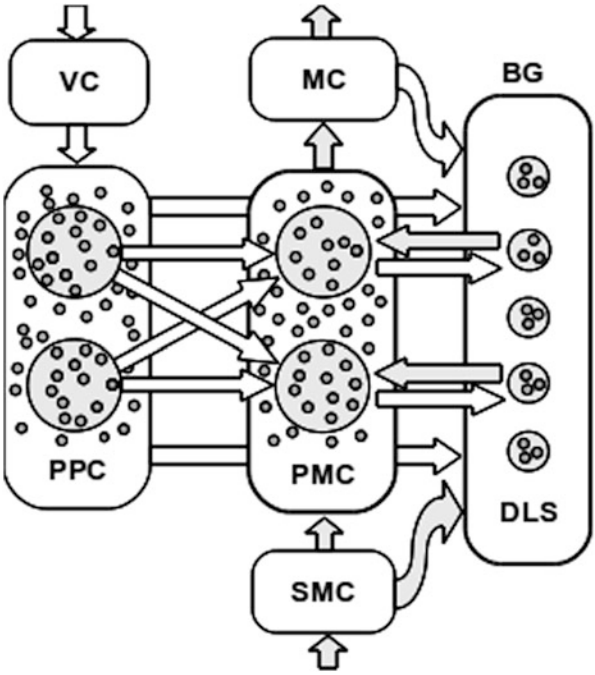

Fig. 16 Mechanisms of interaction involving cortex and basal ganglia (BG). The interactions are explained in the graph with a focus on the BG-cortical loop involving DLS and PMC. With respect to cortex, notice how: (a) PMC is part of a bottom-up information flow involving the VC-PPC-PMC cortical pathway; (b) PMC activity is modulated by a top-down information flow involving the SMC-PMC cortical pathway; (c) the VC-PPC-PMC-MC input–output pathway is "intercepted", at the level of PMC, by a cortico-basal ganglia loop involving DLS and supporting the selections happening within the PMC itself. With respect to basal ganglia: (a) DLS forms a loop with the target PMC cortical region; (b) DLS also receives information from all cortical areas that are linked to PMC (and from other cortical areas not reported here): the ample information gathered through these connections allows DLS to perform a well-informed selection of PMC contents. The graph also schematically illustrates the neural processes happening within and between DLS and PMC: (a) the cortical cell assemblies have comparable size (e.g., those of PPC and PMC in the graph) and exchange all-to-all connections (e.g., the PPC-PMC ones): this allows cortical pathways to perform fine mappings at all levels of cognition; (b) BG receive all-to-all afferent connections from cortex, but exchange connections organised in separate channels with the target cortex (DLS-PMC connections); (c) the BG cell assemblies are much smaller than those of cortex, so implementing a notable funnelling of cortical-to-BG information, and an "abstract control" of BG over the targeted cortex. See Fig. 15 for the acronyms

with sensory low-level cognition processes (Romanelli et al. 2005); the basal ganglia regions involving DMS have some projections to sensory associative areas (PC, IT); those involving DLS have important projections to the frontal cortex area where the brain processes action preparation and execution (PMC, MC); finally those involving DMS and VS have a major target in cortical areas implementing processes related to high-level executive functions (respectively, dlPFC and OFC/vmPFC).

These patterns allow cortex to perform detailed computations at all levels of cognition, while assign to basal ganglia a special role in selecting information at high levels of cognition and close to action preparation and performance.

A third feature is that the highest levels of the basal ganglia hierarchy (involving VS) are also more strongly linked to the sub-cortical areas processing value (e.g., Amg and Hip) with respect to the highest levels of the cortical hierarchy (PFC) (Graybiel and Kimura 1995). The ventral basal ganglia are also more strongly involved in the regulation, and as targets, of dopamine than the related cortical areas (Abercrombie et al. 1989; Berridge and Robinson 1998). These features pose basal ganglia in an ideal position to be informed about the subjective relevance of stimuli, i.e. value, so to perform at best the selection processes.

The last feature involves the learning processes usually ascribed to the two systems. Cortex is usually seen as the locus of unsupervised or associative learning (Doya 2000), as also indicated by the long-term potentiation (LTP) processes happening in it (Iriki et al. 1989; Kirkwood et al. 1996). Basal ganglia, instead, have a micro-architecture that makes it ideal for selection (Redgrave et al. 1999) and for trial-and-error learning guided by dopamine learning signals (Houk et al. 1995; Joel et al. 2002).

We close this section by showing a last aspect of the integrated system-level framework, in particular how it might aid the identification of the local micro interactions of the cortical and basal-ganglia systems where they physically contact. This is illustrated on the basis of Fig. 16. The cortical pathways implement detailed and focussed sensorimotor/cognitive transformations that progressively transform signals from sensory to motor areas (Cisek and Kalaska 2010; Miller and Cohen 2001). Instead, basal ganglia collect a wide range of information, including important information on value (Graybiel and Kimura 1995), and then perform a targeted selection of spatially limited, cortical contents based on focussed output channels involving the striatum-pallidal dishinibition mechanism (Chevalier and Deniau 1990). This *dishinibition* mechanism acts on the cortico-thalamic loops by freeing it from the tonic inhibition of GPi/SNpr so as to *let the information flow of the sensorimotor/cognitive transformations passing through the targeted cortical pathway stage to pass without interruption* (Mink 1996). In particular, information that travels through the various stages of cortex is amplified/refined/preserved by the close loops of such stages with thalamus. These cortico-thalamic loops are by default interrupted (at the level of thalamus) by tonic inhibitory inputs from basal-ganglia outputs. When basal-ganglia release from such inhibition specific cortico-thalamic loops, information is free to be elaborated and flow through the corresponding cortical area. The overall idea is thus that information travels from cortical sensory areas to motor cortical areas via different sensorimotor/cognitive transformation pathways: each of these is formed by various cortical stages within which the various specific contents, part of the information flow, can be either stopped or allowed to be elaborated and pass through by the specific channels of the basal-ganglia loops targeting them.

6 Conclusions

The hierarchical organisation of behaviour requires the implementation of three key functions by the underlying brain hierarchy, namely the implementation of sensorimotor/cognitive transformations at multiple levels of abstraction, the selection of various elements of such transformations, and the guidance of the learning processes. Based on this conceptual grid, the chapter showed how current research on the hierarchical organisation of brain is focussed either on the study of *cortical hierarchies*, or on the study of *sub-cortical hierarchies* formed by basal ganglia and other sub-cortical components processing value and motivations. When taken in isolation, these two approaches have important limitations in accounting for brain hierarchy. In particular, they either tend to ascribe to the systems they study all the three processes needed by the hierarchical organisation of behaviour or overlook some of those processes altogether. Thus, the cortical account often fails to specify the selection mechanisms needed to direct the course of action, and in large part the motivational mechanisms guiding learning processes. On the other side, in most cases the sub-cortical account fails to explain how the detailed sensorimotor and cognitive input–output mappings are learned and expressed.

The type of accounts of the hierarchical organisation of brain given by the two literature threads have been exemplified here through the presentation of two models, focussed, respectively, on the hierarchical processes implemented by the cortex and on the processes implemented by basal ganglia and amygdala. The two models made apparent how the two approaches mainly focus on, respectively, the explanation of sensorimotor/cognitive transformations happening at different levels of abstraction and on the explanation of selection based on value. The two models also highlighted that the computational approaches that back up the cortical and sub-cortical empirical study of brain hierarchy are affected by the same limitations and biases of the related empirical literatures.

Given these limitations, this chapter has proposed a system-level framework of the hierarchical brain within which the cortical and sub-cortical systems form a *whole integrated hierarchical system* and play complementary distinct roles. The principles of this integrated view can be summarised as follows:

– The Cortex is formed by multiple sensorimotor and cognitive pathways that perform fine and detailed information elaborations and transformations from sensations to actions. The capability to perform the elaborations and transformations is acquired mainly through unsupervised and associative learning mechanisms. The major pathways are: (a) a somatosensory-motor pathway to implement learning and performance of motor skills (this mainly involves somatosensory and motor cortex); (b) a dorsal pathway to build affordances and to prepare actions (this mainly involves parietal and premotor cortex); (c) a ventral pathway to identify the resources in the environment and to implement the highest level cognitive processes such as the executive control of goals encoded at multiple levels of abstraction (this mainly involves temporal and prefrontal cortex).

- Basal-ganglia form multiple loops with cortex and select information at multiple levels of abstraction. The ability to perform such selections is acquired on the basis of trial-and-error learning processes. The major basal-ganglia cortical loops are: (a) a sensorimotor loop, important for selecting motor acts (this involves dorsolateral portions of the basal ganglia, and motor cortex); (b) an associative loop, important to select perceptual and high-level cognition contents (this involves medial portions of basal ganglia, and temporal, parietal, and prefrontal cortex); (c) a limbic loop, important for selecting goals and contents with high biological valence, and to regulate the dopamine system (this involves ventral portions of basal ganglia, and orbital and medial prefrontal portions of cortex).
- The highest cortical levels and the ventral basal-ganglia levels have a strong interaction with limbic sub-systems of brain, and so are informed on the motivational and biological value of stimuli, events, and experiences (this involves sub-systems such as amygdala, hippocampus, hypothalamus, and the dopaminergic centres). This information drives and guides the learning processes happening in cortical and basal ganglia systems.
- Information flows and is finely elaborated within the cortical pathways, and within the various stages of such pathways, especially close to the output and at high-levels of cognition, basal ganglia select them by inhibiting them or letting them pass through.

The integrated framework proposed here leads us to overcome the limitations of the cortical and sub-cortical accounts of the hierarchical brain when taken in isolation. Indeed, within the framework the limitations of the cortical theories related to the selection and learning guidance functions are overcome by the fact that such functions are mainly implemented by the sub-cortical systems. On the other side, the limitation of the sub-cortical theories related to the lack of explanation of the fine sensorimotor and cognitive transformations are overcome by the implementation of such functions by cortical pathways.

Apart from these strengths, the hypothesis has still some open problems. We mention a few of these. As explained in Sect. 2, cortical systems are usually assumed to implement two forms of learning, namely unsupervised learning (especially within the perceptual areas) and associative learning (especially within the frontal areas). This raises a problem for the view proposed here when actions or other chunks of knowledge have to be acquired by the cortex on the basis of trial-and-error processes. The framework proposed here offers a solution to this problem. The solution is based on the intriguing idea that, at least when learning happens above a certain level of abstraction, basal-ganglia can acquire the mappings by trial-and-error, and then the information so acquired is progressively transferred to the cortex, which learns on the basis of associative processes under the "instruction" (supervision) of basal ganglia. There is indeed empirical evidence (Carelli et al. 1997; Tang et al. 2007) that when behaviour is first acquired and then automatized (i.e., it becomes "habitual") the basal ganglia show a high initial activation that then decreases with the progress of learning (see also Ashby et al. 2010, for a review

and Ashby et al. 2007, for a model). This hypothesis, however, needs to be further investigated in future work.

The second problem is a specification of the previous one when it is applied to the cortical acquisition of fine and detailed somatosensory-motor cortical transformations (e.g., the mapping implementing a skill). In this case, the mechanism proposed above cannot be exploited because, as shown in Fig. 16, basal ganglia can select cortical neural assemblies only at a gross level given their reduced number of neurons with respect to those of the targeted cortical areas. So, how can such mappings be acquired? This is a problem left open by the framework presented here, and leads us to touch on an issue that can be only introduced here and that should be tackled in future work. A possible solution to the problem might rely upon the *cerebellum*.

The Cerebellum, hosting more than half of the neurons of the brain, plays a critical role in the acquisition and expression of motor behaviour (Houk and Wise 1995; Kawato 1999). The problem mentioned above could be solved by a close interplay between the cortical somatosensory loop and the cerebellum. A possible idea to explore would be that the cerebellum aids the cortex to acquire fine sensorimotor mappings based on the *supervised learning processes* that it implements (Doya 2000; Rolls and Treves 1998). In line with this, some authors propose that the cerebellum plays a key role during learning but then progressively passes the acquired information to the cortex (see Hua and Houk 1997, for a review and a model). We think that the overall motor hierarchy involving the cerebellum would see the basal ganglia, cortex, and cerebellum playing their major roles, respectively, at the top, middle, and lower levels of the hierarchy, so we partially disagree with this view. However, we recognise that it represents a solution to the problem we are considering. These issues need further consideration in the future to be reconciled within the framework proposed here.

The latter observation leads us naturally to highlight a further limitation of the framework proposed here, namely the need to integrate the cerebellum within it. Indeed, aside from the sheer quantitative importance that the cerebellum has within the nervous system, there are strong indications that it forms important loops with the cortex similarly to basal ganglia (Middleton and Strick 2000). Moreover, the cerebellum plays important functions not only for motor behaviour but also for cognitive processes (Ito 2008). These aspects should be accounted for by a system-level framework of the brain hierarchy like the one presented here, a further issue to be investigated in future work.

Although we recognise the existence of these and other open issues, we think that the system-level framework presented here offers a better understanding of how the brain actually implements the three key functions critical for the hierarchical organisation of behaviour—sensorimotor/cognitive transformations, selection, and learning guidance—than the cortical and sub-cortical theories of it. In this respect, the framework is an important theoretical tool usable to formulate new specific theories, to make new predictions and to design new experiments to test them, and to design new computational models on brain hierarchy.

Acknowledgements This research has received funds from the European Commission 7th Framework Programme (FP7/2007–2013), "Challenge 2—Cognitive Systems, Interaction, Robotics", Grant Agreement No. ICT-IP-231722, project "IM-CLeVeR - Intrinsically Motivated Cumulative Learning Versatile Robots".

References

Abercrombie, E. D., Keefe, K. A., DiFrischia, D. S., Zigmond, M. J. (1989). Differential effect of stress on in vivo dopamine release in striatum, nucleus accumbens, and medial frontal cortex. *Journal of Neurochemistry*, *52*(5), 1655–1658.

Alcock, J. (1998). *Animal behavior: an evolutionary approach*, 6th edn. Sunderland: Sinauer Associated.

Alexander, G. E., & Crutcher, M. D. (1990). Functional architecture of basal ganglia circuits: neural substrates of parallel processing. *Trends in Neurosciences*, *13*(7), 266–271.

Alexander, G. E., DeLong, M. R., Strick, P. L. (1986). Parallel organization of functionally segregated circuits linking basal ganglia and cortex. *Annual Review of Neuroscience*, *9*, 357–381.

Ashby, F. G., Ennis, J. M., Spiering, B. J. (2007). A neurobiological theory of automaticity in perceptual categorization. *Psychological Review*, *114*(3), 632–656.

Ashby, F. G., Turner, B. O., Horvitz, J. C. (2010). Cortical and basal ganglia contributions to habit learning and automaticity. *Trends in Cognitive Sciences*, *14*(5), 208–215.

Balleine, B. W., & Dickinson, A. (1998). Goal-directed instrumental action: contingency and incentive learning and their cortical substrates. *Neuropharmacology*, *37*(4–5), 407–419.

Barsalou, L. W. (2008). Grounded cognition. *Annual Review of Psychology*, *59*, 617–645.

Barto, A. G., Singh, S., Chentanez, N. (2004). Intrinsically motivated learning of hierarchical collections of skills. In J. Triesch, & T. Jebara (Eds.), *International conference on developmental learning (ICDL2004)* (pp. 112–119). Piscataway, NJ: IEEE. UCSD Institute for Neural Computation, LaJolla, CA.

Barto, A. G., Sutton, R. S., Anderson, C. W. (1983). Neuronlike adaptive elements that can learn difficult control problems. *IEEE Transactions on Systems Man and Cybernetics*, *13*, 835–846.

Bast, T. (2007). Toward an integrative perspective on hippocampal function: from the rapid encoding of experience to adaptive behavior. *Reviews in the Neurosciences*, *18*(3–4), 253–281.

Berridge, K. C., & Robinson, T. E. (1998). What is the role of dopamine in reward: hedonic impact, reward learning, or incentive salience? *Brain Research Review*, *28*(3), 309–369.

Botvinick, M. M., Niv, Y., Barto, A. (2008). Hierarchically organized behavior and its neural foundations: a reinforcement-learning perspective. *Cognition*, *113*(3), 262–280.

Caligiore, D., Borghi, A. M., Parisi, D., Baldassarre, G. (2010). TRoPICALS: a computational embodied neuroscience model of compatibility effects. *Psychological Review*, *117*, 1188–1228.

Caligiore, D., Borghi, A. M., Parisi, D., Ellis, R., Cangelosi, A., Baldassarre, G. (2012). How affordances associated with a distractor object affect compatibility effects: a study with the computational model tropicals. *Psychological Research*, *77*, 7–19.

Cardinal, R. N., Parkinson, J. A., Hall, J., Everitt, B. J. (2002). Emotion and motivation: the role of the amygdala, ventral striatum, and prefrontal cortex. *Neuroscience & Biobehavioral Reviews*, *26*(3), 321–352.

Carelli, R. M., Wolske, M., West, M. O. (1997). Loss of lever press-related firing of rat striatal forelimb neurons after repeated sessions in a lever pressing task. *Journal of Neuroscience*, *17*(5), 1804–1814.

Chevalier, G., & Deniau, J. M. (1990). Disinhibition as a basic process in the expression of striatal functions. *Trends in Neurosciences*, *13*(7), 277–280.

Cisek, P. (2007). Cortical mechanisms of action selection: the affordance competition hypothesis. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *362*, 1585–1599.

Cisek, P., & Kalaska, J. F. (2010). Neural mechanisms for interacting with a world full of action choices. *Annual Review of Neuroscience*, *33*, 269–298.

Corbit, L. H., & Balleine, B. W. (2011). The general and outcome-specific forms of pavlovian-instrumental transfer are differentially mediated by the nucleus accumbens core and shell. *Journal of Neuroscience*, *31*(33), 11786–11794.

Corbit, L. H., Muir, J. L., Balleine, B. W. (2001). The role of the nucleus accumbens in instrumental conditioning: Evidence of a functional dissociation between accumbens core and shell. *Journal of Neuroscience*, *21*(9), 3251–3260.

Daw, N. D., Niv, Y., Dayan, P. (2005). Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature Neuroscience*, *8*(12), 1704–1711.

Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Cambridge: MIT.

Deco, G., & Rolls, E. T. (2003). Attention and working memory: a dynamical model of neuronal activity in the prefrontal cortex. *European Journal of Neuroscience*, *18*, 2374–2390.

Doya, K. (2000). Complementary roles of basal ganglia and cerebellum in learning and motor control. *Current Opinion in Neurobiology*, *10*(6), 732–739.

Ellis, R., & Tucker, M. (2001). The potentiation of grasp types during visual object categorization. *Visual Cognition*, *8*, 769–800.

Ellis, R., Tucker, M., Symes, E., Vainio, L. (2007). Does selecting one visual object from several require inhibition of the actions associated with nonselected objects? *Journal of Experimental Psychology Human Perception and Performance*, *33*, 670–691.

Erlhagen, W., & Schoner, G. (2002). Dynamic field theory of movement preparation. *Psychological Review*, *109*, 545–571.

Evangeliou, M. N., Raos, V., Galletti, C., Savaki, H. E. (2009). Functional imaging of the parietal cortex during action execution and observation. *Cerebral Cortex*, *19*, 624–639.

Fagg, A. H., & Arbib, M. A. (1998). Modeling parietal-premotor interactions in primate control of grasping. *Neural Networks*, *11*(7–8), 1277–1303.

Felleman, D. J., & Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, *1*(1), 1–47.

Fogassi, L., Ferrari, P. F., Gesierich, B., Rozzi, S., Chersi, F., Rizzolatti, G. (2005). Parietal lobe: from action organization to intention understanding. *Science*, *308*, 662–667.

Fuster, J. M. (2001). The prefrontal cortex—an update: time is of the essence. *Neuron*, *30*, 319–333.

Gazzaniga, M. (2004). *The cognitive neurosciences III*. Cambridge: MIT.

Grace, A. A., Floresco, S. B., Goto, Y., Lodge, D. J. (2007). Regulation of firing of dopaminergic neurons and control of goal-directed behaviors. *Trends in Neurosciences*, *30*(5), 220–227.

Graybiel, A., & Kimura, M. (1995). Adaptive neural networks in the basal ganglia. In J. C. Houk, J. Davis, D. Beiser (Eds.), *Models of information processing in the basal ganglia* (pp. 103–116). Cambridge: MIT.

Graziano, M. S. A. (2011). New insights into motor cortex. *Neuron*, *71*(3), 387–388.

Gurney, K., Prescott, T., Redgrave, P. (2001). A computational model of action selection in the basal ganglia. i. a new functional anatomy. *Biological Cybernetics*, *84*, 401–410.

Haber, S. N. (2003). The primate basal ganglia: parallel and integrative networks. *Journal of Chemical Neuroanatomy*, *26*(4), 317–330.

Haber, S. N., Fudge, J. L., McFarland, N. R. (2000). Striatonigrostriatal pathways in primates form an ascending spiral from the shell to the dorsolateral striatum. *Journal of Neuroscience*, *20*, 2369–2382.

Hall, J., Parkinson, J. A., Connor, T. M., Dickinson, A., Everitt, B. J. (2001). Involvement of the central nucleus of the amygdala and nucleus accumbens core in mediating Pavlovian influences on instrumental behaviour. *European Journal of Neuroscience*, *13*(10), 1984–1992.

Hamilton, A. F., & Grafton, S. (2007). *The motor hierarchy: from kinematics to goals and intentions* (vol. 22, pp. 381–408). Oxford: Oxford University Press.

Hebb, D. O. (1949). *The organization of behaviour*. New York: Wiley.

Heimer, L., Switzer, R. D., Hoesen, V. G. W. (1982). Ventral striatum and ventral pallidum: components of the motor system? *Trends in Neurosciences*, *5*(0), 83–87.

Houk, J. C., Davis, J., Beiser, D. (Eds.), (1995). *Models of information processing in the basal ganglia*. Cambridge: MIT.

Houk, J. C., & Wise, S. P. (1995). Distributed modular architectures linking basal ganglia, cerebellum, and cerebral cortex: Their role in planning and controlling action. *Cerebral Cortex*, *5*(2), 95–110.

Hua, S. E., & Houk, J. C. (1997). Cerebellar guidance of premotor network development and sensorimotor learning. *Learning & Memory*, *4*(1), 63–76.

Humphries, M. D., & Prescott, T. J. (2010). The ventral basal ganglia, a selection mechanism at the crossroads of space, strategy, and reward. *Progress in Neurobiology*, *90*(4), 385–417.

Iriki, A., Pavlides, C., Keller, A., Asanuma, H. (1989). Long-term potentiation in the motor cortex. *Science*, *245*(4924), 1385–1387.

Ito, M. (2008). Control of mental activities by internal models in the cerebellum. *Nature Reviews Neuroscience*, *9*(4), 304–313.

Jeannerod, M. (1999). Visuomotor channels: their integration in goal-directed prehension. *Human Movement Science*, *18*(2–3), 201–218.

Joel, D., Niv, Y., Ruppin, E. (2002). Actor-critic models of the basal ganglia: new anatomical and computational perspectives. *Neural Networks*, *15*(4–6), 535–547.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, *9*(6), 718–727.

Kilner, J. M. (2011). More than one pathway to action understanding. *Trends in Cognitive Sciences*, *15*, 352–357.

Kimchi, E. Y., & Laubach, M. (2009). Dynamic encoding of action selection by the medial striatum. *Journal of Neuroscience*, *29*(10), 3148–3159.

Kirkwood, A., Rioult, M. C., Bear, M. F. (1996). Experience-dependent modification of synaptic plasticity in visual cortex. *Nature*, *381*(6582), 526–528.

Koechlin, E., & Summerfield, C. (2007). An information theoretical approach to prefrontal executive function. *Trends Cognitive Sciences*, *11*(6), 229–235.

Kohonen, T. (2003). Self-organized maps of sensory events. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *361*, 1177–1186.

Lestou, V., Pollick, F. E., Kourtzi, Z. (2008). Neural substrates for action understanding at different description levels in the human brain. *Journal of Cognitive Neuroscience*, *20*, 324–341.

Levy, R., Friedman, H. R., Davachi, L., Goldman-Rakic, P. S. (1997). Differential activation of the caudate nucleus in primates performing spatial and nonspatial working memory tasks. *Journal of Neuroscience*, *17*(10), 3870–3882.

Lewis, S. J. G., Dove, A., Robbins, T. W., Barker, R. A., Owen, A. M. (2004). Striatal contributions to working memory: a functional magnetic resonance imaging study in humans. *European Journal of Neuroscience*, *19*(3), 755–760.

Lisman, J. E., & Grace, A. A. (2005). The hippocampal-vta loop: controlling the entry of information into long-term memory. *Neuron*, *46*(5), 703–713.

MacFarland, D. (1993). *Animal behavior*, 2nd edn. Harlow: Longman Group.

Mannella, F., Mirolli, M., Baldassarre, G. (2010). The interplay of Pavlovian and instrumental processes in devaluation experiments: a computational embodied neuroscience model tested with a simulated rat. In C. Tosh, & G. Ruxton (Eds.), *Modelling perception with artificial neural networks* (pp. 93–113). Cambridge: Cambridge University Press.

Mannella, F., Mirolli, M., Baldassarre, G. (2011). A system-level neural model of the brain mechanisms underlying instrumental devaluation in rats. In *COSYNE—Computational and Systems Neuroscience (2011), Salt Lake City, 24 February 2011*. Available from Nature Precedings: http://precedings.nature.com/documents/5849/version/1.

Meunier, D., Lambiotte, R., Bullmore, E. T. (2010). Modular and hierarchically modular organization of brain networks. *Frontiers in Neuroscience*, *4*, 200.

Middleton, F. A., & Strick, P. L. (1996). The temporal lobe is a target of output from the basal ganglia. *Proceedings of the National Academy of Sciences USA*, *93*(16), 8683–8687.

Middleton, F. A., & Strick, P. L. (2000). Basal ganglia and cerebellar loops: motor and cognitive circuits. *Brain Research Reviews*, *31*(2–3), 236–250.

Miller, E. K., & Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annual Review of Neuroscience*, *24*, 167–202.

Milner, A. D., & Goodale, M. A. (2008). Two visual systems re-viewed. *Neuropsychologia*, *46*, 774–785.

Mink, J. W. (1996). The basal ganglia: focused selection and inhibition of competing motor programs. *Progress in Neurobiology*, *50*(4), 381–425.

Mirolli, M., Mannella, F., Baldassarre, G. (2010). The roles of the amygdala in the affective regulation of body, brain, and behaviour. *Connection Science*, *22*(3), 215–245.

Mirolli, M., Santucci, V. G., Baldassarre, G. (2013). Phasic dopamine as a prediction error of intrinsic and extrinsic reinforcements driving both action acquisition and reward maximization: a simulated robotic study. *Neural Networks*, *39*, 40–51.

Munakata, Y., Herd, S. A., Chatham, C. H., Depue, B. E., Banich, M. T., O'Reilly, R. C. (2011). A unified framework for inhibitory control. *Trends in Cognitive Sciences*, *15*(10), 453–459.

Nachev, P., Kennard, C., Husain, M. (2008). Functional role of the supplementary and pre-supplementary motor areas. *Nature Reviews Neuroscience*, *9*, 856–869.

Oztop, E., & Arbib, M. A. (2002). Schema design and implementation of the grasp-related mirror neuron system. *Biological Cybernetics*, *87*(2), 116–40.

Pavlides, C., Miyashita, E., Asanuma, H. (1993). Projection from the sensory to the motor cortex is important in learning motor skills in the monkey. *Journal of Neurophysiology*, *70*(2), 733–741.

Pitkänen, A., Savander, V., LeDoux, J. E. (1997). Organization of intra-amygdaloid circuitries in the rat: an emerging framework for understanding functions of the amygdala. *Trends in Neurosciences*, *20*(11), 517–523.

Pouget, A., Dayan, P., Zemel, R. (2000). Information processing with population codes. *Nature Reviews Neuroscience*, *1*, 125–132.

Redgrave, P., Prescott, T. J., Gurney, K. (1999). The basal ganglia: a vertebrate solution to the selection problem? *Neuroscience*, *89*, 1009–1024.

Redgrave, P., Rodriguez, M., Smith, Y., Rodriguez-Oroz, M. C., Lehericy, S., Bergman, H., et al. (2010). Goal-directed and habitual control in the basal ganglia: implications for Parkinson's disease. *Nature Reviews Neuroscience*, *11*(11), 760–772.

Rizzolatti, G., & Craighero, L. (2004). The mirror-neuron system. *Annual Review of Neuroscience*, *27*, 169–192.

Rizzolatti, G., Fadiga, L., Gallese, V., Fogassi, L. (1996). Premotor cortex and the recognition of motor actions. *Brain Research*, *3*, 131–141.

Rolls, E. T., & Treves, A. (1998). *Neural networks and brain function*. Oxford: Oxford Unversity Press.

Romanelli, P., Esposito, V., Schaal, D. W., Heit, G. (2005). Somatotopy in the basal ganglia: experimental and clinical evidence for segregated sensorimotor channels. *Brain Research Reviews*, *48*(1), 112–128.

Schrimsher, G. W., Billingsley, R. L., Jackson, E. F., Moore, B. D. (2002). Caudate nucleus volume asymmetry predicts attention-deficit hyperactivity disorder (ADHD) symptomatology in children. *Journal of Child Neurology*, *17*(12), 877–884.

Sobel, I., & Feldman, G. (1968). A 3x3 isotropic gradient operator for image processing. Presentation for Stanford Artificial Project.

Solway, A., & Botvinick, M. M. (2012). Goal-directed decision making as probabilistic inference: a computational framework and potential neural correlates. *Psychological Review*, *119*(1), 120–154.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: an introduction*. Cambridge: MIT.

Tang, C., Pawlak, A. P., Prokopenko, V., West, M. O. (2007). Changes in activity of the striatum during formation of a motor habit. *European Journal of Neuroscience*, *25*(4), 1212–1227.

Thill, S., Caligiore, D., Borghi, A. M., Ziemke, T., Baldassarre, G. (2013). Theories and computational models of affordance and mirror systems: An integrative review. *Neuroscience and Biobehavioral Reviews, 37*, 491–521.

Todorov, E., & Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, *5*(11), 1226–1235.

Tokimura, H., Di Lazzaro, V., Tokimura, Y., Oliviero, A., Profice, P., Insola, A., et al. (2000). Short latency inhibition of human hand motor cortex by somatosensory input from the hand. *Journal of Physiology*, *523*(Pt 2), 503–513.

Ungerleider, L. G., & Mishkin, M. (1982). Two cortical visual systems. In D. J. Ingle, M. A. Goodale, W. R. J. Mansfield (Eds.), *Analysis of visual behavior* (vol. 549, pp. 549–586). Cambridge: MIT.

Venditti, A., Mirolli, M., Parisi, D., Baldassarre, G. (2009). A neural-network model of the dynamics of hunger, learning and action vigor in mice. In R. Serra, M. Villani, I. Poli (Eds.), *Artificial life and evolutionary computation: Proceedings of Wivace 2008* (pp. 131–142). Singapore: World Scientific. Venice, Italy, 8–10 September 2008.

Volkow, N. D., Wang, G.-J., Newcorn, J., Telang, F., Solanto, M. V., Fowler, J. S., et al. (2007). Depressed dopamine activity in caudate and preliminary evidence of limbic involvement in adults with attention-deficit/hyperactivity disorder. *Archives of General Psychiatry*, *64*(8), 932–940.

Voorn, P., Vanderschuren, L. J. M. J., Groenewegen, H. J., Robbins, T. W., Pennartz, C. M. A. (2004). Putting a spin on the dorsal-ventral divide of the striatum. *Trends in Neurosciences*, *27*(8), 468–474.

Wallis, J. D., Anderson, K. C., Miller, E. K. (2001). Single neurons in prefrontal cortex encode abstract rules. *Nature*, *411*, 953–956.

Walsh, V., & Cowey, A. (2000). Transcranial magnetic stimulation and cognitive neuroscience. *Nature Reviews Neuroscience*, *1*(1), 73–79.

Wilson, C. J. (1995). The contribution of cortical neurons to the firing pattern of striatal spiny neurons. In J. C. Houk, J. L. Davids, D. G. Beiser, (Eds.), *Models of information processing in the basal ganglia* (pp. 29–50). Cambridge: MIT.

Yeterian, E. H., & Pandya, D. N. (1995). Corticostriatal connections of extrastriate visual areas in rhesus monkeys. *Journal of Comparative Neurology*, *352*(3), 436–457.

Yin, H. H., & Knowlton, B. J. (2006). The role of the basal ganglia in habit formation. *Nature Reviews Neuroscience*, *7*, 464–476.

Yin, H. H., Ostlund, S. B., Balleine, B. W. (2008). Reward-guided learning beyond dopamine in the nucleus accumbens: the integrative functions of cortico-basal ganglia networks. *European Journal of Neuroscience*, *28*(8), 1437–1448.

Zahm, D. S. (2000). An integrative neuroanatomical perspective on some subcortical substrates of adaptive responding with emphasis on the nucleus accumbens. *Neuroscience & Biobehavioral Reviews*, *24*(1), 85–105.

Divide and Conquer: Hierarchical Reinforcement Learning and Task Decomposition in Humans

Carlos Diuk, Anna Schapiro, Natalia Córdova, José Ribas-Fernandes, Yael Niv, and Matthew Botvinick

Abstract The field of computational reinforcement learning (RL) has proved extremely useful in research on human and animal behavior and brain function. However, the simple forms of RL considered in most empirical research do not scale well, making their relevance to complex, real-world behavior unclear. In computational RL, one strategy for addressing the scaling problem is to introduce hierarchical structure, an approach that has intriguing parallels with human behavior. We have begun to investigate the potential relevance of hierarchical RL (HRL) to human and animal behavior and brain function. In the present chapter, we first review two results that show the existence of neural correlates to key predictions from HRL. Then, we focus on one aspect of this work, which deals with the question of how action hierarchies are initially established. Work in HRL suggests that hierarchy learning is accomplished by identifying useful subgoal states, and that this might in turn be accomplished through a structural analysis of the given task domain. We review results from a set of behavioral and neuroimaging experiments, in which we have investigated the relevance of these ideas to human learning and decision making.

1 Introduction

Many of the activities and tasks faced by humans and animals are hierarchical in nature: they involve tackling a set of nested subtasks, each of varying temporal extension. Problems like navigating involve devising high-level path plans, which are then broken down into smaller sub-planning problems, that can further be

C. Diuk (✉) · A. Schapiro · N. Córdova · J. Ribas-Fernandes · Y. Niv · M. Botvinick
Department of Psychology and Princeton Neuroscience Institute, Princeton University, Princeton, NJ, USA
e-mail: cdiuk@princeton.edu; schapiro@princeton.edu; ncordova@princeton.edu; jf3@princeton.edu; yael@princeton.edu; matthewb@princeton.edu

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_12,

decomposed all the way down to the level of motor primitives. For instance, the task of commuting to work involves deciding whether to take a train, bus or drive, and based on that decision others must be made: taking a train will require navigating to the train station, driving might involve subtasks like filling up the gas tank or checking the state of traffic on the planned route. A hierarchical structure of nested tasks emerges, which will at some level share components like standing up, sitting down, walking, and climbing stairs.

Work in cognitive and developmental psychology has recognized the hierarchical structure of behavior at least since the early 1950s, with the inception of the cognitive revolution. Prior to that watershed, the dominant schools of thought had focused on understanding behavior as a simple chain of stimulus-response associations. Lashley (1951) rejected this idea in favor of understanding behavioral sequences as controlled through a central plan, rather than as simple reflex chains. Following up on this perspective, further pioneering work by Miller et al. (1960) and Schank and Abelson (1977) noted that naturalistic behavior displays a stratified or layered organization, comprising nested subroutines.

In subsequent years, the hierarchical structure of behavior has been taken for granted in psychology and neuroscience. Computational models have been proposed to account for how hierarchically structured procedures are represented and executed (Botvinick and Plaut 2004; Cooper and Shallice 2000; Schneider and Logan 2006; Zacks et al. 2007), and how they are represented in the brain, in particular within the prefrontal cortex (Badre 2008; Haruno and Kawato 2006; Ito and Doya 2011; Koechlin et al. 2003). An important idea, coming primarily out of developmental psychology, is that humans and other animals gradually expand their competence by building up a repertoire of reusable skills or subroutines, which can be flexibly assembled into increasingly powerful hierarchical programs of action (Fischer 1980). The question of how this toolbox of skills is assembled represents one of the toughest questions attaching to hierarchical behavior.

In recent work, we have adopted a novel perspective on the cognitive and neural mechanisms underlying hierarchical behavior, leveraging tools from machine learning research. In particular, we have examined the potential relevance to human behavior and brain function of hierarchical reinforcement learning (HRL), a computational framework that extends reinforcement learning mechanisms into hierarchical domains. A number of intriguing parallels exist between HRL and findings from human and animal neuroscience, which encourage the idea that HRL may provide a useful framework for understanding the biological basis of hierarchical behavior. In the following section, we briefly review the essentials of HRL and summarize some of the potential neuroscientific parallels. We then present results suggesting neural correlates to two key predictions arising from computational HRL models. Next, we focus on a deep and open question: how is hierarchical structure established? What constitutes a "good" task decomposition? One appealing aspect of HRL is that it provides a context within which to consider the "toolbox" question, the question of how useful skills or subroutines are initially discovered or constructed. Following our brief introductory survey, we describe a

set of behavioral and neuroimaging experiments in which we have leveraged ideas from HRL to tackle this question.

2 Hierarchical Reinforcement Learning

Computational reinforcement learning (RL) has emerged as a key framework for modeling and understanding decision-making in humans and animals. In part, this is due to the fact that RL provides a normative computational model of behavior accounting for a host of previous experimental results in classical and instrumental conditioning. But most importantly, its impact has been felt through the discovery of parallels between elements of RL and aspects of neural function. The most critical parallel pertains to midbrain dopaminergic function, which has been proposed to transmit signals comparable to the reward-prediction errors that lie at the heart of RL (Houk et al. 1995; Montague et al. 1996; Schultz et al. 1997). However, other broader parallels have also been proposed, in particular with the so-called actor-critic RL architectures, which have inspired new interpretations of functional divisions of labor within the basal ganglia and cerebral cortex (Joel et al. 2002). Our research asks whether these connections between RL and neurobiology might extend to the setting of hierarchical behavior. Based on the success of standard RL as a framework for understanding the neural mechanisms underlying simple decision making, we hypothesize that HRL may hold similar promise as a framework for understanding the neural basis of hierarchical action.

Computational HRL was born, in part, out of the attempt to tackle the problem of scaling in RL. As researchers in the field recognized early on, one of the problems of basic RL methods is that they cannot cope well with large domains, that is, problems that require learning about large numbers of world states or large sets of possible actions. To make matters worse, RL suffers from what is known as the *curse of dimensionality*, an exponential explosion in the number of states as we increase the number of state variables, or features of the problem, that we want to consider. The result is that any task that requires keeping tabs on more than a handful of variables soon becomes intractable for standard RL algorithms.

A number of computational approaches have been proposed to address the scaling issue. One of them is to reduce the size of the problem at hand by treating subsets of states as behaviorally equivalent, known as *state abstraction*. Consider, for example, that you are walking to the train station, on your way to work. For this task, whether the shops along the way are open or closed is irrelevant, so two states that only differ in the status of a store can be grouped together. On the other hand, if later on you are navigating the same streets with the goal of buying coffee, a different set of variables becomes relevant, and states should be abstracted differently. For different state abstraction methods and aggregation criteria see, Li et al. (2006).

Another approach to addressing the scaling problem—the one taken in HRL—is based on *temporal abstraction* (Barto and Mahadevan 2003; Dayan and Hinton

1993; Dietterich 2000; Parr and Russell 1998; Sutton et al. 1999). The general idea is to expand the standard RL framework to include temporally-extended macro-actions, grouping together sets of simpler actions to form more complex, higher-level routines. Following the example mentioned earlier, the skill of *getting to work* can be thought of as a representation for a set of lower-level sequences like walking to the train station, taking the train and walking from the station to work. Moreover, the same *get to work* skill can encompass more than one set. For example, this skill might consist of not only a set of actions involving the train but also a different set that consists of actions like walking to the car, starting it, driving to work, etc. These multiple representations, abstracted away into the skill of *getting to work*, enable learning and reasoning at a coarser, more tractable granularity.

One particularly influential implementation of HRL, the *options* framework, was proposed by Sutton et al. (1999). The options framework supplements the set of single-step, primitive actions from standard RL with a set of temporally-extended "options". An option is, in a sense, a temporary sub-policy, a mapping from states to actions that does not have the goal of solving the complete problem at hand, but rather some sub-task that is, ideally, a step towards a larger goal. In this formalism, an option is defined by an initiation set, indicating the set of states from which the option can be selected; a termination function, which specifies the set of states that trigger termination of the option; and an option-specific policy (a mapping from states to actions that is in effect while the option is active).

Importantly, in the options framework as in other versions of HRL (Dietterich 2000; Parr and Russell 1998), option-specific policies can map states not only into primitive actions but also into other options, allowing hierarchies of options to be assembled. In the previous example, it is clear that walking to a train station or to the car are not "primitive" actions, but compound, temporally extended behaviors that involve numerous more basic skills and can be achieved in a multiplicity of ways. In an HRL setting, an option for getting to work would call other options for walking to the train station or the car, these would call further options guiding the action of walking, and so forth down to elementary motor commands.

3 Potential Neural Correlates

We see two reasons for considering the potential relevance of HRL to understanding behavior and brain function in humans and other animals. First, if the brain does indeed implement learning mechanisms related to those found in RL, then the RL scaling problem must pertain in neuroscience just as it does in machine learning, raising the question of how RL mechanisms in the brain cope with large-scale tasks. As a computational technique for easing the scaling problem, HRL may furnish clues concerning the brain's ability to select adaptive behaviors in such settings. The second motivation for considering HRL from a neuroscientific perspective is, of course, the pervasively hierarchical structure of human behavior. HRL presents the

possible opportunity to extend our understanding of neural mechanisms for RL so as to engage the issue of hierarchy, significantly widening the scope of current theories.

As a first step toward evaluating the potential neural relevance of HRL, Botvinick et al. (2009) derived a set of predictions from the framework, evaluating the extent to which current scientific knowledge accorded with each of its elements.This work leveraged the existence of proposed parallels between elements of the actor-critic architecture for RL (see Sutton and Barto 1998) and specific brain structures. Botvinick et al. considered what additions or alterations would be required in order to extend the actor-critic architecture for HRL. It turns out that only a handful of modifications are needed, and each of these appears to resonate with established neuroscientific findings.

A key parallel pointed out by Botvinick et al. (2009) relates to the computational requirement, within HRL, of maintaining a representation of the currently selected option. This function seems very closely related to functions commonly ascribed to the dorsolateral prefrontal cortex (DLPFC), and other frontal areas including pre-supplementary motor area (pre-SMA). The DLPFC has been suggested to house representations that guide temporally integrated, goal-directed behavior (Fuster 1997), and recent work has refined this idea by demonstrating that DLPFC neurons play a role in representing task sets: a single pattern of DLPFC activation represents an entire mapping from stimuli to responses (that is, a policy; see Miller and Cohen 2001). Moreover, neurons in several frontal areas (DLPFC, pre-SMA and SMA) have been shown to code for particular sequences of low-level actions, just like options do in HRL. Evidence also shows that areas in frontal cortex represent action at multiple, nested levels of temporal structure (see Badre 2008; Koechlin et al. 2003), akin to the way HRL representations organize tasks into hierarchies, with policies for one option calling other, lower-level options.

The role of options in HRL is to impose an option-specific policy. In translations of RL into neuroscience, policy representations have been proposed to reside at least partially within the dorsolateral striatum. From the point of view of the HRL hypothesis, it is suggestive that DLPFC, SMA, and pre-SMA areas all project heavily into this structure, potentially allowing modulation of policy representations by representations of subtask context. Botvinick et al. (2009) review neurophysiological findings consistent with this idea.

Another computational requirement of HRL is to maintain option-specific value functions. As discussed in Botvinick et al. (2009), this is needed because the value of a state relative to the goals of an option or subroutine may differ from the value of that state relative to top-level goals (i.e., primary reward); option-specific value functions are thus critical for driving the learning of subroutine policies. In work drawing parallels between standard RL and neural structures, an area often linked with state or state-action value representation is the ventral striatum. If HRL mechanisms are relevant, then we might expect to find a neural structure that connects to ventral striatum while at the same time receiving inputs from areas of frontal cortex that carry option representations. An area that meets this criterion is the orbitofrontal cortex (OFC), connecting heavily with both ventral

striatum and DLPFC. As reviewed by Botvinick et al. (2009), research suggests that representations of reward in OFC can be sensitive to shifts in response strategy or task set (O'Doherty et al. 2003; Schoenbaum et al. 1999), linking precisely with the idea that OFC might represent option-specific state values. The OFC also appears to sustain reward-predictive activity over relatively extended periods (Schultz et al. 2000), a function necessary in HRL to support the calculation of reward-prediction errors when options terminate.

As detailed in Botvinick et al. (2009), neural HRL would also impose specific functional requirements on reward-prediction errors, widely believed to be signaled in the brain by phasic fluctuations in dopamine release. Whereas in ordinary RL prediction errors signal whether the selection of single actions turns out better or worse than expected (see Sutton and Barto 1998), under HRL the scope of the prediction error expands to embrace the intervals spanned by options. This resonates with a theoretical analysis of dopamine signaling by Daw et al. (2003), interpreting dopamine function in computational (semi-Markov) terms that also underlie the options framework.

Two key neural predictions arise from HRL. First, in order to sustain learning of option-specific value functions at various levels of a hierarchical task decomposition, multiple prediction error signals are required, sometimes occurring concurrently. Second, HRL predicts that reward prediction errors should occur not only in association with top-level goals (marked by primary reward), but also in connection with *subgoals*. In both cases, previous research provides little to go on. In the next two sub-sections we present work indicating the presence of neural correlates to the two key predictions from HRL. First, we summarize results from an fMRI experiment (Diuk et al. 2012b) which revealed striatal activity correlating with two simultaneous prediction error signals, corresponding to two levels of a hierarchical gambling task. Next, we review work by Ribas-Fernandes et al. (2011) in which we used EEG and fMRI to assay for subgoal-linked reward prediction errors and found activations consistent with these in multiple structures including anterior cingulate cortex, insula, habenula, and amygdala.

Taken together, available neural data encourage the idea that HRL may be relevant to understanding the neural substrates of hierarchical behavior in humans and animals. Even if this turns out to be true, however, there are limits on what present-day HRL research can tell us about brain function, given that computational HRL is associated with its own open questions. Perhaps foremost among these is the problem foreshadowed earlier: how an agent may initially build up a repertoire of useful subroutines (options) from which hierarchical action programs may be composed. This question, which in HRL research has sometimes been referred to as the "option discovery problem", is clearly of equal importance within psychology and neuroscience, and we will dedicate the rest of the chapter to work in which we have begun to address it.

3.1 Two Simultaneous, but Separable, Reward Prediction Errors

Under the HRL options framework, a situation can arise in which the outcome of an action elicits learning at multiple hierarchical levels at the same time. For example, the execution of a primitive action a that leads to a sub-goal state enables learning both about the one-step transition produced by the action, and the temporally-extended subtask that ended with the attainment of the sub-goal. This situation prescribes the presence of two distinct reward prediction errors.

If the brain implements an HRL mechanism, we should be able to measure activity correlating with at least two prediction errors at the same time. In order to test this key HRL prediction, we designed a two-level gambling task (Diuk et al. 2012b), which constitutes a hierarchical extension of the classic bandit task used in previous RL research. The task is summarized in Fig. 1. In each trial, participants first chose between two doors, representing two casinos. Once a casino was chosen, its door opened and a "target" was revealed a number of points (2–10, distributed normally with means 5 and 6 in each of the two casinos) that must be accumulated in order to gain a reward of 10 cents in the casino. Each casino also contained a unique set of four slot machines, of which participants chose two to play. Each slot machine granted 0–5 points, normally distributed, with an independent, slowly drifting mean. If they did not succeed in meeting the target with their two plays, participants lost 10 cents.

This task was designed to elicit learning at two levels: at the slot-machine level (to inform choices within a casino) and at the casino level (to inform choices between the two casinos). In particular, two distinct and coincident prediction errors should occur after playing the second slot machine, when the point outcome of that machine is revealed simultaneously with the win/lose outcome of the casino as a whole. Importantly, in this design these two prediction errors are uncorrelated: It is possible to obtain fewer points than expected on the second slot machine (a negative slot-level prediction error) while at the same time still win the casino as a whole (a positive casino-level prediction error), and vice versa.

We asked 28 participants to play the Casino Task for 120 trials each while undergoing functional Magnetic Resonance Imaging (fMRI) (Diuk et al. 2012b). We modeled the participants' learning under the options framework, where playing the left or the right casino constituted two temporally-extended options and each options' policy consisted of choosing which slot machines to play. We verified that the HRL model best fit the participants behavior when compared to some otherwise plausible alternative models and used the prediction errors generated by this model as regressors to correlate against the registered brain activity. We analyzed in particular the activity in ventral striatum and found it correlated with all three regressors generated by the model, corresponding to prediction errors for the first slot machine ($p < 0.004$), second slot machine ($p < 0.02$), and the casino as a whole ($p < 5.5 \times 10^{-5}$). Note that prediction errors for the second slot machine and the casino occurred simultaneously.

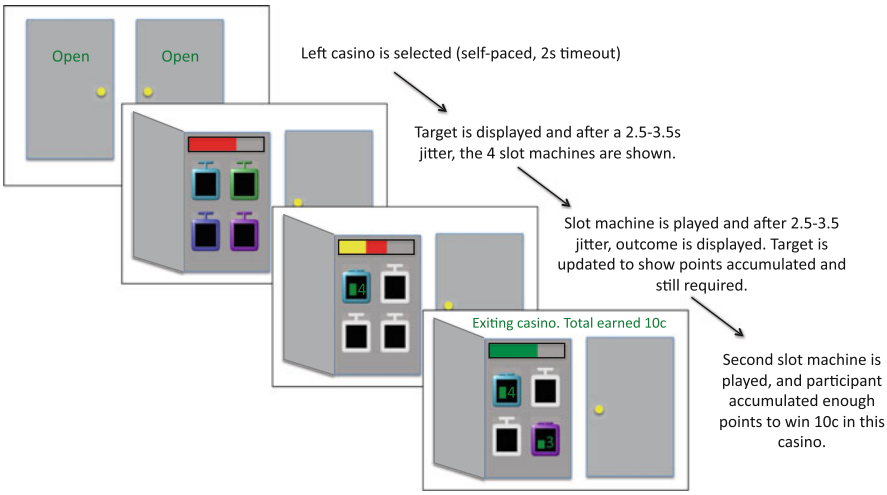

Fig. 1 Sample trial (from Diuk et al. (2012b): The participant chooses to play in the left casino, the door opens and displays a target number of points (indicated by *red bar*). After a few seconds, the four slot machines appear. The participant plays *upper-left slot* and after a few seconds, the points obtained in that machine are shown inside the machine (as a *green bar* plus a Roman numeral). Part of the target turns *yellow*, indicating the points accumulated with the first slot machine play. The rest is still *red*, indicating the points still necessary to win the casino. The participant plays the *bottom right slot* machine and obtains sufficient points to win the casino (10 cents). The target bar turns *green* and a message appears indicating the casino win

These results have two major implications: The first is that the human brain can calculate prediction errors that temporally span over several states and actions, as is required in HRL (Botvinick et al. 2009). The second implication is that more than one prediction error signal may be calculated and employed for learning in the brain, in contrast to original studies which suggested that dopaminergic neurons all report one unitary prediction error signal Schultz et al. (1997). This may not be surprising from a theoretical point of view, as learning about two (or more) separate reward predictions within any given scenario requires the calculation of two separate prediction errors. Such a dual-task situation may be common in daily life. However, reinforcement learning tasks previously examined in laboratory settings did not directly test this prediction.

3.2 Neural Correlates of Pseudo-reward

A second prediction states that prediction errors should occur in connection with task subgoals as well as with top level goals. HRL agents associate a special form of reward with subgoals, dubbed *pseudo-reward* (Sutton et al. 1999). Distinguishing pseudo-rewards from primary reward is crucial: If subgoals were confounded with

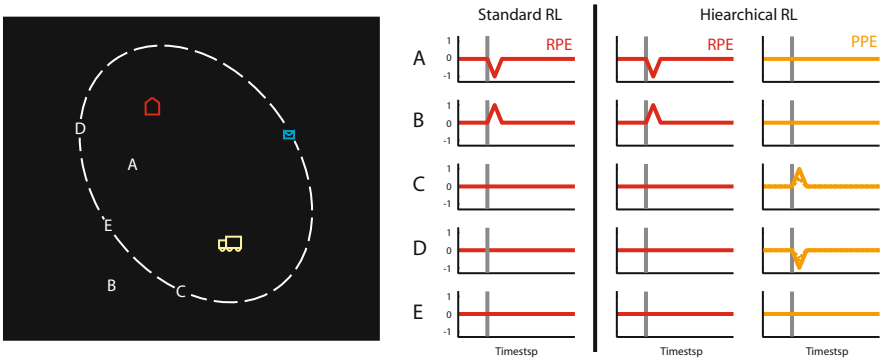

Fig. 2 Task and predictions from HRL and RL (from Ribas-Fernandes et al. 2011). *Left*: Task display and underlying geometry of the delivery task. *Right*: Prediction-error signals generated by standard RL and by HRL in each category of jump event. *Gray bars* mark the time-step immediately preceding a jump event. *Dashed time-courses* indicate the PPE generated in C and D jumps that change the subgoal's distance by a smaller amount

primary reward, the agent might get stuck "chasing" subgoals, even when irrelevant to top-level goals. The distinction between pseudo and primary rewards results in a distinction between two types of prediction errors: ordinary reward prediction errors (RPEs) occur in response to differences in predicted outcomes in progress towards primary goals. Pseudo-reward prediction errors (PPEs) occur in response to outcomes in progress towards subgoals. PPEs are unique to HRL, they do not occur in ordinary, "flat" RL. If HRL is relevant to neural activity, we should expect to see neural correlates of PPEs. Ribas-Fernandes et al. (2011) designed a task to test this prediction, using EEG and fMRI to assay for a neural analogue to the pseudo-reward prediction error.

Figure 2 illustrates the task. Only the colored elements in the figure appear in the task display. The overall objective of the game is to complete a "delivery" as quickly as possible, using joystick movements to guide the truck first to the package and from there to the house. It is self-evident how this task might be represented hierarchically, with delivery serving as the (externally rewarded) top-level goal and acquisition of the package as an obvious subgoal. For an HRL agent, delivery would be associated with primary reward, and acquisition of the package with pseudo-reward.

An additional twist was that on some trials, the package unexpectedly jumped to a new location before the truck reached it. According to RL, a jump to point A in the figure, or any location within the ellipse shown, should trigger a positive RPE, because the total distance that must be covered in order to deliver the package has decreased. (Note that we assume temporal discounting, which implies that attaining the goal faster is more rewarding.) By the same token, a jump to point B or any other exterior point should trigger a negative RPE. Cases C, D, and E are quite different. Here, there is no change in the overall distance to the goal, and so no RPE should

be triggered, either in standard RL or in HRL. However, in case C the distance to the subgoal has decreased. According to HRL, a jump to this location should thus trigger a positive PPE. Similarly, a jump to location D should trigger a negative PPE (note that location E is special, being the only location that should trigger neither an RPE nor a PPE). These points are illustrated in the right panel of Fig. 2, which shows RPE and PPE time-courses from simulations of the delivery task based on standard RL and HRL.

A group of 30 participants performed the delivery task while undergoing fMRI. Here, one third of the trials included a jump of type D (see Fig. 2), predicted to elicit a negative PPE. Neural correlates for such a jump were found in dorsal anterior cingulate cortex and habenula, structures previously suggested to reflect or induce reduced dopaminergic activity.

Because these PPEs are unique to HRL, not occurring in standard RL, Ribas-Fernandes et al. (2011) interpreted these results as providing a neural signature of HRL.

4 The Option Discovery Problem: Identifying Useful Subgoals

In the field of computational HRL, research has focused on the problem of how temporally-extended actions can be incorporated into the standard RL formalism. Some success has been achieved in showing how skills that are provided to the learner as input, or have somehow been previously acquired, can be exploited in order to learn to solve new problems faster (Dietterich 2000; Sutton et al. 1999). However, less work has been done, and less success has been achieved, on the very difficult question of where skill representations come from. How does a learner decide, while performing a task, what components of it are worth incorporating into a collection of skills for future use? This question has added relevance because the wrong set of skills can actually impair learning (Botvinick et al. 2009).

In computational work, option discovery has often been understood to involve the heuristic identification of useful subgoal states. Once a useful subgoal is identified, the learner can then build a strategy to achieve it, turning this strategy or policy into a reusable skill. Note that these subgoal states are not necessarily extrinsically rewarding, that is, the learner might not receive any reward for reaching them. A key assumption of HRL is that the agent is motivated to reach an option's subgoal, once the option gains control of behavior. In HRL, as discussed in the previous section, attaining the subgoal yields a special reward signal, referred to as pseudo-reward, which serves to sculpt the option's policy. However, for this machinery to come into play, pseudo-reward must be assigned to specific outcomes, and therefore the question persists: How are useful subgoal states initially identified?

A number of possible answers to this question can be drawn from both the computational literature and from psychology and neuroscience. One class of

proposals portrays options and subgoals as genetically specified, shaped by natural selection across generations (Elfwing et al. 2007; Schembri et al. 2007a,b). Basic motor behavior, for example, has often been characterized as building upon simple, innate components (Bruner 1975). In a few cases, extended action sequences, such as grooming in rodents, have also been thought of in the animal behavior literature as genetically specified (Aldridge and Berridge 1998). While a role for evolutionary programming seems inevitable, it clearly cannot be the whole story, since both humans and animals obviously discover and incorporate useful behavioral subroutines through learning (Conway and Christiansen 2001; Fischer 1980).

Another approach to explaining subgoal discovery leverages the notion of intrinsic motivation (Baldassarre and Mirolli 2012). The idea here is that certain events or stimuli are inherently interesting to the behaving animal or human. These can be stimuli that display salient perceptual properties or that challenge expectations, eliciting curiosity (Schmidhuber 1991a,b). In an HRL context such states are proposed to be adopted as subgoals, triggering the construction of associated skills or options (see Singh et al. 2005).

The intrinsic motivation perspective provides a compelling account of option discovery. However, without greater specification, it leaves open the question of *which* properties make particular states intrinsically motivating or interesting to the agent. In order to set the scene for our own research in this area, we can consider two general approaches, one based on frequency and the other on problem structure.

Frequency-based methods are based on observed trajectories (that is, sequences of actions that are performed to solve a task). These methods are based on the idea that an animal or human that has experienced a series of interrelated problems, or has had repeated exposure to a problem, is able to extract either subsequences or subgoal states based on their frequent occurrence in trajectories that lead to reward. For example, consider a delivery person distributing packages inside a building. After repeated deliveries, this person might construct some pre-defined ways of traversing certain floors. Furthermore, he might realize that many trajectories involve taking the elevator. He would thus identify reaching the elevator as a useful sub-goal, and construct paths that lead from different offices in a floor to the closest elevator, adding to his repertoire of actions what we could call the *go to elevator* option. Proposals based on this idea can be found in the work of McGovern and Barto (2001); Picket and Barto (2002); Thrun and Schwartz (1995); Yamada and Tsuji (1989).

To introduce structure-based methods it is useful to consider why, in the aforementioned delivery example, the elevator state emerged as special. In this scenario, the elevator state occurs frequently because the elevator is a sort of *bottleneck*: to reach any location on one floor from another floor, one must pass through the elevator. The elevator is thus a location that gives access to an unusually diverse set of other locations. A more formal way of capturing this property can be drawn from graph theory. If we envision the various locations (say, cubicles and offices in our courier's building) as nodes in a graph, with edges connecting immediately adjacent locations, then the elevator location would stand out as a node

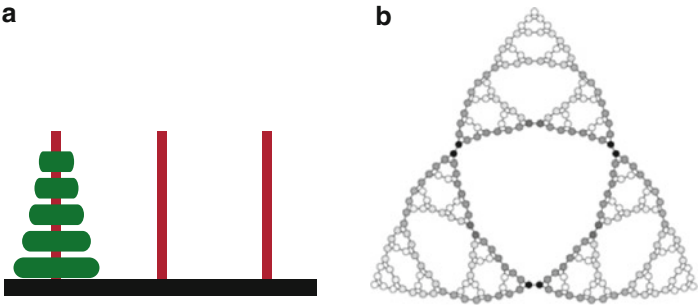

Fig. 3 (**a**) One state of the Tower of Hanoi problem. Disks are moved one at a time between posts, with the restriction that a disk may not be placed on top of a smaller disk. An initial state and goal state define each specific problem. (**b**) Representation of the Tower of Hanoi problem as a graph. Nodes correspond to states (disk configurations). *Shades of gray* indicate betweenness. Source: Şimşek (2008)

with high graph *centrality* (see Opsahl et al. 2010). A particular way of quantifying centrality is via a measure called *betweenness*, which counts the number of shortest paths within the graph that pass through an index node. An illustration, from Şimşek (2008), is shown in Fig. 3.

Şimşek (2008) and Şimşek and Barto (2009) proposed that option discovery might be fruitfully accomplished by identifying states at local maxima of graph betweenness (for related ideas, see also Şimşek et al. (2005); Hengst (2002); Jonsson and Barto (2006); Menache et al. (2002). They presented simulations showing that an HRL agent designed to select subgoals (and corresponding options) in this way, was capable of solving complex problems, such as the Tower of Hanoi problem in Fig. 3, significantly faster than a non-hierarchical RL agent.

As part of our research exploring the potential relevance of HRL to neural computation, we evaluated whether these proposals for subgoal discovery might relate to procedures used by human learners. The research we have completed so far focuses on the identification of bottleneck states, as laid out by Şimşek and Barto (2009). In what follows, we summarize the results of three experiments, which together support the idea that the notion of bottleneck identification may be useful in understanding human subtask learning.

5 Experiments 1 & 2: Humans Identify and Exploit Bottleneck States

In the first experiment we investigated whether humans can identify bottleneck states, when doing so allows them to optimize their performance. We summarize the experiment and its results here; full details are presented in Diuk et al. (2012a).

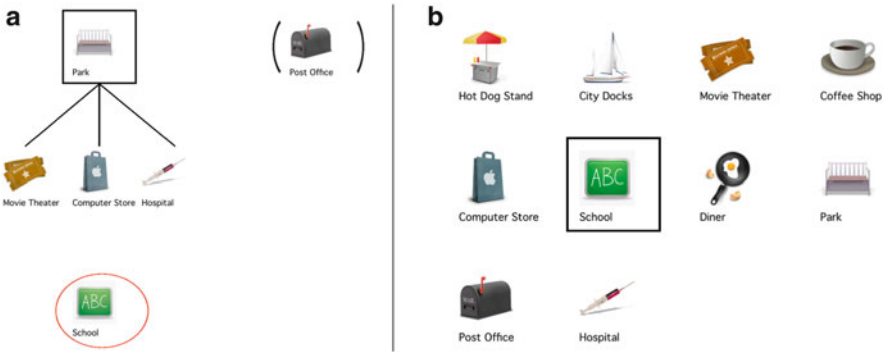

Fig. 4 Interface of experiments 1 & 2. (**a**) At the *top*, the current location (*Park*) is identified along with its three adjacent locations. Circled at the *bottom* is the target destination (*School*), and on the *upper right corner* is the bus-stop location (*Post office*), reachable in one step from any other location. (**b**) The *square* identifies the current location (*School*), and participants must click on its three neighbors

Participants were asked to navigate through a small town, making an extended series of deliveries between landmarks (e.g., school, post office, coffee shop). A new start location and goal location were randomly selected at the beginning of each trial (delivery). Participants were told that they would be paid for each delivery, but that the amount would depend on how many steps they took to reach their goal: each step would subtract a fixed amount from the "full pay." The graphical interface, illustrated in Fig. 4a, indicated the participant's present location, the goal location, and the set of landmarks immediately adjacent to it. Navigation was accomplished by selecting among the latter. Also shown was a "bus stop" location to which the participant could travel from any location using one step. After some experience with the "town," the participant was allowed to choose a new bus-stop location after every five deliveries. Any landmark within the town could be chosen for the bus-stop location. At any time during a delivery, the participant could elect to "jump" to the bus stop, potentially saving costly steps toward the goal.

Underlying the adjacency relations among landmarks in the town was the graph shown in Fig. 5a. Each node corresponds to a landmark, and each edge to an adjacency relation. The graph contains an obvious bottleneck location, which has high graph betweenness. This location represents the best choice for the bus-stop location; given the definition of betweenness, this location lies on the largest number of shortest paths within the graph, and therefore offers the best chance of saving the participant steps toward a delivery to a yet-unknown destination.

Note that participants were never actually shown the graph in Fig. 5a, or any other sort of bird's-eye view of the town. The display only provided information about local adjacencies. Nevertheless, we hypothesized that, with accumulating experience, participants would identify the bottleneck location and exploit it by selecting it as a bus-stop location. Figure 6 summarizes the results of the experiment.

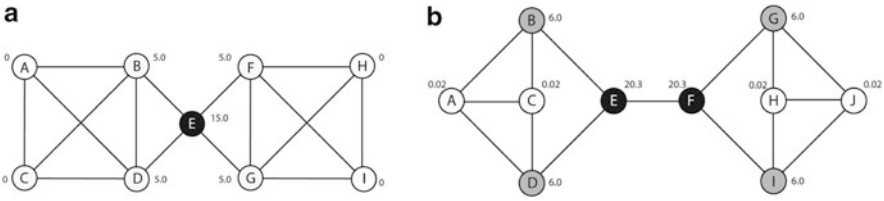

Fig. 5 Graphs underlying the maps of the cities for the first version of the experiment (**a**) and the second one (**b**). Node labels identify the betweenness of each node

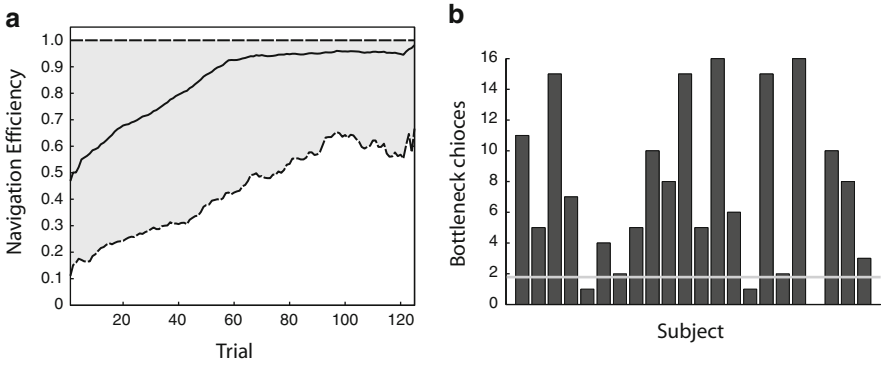

Fig. 6 (**a**) Average performance ratio, over all participants, as a function of trials of experience with the "town." The value on the y-axis in the figure represents the ratio of steps taken to the minimum number of steps, taking into account the optimal bus-stop location. A ratio of 1 indicates optimal performance, i.e., choice of the shortest path from start to goal, assuming an optimal (bottleneck) choice of bus stop. The two dashed data series indicate minimum and maximum score across participants. (**b**) Number of times, out of 16 five-trial blocks, that the bottleneck state was chosen for the bus-stop location. Participants in the x-axis are sorted by performance. The *horizontal line* indicates the expected performance if participants chose bus-stop locations randomly

Panel *a* shows that, over the course of the experiment, participants increasingly picked out the shortest path from start to goal. This simply provides evidence that participants learned something about the layout of the town as they went along. More important are the data in panel *b*, which show the number of blocks (out of a total of 16) in which each participant chose to place the bus stop at the bottleneck location. Although there was some variability across participants, the data clearly confirm a general capacity to detect and exploit the presence of a bottleneck.

The results of this experiment do not, however, allow us to make conclusions about *how* participants identified the bottleneck location. In particular, while we were interested in the possibility that they leveraged structural or topological knowledge, it is possible that participants instead used simple frequency information. Over the course of multiple shortest-path deliveries, the bottleneck location would be

expected to occur frequently, compared with other locations. In order to rule out frequency as the full explanation for our initial findings, we repeated Experiment 1, but with a twist. In this revised version (Experiment 2), participants learned about adjacency relations, but did not ever traverse the town before choosing a bus-stop location. This follow-up experiment was also intended to address a second limitation of the first experiment. Note that in the graph used in the first experiment, not all vertices had the same degree (i.e., the same number of immediate neighbors). While vertices on the "outskirts" of the city had three neighbors, the bottleneck vertex and those adjacent to it had four. In principle, this might have made the bottleneck salient, providing a different explanation for its selection.

Experiment 2, reported in detail in Diuk et al. (2012a), removed the confound between centrality and frequency and used a graph in which all vertices had the same degree (Fig. 5b). Figure 4b illustrates the graphical interface for the task. On each trial, an index location was highlighted, and participants were asked to indicate its three immediate neighbors, receiving feedback concerning the accuracy of their choices. After approximately 20 min on this training task, participants were told they would have to make a delivery between two undisclosed locations, under the same shortest-path conditions as in Experiment 1. Prior to receiving the delivery assignment, participants were asked to choose a location for the bus stop. After they had chosen a location, their knowledge of the underlying topology of the town was tested by asking them to draw a map, indicating adjacency relations between landmarks. Of forty participants tested, 23 drew an accurate map of the town, and of these 23, 18 (78 %) chose one of the bottleneck locations as the bus-stop location, a result far above the chance level of 20 %.

In a further experiment, which we only briefly summarize here, Diuk et al. (2012a) showed that when participants were given a start and goal location, and asked to verify whether a third location would fall on a shortest delivery path, they were especially fast at responding the question when the probe location corresponded to a domain bottleneck. The finding suggests participants formulated delivery plans using bottleneck locations as subgoals.[1]

Together, the foregoing provide support for the idea that humans can identify and exploit bottleneck states in a novel domain, based on an internal model of the domain's structure. Taken on their own, however, they leave open a second question. The computational proposal from HRL was that bottleneck locations provide the anchor for temporal abstractions, representations that treat temporally extended behaviors as a unit. The experiments just reported do not speak to this aspect of the theoretical proposal. However, we can glean some pertinent evidence from a third experiment.

[1]This particular result provides preliminary evidence for "model-based" hierarchical planning in the Diuk et al. (2012a) delivery task.

6 Experiment 3: Bottleneck States and Temporal Abstraction

Our approach in Experiment 3 was based on previous work using event parsing. A standard experimental paradigm in cognitive psychology involves showing an action sequence, and asking participants to "parse" it by pressing a key when they feel that one subsequence or subtask has ended and a new one has begun (Zacks et al. 2007). Consistent with earlier work, we assumed that such parsing responses mark the boundaries of temporally abstract events, i.e., subsequences that the participant views, on some level, as a unit. Based on this assumption, we predicted that if participants were exposed to event sequences that involved bottlenecks, participants would parse those sequences at moments in which a bottleneck was traversed. Details of our experiment are reported in Schapiro et al. (2013); we summarize the work here.

Participants were exposed to a sequence of images presented one at a time over a period of 35 min. During this exposure period, participants were asked to judge whether each image was presented in a canonical orientation, or rotated. The task did not require them to attend to the sequential order of images at all. However, unbeknownst to the participants, that order was highly structured. Specifically, the sequence was generated by a random walk through the graph shown in Fig. 7a. Each of the 15 possible images was assigned to a vertex, and when that vertex occurred in the random walk, the associated image was presented. As is obvious from the figure, the graph contains a subset of bottleneck vertices with high betweenness, namely the vertices that link the three star-shaped clusters. Drawing on the complex network literature, we refer to these clusters as "communities" (see Schapiro et al. 2013).

After performing the orientation judgment task, the sequence of images continued, but participants were asked to perform the standard parsing task, pressing a key when "natural breakpoints" occurred, i.e., when one "subsequence" ended and a new one began. No instruction other than this was given.

Results indicated that participants were significantly more likely to parse at moments where the sequence moved from one star-shaped cluster into another ($p < 0.05$), points corresponding to the traversal of high-betweenness vertices. This result held even when the analysis was limited to Hamiltonian cycles through the graph (traversals of the graph without item repetitions), showing that parsing decisions were not based entirely on item recency judgments or simple effects of priming.

We additionally hypothesized that stimuli grouped together as part of the same event on the basis of community structure might come to be represented more similarly in the brain. Participants in a new experiment were exposed to sequences of stimuli generated from the graph in Fig. 7 in the fMRI scanner. To test the hypothesis that items in the same community would be represented more similarly, we analyzed the similarity of the patterns of voxel activation evoked by the stimuli in searchlights throughout the brain. We found that the patterns of activation for items within a community were more similar than those for items between communities in left IFG, anterior temporal lobe (ATL), insula, and superior temporal gyrus (STG) (see Fig. 8).

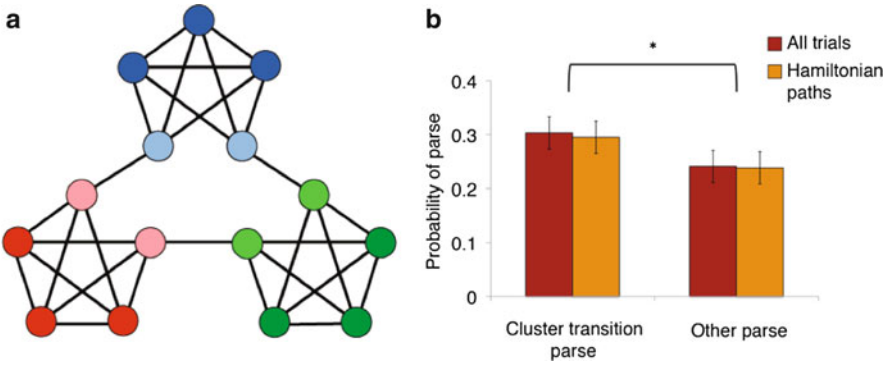

Fig. 7 (**a**) Underlying graph of the task. Each node in the graph is linked to a stimulus used in the sequence. (**b**) Proportion of times participants parsed sequence at cluster-changing points, as opposed to other points in the sequence

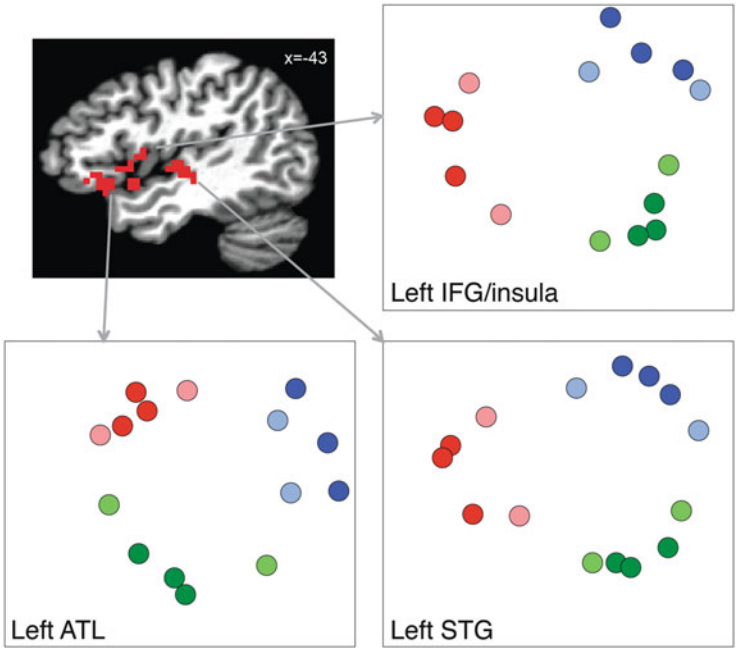

Fig. 8 Pattern similarity effects in left IFG/insula, left ATL, and left STG. Each cluster showed reliable community structure in the BOLD response in a whole-brain search. The similarity structure within each area is visualized using multi-dimensional scaling, with items color-coded in accordance with the graph nodes in Fig. 7

The relation of this experiment to HRL-like action selection is necessarily indirect, given that the task involved observation rather than production of sequences. However, the results are in line with the idea that bottleneck states are not only spontaneously identified by humans, but that bottlenecks provide a basis for the formation of temporally abstract event representations. This is consistent with the proposal that bottleneck states provide anchors for the construction of temporally abstract action representations, i.e., options, although further experimentation will be needed to validate this inference.

7 Discussion

The development of the field of computational RL, together with the discovery of its neural implications, has proven extremely useful in the study of human and animal behavior and brain function. A known limitation of standard RL, however, is its poor scaling to large, real-world problems. Given this limitation, it is unreasonable to expect basic RL principles to account for human learning and decision making in their full complexity. However, the possibility arises of looking at measures proposed by the computational community to deal with the scaling problem, evaluating their possible relevance to the biological case. We reported work that takes this approach, examining one aspect of complex behavior, namely its hierarchical structure. In the work we have reported, the aim was to leverage existing work in HRL, a sub-field developed precisely for tackling the scalability problem, to shed light on how humans might learn to master hierarchically-structured tasks. Our agenda was further reinforced by evidence of potential neural correlates that map nicely with existing HRL frameworks.

One aspect of hierarchical learning, which has provided an important focus for our work, involves the challenge of discovering useful subtask decompositions. On the computational front, this problem has suggested a form of intrinsic motivation, which leads learning agents to identify problem states as sub-goals, constructing the necessary skills to achieve them. The work we have reviewed tested the relevance of this idea to human learning and decision making. In particular, we explored one approach to this problem, based on structural task analysis. We presented three experiments whose results are consistent with the idea that humans are able to learn the topological structure underlying a problem domain, to detect states associated with high centrality (in the graph-theoretic sense), and to adopt them as useful subgoals and as anchors for temporally abstract event representations.

One outstanding question is how subtasks are transferred, once learned. Even though many hierarchical problems share exactly matching subtasks (boiling water for the preparation of both tea and coffee), many other problems faced by humans and animals have only partially overlapping states or actions. A richer understanding of subtask learning should include a mechanism for such less constrained transfer.

Overall, the work we have reviewed, together with convergent evidence available from previous studies, suggests that HRL may provide a useful set of tools for

further investigating the computational and neural basis of hierarchically structured behavior. In this sense, HRL may play the same catalytic role, in the context of hierarchical behavior, that ordinary RL has so fruitfully played in the study of performance in simpler tasks.

References

Aldridge, J. W., & Berridge, K. C. (1998). Coding of serial order by neostriatal neurons: a "natural action" approach to movement sequence. *Journal of Neuroscience*, *18*(7), 2777–2787.

Badre, D. (2008). Cognitive control, hierarchy, and the rostro-caudal organization of the frontal lobes. *Trends in Cognitive Sciences*, *12*(5), 193–200.

Baldassarre, G., & Mirolli, M. (Eds.), (2012). *Intrinsically motivated learning in natural and artificial systems*. Berlin: Springer.

Barto, A. G., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, *13*(4), 341–379.

Botvinick, M., & Plaut, D. C. (2004). Doing without schema hierarchies: a recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review*, *111*(2), 395–429.

Botvinick, M. M., Niv, Y., Barto, A. C. (2009). Hierarchically organized behavior and its neural foundations: a reinforcement learning perspective. *Cognition*, *113*(3), 262–280.

Bruner, J. (1975). Organization of early skilled action. *Child Development*, *44*, 1–11.

Conway, C. M., & Christiansen, M. H. (2001). Sequential learning in non-human primates. *Trends in Cognitive Sciences*, *5*(12), 539–546.

Cooper, R., & Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology*, *17*(4), 297–338.

Daw, N. D., Courville, A. C., Touretzky, D. S. (2003). Timing and partial observability in the dopamine system. In *Advances in Neural Information Processing Systems (NIPS)*. Cambridge: MIT.

Dayan, P., & Hinton, G. E. (1993). Feudal reinforcement learning. In *Advances in neural information processing systems 5* (pp. 271–278). San Mateo: Morgan Kaufmann.

Dietterich, T. G. (2000). Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, *13*, 227–303.

Diuk, C., Cordova, N., Niv, Y., Botvinick, M. (2012a). Discovering hierarchical task structure. *Submitted*.

Diuk, C., Tsai, K., Wallis, J., Niv, Y., Botvinick, M. M. (2012b). Hierarchical learning induces two simultaneous, but separable, prediction errors in human basal ganglia. *The Journal of Neuroscience, 33*(13), 5797–5805.

Elfwing, S., Uchibe, E., Doya, K., Christensen, H. I. (2007). Evolutionary development of hierarchical learning structures. *IEEE Transactions on Evolutionary Computation*, *11*(2), 249–264.

Fischer, K. W. (1980). A theory of cognitive development: the control and construction of hierarchies of skills. *Psychological Review*, *87*(6), 477–537

Fuster, J. M. (1997). *The prefrontal cortex: anatomy, physiology, and neuropsychology of the frontal lobe*, 3rd edn. Philadelphia: Lippincott-Raven.

Haruno, M., & Kawato, M. (2006). Heterarchical reinforcement-learning model for integration of multiple cortico-striatal loops: fMRI examination in stimulus-action-reward association learning. *Neural networks: the official journal of the international neural network society*, *19*(8), 1242–1254.

Hengst, B. (2002). Discovering hierarchy in reinforcement learning with HEXQ. In *Proceedings of the 19th international conference on machine learning*, Sydney, Australia.

Houk, J., Adams, J., Barto, A. (1995). A model of how the basal ganglia generate and use neural signals that predict reinforcement. In J. Houk, J. Davis, D. Beiser (Eds.), *Models of information processing in the basal ganglia*. Cambridge: MIT.

Ito, M., & Doya, K. (2011). Multiple representations and algorithms for reinforcement learning in the cortico-basal ganglia circuit. *Current Opinion in Neurobiology*, *21*(3), 368–373.

Joel, D., Niv, Y., Ruppin, E. (2002). Actor—critic models of the basal ganglia: new anatomical and computational perspectives. *Neural Networks*, *15*, 535–547.

Jonsson, A., & Barto, A. (2006). Causal graph based decomposition of factored MDPs. *Journal of Machine Learning Research*, *7*, 2259–2301.

Koechlin, E., Ody, C., Kouneiher, F. (2003). The architecture of cognitive control in the human prefrontal cortex. *Science (New York, N.Y.)*, *302*(5648), 1181–1185.

Lashley, K. S. (1951). *The problem of serial order in behavior*. New York: Wiley

Li, L., Walsh, T. J., Littman, M. L. (2006). Towards a unified theory of state abstraction for MDPs. In *Proceedings of the ninth international symposium on artificial intelligence and mathematics (AMAI-06)*.

McGovern, A., & Barto, A. G. (2001). Automatic discovery of subgoals in reinforcement learning using diverse density. *Proceedings of the 18th international conference on machine learning*.

Menache, I., Mannor, S., Shimkin, N. (2002). Q-cut-dynamic discovery of sub-goals in reinforcement learning. In *European conference on machine learning (ECML 2002)* (pp. 295–306).

Miller, E. K., & Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annual Review of Neuroscience*, *24*, 167–202.

Miller, G. A., Galanter, E., Pribram, K. H. (1960). *Plans and the structure of behavior*. New York: Adams-Bannister-Cox

Montague, P. R., Dayan, P., Sejnowski, T. J. (1996). A framework for mesencephalic predictive hebbian learning. *Journal of Neuroscience*, *16*(5), 1936–1947.

O'Doherty, J., Critchley, H., Deichmann, R., Dolan, R. J. (2003). Dissociating valence of outcome from behavioral control in human orbital and ventral prefrontal cortices. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, *23*(21), 7931–7939.

Opsahl, T., Agneessens, F., Skvoretz, J. (2010). Node centrality in weighted networks: generalizing degree and shortest paths. *Social Networks*, *32*, 245–251.

Parr, R., & Russell, S. J. (1998). Reinforcement learning with hierarchies of machines. *Advances in neural information processing systems*.

Picket, M., & Barto, A. (2002). Policyblocks: An algorithm for creating useful macro-actions in reinforcement learning. In *Proceedings of the 19th International conference on machine learning*.

Ribas-Fernandes, J. J. F., Solway, A., Diuk, C., McGuire, J. T., Barto, A. G., Niv, Y., Botvinick, M. M. (2011). A neural signature of hierarchical reinforcement learning. *Neuron*, *71*(2), 370–379.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals, and understanding: an inquiry into human knowledge structures.* Hillsdale: Lawrence Erlbaum.

Schapiro, A., Rogers, T., Cordova, N., Turk-Browne, N., Botvinick, M. (2013). Neural representations of events arise from temporal community structure. *Nature Neuroscience, 16*, 486–492.

Schembri, M., Mirolli, M., Baldassarre, G. (2007a). Evolution and learning in an intrinsically motivated reinforcement learning robot. In F. Almeida y Costa, L. M. Rocha, E. Costa, I. Harvey, A. Coutinho (Eds.), *Advances in artificial life. Proceedings of the 9th European conference on artificial life*. *LNAI* (vol. 4648, pp. 294–333). Berlin: Springer.

Schembri, M., Mirolli, M., Baldassarre, G. (2007b). Evolving internal reinforcers for an intrinsically motivated reinforcement-learning robot. In Y. Demiris, D. Mareschal, B. Scassellati, J. Weng (Eds.), *Proceedings of the 6th international conference on development and learning* (pp. E1–E6). London: Imperial College.

Schmidhuber, J. (1991a). A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proceedings of the international conference on simulation of adaptive behavior: from animals to animats* (pp. 222–227).

Schmidhuber, J. (1991b). Curious model-building control systems. *Proceedings of the International Conference on Neural Networks*, *2*, 1458–1463.

Schneider, D. W. & Logan, G. D. (2006). Hierarchical control of cognitive processes: switching tasks in sequences. *Journal of Experimental Psychology: General*, *135*(4), 623–640.

Schoenbaum, G., Chiba, A. A., Gallagher, M. (1999). Neural encoding in orbitofrontal cortex and basolateral amygdala during olfactory discrimination learning. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, *19*(5), 1876–84.

Schultz, W., Dayan, P., Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, *275*(March 1997), 1593–1599.

Schultz, W., Tremblay, L., Hollerman, J. R. (2000). Reward processing in primate orbitofrontal cortex and basal ganglia. *Cerebral Cortex*, *10*(3), 272–84.

Şimşek, O. (2008). *Behavioral building blocks for autonomous agents: description, identification, and learning*. PhD thesis, University of Massachussetts, Amherst.

Şimşek, O., Barto, A. G. (2009). Skill Characterization Based on Betweenness. In D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), *Advances in neural information processing systems 21* (pp. 1497–1504).

Şimşek, O., Wolfe, A. P., Barto, A. G. (2005). Identifying useful subgoals in reinforcement learning by local graph partitioning. In *Proceedings of the twenty-second international conference on machine learning*.

Singh, S., Barto, A., & Chentanez, N. (2005). *Proceedings of Advances in Neural Information Processing Systems 17*.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: an introduction*. Cambridge: MIT.

Sutton, R. S., Precup, D., Singh, S. (1999). Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, *112*, 181–211.

Thrun, S., & Schwartz, A. (1995). Finding structure in reinforcement learning. In G. Tesauro, D. Touretzky, T. Leen (Eds.), *Advances in neural information processing systems (NIPS) 7*. Cambridge: MIT.

Yamada, S., & Tsuji, S. (1989). Selective learning of macro-operators with perfect causality. In *Proceedings of the 11th international joint conference on Artificial intelligence, Volume 1* (pp. 603–608), San Francisco: Morgan Kaufmann Publishers Inc.

Zacks, J. M., Speer, N. K., Swallow, K. M., Braver, T. S., Reynolds, J. R. (2007). Event perception: a mind-brain perspective. *Psychological Bulletin*, *133*(2), 273–293.

Neural Network Modelling of Hierarchical Motor Function in the Brain

Juan M. Galeazzi and Simon M. Stringer

Abstract This chapter discusses computer modelling of hierarchical motor function in the brain. The focus is on dynamical models that utilise biologically plausible neural network architectures with local associative synaptic learning rules. The chapter begins with a review of our own laboratory's work in this area. We present a series of hierarchical motor models and relate these to various areas of brain function. This is followed by a discussion of the limitations of these models and directions for future research.

1 Introduction

Motor function in humans and animals is organized hierarchically, with high level motor programs able to recruit lower level motor primitives (Ghez and Krakauer 2000; Hughlings Jackson 1878). For example, studies with humans have shown that drawing a figure "8" appears to be comprised of a series of distinct line segments, in each of which the angular motion initially increases and then decreases (Lacquaniti et al. 1983). The hierarchical organization of motor function leads to efficient and faster learning of new high level motor skills, which can incorporate previously learned movement primitives. This hierarchical organization may be broadly subdivided into a number of stages of processing within the brain (Gazzaniga et al. 1998) as shown in Fig. 1. The lowest level of motor processing is in the spinal cord, which encodes simple reflexes and rhythmic movements such as walking in cats (Belanger et al. 1996; Pearson and Gordon 2000). The spinal cord receives afferent projections from the motor areas of the cortex and the brainstem. The brainstem is the intermediate stage of motor processing. This area

J.M. Galeazzi (✉) · S.M. Stringer
Oxford Centre for Theoretical Neuroscience and Artificial Intelligence, Department of Experimental Psychology, University of Oxford, South Parks Road, Oxford, OX1 3UD, UK
e-mail: juan.galeazzigonzalez@psy.ox.ac.uk; simon.stringer@psy.ox.ac.uk; www.oftnai.org

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_13,

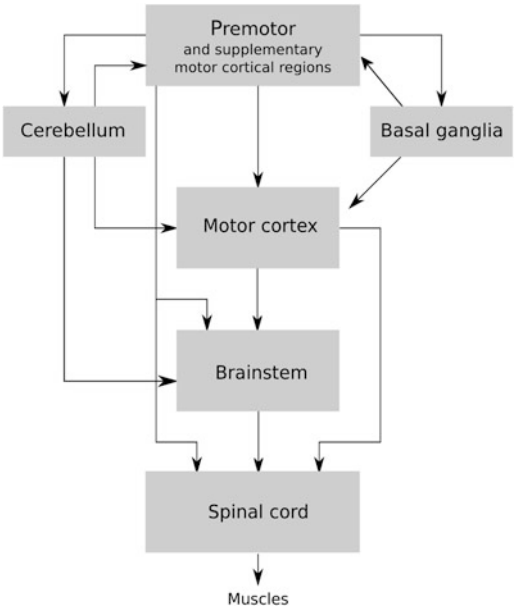

Fig. 1 Hierarchical organization of motor function in the brain

mediates movements of the eyes and head, postural adjustments, and goal-directed movements of the arm and hand. The brainstem receives projections from the motor areas of the cortex, whose activity is modulated by the cerebellum and basal ganglia. The cortical motor areas encode high level voluntary motor programs (Krakauer and Ghez 2000; Passingham 1993). Projections from the motor areas of the cortex to the brainstem and spinal cord are able to recruit elemental motor primitives encoded in these lower stages in order to effect complex sequential motor programs. Each level of the motor hierarchy receives sensory input appropriate to guiding the motor operations at that level. Thus, descending projections from the motor areas of the cortex are able to initiate sequences of motor primitives in the lower stages of the motor hierarchy, which are guided by sensory feedback at those levels. This reduces the complexity and amount of information that would otherwise need to be transmitted from the motor areas of cortex to the lower stages of motor processing (Ghez and Krakauer 2000).

This chapter will begin by reviewing a number of neural network models of hierarchical motor function in the brain (Stringer and Rolls 2007; Stringer et al. 2007, 2003), which our laboratory has developed over several years. Our aim throughout has been to develop time-continuous dynamical models that rely on neural network architectures with local associative synaptic learning rules that are as biologically plausible as possible. A basic model is introduced in Sect. 2, which is built from multiple layers of neurons such as state cells, motor cells, movement selector cells, and high level movement selector cells. High level motor programs encoded in the higher layers of the model are able to recruit sequences of low level motor primitives to effect a desired trajectory through the state space of the

simulated agent. In Sect. 3 the model is extended to show how environmental context, such as the presence of an obstacle, can be incorporated in order to modulate the performance of a high level motor program. This can result in the model generating novel motor sequences not previously performed during learning, and thereby provides a potential mechanism that might contribute to the problem of serial order in behaviour as discussed by Lashley (1951). In Sect. 4 it is shown how some minor modifications to the basic model will allow sequential high level motor programs to be learned with a delayed reward signal, which is only received after the model has performed the motor sequence successfully during training. How can a high level motor program learn a temporal sequence of low level motor primitives if the reward signal that guides learning is only received after all the motor primitives have been performed? Section 5 discusses some of the limitations of the models reviewed in this chapter and directions for future research.

2 A Hierarchical Dynamical Model of Motor Function

Stringer et al. (2003) developed a time-continuous neural network model of motor function that could learn arbitary motor sequences and perform these motor sequences at varied speeds in the absence of sensory inputs. These are important characteristics of biological motor function (Bizzi and Polit 1979; Laszlo 1966, 1967; Polit and Bizzi 1978; Schmidt 1987, 1988). This work has been extended by Stringer and Rolls (2007) to include hierarchical motor function, in which high level movement selector commands learned to drive motor programs composed of temporal sequences of low level motor primitives.

The architecture of the model developed by Stringer and Rolls (2007) is shown in Fig. 2.

The state cells represent the postural or positional state of the agent. For example, if we consider the motor task of reaching to a visual target, the state might be the position of the target with respect to the hand. This important spatial information would be needed to guide reaching. Consistent with this, a number of areas of the posterior parietal cortex and premotor areas have been found to contain neurons that respond to the location of a visual target object in a *hand-centred* frame of reference (Bremner and Andersen 2012; Buneo and Andersen 2006; Buneo et al. 2002; Chang and Snyder 2010; Graziano et al. 1997; Graziano 1999). These neurons provide a direct representation of the distance between the hand and the target object location in a common reference frame, which is necessary to guide reaching to the visual target.

The model architecture shown in Fig. 2 assumes that the state cells derive their firing properties through visual or proprioceptive input. A biologically plausible model of how neurons in the posterior parietal cortex could develop firing responses in a hand-centred frame of reference by visually guided learning has been recently proposed by Galeazzi et al. (2013).

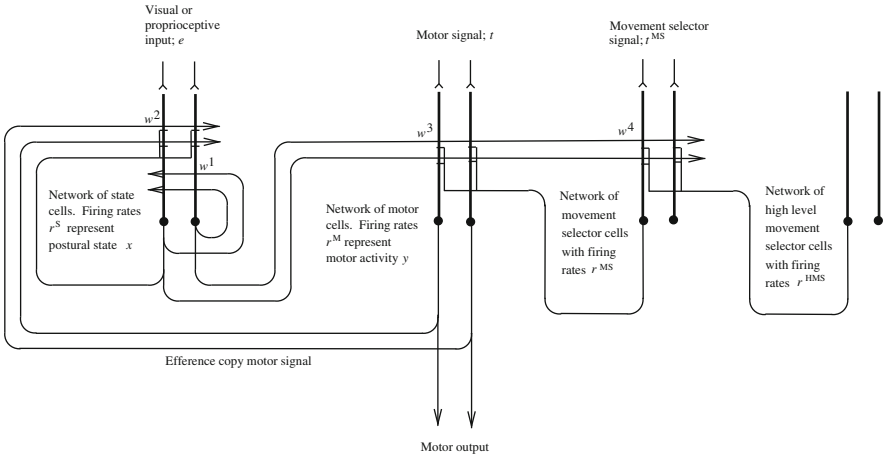

Fig. 2 Architecture of a neural network model of hierarchical motor function. Reproduced with permission from Stringer and Rolls (2007)

Experimental studies have also shown that hand-centred cells in PRR and Area 5 can maintain and update their firing properties in the absence of visual input (Bremner and Andersen 2012; Buneo and Andersen 2012; Chang and Snyder 2010). Consistent with this, the state cells shown in Fig. 2 receive an efference copy signal from the motor cells that permits the firing of the state cells to be updated in the dark.

The motor cells in Fig. 2 are at the lowest level of the motor hierarchy. In the model, the outputs from the motor cells send direct projections to the muscles that effect movement of the arm. These cells correspond to the motor neurons in the spinal cord. The next level of processing in the motor hierarchy shown in Fig. 2 contains movement selector cells. Individual movement selector cells represent the command to perform a particular motor primitive, such as a small movement of the arm. These cells might correspond to more intermediate levels of the motor hierarchy in the brain, perhaps distributed across the spinal cord and the brainstem. For example, neurophysiological studies have shown that the spinal cord encodes simple reflexes and rhythmic movements such as walking in cats (Belanger et al. 1996; Pearson and Gordon 2000). However, Hart and Giszter (2004) have shown that the brainstem can contribute to the performance of motor primitives in frogs by improving the smoothness of movement.

The network architecture implements a *forward model* (Miall and Wolpert 1996; Wolpert and Flanagan 2001) in the synaptic connections w^2. The purpose of the forward model is to continuously and rapidly update the internal representation of the state of the agent from the current motor command. This is useful for fast ballistic motor programs, where there is not enough time for visual feedback to update the internal state representation. Recent neuroimaging studies have demonstrated representations of the outcome of actions in the prefrontal cortex (Hamilton and Grafton 2008). In the model shown in Fig. 2, the synaptic connections

w^2 update the representation in the state cells given the combined firing in the state cells and motor cells.

The network incorporates an *inverse model* (Miall and Wolpert 1996) in the synaptic connections w^3. The function of the inverse model is to fire the correct sequence of motor cells in order to perform a specified motor primitive, where the firing of the motor cells is dependent on the state of the agent. That is, given a specified motor primitive, the inverse model selects the correct sequence of motor cell firing. In Fig. 2, the synaptic connections w^3 drive the motor cells to perform appropriate actions given combined firing in the state cells and movement selector cells.

The highest level of the motor hierarchy shown in Fig. 2 contains the high level movement selector cells. These cells drive the movement selector cells through synapses w^4. The high level movement selector cells represent high level motor programs, which are composed of sequences of motor primitives. The synapses w^4, which drive the movement selector cells, are from combinations of state cells and high level movement selector cells. Hence, the activation of movement selector cells is dependent on the firing of both the state cells and the high level movement selector cells.

The high level movement selector cells correspond to a number of different cell types involved in the execution of higher level motor programs in the brain. Planning, selection, and control of the appropriate motor sequences require the involvement of a number of cortical areas and distributed systems such as the motor cortex, premotor cortex, posterior parietal cortex, and subcortical areas like the basal ganglia and cerebellum (Gazzaniga et al. 1998). For example, Georgopoulos (1995) showed the existence of neurons in the motor cortex that encode the direction of arm movements irrespective of the position of the arm. Moreover, these cells fired before the movement was initiated. Similar cells have been found in other motor-related areas such as the supplementary motor area, primary motor cortex, and basal putamen (Alexander and Crutcher 1990). These authors reported that the cells encoded the direction of arm movement independently of loading, suggesting that the cells did not encode the low level dynamics and muscle movements required. The cells thus appear to encode a high level command to move the arm along a specific trajectory, which is independent of the dynamics and forces required to move it.

The synaptic connections w^2, w^3, and w^4 are known as Sigma-Pi synapses in that each of these synapses combines two afferent inputs in a multiplicative manner. For example, each w^2_{ijk} synaptic connection onto a postsynaptic state cell i combines an afferent input from a state cell j and another input from a motor cell k. The inputs from the state cell j and motor cell k are combined in a multiplicative manner. This means that both inputs need to be active (i.e. greater than zero) for the postsynaptic state cell i to be activated.

The model architecture does not include an explicit dynamical model of the motor plant. However, the model should be able to learn the dynamics of the motor plant during training.

2.1 Learning to Perform Low-Level Motor Primitives

The state cells operate as a *continuous attractor neural network*, which can use its recurrent connections to maintain the firing of its cells in the absence of any external input (Amari 1977; Taylor 1999). The activation h_i^{S} of state cell i is governed by

$$\tau \frac{dh_i^{\mathrm{S}}(t)}{dt} = -h_i^{\mathrm{S}}(t) + \frac{\phi_0}{C^{\mathrm{S}}} \sum_j (w_{ij}^1 - w^{\mathrm{INH}}) r_j^{\mathrm{S}}(t) + e_i + \frac{\phi_1}{C^{\mathrm{S\times M}}} \sum_{j,k} w_{ijk}^2 r_j^{\mathrm{S}} r_k^{\mathrm{M}} \quad (1)$$

where r_j^{S} is the firing rate of state cell j, w_{ij}^1 is the excitatory synaptic weight from state cell j to state cell i, w^{INH} is a constant modelling the effect of inhibitory interneurons between state cells, r_k^{M} is the firing rate of motor cell k, w_{ijk}^2 is the excitatory synaptic weight from state cell j and motor cell k to state cell i, τ is the time constant of the system, and e_i represents an external input to state cell i that may be visual or proprioceptive. The parameters ϕ_0 and ϕ_1 are hand-tuned constants, which are, respectively, scaled by the numbers of w^1 and w^2 synaptic connections C^{S} and $C^{\mathrm{S\times M}}$.

The firing rate r_i^{S} of state cell i is set by the sigmoid activation function

$$r_i^{\mathrm{S}}(t) = \frac{1}{1 + e^{-2\beta(h_i^{\mathrm{S}}(t) - \alpha)}}, \quad (2)$$

where α and β are the sigmoid threshold and slope, respectively.

The motor cells represent the current motor output. The activation h_i^{M} of motor cell i is governed by

$$\tau \frac{dh_i^{\mathrm{M}}(t)}{dt} = -h_i^{\mathrm{M}}(t) + t_i + \frac{\phi_2}{C^{\mathrm{S\times MS}}} \sum_{j,k} w_{ijk}^3 r_j^{\mathrm{S}} r_k^{\mathrm{MS}} \quad (3)$$

where r_j^{S} is the firing rate of state cell j, r_k^{MS} is the firing rate of movement selector cell k, w_{ijk}^3 is the excitatory synaptic weight from state cell j and movement selector cell k to motor cell i, and t_i is the motor signal used during training. The last term of the equation driving the motor activity combines inputs from state cells r_j^{S} and movement selector cells r_k^{MS}. The parameter ϕ_2 is a hand-tuned constant, which is scaled by the number of w^3 synaptic connections $C^{\mathrm{S\times MS}}$.

The firing rate r_i^{M} of motor cell i is determined using the sigmoid activation function

$$r_i^{\mathrm{M}}(t) = \frac{1}{1 + e^{-2\beta(h_i^{\mathrm{M}}(t) - \alpha)}}. \quad (4)$$

During the learning phase, inputs t_i drive the firing of motor cells, which causes the agent to proceed through a set of postural states. This teaching signal implies a form of supervised learning in the network. It remains a matter of speculation where exactly the motor training signals t_i might come from in the brain. Simultaneously, the state cells are directly driven by the external visual or proprioceptive input e_i. During this, the synaptic connections are modified according to the learning rules described next. All of the synaptic learning rules are *local* in the sense that the biological factors used to modify a synaptic weight are related to either the pre- or post-synaptic neurons. This is an important property for biological plausibility.

The recurrent synapses w^1_{ij} are updated according to the Hebb rule

$$\delta w^1_{ij} = k^1 r^{\mathrm{S}}_i r^{\mathrm{S}}_j. \tag{5}$$

Learning rule (5) enables the state cells to operate as a continuous attractor network.

The w^2_{ijk} synapses are updated according to

$$\delta w^2_{ijk} = k^2 r^{\mathrm{S}}_i \overline{r}^{\mathrm{S}}_j \overline{r}^{\mathrm{M}}_k, \tag{6}$$

where $\overline{r}^{\mathrm{S}}_j$ is a memory trace of the firing r^{S}_j and $\overline{r}^{\mathrm{M}}_k$ is a memory trace of the firing of r^{M}_k. The trace value $\overline{r}$ of the firing rate r of a cell is calculated according to

$$\overline{r}(t + \delta t) = (1 - \eta) r(t + \delta t) + \eta \overline{r}(t) \tag{7}$$

where η is a parameter in the interval [0,1] which determines the relative contributions of the current firing and the previous trace. Learning rule (6) allows recent past activity within the state cells and motor cells to be associated with the current activity of the state cells.

The ability of the network to update its internal representation of the state of the agent directly from the motor signal is known as *path integration*. That is, the network continuously integrates the motor velocity signal through time in order to compute the continuous change in the agent's state, and thereby tracks the changing state of the agent through time. This can be done in the absence of external visual signals or other kinds of sensory cue that might provide absolute positional information.

The w^3_{ijk} synapses are updated according to

$$\delta w^3_{ijk} = k^3 r^{\mathrm{M}}_i r^{\mathrm{S}}_j r^{\mathrm{MS}}_k. \tag{8}$$

Learning rule (8) associates the combined activity of the state cells and movement selector cells with the current motor cell activity.

As the agent performs a motor primitive during learning, the state cells are driven by the visual or proprioceptive inputs e_i, the motor cells are driven by the training signal t_i, and the synaptic weights w^1_{ij}, w^2_{ijk} and w^3_{ijk} are updated according to the learning rules described above. After learning, the motor primitive is performed by activating only the relevant movement selector cells.

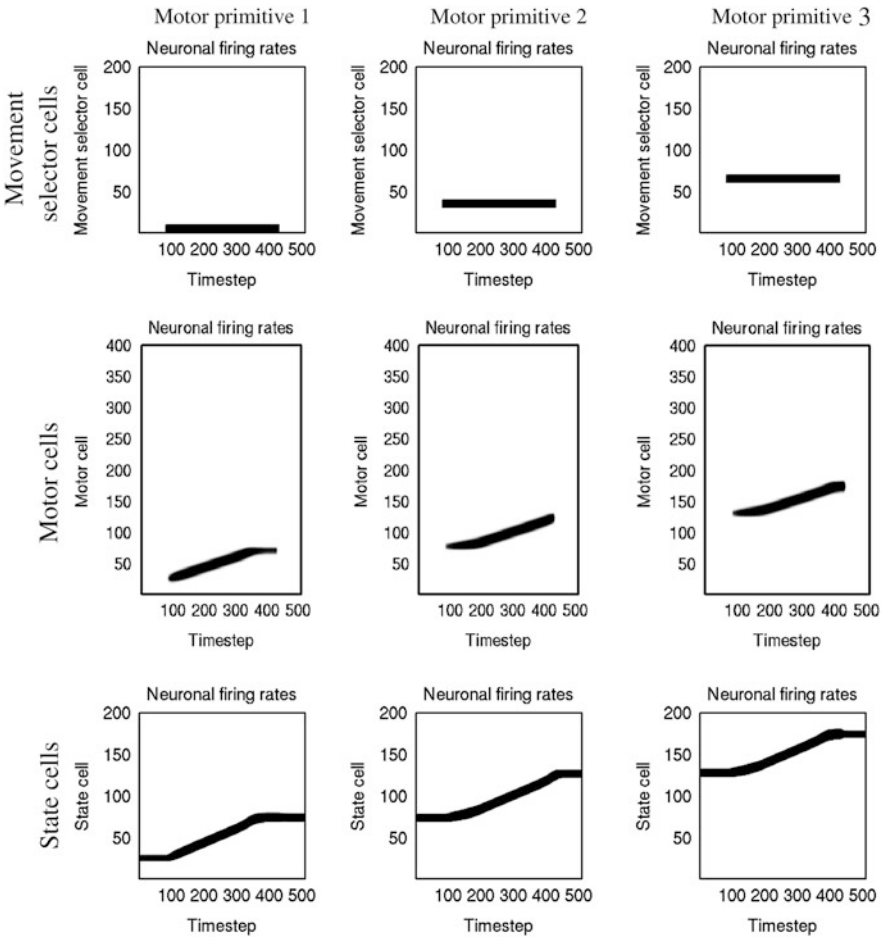

Fig. 3 The network performing motor primitives 1, 2, and 3, which encode small movements of the state of the agent in the positive x direction. Reproduced with permission from Stringer and Rolls (2007)

Numerical simulations showing the performance of the model after learning six low level motor primitives are presented in Figs. 3 and 4. Figure 3 shows motor primitives 1, 2, and 3, which encode small movements of the state in the positive x direction. Each column shows the network performing one of the motor primitives. Within each column, we show the firing rate profiles within the movement selector network, motor network and state network through time as the agent performs the learned motor primitive in the absence of the motor training signals t_i and external (visual or proprioceptive) state inputs e_i. In all plots, high cell firing rates are indicated in black. Figure 4 shows the network performing motor primitives 4, 5, and 6 in the negative x direction.

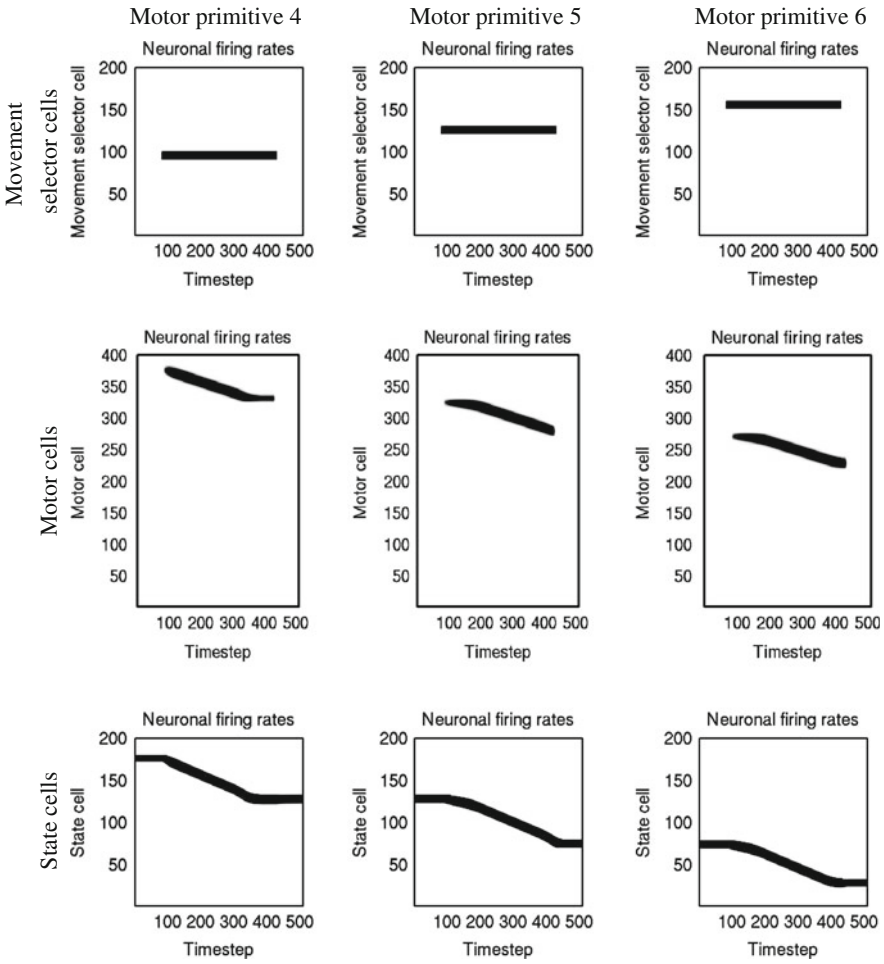

Fig. 4 The network performing motor primitives 4, 5, and 6, which encode small movements of the state of the agent in the negative x direction. Reproduced with permission from Stringer and Rolls (2007)

Of course, the results shown in Figs. 3 and 4 are very elementary and idealised examples of motor primitives. In principle, individual motor primitives could each trace an arbitrary path through the state space of the agent. Moreover, the motor primitives could overlap with each other as they traverse through the state space.

2.2 Learning to Perform High-Level Motor Programs

The movement selector cells are driven through synapses w^4 by combined inputs from state cells and high level movement selector cells. The high level movement

selector cells are able to learn to drive a motor program which consists of a sequence of motor primitives. The execution of each motor primitive in the sequence is gated by the inputs from the state cells.

The equation governing the activation of the movement selector cells is

$$\tau \frac{dh_i^{\mathrm{MS}}(t)}{dt} = -h_i^{\mathrm{MS}}(t) + t_i^{\mathrm{MS}} + \frac{\phi_3}{C^{\mathrm{S\times HMS}}} \sum_{j,k} w_{ijk}^4 r_j^{\mathrm{S}} r_k^{\mathrm{HMS}} \tag{9}$$

where t_i^{MS} is the training signal present during learning for each movement selector cell i. The last term is the input from couplings of the state cells and high level movement selector cells, where r_j^{S} is the firing rate of state cell j, r_k^{HMS} is the firing rate of high level movement selector cell k, and w_{ijk}^4 is the corresponding strength of the connection from these cells. The parameter ϕ_3 is a hand-tuned constant, which is scaled by the number of w^4 synaptic connections $C^{\mathrm{S\times HMS}}$.

The firing rate r_i^{MS} of movement selector cell i is given by the sigmoid activation function

$$r_i^{\mathrm{MS}}(t) = \frac{1}{1 + e^{-2\beta(h_i^{\mathrm{MS}}(t) - \alpha)}}, \tag{10}$$

where α and β are the sigmoid threshold and slope, respectively.

During learning, the synaptic weights w_{ijk}^4 are updated according to

$$\delta w_{ijk}^4 = k^4 r_i^{\mathrm{MS}} r_j^{\mathrm{S}} r_k^{\mathrm{HMS}}. \tag{11}$$

Learning rule (11) associates the current firing patterns within the high level movement selector cells and state cells with the subset of movement selector cells representing the current motor primitive.

As the agent learns a new high level motor program, the movement selector cells are driven by the training signal t_i^{MS} to perform the required motor primitives in the correct temporal order. However, as with the motor training signal t_i, it is not clear where the movement selector training signals t_i^{MS} might come from in the brain. At the same time, a new subset of high level movement selector cells is activated. These high level movement selector cells learn to drive the movement selector cells, which were activated during training, through the w_{ijk}^4 connections. After training, activation of the relevant high level movement selector cells will cause the network to perform the high level motor program in the absence of any training signals t_i^{MS}.

Figure 5 shows the performance of the network after it has learned two different high level motor programs. On the left are results for the first high level motor program. Activating high level movement selector cells 1–10 drives the network through the high level motor program composed of a temporal sequence of three motor primitives 1, 2, and 3.

The high level motor program is performed without the training signal t_i^{MS} driving the movement selector cells which represent the selection of the motor primitives, without the motor training signal t_i, and without the external (visual or

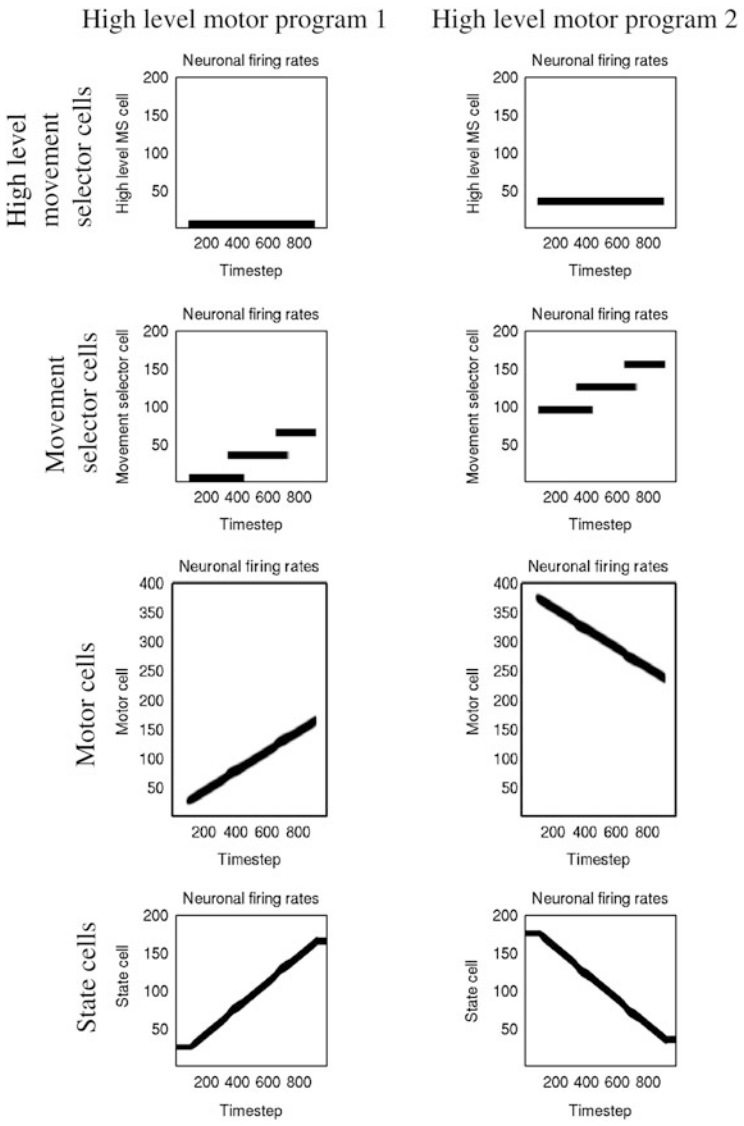

Fig. 5 The performance of the network after learning two high level motor programs. On the left is the first high level motor program composed of a temporal sequence of motor primitives 1, 2, and 3. On the right is the second high level motor program composed of motor primitives 4, 5, and 6. Reproduced with permission from Stringer and Rolls (2007)

proprioceptive) state input e_i. On the right are results for the second high level motor program. Activating high level movement selector cells 31–40 drives the network through the high level motor program composed of a temporal sequence of three motor primitives 4, 5, and 6.

Further experiments have shown that the network can learn high level motor programs in which a number of motor primitives are performed in parallel. Moreover, high level movement selector cells can learn to be associated with arbitrary subsets of movement selector cells that encode the motor primitives. Each motor primitive is executed automatically when the agent is in the appropriate part of the state space. Although, individual motor primitives may depend on a number of different kinds of states simultaneously. For example, the movement of an arm might depend on the positions of a number of obstacles in the environment. In this case, high level movement selector cells can encode any sequence of motor primitives where the primitives are physically consistent with each other along a defined space-time trajectory. The model can thus learn any physically realizable path through the state space, with complex sequences of motor primitives blended together through time.

Two limitations of the model developed by Stringer and Rolls (2007) have been recently addressed as follows.

Firstly, the synaptic connections w^2, w^3, and w^4 are Sigma-Pi connections, which combine inputs from presynaptic cells in a multiplicative manner. This may not be biologically plausible. However, Stringer and Rolls (2006) have shown that the need for Sigma-Pi connections can be eliminated by introducing additional competitive layers of cells, which self-organize to represent combinations of the Sigma-Pi inputs such as state and velocity.

Secondly, the speed of path integration within the layer of state cells, which is driven by the w^2 connections, does not completely self-organize during learning. That is, the network could learn which direction to drive the state in, but could not learn to do this at the correct speed that the network was trained on in the light. This is because the constant ϕ_1 in (1) needed to be tuned by hand to set the speed of path integration. However, this problem has now been addressed by Walters and Stringer (2010), who have shown how the speed of path integration may be learned by utilising natural time intervals present in the brain such as neuronal time constants or axonal conduction delays. Specifically, if such a network incorporates long neuronal time constants or long axonal delays of the order of 50–100 ms, then the network is able to learn explicit associations over these finite time intervals between the velocity signal and a change in the state.

The model presented above in Fig. 2 may be compared with models of hierarchical reinforcement learning (HRL) (Barto and Mahadevan 2003; Botvinick et al. 2009; Dietterich 2000; Parr and Russell 1998; Sutton et al. 1999). HRL models are able to reduce the training time for task domains with a large set of possible states and actions by incorporating a hierarchy of "abstract actions", each defined by a particular mapping from states to actions. Such hierarchies of abstract actions in HRL models can be compared to the different levels of the motor hierarchy in the model developed by Stringer and Rolls (2007) and shown in Fig. 2. A major aim of Stringer and Rolls (2007) was to develop a model of hierarchical behaviour with a biologically plausible neural network architecture, which relied on associative Hebbian learning rules at the synaptic connections. These kinds of learning rules are more biologically plausible than the error correction learning rules typically

found in HRL models. However, a major weakness of the model of Stringer and Rolls (2007) is that these authors did not explain where the training signals t_i and t_i^{MS} might come from in the brain. These signals are used to guide the learning of motor primitives and higher level motor programmes in the model. Nevertheless, HRL models may suffer from a possibly corresponding weakness. Botvinick et al. (2009) recognise that an important challenge for HRL modellers is to explain how HRL models might discover useful action subgoals, which need to be learned and incorporated into the hierarchy of abstract actions, in an automatic and efficient way.

3 The Modulation of High-Level Motor Programs by Context Leads to Novel Movement Sequences

A key challenge in developing neural network models of animal behaviour is to explain how novel sequences of actions can be generated. An influential paper by Lashley (1951) referred to this as *the problem of serial order in behaviour*. One potential mechanism that might be involved is that individual motor primitives are learned in a way that is dependent on context. In this case, a high level motor program composed of a sequence of such motor primitives can automatically adapt to the environmental context resulting in a novel movement sequence.

Stringer and Rolls (2007) showed that the network architecture described in Sect. 2 may be augmented to allow the motor primitives to be modulated by environmental context. An example of environmental context would be the position of an obstacle when reaching to a target location, which might require a different sequence of motor activity to accomplish the task. Figure 6 shows the augmented network architecture. There is an additional layer of context cells that send projections to the motor cells through the w^3 synapses. The context is thus able to modulate the activation of the motor cells through these connections. Equation (3) governing the activation of the motor cells is modified as follows

$$\tau \frac{dh_i^{\mathrm{M}}(t)}{dt} = -h_i^{\mathrm{M}}(t) + t_i + \frac{\phi_2}{C^{\mathrm{S\times MS\times C}}} \sum_{j,k,l} w_{ijkl}^3 r_j^{\mathrm{S}} r_k^{\mathrm{MS}} r_l^{\mathrm{C}} \tag{12}$$

where r_j^{S} is the firing rate of state cell j, r_k^{MS} is the firing rate of movement selector cell k, and r_l^{C} is the firing rate of a context cell l representing some environmental context. w_{ijkl}^3 is the synaptic weight to motor cell i from a combination of state cell j, movement selector cell k and context cell l. The parameter ϕ_2 is a hand-tuned constant, which is scaled by the number of w^3 synaptic connections $C^{\mathrm{S\times MS\times C}}$.

During learning the synaptic weights w_{ijkl}^3 are updated according to

$$\delta w_{ijkl}^3 = k^3 r_i^{\mathrm{M}} r_j^{\mathrm{S}} r_k^{\mathrm{MS}} r_l^{\mathrm{C}}. \tag{13}$$

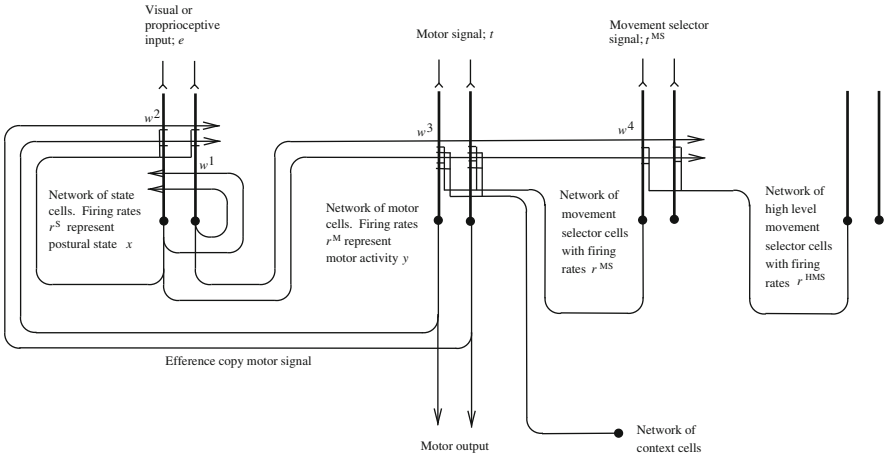

Fig. 6 Network architecture incorporating a postural or environmental context, which is able to modulate how motor programs are performed. Reproduced with permission from Stringer and Rolls (2007)

Each low level motor primitive must be learned for all possible contexts. However, the network needs to learn each high level motor program in only a single context. After learning, the network can automatically adapt its performance of a motor program to the other contexts.

Figure 7 shows the performance of the model with additional context cells. First, the network learned three low level motor primitives similar to those shown in Fig. 3 with the state of the agent moving in the positive x direction. However, this time each motor primitive was learned for two different contexts, each of which required different sequences of activity within the network of motor cells. Specifically, the three motor primitives used motor cells 1–200 in the first context, and motor cells 201–400 in the second context. Next, the network was trained on a high level motor program consisting of a temporal sequence of the three low level motor primitives. However, the network was only trained on the motor program in the first context. After training, the network was tested to see if it could generalise its performance of the motor program across both contexts. The left and right columns in Fig. 7 show the network performing the high level motor program in the first and second contexts, respectively. In each case, the motor program is initiated by activating the relevant subset of high level movement selector cells 1–10. However, the motor activity occurred in motor cells 1–200 in the first context, and motor cells 201–400 for the second context. Therefore, the motor activity depended correctly on the context. Thus, the model was able to generalise its performance across both contexts even though the motor program had only been trained in the first context. Furthermore, the performance of the motor program in the second context gave rise to a new sequence of motor activity to accomplish the task that had never been performed before.

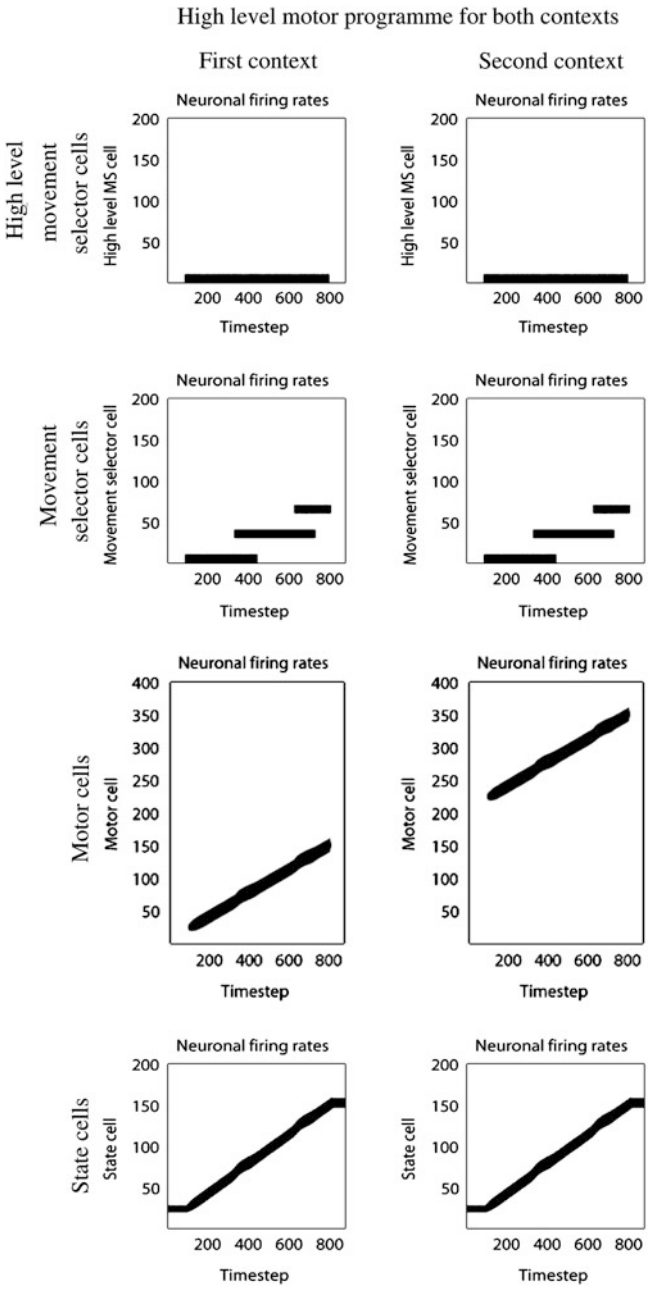

Fig. 7 Performance of the network architecture incorporating additional cells that represent an environmental context. On the *left* is the high level motor program for the first context, and on the *right* is the high level motor program for the second context. Reproduced with permission from Stringer and Rolls (2007)

The ability of the model to generate novel movement sequences demonstrates how hierarchical motor systems incorporating context-dependent motor primitives could shed some light on the problem of serial order in behaviour (Lashley 1951).

4 Learning Movement Sequences with a Delayed Reward Signal

A major issue in reinforcement learning is how an agent can learn a correct sequence of actions if the reward for correct performance is delayed until the sequence has been completed. A leading computational approach to solving this problem is *temporal difference* learning (Suri and Schultz 1998; Sutton 1988; Sutton and Barto 1998). However, the general form of neural network architecture underpinning this approach suffers from a lack of biological plausibility because the synaptic learning utilises an error between predictions of future reward at successive times. It is not clear how such an error signal could be computed and used for synaptic weight modification in the brain.

Stringer et al. (2007) developed an alternative approach that used a hierarchical motor system to learn action sequences with delayed rewards. The task that their model addressed was to learn a particular desired sequence of motor primitives using a delayed reward signal that indicated the success of the model on the current trial. For example, the sequence of motor primitives might be those required to bring a hand to a target location. During training, the network made many random attempts at performing the desired sequence of motor primitives. During this, the learning of synaptic connections within the network was guided by a reward signal that reflected whether the network had performed the desired sequence correctly. If the sequence was performed correctly, then the reward signal was set to 1 and learning took place. In particular, the model used simple associative learning when a reward signal was received for correct completion of an action sequence during training. The effect of training was to associate the currently active high level movement selector cells with all of the movement selector cells representing the motor primitives that had been active during a successful completion of the desired motor sequence. Of course, the reward signal only became available at the end of the task, when it was possible to assess whether the network had performed the correct sequence of motor primitives or not. Nevertheless, the network architecture was able to cope with this delayed reward signal during training. After training, the model was able to perform the action sequence correctly. That is, when the relevant high level movement selector cells are activated, the correct subset of motor primitives are performed. The motor primitives are performed in the right temporal order because their activation is modulated by the state of the agent.

The architecture of the network developed by Stringer et al. (2007) is shown in Fig. 8. The architecture is similar to the model described in Sect. 2 and shown in Fig. 2. However, there are two key modifications in the new model described in this section.

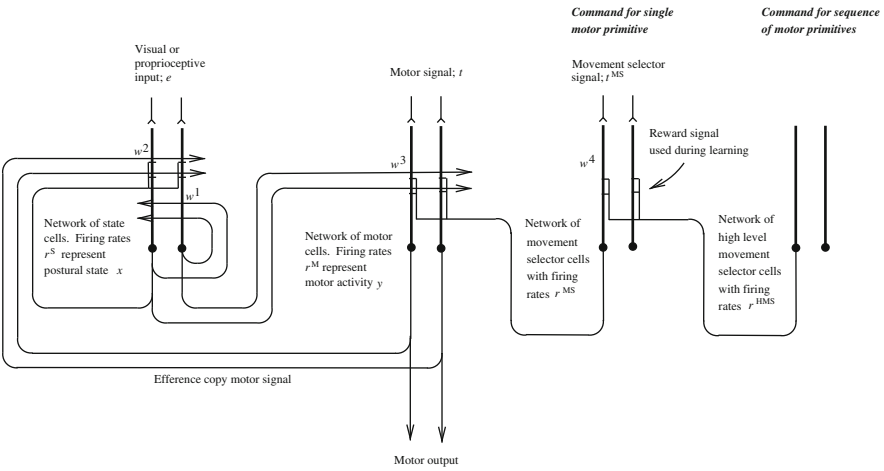

Fig. 8 Network architecture that can learn movement sequences with a delayed reward signal. Reproduced with permission from Stringer et al. (2007)

Firstly, it is shown in Fig. 8 that the w^4 synaptic connections onto the movement selector cells originate purely from the high level movement selector cells. That is, the w^4 connections are not modulated by additional connections from the state cells. Consequently, the high level movement selector cells simultaneously activate, for the entire duration of the motor program, all those movement selector cells r^{MS} that are required during any part of the motor program. These active movement selector cells then send constant driving signals to all of the motor primitives needed to perform the entire movement sequence. However, the motor primitives are still performed in the correct temporal order. This is because the each motor primitive is only performed when the agent is in the correct state because the w^3 connections from the movement selector cells to the motor cells are gated by projections from the state cells.

Secondly, the architecture shown in Fig. 8 incorporates a new reward signal that is used to guide learning at the w^4 synaptic connections from the high level movement selector cells to the movement selector cells. The reward signal is received during training each time the network has successfully completed the desired motor program consisting of a particular temporal sequence of motor primitives. The reward signal strengthens the connections from the relevant subset of high level movement selector cells to all of the active movement selector cells that were responsible for stimulating the correct sequence of motor primitives. In the primate brain, neurons that represent the reward value of environmental states have been found in the orbitofrontal cortex (Rolls 2004; Schultz et al. 2000).

The equations governing the dynamics of the new model are the same as those described above in Sect. 2 except for the two Eqs. (9) and (11) that govern the behaviour of the movement selector cells and modification of the w^4 synaptic weights.

The activation of the movement selector cells is now governed by

$$\tau \frac{dh_i^{\mathrm{MS}}(t)}{dt} = -h_i^{\mathrm{MS}}(t) + t_i^{\mathrm{MS}} + \frac{\phi_3}{C^{\mathrm{HMS}}} \sum_j w_{ij}^4 r_j^{\mathrm{HMS}}. \tag{14}$$

The last term is the input from the high level movement selector cells, where r_j^{HMS} is the firing rate of high level movement selector cell j, and w_{ij}^4 is the corresponding strength of the connection from these cells. The parameter ϕ_3 is a hand-tuned constant, which is scaled by the number of w^4 synaptic connections C^{HMS}.

During training, each time the network completes the correct temporal sequence of motor primitives it receives a reward. The w_{ij}^4 synaptic weights are updated according to the associative (Hebb) rule

$$\delta w_{ij}^4 = k^4 r_i^{\mathrm{MS}} r_j^{\mathrm{HMS}} r^{\mathrm{R}} \tag{15}$$

where r^{R} is 1 if a reward is obtained and 0 if a reward is not obtained.

During training, the high level command to perform a desired motor program is represented by activating a subset of high level movement selector cells. The network then makes repeated attempts at performing the correct sequence of motor primitives needed for the motor program. For each attempt, a random subset of movement selector cells representing different motor primitives is activated by the training signals t_i^{MS}. Each activated motor primitive will be performed only within its relevant part of the state space. If the agent performs the correct temporal sequence of motor primitives to complete the desired motor program, then a reward signal is received and the w^4 weights are updated according to (15). This associates the currently active subset of high level movement selector cells with all of the movement selector cells that represent motor primitives needed at some point by the motor program. After training, the subset of high level movement selector cells is able to drive the network through the correct sequence of motor primitives. Even though the high level movement selector cells simultaneously drive all of the movement selector cells that are needed at some point during the entire motor program, each motor primitive is only performed during its relevant part of the state space because the w^3 connections from the movement selector cells to the motor cells are gated by inputs from the state cells. Thus, the motor primitives are performed in the correct temporal order.

In numerical simulations, the network was first taught the same six low level motor primitives as shown in Figs. 3 and 4. Next, the network was trained to perform two high level motor programs. The first motor program involved the state of the agent moving in the positive x direction by performing motor primitives 1, 2, and 3 in that order. The second motor program involved the state of the agent moving in the negative x direction by performing motor primitives 4, 5, and 6 in that order. On each learning trial, 3 of the 6 motor primitives were randomly selected, and the related movement selector cells were then activated by applying a constant training signal t_i^{MS} throughout the trial. Each time the network performed one of the two motor programs successfully, it received a reward signal and the w^4 weights were updated according to (15). Figure 9 shows the performance of the

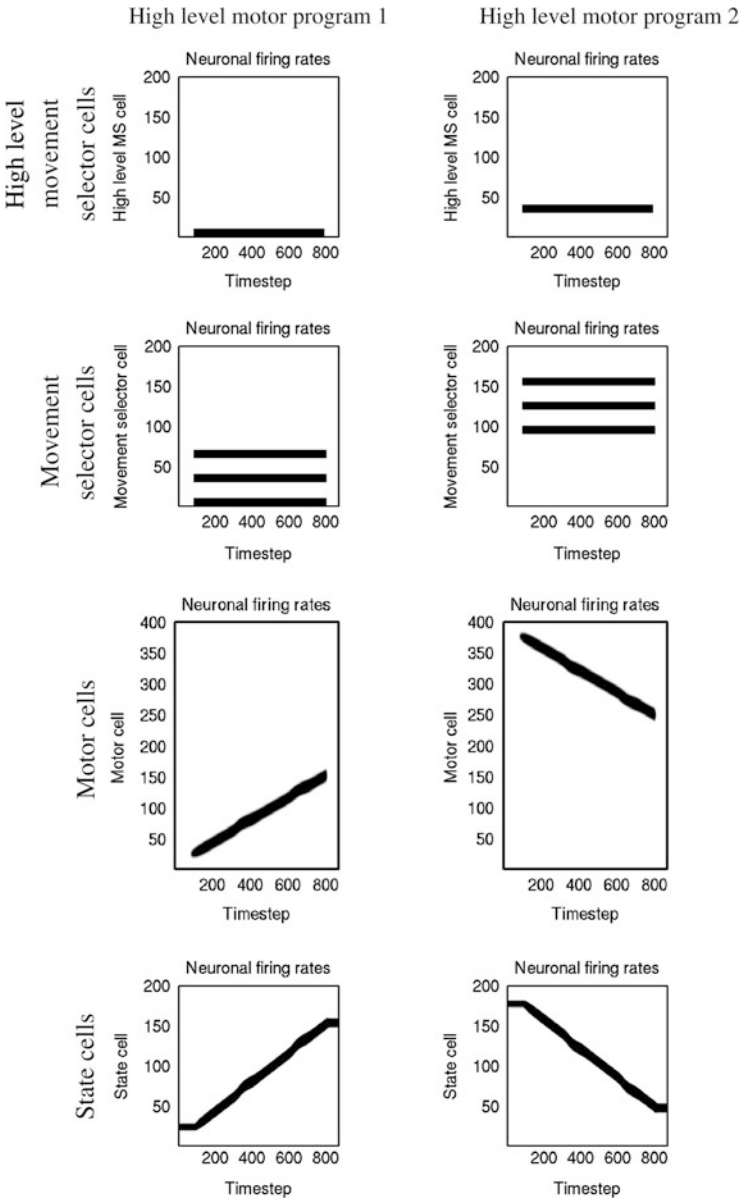

Fig. 9 The performance of the network after learning two high level motor programs with a delayed reward signal. On the *left* are results for the first high level motor program composed of a temporal sequence of the motor primitives 1, 2, and 3. On the *right* are results for the second high level motor program composed of the motor primitives 4, 5, and 6. Reproduced with permission from Stringer et al. (2007)

network after it has learned the two different high level motor programs. It can be seen that the network can be driven through either of the two motor programs by activating the relevant subset of high level movement selector cells. Each motor program simultaneously co-activates all of the required movement selector cells throughout the performance of the motor program. However, the motor primitives are still performed in the correct order because the w^3 connections to the motor cells from the movement selector cells are gated by inputs from the state cells. The ability of the network to perform the two different high level motor programs after training demonstrates how a hierarchical motor system can learn arbitary sequences of motor primitives using a delayed reward signal.

5 Future Challenges in Modelling Hierarchical Motor Function

In this chapter, we have described a series of generic models of hierarchical motor function in the brain. These models could, in principle, implement a variety of different kinds of motor activity, such as reaching for an object and picking it up. Depending on the nature of the task, different neural substrates in the brain would be involved. The models were specifically designed to utilise biologically plausible neural network architectures with local associative learning rules at the synaptic connections. These models are therefore a useful step towards understanding how hierarchical motor function might be implemented in the brain.

However, a major limitation of the models described above is the lack of explanation for the sources of the external signals, e_i, t_i and t_i^{MS}, that are used to drive the activation of cells in the models during training. For example, during learning, the state cells were driven by the visual or proprioceptive signal e_i, the motor cells were driven by the training signal t_i, and the movement selector cells were driven by t_i^{MS}. In future work, we will address more closely how these signals might be generated in the brain.

Within the brain, there is a rich variety of neuronal representations of the state of the animal and its environment. These are important for guiding motor function and behaviour. Different kinds of state representation are localised in specific brain areas and are relevant to specific types of motor task.

As an example, let us consider the task of reaching to a visual target object. A number of areas of the posterior parietal cortex (PPC), including the Parietal Reach Region (PRR) and adjacent Area 5, as well as premotor areas, have been found to contain visually-driven neurons that respond to the location of a target object in a hand-centred frame of reference (Bremner and Andersen 2012; Buneo and Andersen 2006; Buneo et al. 2002; Chang and Snyder 2010; Graziano et al. 1997; Graziano 1999). These neurons provide a direct representation of the distance between the hand and the location of the target object in a common reference frame, which is useful for generating the appropriate motor command to reach for a visual target.

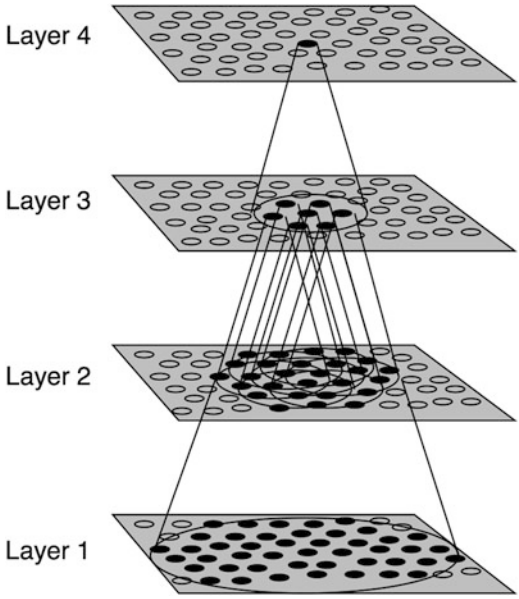

Fig. 10 The VisNet architecture. The network is comprised of a hierarchical series of four competitive networks with feed-forward synaptic connections between successive layers. There is convergence of connections through successive layers, which provides the fourth-layer neurons with information from across the entire input retina

Such hand-centred representations are thought to play a role in visually guiding reaching. The neurons that support these representations in the posterior parietal cortex may be regarded as one potential class of state cells that could be incorporated into the models presented above. In this case, how might a visual signal e_i guide the development of the hand-centred response properties of these PPC cells during training in the light?

Recently, Galeazzi et al. (2013) have proposed a biologically plausible model of how neurons in the PPC might develop their hand-centred responses to visual targets by visually guided learning. The model uses visually guided learning in a biologically plausible neural network model, VisNet, of visual processing in the primate brain (Rolls and Milward 2000; Wallis and Rolls 1997). The VisNet model consists of a hierarchical architecture composed of four layers of competitive networks as shown in Fig. 10. In VisNet, learning is unsupervised, that is, there is no external teaching signal to tell the output cells how they should respond. The feed-forward synaptic weights between layers are updated by a local associative *trace learning* rule (Foldiak 1991; Rolls 1992). The trace rule incorporates a memory trace of recent neuronal activity into the postsynaptic term. This encourages the postsynaptic cell to learn to respond to input patterns that tend to occur close together in time.

The form of trace learning rule used by Galeazzi et al. (2013) to update the synaptic weights at each timestep τ is

$$\delta w_{ij} = k\overline{r}_i^{\tau-1} r_j^{\tau} \tag{16}$$

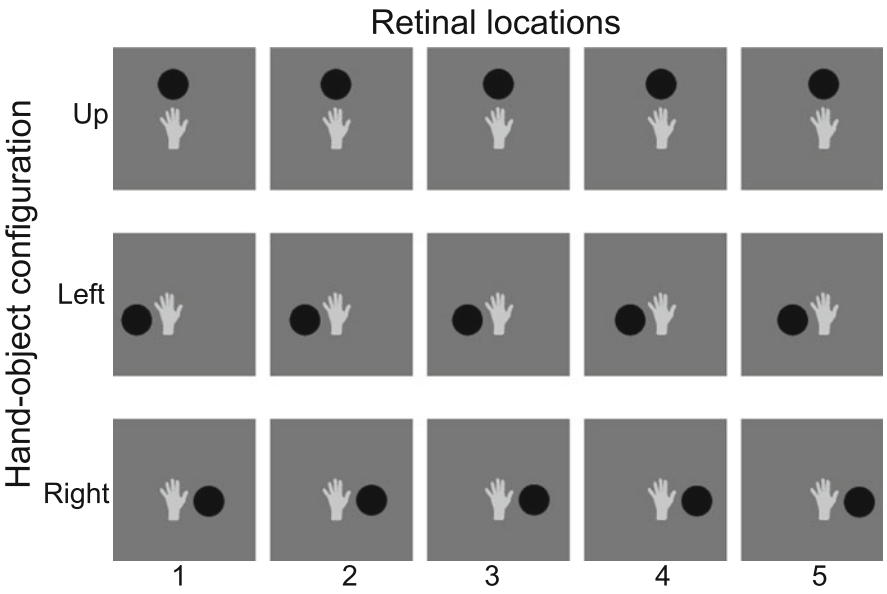

Fig. 11 Three image sequences presented to VisNet in the simulation study of Galeazzi et al. (2013). Each row shows a separate image sequence of a hand and a circular object shifting across the retina together due to (micro)saccades. For each image sequence, the hand and object remain in a fixed spatial configuration that does not change. However, each of the three image sequences has a unique spatial configuration of the hand and object as follows. *Top row*: visual object is shown in the "Up" location with respect to the hand. *Middle row*: visual object is shown in the "Left" location with respect to the hand. *Bottom row*: visual object is shown in the "Right" location with respect to the hand

where w_{ij} is the synaptic weight from presynaptic cell j to postsynaptic cell i. The trace term $\overline{r}_i^{\tau}$ is updated according to

$$\overline{r}_i^{\tau} = (1 - \eta) r_i^{\tau} + \eta \overline{r}_i^{\tau - 1}. \tag{17}$$

The parameter η may be set in the interval $[0, 1]$. Typically, η is set to 0.8.

Galeazzi et al. (2013) hypothesised that during learning, the visual system is exposed to image sequences similar to those shown in Fig. 11. Each image sequence results from the eyes performing (micro)saccades around a visual scene containing the hand and background object(s) in a fixed spatial arrangement. In this case, images of the target object in a particular hand-centred position, although in different retinal positions, will tend to be seen clustered together in time. A trace learning rule may then cause individual output cells to learn to respond to images of a particular spatial arrangement of the hand and the object across different retinal positions. The feasibility of this proposal has been demonstrated in computer simulations carried out by Galeazzi et al. (2013). After training VisNet on the image sequences shown in Fig. 11, cells in the output (fourth) layer of the network had

learned to respond selectively to the location of the visual target in a hand-centred reference frame.

The simulation studies by Galeazzi et al. (2013) have shown one biologically plausible way in which neurons in the brain might learn to represent the location of a visual target with respect to the hand. This particular kind of state information, which is represented in the posterior parietal cortex, is thought to be important for the task of visually guided reaching.

It is perhaps more difficult to identify potential sources in the brain for the training signals, t_i and t_i^{MS}, which are used to guide the firing of the motor cells and movement selector cells in the models described above during training. The question is how can the supervised teaching signals used in these models be replaced with a form of unsupervised learning mechanism, which we assume must be present in the brain? Nevertheless, hierarchical reinforcement learning (HRL) modellers face a comparable challenge to explain how their models might be able to identify and learn behaviourally useful action subgoals in an unsupervised and efficient way (Botvinick et al. 2009).

References

Alexander, G., & Crutcher, M. (1990). Preparation for movement: neural representations of intended direction in three motor areas of the monkey. *Journal of Neurophysiology*, *64*(1), 133–150.

Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, *27*, 77–87.

Barto, A., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, *13*(4), 341–379.

Belanger, M., Drew, T., Provencher, J., Rossignol, S. (1996). A comparison of treadmill locomotion in adult cats before and after spinal transection. *Journal of Neurophysiology*, *76*, 471–491.

Bizzi, E., & Polit, A. (1979). Processes controlling visually evoked movements. *Neuropsychologia*, *17*, 203–213.

Botvinick, M., Niv, Y., Barto, A. (2009). Hierarchically organized behavior and its neural foundations: a reinforcement learning perspective. *Cognition*, *113*(3), 262–280.

Bremner, L., & Andersen, R. (2012). Coding of the reach vector in parietal area 5d. *Neuron, 75*, 342–351.

Buneo, C., & Andersen, R. (2006). The posterior parietal cortex: sensorimotor interface for the planning and online control of visually guided movements. *Neuropsychologia*, *44*(13), 2594–2606.

Buneo, C., & Andersen, R. (2012). Integration of target and hand position signals in the posterior parietal cortex: effects of workspace and hand vision. *Journal of Neurophysiology, 108*, 187–199.

Buneo, C., Jarvis, M., Batista, A., Andersen, R. (2002). Direct visuomotor transformations for reaching. *Nature*, *416*(6881), 632–636.

Chang, S., & Snyder, L. (2010). Idiosyncratic and systematic aspects of spatial representations in the macaque parietal cortex. *Proceedings of the National Academy of Sciences*, *107*(17), 7951.

Dietterich, T. (2000). Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of Artificial Intelligence Research*, *13*, 227–303.

Foldiak, P. (1991). Learning invariance from transformation sequences. *Neural Computation*, *3*, 193–199.

Galeazzi, J., Mender, B., Paredes, M., Tromans, J., Evans, B., Minini, L., & Stringer, S. (2013). A self-organizing model of the visual development of hand-centred representations. *PLOS ONE, 8*(6), e66272.

Gazzaniga, M., Ivry, R., Mangun, G. (1998). *Cognitive neuroscience: the biology of the mind*. New York: Norton & Company.

Georgopoulos, A. (1995). Motor cortex and cognitive processing. In M. Gazzaniga (Ed.), *The cognitive neurosciences* (pp. 507–517). Cambridge: MIT.

Ghez, C., & Krakauer, J. (2000). The organisation of movement. In E. R. Kandel, J. H. Schwartz, T. M. Jessell (Eds.), *Principles of neural science*, 4th edn. (Chap. 33, pp. 653–673). New York: McGraw-Hill.

Graziano, M., Hu, X., & Gross, C. (1997). Visuospatial properties of ventral premotor cortex. *Journal of Neurophysiology, 77*, 2268–2292.

Graziano, M. (1999). Where is my arm? The relative role of vision and proprioception in the neuronal representation of limb position. *Proceedings of the National Academy of Sciences, 96*, 10418–10421.

Hamilton, A., & Grafton, S. (2008). Action outcomes are represented in human inferior frontoparietal cortex. *Cerebral Cortex*, *18*, 1160–1168.

Hart, C., & Giszter, S. (2004). Modular premotor drives and unit bursts as primitives for frog motor behaviors. *The Journal of Neuroscience*, *24*(22), 5269–5282.

Hughlings Jackson, J. (1878). *Selected writings of John Hughlings Jackson 2, Edited by J. Taylor, 1932*. London: Hodder and Staughton.

Krakauer, J., & Ghez, C. (2000). Voluntary movement. In E. R. Kandel, J. H. Schwartz, T. M. Jessell (Eds.), *Principles of neural science*, 4th edn. (Chap. 38, pp. 756–781). New York: McGraw-Hill.

Lacquaniti, F., Terzuolo, C., Viviani, P. (1983). The law relating the kinematic and figural aspects of drawing movements. *Acta Psycholog*, *54*, 115–130.

Lashley, K. S. (1951). The problem of serial order in behaviour. In L. A. Jeffress (Ed.), *Cerebral mechanisms in behaviour* (pp. 112–136). New York: Wiley.

Laszlo, J. L. (1966). The performance of a single motor task with kinesthetic sense loss. *Quarterly Journal of Experimental Psychology*, *18*, 1–8.

Laszlo, J. L. (1967). Training of fast tapping with reduction of kinesthetic, tactile, visual, and auditory sensation. *Quarterly Journal of Experimental Psychology*, *19*, 344–349.

Miall, R. C., & Wolpert, D. M. (1996). Forward models for physiological motor control. *Neural Networks*, *9*, 1265–1279.

Parr, R., & Russell, S. (1998). Reinforcement learning with hierarchies of machines. In M. I. Jordan, M. J. Kearns, & S. A. Solla (Eds.), *Advances in neural information processing systems* (Vol. 10). Cambridge, Massachusetts: MIT Press.

Passingham, R. E. P. (1993). *The frontal lobes and voluntary action*. Oxford: Oxford University Press.

Pearson, K., & Gordon, J. (2000). Locomotion. In E. R. Kandel, J. H. Schwartz, T. M. Jessell (Eds.), *Principles of neural science*, 4th edn. (Chap. 37, pp. 737–755). New York: McGraw-Hill.

Polit, A., & Bizzi, E. (1978). Processes controlling arm movements in monkeys. *Science*, *201*, 1235–1237.

Rolls, E. (2004). The functions of the orbitofrontal cortex. *Brain and Cognition*, *55*(1), 11–29.

Rolls, E. T. (1992). Neurophysiological mechanisms underlying face processing within and beyond the temporal cortical visual areas. *Philosophical Transactions of the Royal Society*, *335*, 11–21.

Rolls, E. T., & Milward, T. (2000). A model of invariant object recognition in the visual system: learning rules, activation functions, lateral inhibition, and information-based performance measures. *Neural Computation*, *12*, 2547–2572.

Schmidt, R. A. (1987). *Motor control and learning: a behavioural emphasis*, 2nd edn. Champaign: Human Kinetics.

Schmidt, R. A. (1988). Motor and action perspectives on motor behaviour. In O. G. Meijer, & K. Roth (Eds.), *Complex motor behaviour: the motor-action controversy* (pp. 3–44). Amsterdam: Elsevier.

Schultz, W., Tremblay, L., Hollerman, J. (2000). Reward processing in primate orbitofrontal cortex and basal ganglia. *Cerebral Cortex*, *10*(3), 272–283.

Stringer, S. M., & Rolls, E. T. (2006). Self-organizing path integration using a linked continuous attractor and competitive network: path integration of head direction. *Network: Computation in Neural Systems*, *17*, 419–445.

Stringer, S. M., & Rolls, E. T. (2007). Hierarchical dynamical models of motor function. *Neurocomputing*, *70*, 975–990.

Stringer, S. M., Rolls, E. T., Taylor, P. (2007). Learning movement sequences with a delayed reward signal in a hierarchical model of motor function. *Neural Networks*, *20*, 172–181.

Stringer, S. M., Rolls, E. T., Trappenberg, T. P., De Araujo, I. E. T. (2003). Self-organizing continuous attractor networks and motor function. *Neural Networks*, *16*, 161–182.

Suri, R. E. & Schultz, W. (1998). Learning of sequential movements by neural network model with dopamine-like reinforcement signal. *Experimental Brain Research*, *121*, 350–354.

Sutton, R., Precup, D., Singh, S. (1999). Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, *112*(1), 181–211.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*, 9–44.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning*. Cambridge: MIT.

Taylor, J. G. (1999). Neural "bubble" dynamics in two dimensions: foundations. *Biological Cybernetics*, *80*, 393–409.

Wallis, G., & Rolls, E. T. (1997). Invariant face and object recognition in the visual system. *Progress in Neurobiology*, *51*, 167–194.

Walters, D. M., & Stringer, S. M. (2010). Path integration of head direction: updating a packet of neural activity at the correct speed using neuronal time constants. *Biological Cybernetics*, *103*, 21–41.

Wolpert, D., & Flanagan, J. (2001). Motor prediction. *Current Biology*, *18*, R729–R732.

Restoring Purpose in Behavior

Henry H. Yin

Abstract The dominant paradigm in the study of behavior today is the linear causation paradigm. This paradigm, inspired by classical physics, assumes that causes precede effects, that the behavior of organisms is caused by antecedent events inside and outside the organism, and that future states such as goals and purposes cannot possibly cause behavior. It is the basis of the general linear model in psychology and the input/output analysis in neuroscience. But linear causation does not apply to any control system with negative feedback. Here I shall argue that organisms are collections of such negative feedback systems that control their perceptual inputs. The chief difference between the behavior of living organisms and that of non-living things is the presence of control. Rather than the effect of some prior cause, behavior is the observable manifestation of control in teleological systems that act on the environment to make inputs match their internal reference values. The previous rejection of control theory in the behavioral sciences was largely based on a misunderstanding of the principles of negative feedback control. I discuss the types of behavioral control enabled by the hierarchical organization and the experimental method for testing the controlled variable.

1 Introduction

> The fifth way begins from the guidedness of things. For we observe that some things which lack knowledge, such as natural bodies, work towards an end. This is apparent from the fact that they always or most usually work in the same way and move towards what is best. From which it is clear that they reach their end not by chance or by intention. For those things which do not have knowledge do not tend to an end, except under the direction of someone

H.H. Yin (✉)
Centre for Cognitive Neuroscience, Department of Psychology and Neuroscience, Department of Neurobiology, Duke University, Box 91050, Durham, NC 27710, USA
e-mail: hy43@duke.edu

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9_14,

> who knows and understands: the arrow, e.g., is shot by the archer. There is therefore an intelligent personal being by whom everything in nature is ordered to this end, and this we call God.

In the above passage, the fifth argument for the existence of God, Aquinas argues that the "guidedness of things" is evidence for some omnipotent agent which directs everything in nature to the appropriate end. For Aquinas, knowledge of ends is needed for movement. Since "natural bodies" do not possess such knowledge, a human-like agent is needed to move them, as the archer shoots the arrow. Galileo and his followers, however, succeeded in explaining the movement of natural bodies, from balls on inclined planes to planetary motion, and the laws of physics have since demonstrated that neither a concept like purpose nor an all-knowing agent like God is necessary to explain how things move. But a gap remains. A pigeon released into the air does not fall like a rock of the same mass, as predicted by physical law. Its flying, and indeed the behavior of any organism, remains a puzzle.

We now know that, at the level of atoms and molecules, there is no fundamental distinction between living and non-living things. Many therefore assume that the physical laws must also suffice to explain the behavior of organisms, which are just collections of atoms like everything else in the universe. The rock falling can be explained by gravitational force; it has no knowledge of the future state of lying on the ground. It is manifestly a violation of the law of causation that a future state can cause anything to happen in the present. There is no place in physics for purpose.

Although Aquinas and modern scientists differ on the question of what causes movement, they all assume that there can only be one kind of "guidedness," for physical things and for living things. For Aquinas it is purpose. For modern scientists it is physical law. Following the example of physics, modern psychology attempted to purge all teleological concepts from its theories. Since Descartes introduced the paradigm of physics into psychology, different schools have looked for the cause of behavior in different places, either external to the organism, in some stimulus, or internal to the organism, in some executive homunculus issuing a command. Over the centuries, these approaches have evolved largely by changing their labels, but little has changed in the underlying assumption that some antecedent event prior to behavior is the cause of behavior. For example, observing someone walking, we may say that the behavior of "walking" is caused by contraction of his leg muscles, which is caused by the release of acetylcholine at the neuromuscular junction, which is caused by firing of alpha motor neurons, and so on, tracing the causal chain backwards until we reach the cause of walking. This view is deemed so self-evident that hardly anyone has ever questioned it.

The dominant approach in the study of behavior is the "input/output" approach. The experimenter manipulates input to the organism, in the form of stimuli, and measures output, in the form of behavior. The input is the independent variable, and the output the dependent variable. The goal is to identify a function that relates the input to the output. So far, however, all attempts to find such a function have failed, despite considerable progress in our understanding of the structure and physiology of the brain. It is clear to the engineer attempting to build a behaving robot that the

diagrams found in neuroscience textbooks do not actually produce viable behavior, even the behavior of a worm or of a fly.

Students of behavior have resorted to two major strategies. In neuroscience, researchers have often used "preparations" that yield high correlations between stimulus and response. For example, when Sherrington, a pioneer of the input/output analysis in neuroscience, used awake and intact animals as his experimental subjects, the behavior he observed was simply too variable and the relationship between brain stimulation and behavioral output too uncertain (Leyton and Sherrington 1917). Instead he often relied on the decerebrate preparation, which removes the influence of the brain to yield behavioral data that more closely approximate the predictions of the linear causation hypothesis (Sherrington 1906). Scientists like Sherrington insist that the organism yield a high correlation between the stimulus and the response, even if it involves removing the brain, keeping the animal anesthetized, or whatever it takes to arrive at a "preparation" that will satisfy the rigor of the physical sciences. In experimental psychology, on the other hand, the strategy is to move beyond the individual as the basic unit of analysis, by averaging across multiple individuals. The statistically average animal appears to show the stimulus-response correlation that satisfies the experimenter, even if the individual animal does not. Whereas neuroscience has focused on the "partial animal," so that whatever is left can "behave" according to the experimenter's a priori assumptions about what behavior should be, psychology studies the "average animal," or the "Gaussian animal," a creature not found in the woods.

Consequently, students of behavior today often share a set of implicit assumptions about the study of behavior. Above all, the experimenter must be able to manipulate input and measure output, so that some function can be found to describe the output in terms of the input. When behavior is variable, the low correlation between stimulus and response is attributed to noise, or changes in "conditions," "contexts," or whatever the experimenter cannot explain. In addition, because more subjects in a study means more statistical power, a study with many subjects is better than a study with a few subjects, single-subject designs or self-experimentation having been largely excluded from acceptable scientific practice. These assumptions all stem from the most basic assumption: that behavior is the effect of some antecedent cause. Something must act on the organism, or more exactly its nervous system, resulting in a chain of transformations within that ultimately leads to behavioral output. This something can come from within, in the form of reasons or homunculus, or it may come from without, in the form of stimuli and contexts.

2 The Calculation Problem

In 1935, Nicholai Bernstein pointed out a problem with the assumption that the output of the nervous system causes movement. Clearly some neural output leads to behavior. That much is seen. What is not seen, however, is additional forces acting on the muscles to generate what we observe as behavior. For example, in

opening a door, the stiffness of the door handle, the wind outside, the weight of the door, the oil in the hinge all these variables are independent of the neural activity, yet they contribute to the observed behavior. Bernstein realized that, for natural movements to be possible, there cannot be an unequivocal relationship between neural output and behavior, because the environmental disturbances are always present and unpredictable, and together with neural output they jointly determine the behavior observed (Bernstein 1967). Just as the movement of a car is not determined by the engine output alone, so the movement of the pedestrian is not a result of his neural activity alone. It does matter whether there is a hill or how strong the wind is, for the car or the pedestrian. Because disturbances can accumulate quickly, the more complex the movements, the less correspondence there is between neural output and actual behavior, as more disturbances are added to the neural outputs, at each turn, to produce behavior as observed. Consequently, the sequence of neural activity recorded while you walk home, if repeated, will not take you home a second time. As Bernstein explained: "A trained athletes consecutive running steps are as identical as coins of the same value, but this identity results, not from the brains ability to send absolutely identical motor impulses to the muscles, but only from the faultless work of sensory corrections. . . . even if the muscles received ten absolutely identical motor impulses in a row, there would be, in the very best case, 10 ugly steps, each one different from the others and with a result quite different from running" (Bernstein 1967 p. 180).

If behavior as observed is determined by both neural output and environmental disturbances, then how can it be reproduced when only one of these is generated by the brain? How do organisms generate output that is specifically calibrated to oppose unknown and unpredictable disturbances? It is easy to overlook this "calculation problem" because we are accustomed to observing living organisms "behave." What we observe is the achievement, in the Brunswikian sense (Hammond and Stewart 2001), but we do not see the hidden disturbances that also influence the behavior as achieved.

3 The Nature of Control

The calculation problem, that of calculating inverse kinematics and dynamics of behavior, is widely acknowledged, but there is no consensus on how it can be solved. One possible solution, popular today, is to predict exactly what the disturbances will be and to calculate exactly what is needed to overcome such disturbances (Franklin and Wolpert 2011; Shadmehr et al. 2010). Some believe that, given unlimited computational power, which is often attributed to the brain, the organism can predict exactly what is needed to overcome the disturbances. But these models are based on unwarranted assumptions and a misunderstanding of negative feedback systems. Consequently, much effort is devoted to the solution of the calculation problem using advanced mathematics. The "feedforward" solution to the calculation problem may appear simple, but in practice even the most powerful computers have yet to

succeed using this approach to solve the calculation problem. Calculating output requires an understanding of the basic laws of physics, which are not known to the organisms. To compute the right outputs, the brain needs information about the masses and moments of inertia of the arm segments, the properties of muscle contraction, the properties of nonlinear muscle springs, the variations in mechanical advantage as the joint angles change, the physics of dynamic movements, the trigonometry of spatial relationships, and the initial state of the effectors, among other things. And all these elaborate computations must be performed in real time (Powers 1978).

There is, however, a far simpler alternative that does not require the calculation of inverse kinematics and dynamics (Powers 1973b). For the calculation problem is not unique to the field of behavior. Engineers face it every time they try to design a machine that can resist unexpected disturbances. Although they repeatedly stumbled upon the solution, only in the twentieth century did they finally understand the properties of a negative feedback control system (Black 1934). In their terminology, the phenomenon of resistance to disturbances is called control, and the principles that achieve control without calculating inverse kinematics are known as classical control theory.

The key principle in classical control theory is closed loop negative feedback. Although for engineers negative feedback is simply a convenient way of solving practical problems, Rosenblueth and Wiener, the founders of cybernetics, realized that it has important implications for psychology and the life sciences. Unfortunately, because they were not familiar with how control systems actually operate, the cybernetic models were fatally flawed (Rosenblueth et al. 1943; Wiener 1948). They introduced the vocabulary of control theory, without explaining correctly how it could be applied to the study of behavior. Partly for these reasons, classical control theory was abandoned by students of behavior long before they even understood it. Had they studied it carefully, they would have discovered that the phenomenon of control creates insurmountable challenges for the standard paradigm in neuroscience and psychology.

The major implication of negative feedback control is that linear causation is always wrong in explaining the behavior of control systems. For example, the thermostat is a simple negative feedback control system. If the room temperature deviates from the set temperature, the thermostat will act to oppose the change. If we apply the standard input/output analysis to the thermostat, we will obtain data similar to what is shown in Fig. 1. The room temperature, the input to the thermostat, is relatively stable, whereas the output, measured by the utility bill, varies a great deal. It is impossible to find a function to describe the relationship between these two sets of values. In fact, the major theories in the history of psychology all fail when applied to the analysis of the humble thermostat, precisely because all such theories attempt to find the function that relates the input to the output. The major implication of classical control theory is that all such attempts are necessarily in vain.

The thermostat senses input in the form of room temperature; it compares this input with a reference signal specifying what the temperature should be; and it

Month	Input (temperature in Celsius)	Output (Utility bill in dollars)
Jan	20	150
Feb	20	120
Mar	20	110
Apr	20	70
May	21	60
Jun	22	200
Jul	25	170
Aug	25	140
Sep	25	130
Oct	21	80
Nov	19	100
Dec	19	150

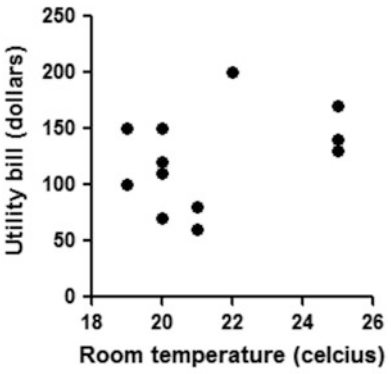

Fig. 1 Hypothetical data set from the input and output measurements of a thermostat

generates some error signal, the difference between current temperature and desired temperature, which is then converted into the output. The controlled variable is the room temperature, and a working thermostat will keep this variable as close to the set temperature as possible, despite fluctuations in the temperature outside. When the current temperature is the same as the desired temperature, there is no error, and the output is turned off. Because the output acts to keep the input close to the set temperature, there is no consistent relationship between the input and the output. Rather the output is correlated with the disturbances, i.e. outside fluctuations in temperature. The thermostat does not sense the disturbances directly. It only senses one input, current temperature, which is determined jointly by its own output and a variety of disturbances.

The thermostat could have used a "feedforward" approach popular in the motor control literature to predict disturbances to the temperature, calculating outputs needed to maintain the desired temperature, based on an inspection of the visual scene outside, the color of the leaves, the time of the year, and so on, and calculate the exact amount of output needed at any given moment. But even if it was equipped with a very powerful computer, it would not have been able to control room temperature.

Closed loop negative feedback control system has special properties not found in open loop systems. An example of an open loop system is the electric fan, which controls its output rather than its input. The thermostat, by contrast, controls its input—the sensed temperature in the room. It does not need to know anything about the disturbance to the controlled variable. The fluctuations in temperature could be caused by the fire in the bedroom or the snow on the roof. Whatever the sources, the thermostat only has to sense the variable being controlled, namely the current

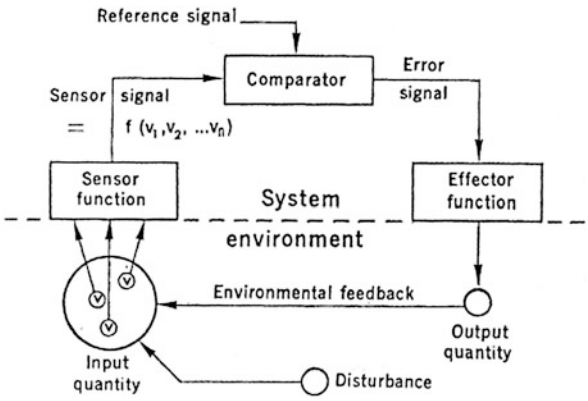

Fig. 2 The basic organization of a closed loop control system (from Powers 1973)

temperature at its sensor, to bring the input variable close to the value specified by the reference signal.

Although teleology has become a dirty word, the thermostat is a teleological machine. If you set it at 25 degrees Celsius when it is currently 10 degrees, the room temperature will eventually reach 25 degrees. The set temperature is a representation of a future state, but it contains no information about how the system should behave. We cannot predict the thermostat's behavior if we only know the reference signal. Nor can we predict its behavior if we only know the room temperature. To predict its behavior we need to know both the current temperature and the reference temperature, because the difference between the two generates the output.

3.1 The Organization of Control

A negative feedback control system, then, solves the calculation problem without performing calculations of the actual disturbances. It merely adjusts its output according to the difference between the present value of the controlled variable and the desired value. Control is an emergent property of the negative feedback loop. The linear cause and effect model breaks down in the analysis of such a system, because the output is acting on the input at the same time that the input is acting on the system. The closed loop is a loop of circular causation (Powers et al. 1960).

As illustrated in Fig. 2, a closed-loop control system has three main components: input function, comparator, and output function. The input function represents some variable to be controlled; the comparator computes the difference between a reference signal that represents the desired value of the variable and the input signal; the output function converts the error signal into an output that acts on the environment. A successful control system will keep the value of the controlled variable close to the value of the reference signal, by varying the output to counter the effects the effects of disturbances on the controlled variable. In traditional open

loop models of behavior, the input is not controlled; outputs are generated either by transforming inputs, or spontaneously. In a negative feedback control system, only the input is controlled.

The comparator receives two types of signals, reference signals and input signals. Either could be positive or negative, but they should be opposite in sign. Thus, if the reference is positive, then increasing the reference signal also increases error if the input remains the same. The output will be generated continuously until that error is reduced.

Reference signals are internal to the organism. This is the chief difference between biological organisms and existing man-made control systems. In the thermostat, the reference signal (desired temperature) is set by the user. This is not possible in an organism, because its reference signals are its own. We can only influence reference signals indirectly, e.g. by depriving the animal of what is essential to it, or by manipulating their nervous system if we understand how the comparator functions are implemented.

The concept of set point (Sollwert) in physiology comes closest to the reference signal. The maintenance of a stable internal environment (milieu interieur) in various physiological systems was first noted by Claude Bernard. Indeed, homeostasis, the popular term coined by Cannon, has become a core concept of physiology. Through Cannon's student Rosenblueth, it inspired the field of cybernetics. Surprisingly, none of these concepts had a major impact on the study of behavior, with the possible exception of the work on feeding systems (Staddon 1983). Homeostasis refers to a control system with a relatively fixed reference signal. A common experiment in nineteenth century physiology was to study the wiping reflex of frogs by leaving acid on the skin of frogs with transected spinal cords. The wiping is easily elicited, presumably because the tolerance for acid (reference signal) is close to zero. Whenever the reference signal is negative and close to zero in magnitude, a small perceptual signal can generate an error signal, thus generating an output. Influenced by the work on reflexes, many physiologists implicitly assume a constant reference signal of zero. Freud, for example, often assumed that the primary purpose of behavior is to remove sensory stimulation altogether (Freud 1915). Such systems have the appearance of a simple input causing a simple output. But the appropriate stimulus is simply a disturbance to a controlled variable, creating an error signal from the constant reference signal. Thus homeostasis is a particular type of control phenomenon, found in control systems with relatively constant reference signals, e.g. body temperature in mammals. In many behaviors, the reference signal can change quickly, though a similar negative feedback organization is still found.

The comparator sends an error signal to the output function, which transforms it into some output. There is not necessarily a one-to-one correspondence between error signal and output. Even a simple operation like integration can make the output appear very different from the error signal (Robinson 1989). The output of a control system, however, should not be equated with the behavior per se. Most control systems do not generate outputs that correspond to what we observe as the behavior of the organism. Their output functions are located inside the organism, or more

exactly in the nervous system, and do not affect motor neurons directly. In fact, as we shall discuss below, they can act on other control systems in a hierarchical arrangement.

The input function, output function, and comparator are all located inside the control system, but the feedback, which describes the input as a function of the output, is located outside, in the environment. The environment is everything that is outside the control system. With feedback the loop is closed, so that the output will ultimately reduce the error signal—the definition of negative feedback. A positive feedback system, by contrast, will increase the error and moves input away from its desired state. Such a system is not desirable for controlling any variable, as it will only amplify the effects of disturbance rather than resist them (Powers 1978).

3.2 Wiener's Error

Although words like feedback and control have become a part of the everyday vocabulary, the principles outlined above are misunderstood. For by the time engineers discovered the principles of control, the life sciences were already entrenched in the paradigm of linear causation, imported by Descartes from physics. Descartes himself understood the logical consequence of his position. On the one hand, the explanation of the movement of natural bodies offered by physics can be adopted to explain animal behavior, using the model of the reflex arc. On the other, subjective experience convinced him that he was able to act voluntarily, independent of environmental stimuli. These two possibilities being incompatible, Descartes was forced into the dualism for which he is known. Where a clear antecedent is found, the reflex arc is proposed to transform external energy into motor output; where no stimulus can be identified as the cause of behavior, it is assumed that, somewhere within the brain, a homunculus issues commands which are ultimately sent to the muscles. The homunculus, on this account, cannot possibly be material, for that would place it in the chain of linear causation. Behind dualism is the assumption of linear causation. Descartes was therefore quite consistent. Only those who came after him became confused. Instead of accepting his dualism, most chose either the stimulus-driven reflex arc model or the command-driven model. These two positions were eventually transformed into pairs of opposing schools: rationalism and empiricism, vitalism and mechanism, associationism and idealism, and most recently, behaviorism and cognitivism. As is common in intellectual history, the opponents typically share key assumptions, in this case the linear causation model.

Today, many students of behavior would be surprised to hear that they are following Descartes. It is therefore important to understand how they came to be Cartesians without knowing it, by exposing their unquestioned assumptions. Both the stimulus-driven and the command-driven models assume that behavior is open loop. Even the founders of cybernetics, who introduced control theory to the general public, could not free themselves from the grip of the linear causation paradigm (Rosenblueth et al. 1943; Wiener 1948).

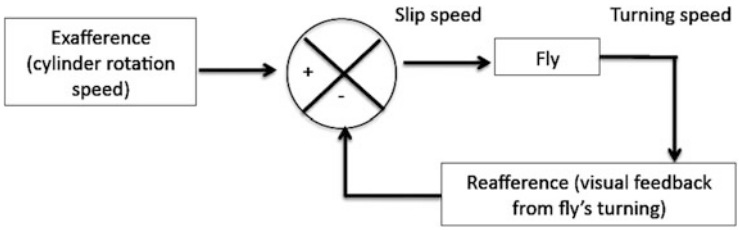

Fig. 3 The wrong model of closed loop organization influenced by Wiener (after Camhi 1984). In this model, the comparator function is located in the environment rather than in the organism

Wiener, in particular, was responsible for the widespread misunderstanding of negative feedback in the life sciences. In Wiener's model, the organism is not a control system, but merely the output function of a control system (Wiener 1948). He placed the comparator outside of the organism, in the environment (Fig. 3). The organism simply receives the error signal and transforms it into output—a stimulus-response device. Wiener's error is not easy to discern, for his model contains exactly the same components as a negative feedback control system. His mistake is in the assignment of the components of the control system to the organism–environment relationship. Instead of placing the collection of control systems inside the organism, Wiener simply turned the organism into an output function (Fig. 3). In itself, of course, the output function is not a control system, because control is an emergent property of a collection of components in a particular arrangement.

Wiener was probably misled by man-made control systems, in which the reference signal is specified by the human user. The engineers have made a control system seem to act like any other input–output system. Although it has the negative feedback control loop, it does not have its own reference signals. That is why one can adjust the setting on thermostat, but the thermostat itself does not have a preferred temperature. The preference belongs to the user. The reference signal in the man-made control system is normally designed to be accessible to the human user. Biological organisms do not serve any user. They serve their own essential variables, and their reference signals belong to them alone. Environmental influences act as disturbances to the controlled variable.

Unfortunately, Wiener's error became his major legacy, resulting in nearly universal confusion regarding the use of negative feedback in explaining behavior. A good example is standard textbook explanation of the optomotor turning response (Camhi 1984). In the typical experiment, a fly is placed on a platform inside a cylinder with a striped pattern. If the cylinder is rotated, the fly will turn in the same direction as the cylinder. The difference between the fly turning and the cylinder turning is called "slip speed." According to Camhi, the slip speed serves as the error signal in this closed loop control system; it is needed for the fly to generate the output of turning. In other words, slip speed is used as a stimulus which "causes" the response.

Moreover, Camhi assumes there are two inputs, one that is independent of self-motion (exafference) and another that is self-induced (reafference). The turning of the cylinder itself is the exafference, which is then compared with the reafference, visual motion as a result of turning (Von Holst and Mittelstaedt 1950). But Camhi fails to see that the slip speed, the difference between the cylinder rotation and the self rotation, is the perceptual input—the actual visual motion sweeping across the retina. The slip speed is not an error signal, but the input to the optomotor control system, and the variable being controlled. What Camhi calls the input, the cylinder rotation, is really the input from the perspective of the experimenter, not the actual image velocity on the retina of the fly. The comparator is located somewhere in the fly brain; and it compares the perceived visual motion with the reference signal, which specifies how much "slipping" of the cylinder is permitted. If the latter is zero, then any perceived slip speed will create an error signal (i.e. cylinder is moving too quickly). The output ultimately generated is turning in the same direction as the cylinder, which reduces the value of the controlled variable.

Camhi writes: "one surprising feature of the optomotor feedback loop is that it is actually impossible for the fly to keep up perfectly with the cylinder's rotation; for to do so would . . . create a slip speed equal to zero. But a slip speed of zero would produce zero behavior, so that fly's turning would, of necessity, stop. Given this circumstance, the best that the fly can do is to keep its turning speed very close, but not equal, to the cylinder's rotation speed. This it does, especially for slow cylinder rotations of up to about 20 per second." (Camhi 1984). Camhi assumes that closed loop feedback control requires the animal to generate sufficient error in order to generate behavior. This he considers a flaw in the system. But if we assume that the reference signal is zero (no slipping allowed), then error and output will be proportional to the input. Turning continues because additional error is created by the disturbance introduced by the experimenter (i.e., cylinder rotation), not by the fly. So long as the cylinder is rotating, the error is continuously generated.

4 External Causation and the Reflex Arc

The stimulus-response model is the modern day version of classical empiricism: "sensory vibrations which are excited in the external organs, and ascend towards the brain by agitating the small particles of the muscular fibers . . . excites them to contraction" (Hartley 1749). This model still dominates psychology and neuroscience. Even many models in cognitive psychology (e.g., connectionist neural network models) are just stimulus-response models with intervening variables or hidden layers (Hull 1943; Millerm and Cohen 2001). But few have asked whether the stimulus-response model can explain any behavior in any organism.

By universal consensus, the reflex arc is the best example of a stimulus-response mechanism in its purest and simplest form. According to Sherrington, the reflex arc, comprising a receptor, a conductor, and an effector, is the fundamental unit of integrative activity in the nervous system (Sherrington 1906). Such a model

appears similar to the control system, the receptor resembling the input function and the effector the output function. But there is no feedback in the reflex arc, and the conductor does not receive reference signals for desired value of the input. It assumes that nothing happens to the value of the stimulus, once the response is initiated.

Sherrington did not hesitate to title his book "The Integrative Actions of the Nervous System" even though it only describes his work on spinal reflexes using the decerebrate preparation. In this preparation, the spinal cord is cut to remove the influence of the brain. Sherrington focused on the scratch reflex in dogs. Since the dog can no longer behave, it will not respond to the stimulation as it would normally do (which might include attacking the experimenter). Having removed the brain, Sherrington could better manipulate the input (e.g., drops of acid or electrical stimulation) and measure output (a scratching of the hindleg). But even in his reduced preparation, the relationship between input and output was far from clear. The observed irregularities (e.g., variable latency and after-discharge) was attributed to synaptic transmission. As a leading proponent of the neuron doctrine, Sherrington was convinced of the existence of junctions between neurons. He therefore concluded that his data on reflexes not only supports the reflex arc model but also suggests the existence of synapses (Sherrington 1906).

4.1 The Stimulus-Response Illusion

What Sherrington neglected is the feedback acting on the controlled variable. The "stimulus specificity" of reflexes provides a clue about the controlled variable, which is always disturbed by the "effective" stimulus. For the scratch reflex, it is the amount of irritation on the skin; for the pupillary light reflex, the amount of light on the retina. The reference signal is negative for protective reflexes, determining what is "excessive" activation of the relevant receptors. In the case of protective reflexes, the output usually reduces the value of the controlled variable.

Sherrington's input/output analysis only works for open loop systems. When applied to control systems, it can produce a powerful illusion. The illusion of a stimulus-response system emerges when the observer can identify a clear antecedent stimulus event and a behavior that follows (Powers 1978). The antecedent event appears to be the cause of behavior. The fire appears to cause withdrawal of the hand; the acid on the skin appears to cause scratching; and the nervous system appears to transform the sensory input into motor output. If there is any correlation between the stimulus and the response, then the function defining this relationship appears to be a function of the organism.

Appearance, however, can be deceiving. For stimuli to be effective, they must act as disturbances to the input variable being controlled. Since the error signal from the comparator can only be produced by variations in the input and the reference signal, when the reference signal is constant and low (e.g., zero), the error signal is largely determined by the disturbance. Thus with a fairly constant reference signal,

we can observe a regular relationship between stimulus and response, thus creating the illusion that stimuli cause responses.

Since the definition of control is systematic resistance to disturbance, the disturbance and the output are highly correlated. Yet this correlation is not a function of the organism. It is a result of disturbance and output simultaneously acting on the same input variable, but in opposite directions, one pushing and the other pulling. For example, if I am trying to hold my umbrella in the rain, the wind is a disturbance to the position of my umbrella. My arm counters this disturbance to keep my umbrella straight. Recording activity from my arm muscles and the wind simultaneously, we can find a high correlation. The wind is transformed by my nervous system into muscle contraction in my arm—so it appears. But there is no such transfer function inside of me. For instead of a "wind to muscle" sensorimotor transformation, there is the connection between the wind and the umbrella, which is a feature of the environment, and there is the connection between the arm and the umbrella, which is another feature of the environment. These two functions are correlated, so that the latter cancels the effects of the former. Neither connection belongs to the organism, and neither can be found inside the brain. Both the wind disturbance and the neural output act on the same variable—umbrella position—which is perceived by my nervous system. So long as the umbrella position is being controlled, the wind input will be correlated with the arm activity, creating the illusion of a stimulus-response system. But as soon as I stop controlling the umbrella, the correlation disappears. This illusion is the chief source of spurious sensorimotor transformations in many studies.

4.2 Fixed Action Patterns, Central Pattern Generators, and the Cognitive Homunculus

Some might argue that more enlightened "cognitive" theories of behavior have long ago replaced the simple reflex arc models. Most cognitive models are variants of the command-driven model, which is on the other side of Cartesian dualism. On this view, the cause of behavior is not found outside the organism, in some environmental stimulus, but within the organism itself. Independence from sensory inputs is considered the basis for cognition. Voluntary actions are produced by some internal, spontaneously active process, a homunculus issuing commands to output functions. This view has many branches, such as rationalism and its modern variants in linguistics and cognitive science (Chomsky 1965; Miller et al. 1960).

In neuroscience, the command-driven model is based on the concept of central pattern generators, which originated from the work of Graham Brown, one of Sherrington's students (Graham Brown 1911). Using the same spinal preparation, Graham Brown showed that the removal of sensory afferents to the spinal cord did not eliminate locomotion in the cat, suggesting the existence of intrinsic mechanisms capable of generating rhythm independently of sensory afferents (Lashley 1951). That the nervous system, like the heart, can generate spontaneous behavior

independent of inputs was not a novel idea. For the scratch reflex, there is no one-to-one correspondence between stimulus and response; the rhythm of scratching, at about four times per second, is independent of the rhythm of stimulation (Sherrington 1906). Sherrington himself compared the fixed rhythm of the scratch reflex to the beating of the heart. But Graham Brown argued that the independence of such intrinsic rhythm from sensory afferents falsifies the reflex arc theory. We now know of mechanisms which allow neurons themselves to act as pacemakers, even in the absence of synaptic inputs (Llinas 1988), which appears to support the idea that intrinsically generated patterns can generate behavior independent of sensory inputs.

The command-driven model is intended to replace the reflex arc model. But like Descartes' homunculus, it shares the same underlying assumption of linear causation. The cause is internal, and the effect or observed behavior is external—an inside-out version of the stimulus-response reflex arc paradigm. And to explain the unpredictability of behavior, what occurs inside the command-issuing center is often assumed to be some random process (Glimcher 2005; Neuringer 2002; Neuringer and Jensen 2010). But the conceptual confusion of linear causation remains. If the behavioral patterns control specific variables, then the intrinsic rhythm may simply reflect properties of the output function, which must still vary as a function of the relevant error signal. If the spontaneously generated outputs do not control any input variable, then the calculation problem cannot be solved at all.

5 Hierarchy of Control

As I write, my head position, body temperature, blood sugar, and direction of my gaze are all being controlled. A single control system cannot possibly control all these variables. The control system described above is therefore only the basic building block out of which the organism is constructed.

The comparator in any control system must receive input signal from the input function and reference signal from somewhere else, and its output must also go somewhere else. Where does the reference signal come from? And where does the output signal go? According to Powers, the level that supplies the reference signal is another control system (Powers 1973b). As shown in Fig. 4, a control system can send a reference signal to the comparator of another control system. The system that sends the reference signal is hierarchically higher. In other words, the higher level supplies the purpose for the lower level.

In traditional hierarchical models, "commands" are elaborated by lower levels into more detailed commands (Fuster 1995). Each level is open loop (Millerm and Cohen 2001). By contrast, descending signals in a control hierarchy do not specify the output of the lower system (Marken 1993). Instead of telling lower systems what to do, the higher systems tell lower systems what to sense. The lower output is proportional to the difference between the top down reference signal and the input

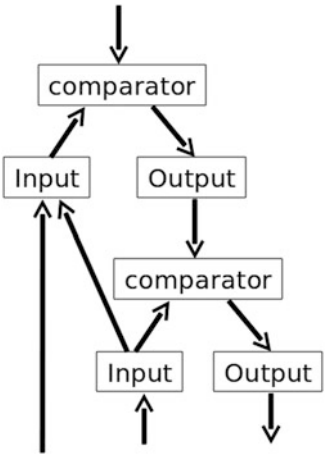

Fig. 4 Hierarchical organization of control systems. The key feature of the hierarchy is that a higher control system may send outputs to the comparator of a lower control system, thus changing its reference signal

to the lower control system. Consequently there need not be any correlation between the top-down reference signal and the output of the lower systems.

It is not possible to specify the output of lower systems by directly activating their output functions. If the reference of the lower control system remains the same, the direct command to the output function will change the input to the lower system, which will produce an error signal, and an output from the lower system that cancels the effect of the command. To use the lower system, the higher system must change the reference signal, to produce the needed output to reduce its own error signal. The lower system knows nothing about the variable being controlled by the higher systems. But it will counter any disturbance to its controlled variable, the value of which is determined by the top down reference signal. It just matches its input with the reference signal it receives. The higher system does not sense the effect of such disturbances to the lower controlled variable, if the inputs at the lower level are already controlled.

5.1 Essential Variables and Homeostatic Control

As we ascend the control hierarchy, we ask why a certain output is produced, which is a question about the reference signal and the variable being controlled (Powers 1973a). Conversely, when we descend the hierarchy we ask how a variable is controlled. If we keep asking the "why" question as we ascend the hierarchy, i.e. look for the source of the reference signal for the present level, there must be a level at which we cannot go any higher. This would be the top level, for the sake of which other control systems operate. Ultimately, why do living organisms do what they do? The concept of "survival" comes to mind. But survival is merely an abstraction,

comprising a collection of variables, which Ashby called "essential variables." They are essential, because death results if these variables are not controlled (Ashby 1960). The traditional examples of homeostatic control, e.g. blood glucose, body temperature, are all essential variables (Cannon 1932).

For any essential variable, there is some innately organized control system that opposes the effects of disturbances. For example, blood pressure is sensed by baroreceptors located in the carotid sinus. When blood pressure increases, activating the baroreceptors, the signal is sent to the brainstem, where specialized neurons with a built-in discharge rate serving as the reference signal are located. The difference between the input and the desired rate is the error signal, which can change the heart rate. Such autonomic functions are performed by many control systems specialized for the control of specific variables. At any given moment, all of these specialized control systems must function for the organism to survive. But here is a paradox. If the essential variables are controlled by first order control systems, each associated directly with the appropriate output functions that can act to control the value of the variable, then why is a hierarchy needed at all?

Although all control systems within the organism must somehow serve the essential variables, this organization is more complex than it appears at first glance. Merely having the first order homeostatic control systems is not enough. The new born infant is already equipped with functional control systems for essential variables, yet it is quite helpless, its ability to resist perturbations to these variables limited. For example, it can sweat when the body temperature increases, but it cannot escape from a fire, or to extinguish the fire with water. It is instructive to understand the limitations of first order control systems.

In simple negative feedback systems, the sensors sense the state of the essential variable directly. For example, a body temperature control system does not get information about the disturbance to the temperature until the disturbance has already started to change the value of the controlled variable. Should a fire be started in the room, it would know nothing about the fire. As far as the first order control system is concerned, there is no such thing as fire. There is only the sensed temperature. No other input is available to this control system—not the smoke or the flame. The simple control system is therefore limited by the sensory variables it can detect. To detect other variables, signals from various sensory receptors must be combined. A hierarchical organization is required for the progressive transformation and recombination of "primitive" sensory inputs (Konorski 1967). At the same time, the outputs of the control systems for essential variables are also limited. The autonomic outputs (e.g., sweating) are not general purpose output functions. One cannot use them to walk. As we shall see, such limitations in the output function also requires hierarchical organization and the development of a general purpose skeletomotor system.

5.2 *Anticipatory Control*

Facing a fire, most organisms will not wait for the flame to engulf them. They will act as soon as they detect the fire. To many, the fact that organisms can anticipate disturbances suggests that the nervous system makes predictions about the future (Wolpert et al. 2011). But before jumping to this conclusion, we must first understand exactly what "prediction" means, and how the anticipatory behavior is produced.

Starting with Pavlov, anticipatory control has been studied extensively. In Pavlovian conditioning, the unconditional stimulus (US) can be a drop of acid, and the UR is salivation, which dilutes the acid. The controlled variable is irritation of the mouth cavity. The US is merely a disturbance that changes the value of some essential variable, and the action of the control system is the unconditional responses (UR). The conditional stimulus (CS) is a neutral stimulus, such as the sound of a metronome, which does not evoke any response. But after pairing with US, it will reliably evoke a "conditional response" or (CR). The CR, then, is the anticipatory action. In Pavlov's original studies, the same output—salivation—simply occurs earlier in time. As a result of conditioning, the body will react before the actual disturbance or the US occurs. The CR is an anticipatory version of the UR and prepares the animal for the US. Pavlov called this process stimulus substitution, since the CS becomes a substitute for the US in its ability to elicit the same response (Pavlov 1927).

If the US is a disturbance to some controlled variable, then the CS, as the predictor of the disturbance, can act like the disturbance itself. For example, the introduction of glucose into a rat's stomach will cause a rise in blood sugar. This increased blood glucose is not the UR. It is merely a consequence of the glucose absorption. Repeated introduction of glucose following a CS will result in the development of a hypoglycemic response to the CS. The reduction in blood sugar will compensate for the glucose disturbance. The CR attenuates the impact of the US (Dworkin 1993). This can be accomplished by redefining the input function, so that US input function is now responsive to the CS input. The weight of the existing connection may be negligible, which explains why the CS is originally "neutral", i.e. does not evoke any response. Pavlovian conditioning can therefore strengthen this connection, adding a new source of signal to the input function of the UR control system. This type of mechanism can explain some of the constraints on learning. For example, the closer together the two input functions are, the better will be the consequent conditioning. Obviously, if the CS is in the same sensory modality as the US, the conditioning is easier because there may already be existing weak connections between the CS input function and the motor neurons producing the UR.

It is beyond the scope of this essay to discuss the mechanisms of learning in any detail, but it is important to point out the critical role of the homeostatic control systems in driving the reorganization. The error signals from the essential variables can produce widespread random fluctuations in the parameters of other

control systems (Powers 1973b). With the reflexive withdrawal of a hand from the approaching fire, the controlled variable is sensed temperature, and the UR reduces that temperature. When there is a burn, the error signal can randomly change the weights of the system. And a reduction in the error signal can stop this reorganization process. For example, if there is a CS that predicts the fire, the weight connecting this signal to the same motor neurons that produce the UR will sometimes be larger due to the random fluctuations, producing earlier withdrawal from the heat source and a reduction of error, which in turn slows down the reorganization process, retaining the increased weight.

But pure stimulus substitution of this type cannot be common for a number of reasons. A predictor of the disturbance can be detected by sensors from many modalities, which are not located in the local spinal circuitry critical for the withdrawal UR. The projections from these detectors to the detector of the US are not usually direct. Consider the noise of the crackling fire, the smell of the smoke, or the sight of the fire—in all these cases the sensory signal cannot go directly from the perceptual channels to the spinal motor neuron innervating the flexors. The expansion of the input function in stimulus substitution is not sufficient. The input function must now associate a high level perception, e.g. sight of fire, to the low level perception of the US disturbance.

One possibility is that the output function of the higher-order system that receives the CS can be modified, so that a projection is sent to the lower system to alter its reference signal. This is supported by experimental evidence. In eyeblink conditioning, an air puff to the eye elicits a reliable blink, but if the air puff is predicted by an auditory stimulus, the blink will occur earlier in time, following the CS. Although the CR and the UR are similar, pharmacologically blocking the red nucleus prevents expression of the CR but not the UR (Thompson et al. 1998). The CR circuitry can thus be dissociated from the UR circuitry, though they share the final common path, using the motor neurons in the cranial motor nuclei. The CR can be generated by a hierarchically higher level sending a reference signal to the motor neuron generating the UR. The higher system, which involves the interpositus nucleus of the cerebellum, generates an output that, via the red nucleus, changes the reference signal for the motor neuron, which serves as a comparator. A change in the reference signal can also produce the blink, in the same way that a change in the perceptual signal does. The reference signal "simulates" actual disturbance. The comparator takes the difference between the actual disturbance to the controlled variable (proportional to US magnitude) and the reference signal from the higher system, generating an error signal that produces the CR. The CR reduces disturbance, and as a consequence there will be less error from the output function when the US arrives. The UR is reduced, since the effect of US is already resisted by the CR. This phenomenon is commonly called conditional diminution of the UR, an example of US processing (Domjan 2005), but it is just due to the disturbance resisting effect of the CR.

Traditionally, the term Pavlovian conditioning is simply used to describe whatever happens when some neutral stimulus is paired with some disturbance to an essential variable. But this classification is not based on any common underlying

mechanism. In some cases, whether by redefining the input function or by changing the reference signal from the output of a higher level, the CR and the UR share the final common path. This is true for salivation and eyeblink conditioning. But in other cases, the CR and the UR are different. For example, in fear conditioning, an auditory tone predicts electrical shock to the feet. The URs to shock include jumping up and down. The CR is freezing, a stereotypical defense reaction to danger in rodents. When the predator approaches and there is nowhere to escape, the animal freezes. The sight of predator is a danger signal, not a pain signal, and the behavior in response to danger and to pain can be quite different. Of course, when the predator is attacking the prey, the prey does not freeze—it struggles to escape, much as the shock elicits avoidance behavior. If the sight of the predator actually triggers the unconditional response to pain, as in stimulus substitution, then the animal will simply alert the predator.

Thus, it is not enough just to produce same output earlier in time, in response to a predictor of the disturbance. A different type of output is required to affect the disturbance in question. Instead of resisting the same variable as the UR does, the CR controls a different perceptual variable. By controlling a different aspect of the disturbance, it is able to reduce or prevent the error signal in the essential variables. It is by not alerting the predator that the rat can avoid injury and death altogether.

As we have seen, anticipatory control, as revealed by Pavlovian conditioning, can involve three possible mechanisms. First, conditioning can redefine the input function within the same control system that generates the UR. The US is always a disturbance to some essential variable under homeostatic control, and the additional input from the CS detector to the UR motor neuron can become stronger. This mechanism requires the CS input function to be at the same level as that of the US. No hierarchical organization is needed, since the same system that controls for the effect of the US can be modified to produce the anticipatory output. Second, a higher system detecting the CS sends an output to the lower system which controls the variable disturbed by the US. The CR and the UR share the final common path, but the CR is generated by changing the reference of the US control system. To the motor neuron, such a change in the reference signal is similar to a change in perceptual input caused by the disturbance. The only difference is in the timing. The CR is produced earlier and consequently reduces the impact of the US disturbance when it actually arrives, thereby also reducing the UR. Finally, it is possible for higher control systems to alter the reference of alternative control systems that can control for different perceptual variables and generate different behavioral outputs. As a result of controlling these other perceptual variables, the error signal in the US control system is reduced or even prevented.

5.3 Instrumental Control

In Pavlovian conditioning, the animal is usually prevented from affecting the CS, which is under the control of the experimenter. If it is free to behave as it normally

would, without the interference of the experimenter, the animal will attempt to control the CS. Imagine what would happen to a dog if acid is introduced into its mouth. It is only under very artificial conditions that we can measure the anticipatory salivation and repeat the acid treatment. The restrained animal simply cannot behave freely. It would be a very unusual dog indeed if it simply accepts this state of affairs. Had it not been restrained, it might not be so kind to the experimenter who is injecting acid into its mouth. The way to avoid acid is not to dilute it with saliva, but to attack the man in the lab coat. This type of behavior, acting on the source of the disturbance and exerting control over it directly, is familiar to us. More than the reflexes and simple anticipatory behaviors, it is the most common behavior in humans. Such goal-directed, voluntary behavior requires yet another type of control, which I shall call "instrumental control," often studied in instrumental or operant conditioning experiments.

To understand the distinction between instrumental control and anticipatory control discussed above, consider the example of approach behavior. Animals will often approach predictors of reward. In the well-known experiments on pigeons, a key light was lit before food presentation—a basic Pavlovian experiment with key light as the CS and food as the US (Brown and Jenkins 1968). Pigeons learned to approach and peck the key light that predicted food delivery. The US is not contingent upon the pecking: there is no feedback function between the anticipatory behavioral output and the food delivery. Then the experimenter reversed the contingency between the pecking and the food delivery. Now food was delivered following the CS as before, but any pecking of the key light would cancel it. Surprisingly, when this "omission" contingency was imposed, the pigeons did not immediately reduce their pecking behavior. The CR persisted, even though it canceled the reward.

Before calling the persistence of the CR maladaptive, it is worth noting that, in the natural environment outside the laboratory, sign tracking is very effective. Approaching the sound of the apple falling from the tree, e.g., usually gets you closer to the apple. There is a stable relationship between the sign and the goal, so that proximity to the sign means proximity to the goal itself. Of course, it is possible to imagine a world with a different arrangement, but evolution does not operate on imaginary worlds.

Once the animal has acquired the anticipatory tracking behavior, it cannot withhold it even when doing so increases the error signal. This is trivially true for first order control systems. One cannot refrain from sweating in order to cool the body. If some mischievous experimenter arranged the contingency so that sweating will increase room temperature, then the body will still continue to sweat, even though it is increasing the error. The error signal that represents "too much heat" can only be translated into sweating. It cannot be used by the same control system to stop sweating. A first order control system cannot adapt to a reversal in the polarity of the feedback function, though of course the error signal can engage different control systems in a hierarchy that allows one to leave the heated room. More importantly, the control hierarchy that produces anticipatory control also cannot cope with this type of reversal, as revealed by the omission test. The ability to alter behavior

following a reversal in feedback function is a distinctive feature of instrumental control. A typical instrumental contingency could be "push the door to open it," or "pull the door to open it." The same goal is reached with different actions. The same error signal can be used to perform two opposite actions. Distinct lower level control systems can be selected depending on the feedback.

What is the difference between instrumental control and anticipatory control? If we hear the sound of an apple falling, we may approach the location of the sound. The fact that we are closer to the apple when we approach the sound is due to the environmental contingency. Apples normally stay close to where they fall—a feature of the environment independent of any organism, though the efficacy of the sign tracking behavior depends upon it. As a stable feature of the environment, it is probably responsible for the evolution of the sign tracking behavior. Note, however, that neither salivation nor approach changes the rate of the apple falling. These behaviors control for variables related to the apple—proximity or ease of digestion, but they do not cause the falling of the apple. They do not operate on the distal environmental variable that is responsible for the apple falling. On the other hand, instead of waiting for the apple to fall from a tree, we may shake the tree to make more apples fall, or we can climb the tree to get the apples. We can even grow apple trees. These are all examples of instrumental actions.

A key feature of instrumental control is the diversity of the means to achieve the same end. Wittgenstein once observed that the category of games have nothing in common, in the sense that there is no Platonic essence in which all games partake. He described the relationship among members of this category as "family resemblance" (Wittgenstein 1953). But Wittgenstein failed to consider the function or the goal of the game. Similarly, Skinner has defined the "operant" as a family of movements whose only common feature is that they effectively earn the reward. For example, there are many ways to move the lever in order to produce food delivery. One cannot even define a priori what these movements will be, just as one cannot predict ahead of time all the members of the category of game. If we carefully measure the physical attributes of each lever press in terms of the kinematics, or the pattern of neural signals sent to the muscles, we will not find some essence that all of them have in common, except their goal. But the goal is not a property to be abstracted from the physical properties of the movements, just as the reference signal is not something you can extract from the behavior of the thermostat.

Anticipatory behavior is based on a stable environmental contingency—approach the location of the sound, you approach the apple itself. The CR is generated by a control system controlling for a particular perceptual variable with a fixed relationship with the essential variable controlled by the UR system. The only way to reduce the CR is to change the relationship between the CS and the US. For example, the organism does not affect the reliability of the key light in predicting food presentation. One can degrade this relationship and reduce the CR (Schwartz and Gamzu 1977). By contrast, the instrumental action is based on the contingency between the action and the goal. The rate of the action can itself be a controlled variable. The system that controls the rate of reward, e.g., can send a reference

signal to the lower system that controls the rate of action, which in turn tells other lower systems what to sense in completing the action.

During instrumental conditioning, by trial and error the animal identifies the mechanism underlying this environmental contingency (that apple falls from the tree), and acts on it by shaking the tree or climbing it. There is therefore learning about the action-outcome or instrumental contingency (Balleine and Dickinson 1998). The goal is related to the error signal of some essential variable. A reduction in blood sugar, e.g., may produce an error signal in the homeostatic control system, triggering a number of autonomic responses, but at the same time such an error signal may be transformed into higher level perceptual signals, related to hunger and appetite, or to specific representations of goals like apples. Coupled to knowledge of the instrumental contingency between actions and outcomes, the higher levels may choose from multiple action systems, which are themselves hierarchically organized, and use these as output functions to control a variable like the rate of apples obtained.

5.4 Perception and Action

Hierarchically higher control systems can send reference signals to the homeostatic control systems. Consider the behavior of withdrawing your hand from excessive heat. Let us assume that the reference signal has a net inhibitory effect on the motor neurons innervating the flexors. In the lower control system, the pain signal, which exceeds the value allowed by the reference signal, can produce the withdrawal reflex. The reference signal can be translated as "do not let the input exceed this value" and the output reduces the input, by moving the hand away from the source of pain. But we can hold a hot cup without dropping it. Thus changing the reference signal can increase pain tolerance of the flexion withdrawal system. The reference signal of the higher level might be "do not drop this cup"; and of a still higher one, "do not embarrass yourself."

When a higher control system sends a reference signal to a lower one, the lower one essentially serves as an extension of the output function of the higher level. Yet the higher systems can only have limited influence on the reference signals, because the value of the essential variables must stay within a range before survival is threatened (Mrosovsky 1990). Although the top down reference signals to the homeostatic control system can change their outputs, the autonomic outputs are not sufficient to serve as an extension of their output function. This may explain the difficulty with operant conditioning of the autonomic outputs (Dworkin 1993). Moreover, the homeostatic control systems can only have very limited effects on the environment. To act on the environment, to extinguish the fire or to climb the tree for apples, the skeletomotor system is needed, which has a parallel role in extending the output function of higher control systems.

The skeletomotor system is itself a hierarchy of control systems. The last output function of this hierarchy is the final common path, comprising the alpha motor

neurons and the muscles they innervate. The force of the muscle contraction, a function of the motor neuron output, is sensed by the Golgi tendon organs. The reference signal for force control can come from multiple sources, such as the Ia afferent and descending projections to the alpha motor neuron. The output produced by the alpha motor neuron is sensed by the Golgi tendon organs, and the sensed signal representing load in turn is compared to the sum of the force reference signals. The alpha motor neuron can receive reference signals from multiple higher sources. The brain cannot send a command to specify the tension of the muscle. It can only specify how much tension is to be sensed. This does not determine how tense the muscle will be, for if there is a disturbance, the muscle will adjust its degree of contraction to compensate for the disturbance. If the sensors are activated to the desired value artificially, e.g. by pulling the tendon organs, the muscle tension will not increase because the reference signals merely specify the requisite level of sensed tension.

Movement per se involves multiple levels controlling for multiple variables such as muscle load, length, joint angle, and sequential activation of agonist and antagonist. But distal senses like olfaction, audition, and vision are not necessary for these types of control. The modality of the perceptual input is primarily proprioceptive, which come from specialized sensors that report the current state of the muscles, tendons, and joints. The distal senses play an auxiliary role in movement. They are critical for goal-directed behavior, particularly for instrumental control and anticipatory control, but they are not necessary for the very character of movements themselves. The spinal cord, brainstem, and cerebellum are sufficient for the control of the proprioceptive inputs, producing the needed output from motor neurons. Thus, it is not surprising that anticipatory control and instrumental control generally require the cerebral hemispheres. Decerebrate animals can still move, but they can no longer control abstract variables such as proximity to predator or the rate of food reward. Despite the enormous complexity in the skeletomotor system, it can be viewed as an extension of the output function of higher order control systems, which use the skeletomotor system to reach specific goals.

Likewise, the perceptual hierarchy for the distal senses like vision and audition are also extensions of the input function for the essential variables. They permit the perception of distal aspects of disturbances and predictors of disturbances. Whereas first order control systems (simple reflexes) receive the first order inputs directly from receptors in different sensory modalities, hierarchical control requires the construction of new perceptual variables. From homeostatic to instrumental control, there is a continuous spectrum of perceptual inputs. They permit the representation of abstract variables like the rate of reward by combining lower level inputs. New variables continue to be formed through experience, becoming capable of being controlled by the skeletomotor outputs through the diverse and ever changing feedback functions in the environment. Note that neither anticipatory nor instrumental control is accomplished by a feedforward model making detailed calculations about the outputs needed ahead of time. In spite of the tremendous complexity in the hierarchical organization, each control system still operates by negative feedback and control of input.

6 Test for the Controlled Variable

As mentioned above, traditionally the input is introduced and manipulated precisely, while output is measured. No question is asked about what the impact of the output is on the controlled variable, because the existence of negative feedback is not recognized. Based on the linear causation paradigm, many experiments were designed to turn organisms into open loop systems. For example, the cuttlefish attacks prey by ejecting its tentacles. If the target is pulled away just after the cuttlefish has begun to eject its tentacles, the direction of the strike is not adjusted, and the tentacles miss their target (Camhi 1984). This type of manipulation is too abrupt and tells us little about the behavior in question. Any control system can fail, especially when the environment presents disturbances that exceed the capacity for control. During a hurricane, the behavior of a pigeon may be indistinguishable from that of a rock of similar mass; but that does not mean that the pigeon is an open loop system. It could be an example of open loop control, but it could also be an example of failed closed loop control. Manipulations that do not take into account the timescale and the capacity for control just force the control systems to fail. It is like killing an animal to show that it cannot control its body temperature.

The principles of negative feedback control, however, suggest a very different way of studying behavior. To study any behavior, it is necessary first to identify the controlled variables. This has been called the test for the controlled variable. Rather than asking how an organism responds to a stimulus or generates intrinsic activity, we ask which variable it is trying to control, i.e. what the purpose is.

Fatigue, e.g., can be a consequence of strenuous activity, but it is not usually the purpose. If we hypothesize that fatigue is the purpose of someone moving a sofa from a truck to his living room, we can introduce disturbances systematically and measure resistance to the disturbances to the controlled variable. We can move the sofa for him, thus preventing fatigue unless he insists on moving it himself. If the controlled variable is fatigue, he will resist the attempt to reduce fatigue for him. On the other hand, if the controlled variable is not fatigue but the position of the sofa, he would gladly accept the offer. Introducing disturbance to the "sofa position" variable would then encounter predictable resistance.

To test for the controlled variable, then, we can manipulate the disturbance to a hypothetical variable or the feedback function (Marken 2001). Systematic resistance to such manipulations is evidence for control. In the scratch reflex, to terminate the irritation on the skin, it is necessary for the scratching movement to target specifically the area being stimulated. To discover the controlled variable, we can manipulate the relationship between the scratching and the sensory input. For example, a frog with its brain disconnected from its spinal cord can still remove a chemical stimulus from the skin of the forelimb by wiping with the ipsilateral hindlimb (Fukson et al. 1980). When the experimenter moves the forelimb, the coordinates of the stimulation site in relation to the hindlimb are altered. Since the same receptors are stimulated, the reflex arc model predicts that the same output will

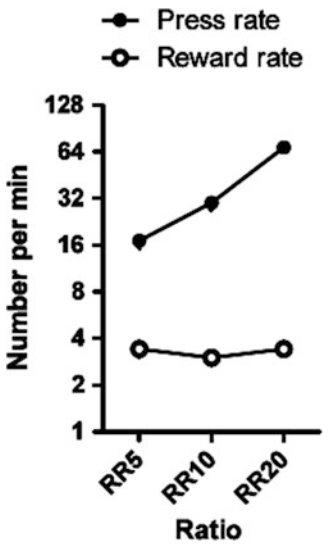

Fig. 5 Control of reward rate in operant conditioning. We used a random ratio (RR) schedule in which the ratio varies randomly around a mean specified ratio value. As we manipulated the average ratio (RR5, RR10, and RR20), the rate lever pressing changed accordingly, while the rate of pellet delivery remained relatively constant. Shown are the steady state values of the rate of lever pressing and the rate of pellet delivery after at least two sessions of training on each ratio ($n = 6$, error bars represent standard error of the mean). The rate of reward appears to be the controlled variable

be elicited, though of course repetition of the same scratching movement would have missed the target region. But in fact the spinal frog successfully moves the hindlimb to the correct target region, the output compensating for the effect of the disturbance (change in stimulation location). For example, wiping at the same site occurs not only when the forelimb is parallel to the body but also when it is perpendicular to the body. The outputs necessary for targeting these two locations are quite different, but the end result is the same. In this study, therefore, the researchers simply manipulated the feedback function, so that a very different output was required to reach the same physical location.

Another illustration comes from our own work on operant conditioning. We trained six hungry rats to press a lever for food pellets. We manipulated the schedule of reinforcement, the feedback function that links the behavioral output to food input. The ratio of a schedule defines how many lever presses must be produced before a food pellet is delivered. As we increased the number of presses required to earn a food pellet from 5 to 20, the rats increased their rate of pressing accordingly, but the overall rate of reward remains relatively constant (Fig. 5). Thus by varying the feedback function, we can show that the rate of reward is the controlled variable. Of course, if the ratio is too high, then the rats will not be able to press rapidly enough to defend their preferred rate of reward. The

cost of lever pressing is assumed to be negligible so long as the ratio is not too high, which appears to be the case when the average number of presses required per reward (45 mg pellet) is less than 20. The controlled variable in this case is therefore the rate of pellet delivery, and the output (rate of lever pressing) simply varies in order to defend the preferred rate of reward. Such results cannot be explained by traditional models based on the concept of reinforcement (Sutton and Barto 1998). These models inevitably predict that decreasing the amount of reinforcement will also reduce the behavior that is reinforced, whereas we observed the opposite. Nor do reinforcement-based models predict a constant rate of reward—the controlled variable being defended as a result of variable behavioral output.

7 Conclusion

Behavior is not usually recognized as the manifestation of control. Although we can easily observe the movements of animals and inanimate objects, we cannot just as easily observe purpose. Folk psychology, though much maligned, is not so absurd as to deny the existence of purpose.

Words like purpose and goal are of course used often, but the verbal acknowledgment of such ideas is deceptive. Underneath these models we still find the same linear causation paradigm. Many, e.g., argue that internal representations of the goals can cause behavior. This type of "cognitive" or "purposive" model is meant to replace the S-R model. But as we have seen, this is an inside-out S-R account. It is incorrect to say that the purpose or representation of the goal causes behavior. In the analysis of negative feedback control systems, there are two simultaneous equations to be solved, because the output acts on the input at the same time that the input is acting on the output. To account for purposive behavior, it is necessary to take into account both equations.

In neuroscience, on the other hand, the function of the brain is often described as a "sensorimotor transformation." As the information from the peripheral receptors travels to the brain, at each step some transformation occurs, ultimately resulting in the commands from the brain to generate the behavior (Sherrington 1906). Regardless of how complex the transformation is in the brain, this is also a linear causation model.

Thus whether or not words like "purpose" or "goal" or "prediction" are used is quite irrelevant. We can always find out exactly what the underlying assumptions are by examining the experimental methodology. Does the experimenter attempt to identify the controlled variable (purpose)? Or does he simply manipulate sensory input while recording behavioral output?

Purpose must also be distinguished from equilibrium or consequence. The equilibrium point is not a controlled variable. In a pendulum, the restoring force can be predicted precisely given the disturbance. The displacement itself is solely responsible for the corrective oscillation of the pendulum. In a control system,

the resistance to disturbance cannot be predicted from the disturbance, because it cancels the disturbance in relation to an internal reference condition. How much resistance is offered depends on the reference signal as well as the disturbance. All the energy used to correct a deviation of a variable from the equilibrium point comes from the disturbance that caused the deviation. The loop gain is very low. In a control system, the energy used to resist the disturbance does not come from the disturbance itself. In animals, it comes from food. The closed loop system has a high loop gain, which offers a lot of resistance in response to a little disturbance.

Descartes, impressed by the machines of his day, tried to apply the mechanistic reasoning to explain the behavior of organisms. Unfortunately, he only knew about open loop machines, and it was this narrow concept of mechanism, based on linear causation, that has become synonymous with science. In physics, Galileo was right to reject purpose—since natural bodies are not control systems. The success of physics, however, has led to the erroneous conclusion that purpose does not exist.

Until the formal analysis of negative feedback control was properly introduced into the study of behavior, no one understood exactly what a purpose is (James 1890; Tolman 1932). Most conceptual confusions are created by the campaign to remove teleological explanations of any biological phenomenon. After all, modern physics owed its beginning to a victory—no easy victory—over teleological thinking. Yet purpose is not a metaphysical fiction, though the reflex-arc increasingly appears to be one. Nor is it an illusion created to satisfy our vanity, though the illusion of a stimulus-response transformation can be demonstrated easily. And ironically even the argument that the illusion of purpose is created in order to satisfy our vanity is a teleological explanation. Not only is purpose an operationally defined and experimentally testable concept, it is the only one that can make exact experiments on behavior possible for the first time. A restoration of purpose is therefore a new beginning for the sciences of behavior.

References

Ashby, W. (1960). *Design for a brain*, 2nd edn. New York: Wiley.

Balleine, B. W., & Dickinson, A. (1998). Goal-directed instrumental action: contingency and incentive learning and their cortical substrates. *Neuropharmacology*, *37*(4–5), 407–419.

Bernstein, N. (1967). *The coordination and regulation of movements*. Oxford: Pergamon.

Black, H. S. (1934). Stabilized feedback amplifiers. *Electrical Engineering*, *53*, 114–120.

Brown, P. L., & Jenkins, H. M. (1968). Auto-shaping the pigeon's key peck. *Journal of the Experimental Analysis of Behavior*, *11*(1), 1–8.

Camhi, J. M. (1984). *Neuroethology*. New York: Sinauer.

Cannon, W. (1932). *The wisdom of the body*. New York: W. W. Norton.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge: MIT.

Domjan, M. (2005). Pavlovian conditioning: a functional perspective. *Annual Review of Psychology*, *56*, 179–206.

Dworkin, B. (1993). *Learning and physiological regulation*. Chicago: University of Chicago Press.

Franklin, D. W., & Wolpert, D. M. (2011). Computational mechanisms of sensorimotor control. *Neuron*, *72*(3), 425–442.

Freud, S. (1915). Instincts and their vicissitudes. In *Collected papers*. New York: Basic Books.
Fukson, O., Berkinblit, M. B., Feldman, A. G. (1980). The spinal frog takes into account the scheme of its body during the wiping reflex. *Science*, *209*(4462), 1261–1263.
Fuster, J. M. (1995). *Memory in the cerebral cortex*. Cambridge: MIT.
Glimcher, P. W. (2005). Indeterminacy in brain and behavior. *Annual Review of Psychology*, *56*, 25–56.
Graham Brown, T. (1911). The intrinsic factors in the act of progression in the mammal. *Proceedings of the Royal Society of London*, *84*, 308–319.
Hammond, K. R., & Stewart, T. R. (2001). *The essential Brunswik*. New York: Oxford University Press.
Hartley, D. (1749). *Observations on man*. Bath: Leake and Frederick.
Hull, C. (1943). *Principles of behavior*. New York: Appleton-Century-Crofts.
James, W. (1890). *The principles of psychology*. New York: Henry Holt.
Konorski, J. (1967). *Integrative activity of the brain*. Chicago: University of Chicago Press.
Lashley, K. S. (1951). The problem of serial order in behavior. *Cerebral mechanisms in behavior: The Hixon symposium* (pp. 112–146). New York: Wiley.
Leyton, A. S. F., & Sherrington, C. (1917). Observations on the excitable cortex of the chimpanzee, orangutan, and gorilla. *Experimental Physiology*, *11*(2):135–222.
Llinas, R. (1988). The intrinsic electrophysiological properties of mammalian neurons: insights into central nervous system function. *Science*, *242*, 1654–1664.
Marken, R. (1993). The hierarchical behavior of perception. *Closed Loop*, *3*, 33–54.
Marken, R. (2001). Controlled variables: psychology as the center fielder views it. *American Journal of Psychology*, *114*(2), 259–281.
Miller, G. A., Galanter, E., Pribram, K. H. (1960). *Plans and the structure of behavior*. New York: Holt.
Millerm, E. K., & Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annual Review of Neurosciences*, *24*, 167–202.
Mrosovsky, N. (1990). *Rheostasis: the physiology of change*. Oxford: Oxford University Press.
Neuringer, A. (2002). Operant variability: evidence, functions, and theory. *Psychonomic Bulletin and Review*, *9*(4), 672–705.
Neuringer, A., & Jensen, G. (2010). Operant variability and voluntary action. *Psychological Review*, *117*, 972–993.
Pavlov, I. (1927). *Conditioned reflexes*. Oxford: Oxford University Press.
Powers, W. (1978). Quantitative analysis of purpose systems. *Psychological Review*, *85*, 417–435.
Powers, W. T. (1973a). Feedback: beyond behaviorism. *Science*, *179*, 351–356.
Powers, W. T. (1973b). *Behavior: control of perception*. New Canaan: Benchmark Publications.
Powers, W. T., Clark, R. K., McFarland, R. L. (1960). A general feedback theory of human behavior. *Perceptual and Motor Skills*, *11*, 71–88.
Robinson, D. A. (1989). Integrating with neurons. *Annual Review of Neurosciences*, *12*, 33–45.
Rosenblueth, A., Wiener, N., Bigelow, J. (1943). Behavior, purpose, and teleology. *Philosophy of science*, *10*, 18–24.
Schwartz, B., & Gamzu, E. (1977). Pavlovian control of operant behavior. In W. Honig, & J. E. R. Staddon (Eds.), *Handbook of operant behavior* (pp. 53–97). Old Tappan: Prentice Hall.
Shadmehr, R., Smith, M. A., Krakauer, J. W. (2010). Error correction, sensory prediction, and adaptation in motor control. *Annual Review of Neurosciences*, *33*, 89–108.
Sherrington, C. S. (1906). *The integrative action of the nervous system*. New Haven: Yale University Press.
Staddon, J. E. R. (1983). *Adaptive behavior and learning*. Cambridge: Cambridge University Press.
Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning*. Cambridge: MIT.
Thompson, R. F., Thompson, J. K., Kim, J. J., Krupa, D. J., Shinkman, P. G. (1998). The nature of reinforcement in cerebellar learning. *Neurobiology of Learning and Memory*, *70*(1–2), 150–176.
Tolman, E. C. (1932). *Purposive behavior in animals and man*. New York: Macmillan.

Von Holst, E., & Mittelstaedt, H. (1950). The reafference principle. In *The collected papers of Erich von Holst*. Coral Gables, FL: University of Miami Press.
Wiener, N. (1948). *Cybernetics*. Paris: Hermann and Cie Editeurs.
Wittgenstein, L. (1953). *Philosophical investigations*. London: Blackwell.
Wolpert, D. M., Diedrichsen, J., Flanagan, J. R. (2011). Principles of sensorimotor learning. *Nature Reviews Neuroscience*, *12*, 739–751.

Index

A
Abelson, R.P., 272
Adaptive behavior, 7
Amarel, S., 39
Anatomical/hardware constraints, 181
Aquinas, T., 320
Arbib, M., 48, 69, 70
Artificial system, 3
Ashby, W., 334
Atkeson, C., 167
Autonomous representation learning, 4–5
  define event, 65–66
  event identification
    contingency, 66–67
    model creation, 67
    self-supervised learning, 67–68
  landmarks creation, 65–66
  principles
    break environment, 65, 78
    create representation, 65, 78
    exploit synergy, 63–64, 78
    generate representation, 64–65, 78
  QLAP (*see* Qualitative learner of action and perception (QLAP))

B
Back-propagation through time (BPTT) algorithm, 59
  MTRNN, 54
  RNNPB, 50
Bakker, B., 18
Baldassarre, G., 1–9, 19, 237–265
Ballard, D.H., 5, 99–123
Baraduc, P., 207
Barto, A.G., 4, 13–40, 113–115, 168, 281, 282
Basal-ganglia hierarchies
  amygdala and dopaminergic system, 256–258
  associative loop, 253
  cortico-striatal regions and interconnections, 251, 252
  devaluation experiment
    architecture, 254, 256
    simulated rats behavior, 257, 258
    simulator, 254, 255
    training and test phases, 254, 255
  Hebbian learning rules, 256
  internal globus pallidus (GPi), 251
  limbic loop, 251–252, 258
  micro architecture, 251, 252
  sensorimotor loop, 253
  striato-nigro-striatal spiral pathway, 253–254
  striatum, 251
  substantia nigra pars reticulata (SNpr), 251
  sub-thalamic nucleus (STN), 251
  trial-and-error learning processes, 257–258
Bayesian decision theory, 133
Becker, J.D., 207
Behavioral hierarchy, 4
  description, 14
  ensembles of tasks, 40
  exploratory behavior, Light Box Environment
    active learning, 25–26
    causal graph, 20–21
    configuration, 20
    multiple task skills, 26–28
    option policies, 21, 22
    structure learning, 21–24
  learning in large, complex domains
    lightweight options (*see* Lightweight options, Pinball Task)

G. Baldassarre and M. Mirolli (eds.), *Computational and Robotic Models of the Hierarchical Organization of Behavior*, DOI 10.1007/978-3-642-39875-9,

skill-specific abstraction (*see* Skill-specific abstraction)
value functions, 27–28
many-level skill hierarchies, 39–40
option creation, 39
parameterized options, 39
real-world applications, 40
representation and abstraction, 39
skills, 15
transfer learning, 15
Behavior, cause of
calculation problem, 321–322
control hierarchy
anticipatory control, 335–337
commands, 332
essential variables and homeostatic control, 333–334
higher system, 333
instrumental control, 337–340
lower system, 333
perception and action, 340–341
reference signal, 332
controlled variable, 342–344
control system
classical control theory, 323
closed loop control system, 325
controlled variable, 324
feedforward solution, 322
negative feedback control, 323
open loop system, 324
organization of, 325–327
teleology, 325
thermostat, 323, 324
Wiener's error, 327–329
external causation and reflex arc
central pattern generators, 331
command-driven model, 331
decerebrate preparation, 330
homunculus, 331
scratch reflex, 332
stimulus-response models, 330–331
synaptic transmission, 330
input/output approach, 320
natural bodies, 320
neuroscience, 321
Behavioral flexibility, 1–2
Bernard, C., 326
Bernstein, N., 321, 322
Bickhard, M., 182
Billard, A., 149
Bio-inspired robots, 3
Bongard, J., 181
Botvinick, M., 7, 271–289, 305
Braun, D., 161
Brown, G., 331, 332
Bruner, J., 179
Butz, M.V., 5, 129–150

C
Caithness, G., 161
Caligiore, D., 7, 237–265
Calinon, S., 149
Callebaut, W., 14
Camhi, J.M., 328, 329
Campbell, R., 182
Cannon, W., 326
Catastrophic interference, 2
Chang, Y.-H., 103, 105, 112–114
Chunking, 2
Cisek, P., 244
Clifton, R., 199
Cognitive/computational constraints, 181
Cohn, D., 105, 113, 115–117, 122
Compositionality, 48–49
Concrete operational period, 179
Contingency, 66–67
Continuous time recurrent neural network (CTRNN), 171
Control system
classical control theory, 323
closed loop control system, 325
controlled variable, 324
feedforward solution, 322
negative feedback control, 323
open loop system, 324
organization of, 325–327
teleology, 325
thermostat, 323
Wiener's error, 327–329
Córdova, N., 7, 271–289
Cortical and basal ganglia systems
cognition levels, 260
input/output ratio, 260
integrated view, 259, 260
interaction mechanisms, 260, 261
"inter-loop" mechanisms, 259
learning processes, 262
realm of activity, 260–261
sensorimotor/cognitive transformations, 259
striatal neurons, 260
striatum-pallidal dishinibition mechanism, 262
sub-cortical areas processing value, 262
Cortical hierarchies
dorsal and ventral streams, 245
dual route hypothesis, 244

prefrontal cortex, 245
top-down biasing, 244
TRoPICALS model
AIP-PMCl and PRR-PMCd streams, 247
architecture, 246
cortical area activation, 247, 248
dynamic neural fields, 248
guidance/biasing, 245
Hebbian-based reinforcement learning mechanism, 250
input and output, 246
PFC activation, 247, 249
PMCl activation, 247, 250
robotic set-up, 246, 247
Credit assignment
bootstrapping process, 111
individual behaviors, 109
modular architecture, 110
module activation protocol, 110
proof of convergence, 111–112
SARSA algorithm, 111
uncertainty, reward model, 112–113
Criscimagna-Hemminger, S., 160, 161
Crosstalk, 2
CTRNN. *See* Continuous time recurrent neural network (CTRNN)

D
DAC architecture. *See* Distributed adaptive control (DAC)
Daw, N.D., 102, 276
Degris, T., 23
Demiris, Y., 5, 81–95
Descartes, R., 320, 327, 332, 345
Developmental psychology, cumulative learning robots
action formation
eye-saccade, 205
image, gaze and arm maps, 203–204
integrated reaching, 201, 202
partial motor development sequence, 203
stage transitions, 203
time-line chronicling, 201
visual stimuli, 204
autonomous robots, 177
bio-inspired robotics, 178
constraints
bias/*a priori* assumptions, 180
in newborn, 181
sensory bandwidth reduction, 180
types, 181–182
developmental stages, 179–180
infant grasping, 207
infant's impressive cognitive growth, 207
LCAS approach (*see* Lift-constraint, act, saturate (LCAS) approach)
novelty as motivation, 200–201
research challenges, 205–206
Dimitrakakis, C., 6, 155–173
Distributed adaptive control (DAC), 7
adaptive layer
actions, 217
Hebbian learning rule, 218
internal data representation, 216
weight matrix, 217–218
classical and operant conditioning, 215–216
contextual layer
collector, 219–220
CS prototype, 218–219
foraging tasks, 222–223
perception-action sequences, 222
sequence, 219
STM and LTM structures, 218, 219
stored patches, 223
trajectory plots, 223
cumulative learning models, 231
intrinsic motivation, 231
Khepera robot, 221
reactive layer
actions, 217
data sampling, 216
homeostasis, 221
self-regulatory process, 221, 222
robot-based neuronal model, 215
self-contained learning system, 230
spatial information
bimodal integration approach, 229
bimodal sensor fusion, 224
egocentric sensory inputs, 224
LTM memory, 226
MC-POMPD method, 228, 229
memory smoothing, 227–228
performance and memory content, 227
sequence fidelity and goal fidelity, 225–226, 229
spatial scale, 227–228
Diuk, C., 7, 23, 271–289
DLPFC. *See* Dorsolateral prefrontal cortex (DLPFC)
Dopaminergic systems, 241
Dorsal neural pathway, 239, 259
Dorsolateral prefrontal cortex (DLPFC), 275
Doya, K., 102

Drescher, G.L., 65, 68, 207
Duff, A., 7, 213–231

E
Ehrenfeld, S., 5, 129–150
Embodied action perception, 5
  computational principles
    action execution knowledge reuse, 83–84
    active perception, 82–83
    hierarchical action representations, 82
    sensory action consequences prediction, 83
  HAMMER (*see* Hierarchical attentive multiple models for execution and recognition (HAMMER))
External/environmental constraints, 181–182
Extrinsic motivation, 18–19

F
Fayyad, U., 68
Ferguson, K., 167
Flexible human motor control, 5–6
Formal operation period, 179
Freud, S., 326
Friedman, N., 64
Functional magnetic resonance imaging (fMRI), 277
Furukawa, T., 60
Fuster, J.M., 239, 240

G
Galeazzi, J.M., 8, 293–315
Galileo, G., 320, 345
Gasser, M., 180
Generalisation, 2
Georgopoulos, A., 297
Gershman, S., 101
Giszter, S., 48
Goal-directed behavior, 8
Goldberg, M., 189
Goldszmidt, M., 64
Gomez, G., 188
Goodman, N., 64
Graziano, M.S.A., 137
Grupen, R., 18, 39
Guestrin, C., 15

H
Haas, H., 59
Haber, S.N., 253
HAMMER. *See* Hierarchical attentive multiple models for execution and recognition (HAMMER)
Hartland, C., 222
Hart, S., 18
Haruno, M., 149
Hebbian learning process, 186, 247
Hengst, B., 282
Herbort, O., 5, 129–150
Hierarchical attentive multiple models for execution and recognition (HAMMER), 5
  action-execution, 84, 85
  action-recognition, 84–85
  complex inverse models, 85–86
  high-level inverse model, 87–88
  inverse-forward model pair, 84
  learning
    by demonstration, 90–91
    motor babbling, 89–90
    OSILA, 91–92
    SCFG, 92–94
  primitive inverse models, 86–87
  schematics, 85
Hierarchical brain and behavior
  bio-behavioral literature, 240–242
  cognitive neuroscience literature, 239–240
  computational literature, 243
  empirical literature, 243
  limitations, 264–265
  principles, 263–264
  unsupervised learning, 264
Hierarchical knowledge accumulation
  behavioral learning, 215
  cumulative learning, 214
  DAC (*see* Distributed adaptive control (DAC))
  definitions, 214
  foraging tasks, 214
  machine learning, 215
  multi-modal integration, 230
  perceptual learning, 215, 216
  rational decision making, 214
Hierarchical motor function, brain
  architecture, 296
  brainstem, 293
  cortical motor areas, 294
  forward model, 296
  hand-centred frame, 295
  high-level motor programs, 301–305
  image sequences, 314
  inverse model, 297
  low-level motor primitives, 298–301
  motor cells, 296

neurons, 295
PPC, 312
PRR, 312
trace learning rule, 313
VisNet architecture, 313
visual target object, 312
Hierarchical reinforcement learning (HRL), 4, 7–8. *See also* Behavioral hierarchy
abstract action, 17
actor-critic RL architectures, 273
bottleneck states and temporal abstraction, 286–288
closed-loop policies, 17
computational approaches, 273
elements, 16
exploitation and exploration, 16
getting to work skill, 274
hierarchical behavior, 273
humans, bottleneck states, 282–285
large-scale problems, 16–17
option discovery problem, 280–282
option model
creation, 19
function, 17
goal states, 18
reward signals, 18–19
options framework, 274
policies, 16
potential neural correlates
brain's ability, 274
Casino Task, 277
DLPFC, 275
dopaminergic neurons, 278
human behavior, 274
human brain, 278
OFC, 276
option-specific policy, 275
pseudo-rewards, 278–280
reward-prediction errors, 276
sample trial, 278
slot-machine level, 277
state abstraction, 273
stochastic optimal control, 16
temporal abstraction, 273
Homeostasis, 326
Honey, C., 58
HRL. *See* Hierarchical reinforcement learning (HRL)
Hülse, M., 6, 177–208
Human motor control
artificial neural networks, 172
Bayesian models, 172
interference and generalization, biological systems
Bayesian model, 162
catch trials, 160
CNS, 159
equilibrium point hypothesis, 159
internal model hypothesis, 160
motor memories, decay of, 162
structural learning, 163
memory systems, 173
motor redundancy, 134–135
multiple motor tasks, 173
multi-task learning
decision theory view, 164–166
motor control models, 168–172
reinforcement learning, 166–168
neurobiological correlates, 172–173
procedural memories
consolidation, 158
experimental paradigms, 157
eyelid conditioning, 158
finger tapping task, 157
memory trace, 158
novel motor skills, 156
off-line learning, 158
retroactive and proactive interference, 159
savings, 158
rewards, 173
sensory redundancy
multi-modal goals, 133–134
multi-modal information, 132–133
uncertainty, 135–136
Human primates, 238
Human visuomotor behavior
actor-critic-type architecture, 102
behavioral goals, 100
composite tasks, 103
continual learning, 102
credit assignment, modular behaviors
bootstrapping process, 111
individual behaviors, 109
modular architecture, 110
module activation protocol, 110
proof of convergence, 111–112
SARSA algorithm, 111
uncertainty, reward model, 112–113
independent modules, 101
learning module activation, 107–109
Markov decision process, 103, 122
modular learning model, 101
multiple modules, individual task solutions, 104–107
on-policy learning method, 121
Q-learning rule, 104
RL algorithms, 100

SARSA, 104
sequential perception and action, 100
simulation results
individual learners, 113
learning progress, 114
multi-tasking problem, 113
predator-prey type problem, 113
Q values, 114–115
single-agent problem, 115–118
walkway navigation task, 118–121
temporal difference learning, 104

I
Integrated hypothesis, 7
Intelligent adaptive curiosity (IAC), 71
Internal globus pallidus (GPi), 251
Intrinsic motivation, 18–19
Inverse-forward model pair, 84. *See also* Hierarchical attentive multiple models for execution and recognition (HAMMER)
Irani, K., 68

J
Jaeger, H., 59
Johnson, M., 69
Jonsson, A., 22–25, 34, 282

K
Kalaska, J.F., 244
Kantak, S., 161
Kaplan, F., 200
Keil, F., 180
Knowlton, B.J., 252
Konidaris, G., 4, 13–40
Krakauer, J., 161, 162
Kuipers, B., 18, 23, 63–79

L
Lakoff, G., 69
Lashley, K.S., 272, 295, 305
Law, J., 6, 177–208
LCAS approach. *See* Lift-constraint, act, saturate (LCAS) approach
Lee, K., 5, 81–95
Lee, M., 6, 177–208
Lift-constraint, act, saturate (LCAS) approach, 6
"Act" stage, 183
constraint lifting, 182, 183
content-neutral methodology, 182
developmental stages transition, 182
global states, 182–183
hand/eye interaction
behavioral patterns, 198
early stage trace, 196
exploratory arm motions, 197
final stage trace, 196
growth rates of saccade behavior, 196, 197
intermediate stage trace, 196
kinaesthetic sense, 198–199
proprioception, 195
saccade learning, 195–196
tactile and somatic sensing, 199
tactile sensor, 198
visual gaze space, 198
representations, 183–184
sensory-motor mapping model (*see* Sensory-motor mapping model)
Lightweight options, Pinball Task
goals, 30–31
performance, 32–33
sample solution trajectories, 33
skill chaining, 31–32
Li, L., 273
Limbic cortico-striatal loop, 241
Lindley, D.V., 165
Liu, Y., 15
Locally weighted learning (LWL), 170
Locally Weighted projection regression (LWPR), 170
Lonini, L., 6, 155–173
LWL. *See* Locally weighted learning (LWL)
LWPR. *See* Locally weighted projection regression (LWPR)

M
Macros, 17
Maes, P., 207
Mahadevan, S., 17, 29, 39, 167
Mandler, J., 69
Mannella, F., 7, 237–265
Marcos, E., 7, 213–231
Markov decision process (MDPs), 69–70, 103, 122, 166
Mathews, Z., 231
Maturational constraints, 181
McCallum, A.K., 34
McGovern, A., 281
MDPs. *See* Markov decision process (MDPs)
Mehta, N., 15
Menache, I., 282

Metta, G., 201
Meuleau, N., 122
Miller, E.K., 272
Mirolli, M., 1–9, 19
Mixture of experts (MEX), 169
Modayil, J., 65
Modularity. *See* Behavioral hierarchy
Modular modality frame (MMF) model, 6
  body state maintenance, 143–145
  exemplar performances, 145–147
  kinematic arm chain, 142
  limb orientations, 143
  movement planning, 145
  reference frame, 142
Modular network self-organizing map (mnSOM), 60
Modular selection and identification for control (MOSAIC), 148, 170
Module activation, 107–109
Morgan, R., 161, 162
MOSAIC. *See* Modular selection and identification for control (MOSAIC)
Motor redundancy, 6
MTRNN. *See* Multiple timescale recurrent neural network (MTRNN)
Mugan, J., 18, 23, 63–79
Multimodal arm control models
  body schema, 130
  brain, modularity and hierarchy
    modular representations, 136–137
    planning and control, 137–138
  dexterous skills, 130
  modular and hierarchical models
    MMF, 142–147
    SURE REACH, 138–142
  motor control, computational problems
    motor redundancy, 134–135
    sensory redundancy, 132–134
    uncertainty, 135–136
Multiple cortical-striatal macro-loops, 240
Multiple timescale recurrent neural network (MTRNN), 4
  BPTT algorithm, 54
  continuous time characteristics, 53
  goal states, 56–57
  incremental learning, 56
  learned behavior sequences, 54–55
  position-dependent changes, 55–56
  visuo-proprioceptive (VP) state, 53–54
Multi-task learning
  decision theory view, 164–166
  motor control models, 168–172
  reinforcement learning, 166–168
Mussa-Ivaldi, F., 169

N
Negative feedback, 8
Nehmzow, U., 206
Neto, V., 206
Neumann, G., 31
Neural network models, 8
  delayed reward signal
    correct temporal order, 309
    Hebb rule, 310
    high level movement selector cells, 310
    motor primitives, 309
    network architecture, 309
    synaptic connections, 308
    temporal difference learning, 308
  hierarchical motor function, brain
    architecture, 296
    brainstem, 293
    cortical motor areas, 294
    forward model, 296
    hand-centred frame, 295
    high-level motor programs, 301–305
    image sequences, 314
    inverse model, 297
    low-level motor primitives, 298–301
    motor cells, 296
    neurons, 295
    PPC, 312
    PRR, 312
    trace learning rule, 313
    VisNet architecture, 313
    visual target object, 312
  novel movement sequence
    environmental context, 305
    motor activity, 306
    motor primitives, 305
    network architecture, 306, 307
    problem of serial order in behavior, 305, 308
Niv, Y., 7, 271–289
Nolfi, S., 53
Non-human primates, 238

O
OFC. *See* Orbitofrontal cortex (OFC)
Ognibene, D., 5, 81–95
One-shot imitation learning algorithm (OSILA), 91–92
Orbitofrontal cortex (OFC), 275
OSILA. *See* One-shot imitation learning algorithm (OSILA)
Osu, R., 161, 163
Oudeyer, P.-Y., 71, 200, 206

P
Paine, R., 53, 58
Parietal reach region (PRR), 312
Parr, R., 17
Pavlov, I., 335
Pavlovian conditioning, 335–337
Perceptual meaning analysis, 69
Pfeiffer, R., 181
Piaget, J., 63–65, 179, 180, 207
Picket, M., 281
Pierce, D., 65
Poeppel, D., 53, 58
Policies, 16
Posterior parietal cortex (PPC), 312
Powers, W.T., 332
PPEs. *See* Pseudo-reward prediction errors (PPEs)
Preoperational period, 179
Prescott, T., 206
Procedural memory, 6, 156
Pseudo-reward prediction errors (PPEs), 279

Q
QLAP. *See* Qualitative learner of action and perception (QLAP)
Qualitative learner of action and perception (QLAP), 4–5
  action plans
    image schemas, 69
    learned representations, 71
    Markov decision process, 69–70
    models to plans conversion, 70
    qualitative variable and values, 68
  contingency, 66–67
  cyber security, 78–79
  description, 65
  evaluation
    learned representations, 75–77
    Open Dynamics Engine, 72–73
    QLAP *vs.* tile coding performance, 75–77
    supervised learner, 74–75
  event identification
    improving model, 67–68
    model creation, 67
  exploration, 70–71
  learned representations
    generate-and-test approach, 77–78
    type system, 78

R
Recurrent neural network with parametric biases (RNNPB), 4
  back-propagation through time algorithm, 50
  behavior recognizer and generator, 50–51
  learned movements, 50, 51
  PB regression, 51–52
Redgrave, P., 241
Reflex arc
  central pattern generators, 331
  command-driven model, 331
  decerebrate preparation, 330
  homunculus, 331
  scratch reflex, 332
  stimulus-response models, 330–331
  synaptic transmission, 330
Reinforcement learning (RL)
  actor-critic-type architecture, 102
  behavioral goals, 100
  composite tasks, 103
  continual learning, 102
  credit assignment, modular behaviors
    bootstrapping process, 111
    individual behaviors, 109
    modular architecture, 110
    module activation protocol, 110
    proof of convergence, 111–112
    SARSA algorithm, 111
    uncertainty, reward model, 112–113
  independent modules, 101
  learning module activation, 107–109
  Markov decision process, 103
  MDPs, 122
  modular learning model, 101
  multiple modules, individual task solutions, 104–107
  on-policy learning method, 121
  Q-learning rule, 104
  SARSA, 104
  sequential perception and action, 100
  simulation results
    individual learners, 113
    learning progress, 114
    multi-tasking problem, 113
    predator-prey type problem, 113
    Q values, 114–115
    single-agent problem, 115–118
    walkway navigation task, 118–121
  temporal difference learning, 104
Rescorla, R., 218
Reservoir computing, 59
Reuse of knowledge, 2
Reward prediction errors (RPEs), 279
Ribas-Fernandes, J., 7, 271–289
Ringwald, M., 7, 213–231
RL. *See* Reinforcement learning (RL)

RNNPB. *See* Recurrent neural network with parametric biases (RNNPB)
Robertson, E., 159
Rolls, E.T., 295, 296, 300, 301, 303–307
Rosenblueth, A., 323, 326
Rothkopf, C.A., 5, 6, 99–123, 155–173
Russell, S., 17, 106

S
Sakai, K., 48
Sánchez-Fibla, M., 7, 213–231
Santamaria, J., 167
SARSA, 104, 111
SCFG. *See* Stochastic context free grammars (SCFG)
Schank, R.C., 272
Schapiro, A., 7, 271–289
Schmidhuber, J., 18, 19, 24, 71, 200
Schultz, W., 278
Schwartz, A., 281
Self-organized functional hierarchy, 4
  compositionality, 48–49
  generalization and segmentation, 59
  mnSOM, 60
  motor primitives, 48
    gated modular networks, 49–50
    RNNPB (*see* Recurrent neural network with parametric biases (RNNPB))
  multiple spatial scales, 57–58
  multiple timescales
    hierarchical functional differentiation, 58
    MTRNN (*see* Multiple timescale recurrent neural network (MTRNN))
    neural synchrony, 52
    and spatial organization, 58–59
    temporal integration windows, 53
  schema theory, 48
  topological structure, 60
Sensorimotor mapping, 1, 239, 259
Sensorimotor unsupervised redundancy resolving architecture (SURE_REACH), 6
  extrinsically defined constraint, 140
  Hebbian learning, 139
  kinematic model, 139
  motor commands, 141
  motor controller, 139
  population-encoded representation, 141
  proprioceptively defined constraint, 140
  visual feedback loop, 141
Sensory-motor constraints, 181
Sensory-motor mapping model
  adaptation and plasticity, 193
    local adaptation, 194
    re-calibration and realignment, 194–195
  2.5D architecture, 185
  field distributions and overlap, 185–186
  field generation
    configuration parameters, 186–187
    stimulus point, 187
    uniform size fields on regular grid, 187–188
    uniform size fields with irregular locations, 189–190
    variable sized fields on regular grid, 188–189
    variable sized fields with irregular locations, 190
  populating fields
    boundary distortion, 193, 194
    boundary fields, 191, 192
    internal node displacement, 193, 194
    local distortions and global warpings, 192
    motor babbling, 191–192
    muscle-based sensor systems, 193
  topological mapping method, 184
  two-dimensional structures, 184
Sensory-motor period, 179, 180
Sensory redundancy, 5
Serial reaction time (SRT) task, 157
Shadmehr, R., 48, 160, 161
Shea, J., 161, 162
Sherrington, C.S., 321, 329–332
Silva, B.C., 39
Simon, H.A., 13
Şimşek, O., 282
Singh, S., 34, 39, 40, 105, 113, 115–117, 122, 168
Skeletomotor system, 340
Skill chaining, 31–33
Skill-specific abstraction, 29–30
  Bayesian information criterion, 34
  Continuous Playroom
    abstraction library, 36–37
    configuration, 34–35
    learning curves, 36, 37
    task function, 35–36
  new option policy, 34
Slip speed, 328
Smith, L., 180
Smith, M., 161
Soft constraints, 182
Soni, V., 39
Spatial attention, 224–228

Spatial integration, 224–228
Sprague, N., 106, 109, 111, 116, 118, 119
Stochastic context free grammars (SCFG), 89, 92–94
Stone, P., 15
Strehl, A.L., 23
Stringer, S.M., 8, 293–315
Structured brain architecture, 238
Structure learning
  active structure learning, 23–24
  GLOBAL agent, 24
  VISA, 21–22
Sub-cortical hierarchy. *See also* Basal-ganglia hierarchies
  associative loop, 241
  basal ganglia-cortical loops, 240–241
  cross-loop mechanisms, 242
  learning processes, 241
  limbic loop, 241
  sensorimotor loop, 241
Substantia nigra pars reticulata (SNpr), 251
Sub-thalamic nucleus (STN), 251
SURE_REACH. *See* Sensorimotor unsupervised redundancy resolving architecture (SURE_REACH)
Sutton, R.S., 17, 38, 113–115, 274

T
Tani, J., 4, 47–60, 171, 172
Task decomposition, 7–8
Taylor, M.E., 15
Tedrake, R., 31, 39
Thelen, E., 182
Thoroughman, K., 48
Thrun, S., 281
Tokunaga, K., 60
Tolman, E., 222
Torrey, L., 15
Toutounji, H., 122
Transfer learning, 15
Triesch, J., 6, 155–173
Tsuji, S., 281
Turing, A., 178

U
Uncertainty, 6

V
van Seijen, H., 34
Variable influence structure analysis (VISA), 21–22
Ventral neural pathway, 239, 259
Verschure, P.F.M.J., 7, 213–231
Vigorito, C., 4, 13–40

W
Wagner, A., 218
Walker, M., 161
Walters, D.M., 304
Whitmyer, V., 182
Wiener, N., 323, 327–329
Wittgenstein, L., 339
Wurtz, R., 189
Wu, Y., 5, 81–95

Y
Yamada, S., 281
Yamashita, Y., 4, 47–60, 171, 172
Yin, H.H., 8, 252, 319–345

Z
Zimdars, A.L., 106