TOPICS IN INTELLIGENT ENGINEERING AND INFORMATICS

Radu-Emil Precup
Szilveszter Kovács · Stefan Preitl
Emil M. Petriu
Editors

Applied Computational Intelligence in Engineering and Information Technology

Revised and Selected Papers
from the 6th IEEE International Symposium
on Applied Computational Intelligence
and Informatics
SACI 2011

Springer

Topics in Intelligent Engineering and Informatics 1

Editorial Board

For further volumes:
http://www.springer.com/series/10188

Aims and Scope

This book series is devoted to the publication of high-level books that contribute to topic areas related to intelligent engineering and informatics. This includes advanced textbooks, monographs, state-of-the-art research surveys, as well as edited volumes with coherently integrated and well-balanced contributions within the main subject. The main aim is to provide a unique forum to publish books on mathematical models and computing methods for complex engineering problems that require some aspects of intelligence that include learning, adaptability, improving efficiency, and management of uncertain and imprecise information.

Intelligent engineering systems try to replicate fundamental abilities of humans and nature in order to achieve sufficient progress in solving complex problems. In an ideal case multi-disciplinary applications of different modern engineering fields can result in synergistic effects. Informatics and computer modeling are the underlying tools that play a major role at any stages of developing intelligent systems. Soft computing, as a collection of techniques exploiting approximation and tolerance for imprecision and uncertainty in traditionally intractable problems, has become very effective and popular especially because of the synergy derived from its components. The integration of constituent technologies provides complementary methods that allow developing flexible computing tools and solving complex engineering problems in intelligent ways.

Radu-Emil Precup, Szilveszter Kovács,
Stefan Preitl, and Emil M. Petriu (Eds.)

Applied Computational Intelligence in Engineering and Information Technology

Revised and Selected Papers from the 6th IEEE International Symposium on Applied Computational Intelligence and Informatics SACI 2011

Springer

Editors
Radu-Emil Precup
Faculty of Automation and Computers
Department of Automation and Applied Inf.
"Politehnica" University of Timisoara
Romania

Stefan Preitl
Faculty of Automation and Computers
Department of Automation and Applied Inf.
"Politehnica" University of Timisoara
Romania

Szilveszter Kovács
Department of Information Technology
University of Miskolc
Hungary

Emil M. Petriu
School of Information Technology and
Engineering (SITE)
University of Ottawa
Canada

ISSN 2193-9411
ISBN 978-3-642-28304-8
DOI 10.1007/978-3-642-28305-5
Springer Heidelberg New York Dordrecht London

e-ISSN 2193-942X
e-ISBN 978-3-642-28305-5

Library of Congress Control Number: 2012932488

Printed on acid-free paper

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

Preface

This volume presents a next generation of computational intelligence techniques that will allow future engineering and information technology applications that compute with words, or involve evolving intelligent systems, co-evolution, nature-inspired optimization and artificial immune systems.

The volume includes a selection of extended papers presented at the 6th IEEE International Symposium on Applied Computational Intelligence and Informatics SACI 2011, held in Timisoara, Romania, from 19 to 21 May 2011.

The first chapter includes extensions of a plenary lecture given at SACI 2011. Hluchý et al. argue in "Towards More Realistic Human Behaviour Simulation: Modelling Concept, Deriving Ontology and Semantic Framework" in favor of semantic methods and approaches to human behavior modeling.

Preitl et al. treat in "2-DOF and Fuzzy Control Extensions of Symmetrical Optimum Design Method. Applications and Perspectives" theoretical results concerning the Symmetrical Optimum method, linear two-degree-of-freedom and fuzzy control extensions, perspectives and applications. Speed and position control of rapid plants are given as applications.

Georgescu and Kinnunen deal in "Mixed Multidimensional Risk Aversion" with the risk aversion of an agent in front of a situation of uncertainty with many risk parameters. A general model of risk aversion is probabilistically described and other models are possibilistically described in this chapter.

Gaskó et al. discuss in "Strong Berge and Strong Berge Pareto Equilibrium Detection Using an Evolutionary Approach" strong Berge and strong Berge Pareto equilibria are important refinements of the Nash equilibrium. An evolutionary technique based on non-domination is suggested in order to detect these equilibria.

Butka et al. propose in "A Proposal of the Information Retrieval System Based on the Generalized One-Sided Concept Lattices" conceptual models based on the generalized one-sided concept lattices, which are locally created for subsets of documents represented by object-attribute table. These local concept lattices are combined to one merged model using an agglomerative clustering algorithm based on the descriptive representation of particular lattices.

József Tick presents in "Visualization and Simulation Tool for Analyzing P-graph based Workflow Systems" the concept of visualization and simulation tool supporting workflow analysis and examination. It is proved that the analysis based on this concept makes the determination of resource constraints, bottlenecks and redundancies possible, and it enables more efficient operation.

Johanyák and Papp examine in "Benchmark Based Comparison of Two Fuzzy Rule Base Optimization Methods" the performance of several fuzzy systems obtained by applying two different optimization methods, the cross-entropy

method and a hill-climbing based method. The two methods are compared in four benchmarking problems.

David et al. discuss in "Three Evolutionary Optimization Algorithms in PI Controller Tuning" three evolutionary optimization algorithms employed in the optimal tuning of proportional-integral (PI) controllers dedicated to a class of second-order processes with an integral component and variable parameters. The PI controllers are tuned such that to ensure a reduced sensitivity with respect to the parametric variations of the small time constant of the process.

Gal and Terdik propose in "Statistical Analysis of Next Generation Network Traffics Based on Wavelets and Transformation ON/(ON+OFF)" several models for the qualitative and quantitative evaluation of physical phenomenon supervened on different OSI layers at the routers and switches. Aspects concerning the common usage of wavelet and ON/(ON+OFF) transformations in network traffic analysis are evaluated.

Eredics and Dobrowiecki show in "Data Cleaning and Anomaly Detection for an Intelligent Greenhouse" that the effectiveness of greenhouse control can be improved by the application of model based intelligent control. Problems of cleaning the measurement data collected in a well instrumented greenhouse are discussed, and solutions for various kinds of missing data and anomaly detection problems are offered.

Hajjar and Hamdan present in "Clustering of Interval Data Using Self-Organizing Maps – Application to Meteorological Data" a self-organizing map to do unsupervised clustering for interval data. The map uses an extension of the Euclidian distance to compute the proximity between two vectors of intervals where each neuron represents a cluster.

Chirilă and Crețu suggest in "A Set of Java Metrics for Software Quality Tree Based on Static Code Analyzers" a fault density based quality model that relies on static source code analyzers and on a set of language specific metrics. The fault ratio for each static analyzer rule is computed.

Andrea Tick presents in "VLearning, a New Direction for eLearning Challenges" the development the vLearning technology. The most significant components and challenges of this technology are analyzed.

Kiss and Tevesz propose in "A Receding Horizon Control Approach to Navigation in Virtual Corridors" the Global Dynamic Window Approach with Receding Horizon Control (GDWA/RHC. The concept of virtual corridors is presented, and it is shown that this concept reduces the computational cost of navigation function evaluation.

Caprita and Popa offer in "Multithreaded Peripheral Processor for a Multicore Embedded System" a peripheral processor architecture implemented in multithreaded technology. This processor is able to handle more tasks concurrently.

Amaricai et al. present in "Using Cycle-Approximate Simulation for Bus Based Multi-Processor System-on Chip Analysis" a cycle approximate simulator for multi-processor system-on chip. This simulator has high flexibility, it offers accurate modeling of features specific to multiprocessor systems, accurate implementation of a wide range of performance metrics and power consumption estimates and high simulation speed.

Parri et al. exploit in "A Case Study on Hardware/Software Codesign in Embedded Artificial Neural Networks" the hardware/software codesign landscape in the artificial neural network problem space. Design space exploration options are discussed here to achieve better software/hardware partitions using instruction-set extensions and coprocessors.

Boraci et al. present in "Pragmatic Method to Obtain Optimal Control Laws for Small Windgenerators" a set of pragmatic methods to obtain optimal control methods of variable speed fixed blades small windgenerators. Available information values about the control object (wind speed, rotation speed, air density, blade's position in the air flow) are taken into consideration.

Szalay and Kovács consider in "Applicability of Asymptotic Tracking in Case of Type 1 Diabetes" that the treatment of diabetes mellitus can be represented by an outer control loop, to replace the partially or totally deficient blood glucose control system of the human body. The linear controllers are extended from the neighborhood of a working point to a larger subset of the state-space bounded by specific constraints.

Robu et al. develop in "Optimal Energetic Conditions for Cell Seeding of Scaffolds" computational models of biological systems formed by a cellular aggregate located on the plane surface of a biomaterial, respectively by a cellular aggregate located on a porous scaffold. The evolution of a cellular aggregate on the biomaterial's surface is simulated, and the energetic conditions that lead to uniform and rapid cell spreading are identified.

Pozna and Precup present in "Ideas on a Pattern of Human Knowledge" some ideas that concern a pattern of human knowledge based on the experimentation of causal relations. An application scenario concerning a robot integrated in a cognitive system is given.

Krizsán and Kovács offer in "Structural Improvements of the OpenRTM-aist Robot Middleware" some structural and implementation details of the OpenRTM-aist. Attractive results of the experiments done for the performance comparison of the original and extended system are included.

Horváth and Rudas propose in "Decision Support at a New Global Level Definition of Products in PLM Systems" a method that establishes a global level of the decision making on product object parameters. This chapter gives contextual chains along which communication is done from human thinking to product model entity parameter generation.

Hermann and Tomanyiczka discuss in "Autocollimator Calibration Using a Tangent Bar" a new low cost angle generator intended for the calibration of autocollimator. The description of the working principle is accompanied by a detailed calibration procedure which is based on the comparison principle.

The last chapter is an extension of another plenary lecture given at SACI 2011. Ionescu et al. offer in "Gesture Control: A New and Intelligent Man-Machine Interface" a survey of the methods and technologies used in the gesture control area. A new and intelligent user interface based on a sequence of gestures linked in a gesture language through a sign grammar is introduced and described.

The editors are grateful to the authors for their excellent work and to the Editors-in-Chief of this new Springer series, Prof. Imre J. Rudas and Prof. János Fodor, Óbuda University, Hungary, for the fruitful cooperation the structure, organization and contents of this book. Many thanks are also due to Dr. Thomas Ditzinger and to Mr. Holger Schäpe from Springer for their editorial assistance and their strong effort in bringing out the volume nicely in time. We do hope that this first volume in the Springer series on Topics in Intelligent Engineering and Informatics will be appreciated by the readers.

Radu-Emil Precup
Szilveszter Kovács
Stefan Preitl
Emil M. Petriu

Contents

Towards More Realistic Human Behaviour Simulation: Modelling Concept, Deriving Ontology and Semantic Framework

Ladislav Hluchý[1], Marcel Kvassay[1], Štefan Dlugolinský[1],
Bernhard Schneider[2], Holger Bracker[2], Bartosz Kryza[3], and Jacek Kitowski[3]

[1] Institute of Informatics, Slovak Academy of Sciences,
Dubravska cesta 9, 845 07 Bratislava, Slovakia
hluchy.ui@savba.sk, marcel.kvassay@savba.sk,
stefan.dlugolinsky@savba.sk
[2] EADS Deutschland GmbH, Landshuter Straße 26,
85716 Unterschleißheim, Germany
bernhard.schneider@cassidian.com,
holger.bracker@cassidian.com
[3] Academic Computer Centre CYFRONET,
University of Science and Technology in Cracow,
ul. Nawojki 11, 30-950 Kraków, Poland
bkryza@agh.edu.pl, kito@agh.edu.pl

Abstract. This chapter argues in favour of semantic methods and approaches to human behaviour modelling. Semantic perspective can provide a seamless bridge between theoretical models and their software implementations, as well as contribute towards elegant and generic modular structure of the resulting simulation system. We describe our work in progress regarding highly realistic models of human behaviour and the impact of ontological reasoning on real-time simulations. We illustrate our approach in the context of the EDA project A-0938-RT-GC EUSAS, where we plan to implement it.

1 Introduction

The principal dilemma facing human behaviour modellers is how to achieve a reasonably realistic approximation of human behaviour while still preserving the simplicity and elegance of the underlying theoretical models. In this chapter we build on our earlier work [1] concerning the complexity of highly realistic behaviour models, and validate our approach by modelling the emergence of collective aggression in the context of the EDA project A-0938-RT-GC EUSAS.

The EUSAS project deals with asymmetric security threats in urban terrain and aims at improving and combining mission analysis capabilities with virtual

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 1–17.
springerlink.com

training of security forces in a highly realistic 3-D cyber environment. This will be achieved by a detailed modelling and simulations of the behaviour of individuals and crowds on the basis of latest findings deriving from psychology.

During simulations, behaviour model components are to be coordinated through an intermediation infrastructure and learning functionality. The main idea of the project is to use these common elements (behaviour models, intermediation infrastructure and learning functionality) for both virtual training and mission analysis.

From the technical perspective, the PECS reference model will serve as the modelling basis. The acronym stands for Physical conditions, Emotional state, Cognitive capabilities and Social status [2]. According to [3], "PECS is a multipurpose reference model for the simulation of human behaviour in a social environment," with emphasis on the "emergent behaviour… typical of the formation of groups and societies". Following the general system theory, PECS agents are structured into input, internal state and output. Input is represented by the Sensor and Perception components, output by the Behaviour and Actor components, and the internal state by the Social Status, Cognition, Emotion and Physis components. These components are interconnected by an intricate network of causal dependencies and information flows. The global state transition function of such a PECS agent consists of a collection of the state transition functions of its individual components. Their mathematical form is not constrained, giving the modeller an almost total freedom in shaping the agent behaviour. In this way, the PECS reference model serves as a blueprint for a whole class of systems with a common deep structure that can be adapted to individual modelling needs by filling in the provided architectural slots (components) with state variables and their state transition functions, and by leaving out the slots that are not needed.

A more detailed description of the PECS reference model is provided in [2, 3]. Broadly following [1], the remaining portion of the Introduction recapitulates how we filled the PECS reference frame with content derived from the modelling requirements of the EUSAS project. In Section 2 we show how these elements can be used to model the emergence of collective aggression. In Section 3 we describe the salient features of our semantic intermediation framework with particular focus on the intermediation ontology, which represents its core. Finally, in Section 4 we experimentally validate our approach on the computational model of collective aggression presented in Section 2.

Human Information Processing. In simulation models aiming at representing human behaviour, the main challenge is to capture the relevant aspects of human information processing at the right level of granularity. This includes information intake, construction and update of a human being's mental world view, decision making and behaviour control. To date, there is a variety of attempts to model human information processing and behaviour control. Some focus on the interplay between emotion and the quality of human decision making. Belavkin [4] investigated effects of emotion during problem solving based upon ACT-R cognitive architecture. Marinier [5] combined a cognitive theory of behaviour control and a theory of emotion in a unified computational model. Besides, there are approaches

to model arousal's impact on memory. Cochran [6] proposed a framework for emotion to be included in an integrated cognitive architecture that originally did not account for emotions.

Our global model consists of five main modules: *Perception, Memory, Behaviour Control, Acting* and *Stress*. Each module comprises a set of cognitive processes. Fig. 1 shows the flow of information in a single simulation step, starting from the environment and flowing through the different modules.

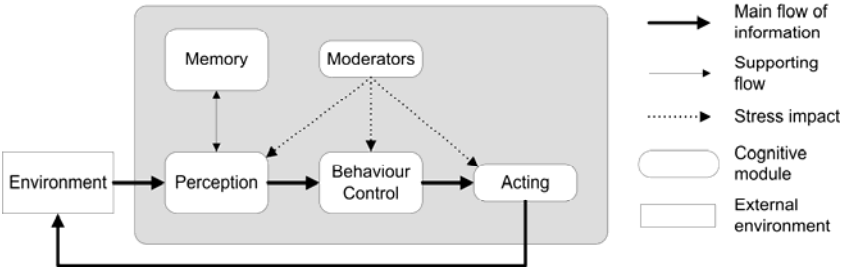


Fig. 1. Modules and flow of information.

Stress and Rule Sets. We consider three major factors that seem to influence the human capability to process information most: emotional arousal, exhaustion and time pressure. Emotional arousal is expressed by a set of single state variables, e.g. *Fear*. The modelling of the dynamics of an agent's fear is founded on the *Theory of Cognitive Appraisal for Emotions* described by Cañamero [7]. The value of the state variable *Fear* decreases continuously over time if the agent does not recognize any fear-inducing stimulus in his environment. If the agent receives information on critical events near him, the value of the state variable increases discretely. This can be seen as cognitive evaluation of a perceived danger.

The degree of exhaustion of an agent is suggested to be proportional to the span of time that the agent follows his mission. The cognitive resources parameter expresses the maximum cognitive performance of an agent at a given time. The amount of cognitive resources is connected to the degree of exhaustion: as the degree of exhaustion rises, the amount of cognitive resources decreases continuously. Computation of time pressure shall take into account the number of tasks to be handled by the agent, and the ratio between the estimated time to task completion and the maximum available time to completion per task waiting for execution.

As shown in Fig. 1, the proposed model consists of different cognitive modules, each responsible for performing a different set of tasks. Each of these tasks can be performed by real human beings in different ways, for example quick, easy and vague or time-consuming but precise. The agent's performance depends on the requirements of the current situation and the agent's internal state. In order to take this into account, rule systems were introduced, which are based on the concept of classifier-systems (see [8]). Accordingly, a set of rules is specified, which relates the way of information processing in mental processes to the influencing factors: cognition intensity, time pressure and emotional arousal. The rule system *RS*

which is connected to a generic cognitive process is formally defined as set of quadruples *{(T, S, I, R)}* with the following interpretation: for a given cognitive task *T*, a logical rule *R* assigns a strategy *S*. The parameter *I* defines a set of concrete values for the influencing factors (the description of *R* contains correspondent variables). In the choice of a strategy, the rule including the most restricting conditions according to given requirements is preferred. A strategy, which is chosen to perform a task, is said to be active. In the case of time pressure, quick strategies should be preferred for cognitive tasks connected to a rule system. Hence, only simple problem solving strategies are considered by the agent, if the cognitive resources are low. Otherwise, complex and precise strategies are preferred.

Memory. The *Memory* module shall be responsible for storing and providing the set of cognitions in the agent's mental world view. Cognitions of an agent are mental representations of real world objects as well as of the relations between them, such as *approaching, communicating, threatening,* etc. Besides, the mental world view contains background information relevant for the agent, e.g. mental representations of all known buildings and streets in the environment. The agent uses two different kinds of mental representations: observations and expectations. Observations are actually perceived objects that are transformed to representations by the perception process. Expectations are generated by mental simulation and represent possible future states of real world objects.

Perception. The *Perception* module is mainly responsible for creating mental representations of perceived objects in the environment, including characterizing information like position, type, etc. The perception process compares single perceived objects to accessible observations and expectations already stored in memory in order to identify the perceived objects. Perceived objects for which no mental representations exist in memory – that means, they are seen for the first time – are marked as "new objects". This allows emulation of a possible error: confusion of perceived objects in the agent's mental world view. A filter process deletes mental representations with low significance in terms of low salience and low relevance depending on the cognitive resources of the agent. Salience means the accentuation of a stimulus out of its context, which makes this stimulus more easily accessible for the human consciousness. A salient stimulus automatically attracts attention. Perceptions with low salience and relevance are not deleted, if the amount of cognitive resources is high enough, that means a filter threshold decreases with increasing cognition intensity. In consequence, different perceptions can be important for the agent depending on the situation.

Behaviour Definition. The model elements described above enable us to capture the emergence of collective aggression, as we show in Section 2. They also help us model errors in human cognition and decision making: the delay of information intake, mental confusion of real world objects, disregarding important information and the resulting inaccuracies in the mental world view, or the selection of inappropriate problem solving strategies, behaviour patterns and actions to execute.

2 Computational Model of Collective Aggression

This section describes a social-psychological model of collective aggression emergence that is based on empirical psychological findings. The model intends to enable an analyst to investigate the evolvement of missions especially in the environs of stabilization operations based on the simulated interplay between soldiers and civilians. This allows for example to analyze the dependencies between the soldiers' behaviour and the escalation or de-escalation of a situation.

This aggression emergence model focuses on modelling collective aggression on the civilian side. In Summers [9], collective violence is defined as follows: "'collective violence' refers to situations in which people are harmed by the joint contributions of others. The number of perpetrators can range from a small group to an entire society. The number and type of victims can also range widely." Thus, collective violence describes the instrumental use of violence by people who identify themselves as members of a group against another group or set of individuals, in order to achieve political, economic or social objectives.

The model characterizes civilian agents by several personality factors and internal processes that generate a situation-dependent behaviour driven by their motivational and emotional state. The modelling is persistently based on empirical findings from the psychological research on aggression. Basic literature was Berkowitz [10], Prentice-Dunn et al. [11], and Staub [12].

The aggression model contains three basic processes which determine the individual level of aggression (see Fig. 2): (1) Needs and emotions influence the level of aggression, (2) a process of de-individuation decides how strongly personal norms of anti-aggression come into play, and (3) a process of social influence forces agents to adjust their behaviour according to the role models from their in-group with regard to emotions and readiness for aggression.

Following the blocks in Fig. 2, the aggression model can be described as follows. The occurring events or actions are assessed by the agent (*event assessment*). Actions of other agents refer here for example to actions performed by the soldiers like pacifying, threatening or defending. Of course, actions performed by other civilians are evaluated as well. All these actions may evoke fear, anger and arousal, depending on the cognitive assessment of the situation.

The *social influence by other agents* is modelled according to Latané's formula of strength, immediacy and number of other agents [13]. This means that the social influence is exerted by the sum of the influence strengths and the proximity of each in-group agent. *Fear*, *anger* and *readiness for aggression* are related to the social influence of agents of the in-group of the agent. This means, for example, that by observing other agents with high fear, the agent's own fear will increase. If the agent observed has a certain social status, like being a respected mayor, the "strength" in Latané's formula is higher, so the influence in increasing the fear of other agents in our example will also be higher.

The agent's behaviour mainly depends on his motivation. Motives are seen as psychological forces that drive an organism to show certain behaviour in order to reach a goal that is connected to the corresponding motive. At each point of time in the simulation run, the motive with the highest motive intensity determines the behaviour of an agent. This motive is called *action-guiding* or *action-leading*.

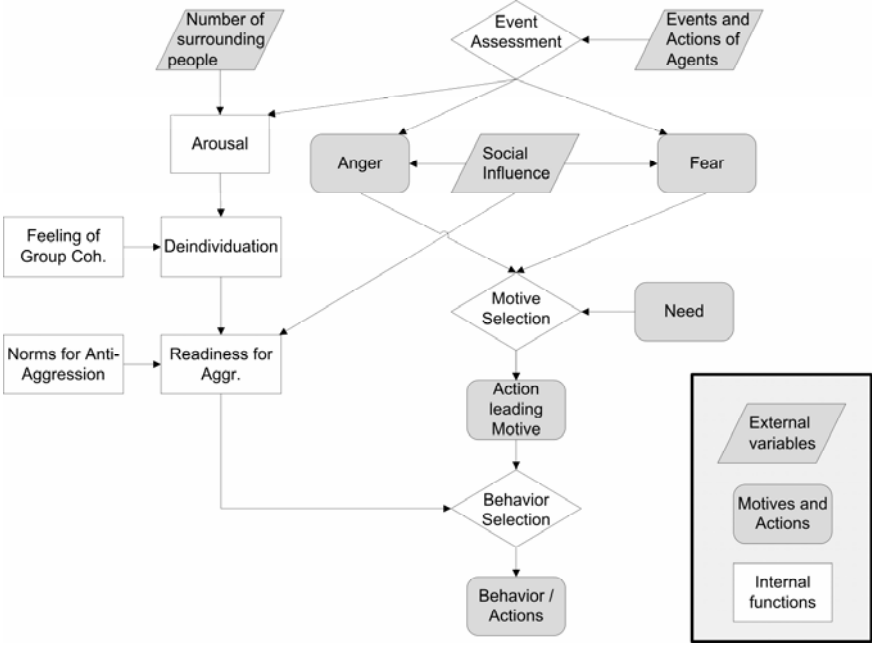


Fig. 2. Socio- psychological model for the emergence of aggression.

The modelling approach suggests a connection between motives and behaviours in the sense that each motive can, dependent on the motive intensity, trigger the execution of different predefined behaviour patterns. If e.g. the *fear*- motive is action guiding, it depends on the fear intensity if the agent will show a conscious and controlled withdrawal from a dangerous scene (in case of low motive intensity) or the agent will show a panic flight (in case of high motive intensity).

The course of a motive's intensity is defined by a continuous part and a discrete calculation part. The continuous part describes an increase or decay of the motive intensity over time regarding dynamically changing environment aspects (e.g. number of agents in the environment or the strength of social influences generated by the present agents). The discrete part of the intensity computations is based on the theory of cognitive appraisal for emotions as described by Cañamero [7]. Accordingly, individuals constantly evaluate perceptions concerning events taking place in the environment. The evaluation result can have an impact on the motive intensity by means of a discrete increase or decay of the corresponding motive

intensity. In such a manner, the detection (or assumption) of a possible threat can cause a rise of the emotional motive *fear* in the form of a sudden discrete increase.

According to Schmidt [14], a single emotional state like *fear* can, in keeping with the system- theoretical modelling approach, be modelled by a single state variable that does not directly depend on other internal states of an agent.

Besides *fear*, the motives *anger* and *need* are introduced. *Need* is a motive to satisfy a certain need (e.g. to enter a certain location in order to fulfil a certain task). The main driver for the behaviour of an agent is the *action leading motive*. The motive with the highest value at a particular point of time is selected as action leading and predetermines the main direction of the behaviour. That means for example, if the action leading motive is *fear*, the agent will – depending on the possibilities and the strength of the *fear* – try to achieve more safety.

The internal *arousal* is influenced by the event assessment and the number of people surrounding the agent. The higher the number of people surrounding the agent, the stronger is the increase of an individual's arousal.

The *de-individuation* describes a state of the agent were he considers himself as part of a crowd and not as an individual. So the higher the agent's arousal is and the more he feels like belonging to the surrounding people (intensive feeling of group cohesiveness), the higher is the de-individuation.

The *readiness for aggression* is determined by norms for anti-aggression, de-individuation, and social influence in the following way: (a) the higher the norms for anti-aggression, the lower the readiness for aggression; (b) the higher the de-individuation the higher the readiness for aggression; and (c) the more the social influence tends to aggression the higher the readiness for aggression.

It is important to note that, according to psychological theories, the execution of aggressive actions primarily depends on the existence of readiness for aggression. Without readiness for aggression, a human being, even if very angry, would not try to harm others or cause damage. We capture it in our model as follows: If *anger* is action-guiding, then the level of the readiness for aggression and the intensity of *anger* jointly determine the extent of an individual's aggressive behaviour.

These mechanisms are further influenced by various moderators, such as emotional arousal (reflecting the intensity of the agent's own emotions) and the exhaustion (reflecting the lack of physical energy). While the exhaustion moderator primarily affects the duration of the agent's actions and its speed, the emotional arousal moderator affects its sensory perception: when it crosses a certain event type-specific threshold, the events of that type are no longer perceived by the agent.

3 Semantic Intermediation Framework

As we stated in the EUSAS architecture overview in [1], the core of our intermediation infrastructure is a common ontology, which guarantees a unified semantic description of all the behaviour models based on RDF/OWL standards. The

architecture of the EUSAS system splits the behaviour-related knowledge into two parts. The static part is captured in the ontology, the dynamic one in the software code of the objects (classes) representing the ontological entities in the simulation. In the following sections we focus on the salient aspects of the EUSAS system regarding the actual or potential use of semantic methods and approaches, starting with the role and the structure of the ontology itself.

Role and Structure of Intermediation Ontology. Put briefly, the role of ontology in the EUSAS project is to support the agent-type creation from the existing behaviour elements, enhance matchmaking capabilities and context-specific object identification, and to support discovery and composition of action plans by agents based on their current motive and location in the environment. In later project phases it will also support behaviour cloning techniques. Although some attempts at supporting the environment and behaviour modelling with ontologies in multi-agent crowd simulation already exist [15, 16, 17], they do not provide a concise and universal way for application in simulation frameworks requiring both 2D and 3D simulations and complex behaviour models. Additionally, none of them covers the concepts necessary for military contexts, where specific types of weapons, actions and behaviour patterns must be modelled.

Our intermediation ontology is divided into two major parts: environmental and behavioural. The environment ontology provides the means for annotating the elements of the terrain and urban area, the dynamic objects which can be found and used by agents, weapons and control force facilities, as well as possible actions which can be performed on the objects in a given context and numeric properties about the objects such as energy cost or breakability. The behaviour ontology on the other hand provides means for annotating the different elements of agents in order to allow users to configure/build different agent types according to the requirements of a particular scenario. This entails, among others, concepts representing inputs and outputs of different behaviour model elements, or verification whether two parts can be combined in one agent type. Another important aspect of this ontology will be the modelling of typical civilian and military behaviour patterns.

The core concept which merges all the aspects of the ontology including objects, actions and motives is the ContextAction, which allows to define types of possible actions by putting constraints on the range of its properties. An example of such a concept is presented in Fig. 3, where we can see AttackWithFirearm action which is an indirect subclass of ContextAction that requires objects to be of type FireArm, subjects of type Agent and elementary actions of type EffectiveShot. The possible results include wounding or killing the agent. Such a structure allows to model actions and objects separately and to define custom action types which connect objects and actions including effects and preconditions, and to significantly limit the amount of explicit information which has to be introduced by the system, while all the valid combinations of actions, object and subject along with the numerical properties for their energy cost and effectiveness can be obtained automatically through ontological inference.

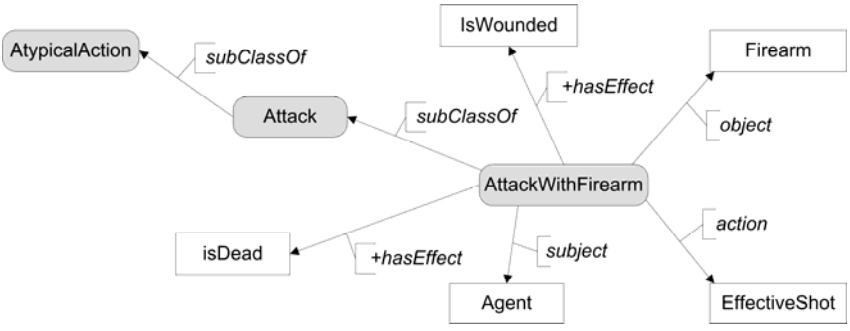


Fig. 3. Excerpts from the intermediation ontology.

User-defined Agent Types. The importance of the intermediation ontology becomes apparent when we consider the process of defining new agent types by the users who are not technical experts. As we explained in [1], we designed our agents to be compatible with the PECS reference frame, but also to be as generic as possible. With this end in view, we split the agent's internal state into two loosely structured collections. One is a set of motivational factors (motive components) broadly corresponding to human emotions (e.g. fear, anger), physical drives (thirst, hunger, etc.) and cognitive motives. The other is a set of behaviour patterns triggered by these motives. In our architecture, it is the task of Behavioural Cognition (or Behavioural Intelligence) to decide which behaviour pattern should be activated based on the inner state of the motive components. Users define new agent types by choosing appropriate motive components and behaviour patterns from a repository, and by setting their parameters, as shown in Fig. 4. Users may also add new motive components and behaviour patterns to the system, but doing so requires a certain level of technical expertise.

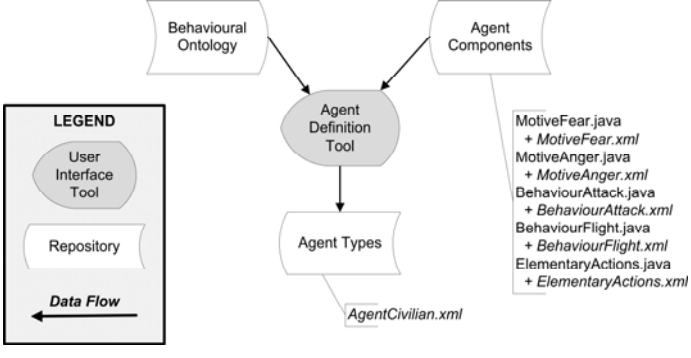


Fig. 4. Definition of New Agent Types.

The Agent Definition Tool scans all the available agent components (motives and behaviours) in a repository and offers them to the user. Each component consists of two files: a JAVA file implementing the functionality of the component,

and an XML file containing the values of its parameters. If the user includes a particular component in the new agent type, he or she will be given a chance to adjust its parameters. Since the execution of behaviour patterns during the simulation consists in generating a sequence of elementary actions (the smallest uninterruptible action units that an agent may perform subject to certain preconditions), the available repertoire of elementary actions can be considered as automatically included in each agent type. Behavioural ontology contains the details of the logical coupling between the motives and the behaviours, so it can provide the reasoning services for verifying the consistency of the new agent type:

1. Along with each behaviour pattern, the motives that trigger it must also be included in the agent definition, or else the behaviour would never be activated. This is illustrated in the figure by showing two motives – Anger and Fear – along with the two behaviours they trigger: Attack and Flight to safety.
2. Along with each component, the components on which it depends must also be included. Thus, for example, the physical energy component must always be present since the basic precondition for executing elementary actions is that the agent must have enough energy to perform them.

The resulting agent type is saved as an XML file containing references to the JAVA component classes that need to be instantiated at runtime. This helps to preserve the light-weight character of the simulation: although the repository of available agent motives and behaviours can grow large over time, only the needed components will be actually instantiated at runtime.

Supporting Behavioural Intelligence. Another task for the intermediation ontology is to support the process of activating the behaviour patterns by the motives during the simulation. If the coupling between the agent's motives and behaviours were always straightforward, then the ontological reasoning would not be needed. But our aim of realistic human behaviour modelling led us to the point where the coupling was no longer trivial. A good example is human aggression. To non-experts it might seem that aggression is triggered by anger, and the angrier we are, the more aggressively we behave. A deeper psychological inquiry, however, belies such simplistic notions. More sophisticated models need to take into account additional factors. In our model of collective aggression, anger really has to be action-leading, but the actual form of aggression is determined by another composite factor termed "readiness for aggression". This factor reflects, among other things, the agent's level of de-individuation, model influence of others, as well as his personal norms for anti-aggression and expectation of negative consequences.

In this respect we still face challenges. Our earlier experiments in [1] showed that ontological reasoning required substantial time and might not be feasible for simulations that needed to run in real time. That was unfortunately the case of our project, where the Agent-based Simulation Component is to be included in a 3-D cyber environment for the purpose of real-time virtual training of security forces. As we explained and partially validated in [1], our proposed solution is to invoke the reasoning at the initialisation stage (before the simulation starts) and store its

results in efficient in-memory structures (e.g. key-value maps) for quick access during the simulation. To design such structures for storing non-trivial reasoning results is a task that we have yet to tackle in subsequent stages of the project.

Modular Structure of Simulation Scenarios and Generic Agents. Semantic approach to software development requires adequate support in the underlying platform. JAVA programming language, besides its support of XML, also offers JAVA reflection and JAVA annotations. Although the scope of this chapter does not permit details, we shall briefly outline how we used them to streamline the structure of our simulator.

As mentioned before, we store the values of component parameters in XML files. The same applies to agent types, physical environments and simulation scenarios – each is serialised to (and deserialised from) a special kind of XML file. Just as each agent type XML refers to the XML files of its components, so each simulation scenario XML refers to the XML files of the physical environment and the agent types that are instantiated in it. Our simulator first reads the scenario XML file. Next, it loads all the referenced XML files and cascades further down the reference chain until all the needed classes are loaded. Then a "parameter override" (a special sub-phase of the simulation initialisation) takes place.

"Parameter override" concerns mainly the motive components and allows the user to adjust their parameters without losing the default values recommended by the component creators. Since components may have dozens of parameters of which typically only one or two need to be adjusted for a given simulation, we preferred to avoid making separate copies of the whole component XML files. Instead we chose to develop a new feature enabling us to specify the new values of component parameters directly in the simulation scenario XML file itself. To this end we included there an optional element corresponding to the component parameter that we wish to modify. We use a generic element for all the components and identify each component parameter by a reference string – the name of the component JAVA class, followed by the name of the parameter variable. During the "parameter override", we invoke JAVA reflection to override its default value. In this form there is nothing particularly semantic about our approach except that we use JAVA reflection, which qualifies as a kind of meta-information about JAVA classes. However, as we extended this mechanism to kindred purposes, its semantic dimension became more prominent.

This is particularly true of the dependence of some components on others. The real challenge is not when the dependence is on a unique component, for then the previous method – the reference string with the name of the component class and the name of the variable – does the job. The difficult case is when the dependence is meant to be more abstract. For instance, there can be alternative models of how the agent's available physical energy changes over time depending on the actions he has performed and suffered. Assuming the competing models can provide energy in the same units (e.g. in percents, with 100% representing maximum energy, 0% representing immobility or death), we may wish to be able to say: "I need the current percentage value of the available physical energy, whatever the name of

the component and the variable holding it." In this case, clearly, the previous kind of reference string will not work. It is here that the utility of the intermediation ontology comes to the fore. The alternative energy models can annotate their energy status variables with a Unified Resource Identifier (URI) that points to the corresponding concept in the ontology. This goes beyond the notion of a common software interface, since the ontology can describe additional attributes of the concept (e.g. its units, minimum and maximum) and by reasoning derive further information about it far beyond what could be expected from a typical software interface.

4 Experimental Validation

This chapter describes our work in progress and not all the design ideas presented above could have been experimentally verified yet. Moreover, our validation of the model of collective aggression had to be largely qualitative – on the basis of expert opinion – since it is notoriously difficult to predict the behaviour of multi-agent systems on a rigorously quantitative basis. Nevertheless, our preliminary results justify optimism regarding the utility of the proposed approach for realistic human behaviour modelling. Regarding our modular architecture based on "pluggable" agent components, we had several opportunities to appreciate its flexibility. The first came as we moved from the simple exemplary scenario, which we used to evaluate the existing simulation frameworks in [18], to the much more sophisticated model of collective aggression presented here. It turned out that our modular framework required surprisingly few adjustments to accommodate the new model. The second opportunity presented itself as we tackled the logging of events and key variables during the simulation for subsequent data analysis. After a thorough deliberation we realised that the best way to implement the logger was in the form of an agent component – as if it were just another motive like fear or anger. We were again surprised at the brevity and elegance of the resulting implementation.

Below we present our validation scenario and the findings that were the basis for our conclusion. The overall setting is captured in Fig. 5:

The numbered circles represent agents: those numbered 1-5 in the upper part are soldiers/guards; the others numbered 6-25 are civilians. The area in the top left corner represents the inside of a military base; the dark border around it is the wall. Its upper part enlarges into a watchtower manned by the guards Nos. 1-3. The pedestrian entrance is shown as a thinner wall just below the watchtower and is secured by the guards Nos. 4 and 5. The civilians move and interact with soldiers in the central portion of the figure. The bottom right shows the safety area where civilians flee when fear dominates their behaviour. Fig. 5 captures the situation after the escalation of civilian aggression, when the soldier responses (mainly warning shots in the air) triggered panic in some civilians. The panicking agents (Nos. 12, 14, 8, 15, 20, and 10) are seen flying towards the safety area (bottom right).

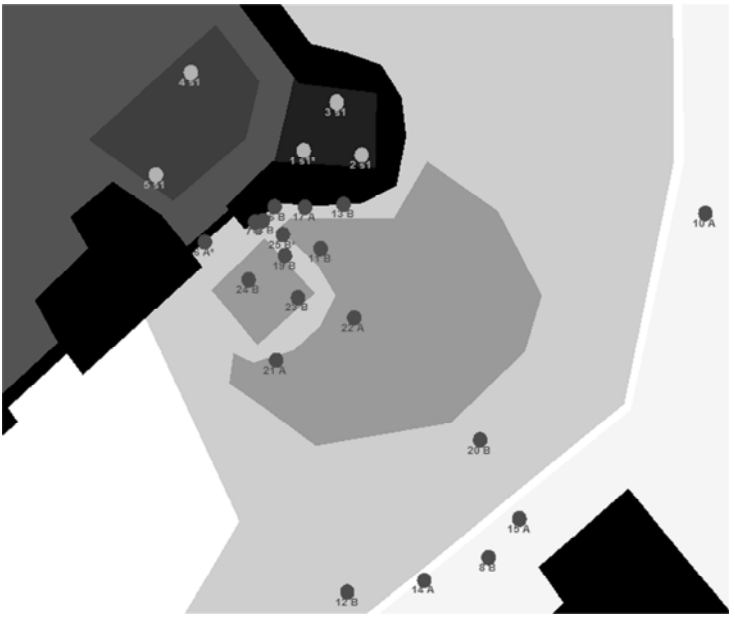


Fig. 5. Validation scenario snapshot.

Our scenario is broadly modelled along the lines of the current ISAF Mission in Afghanistan. The civilians represent daytalers and employees of the base waiting for their entry into the base. They are divided into two groups (A and B) along with their respective leaders (agents 6 and 7). At the beginning of the simulation, the civilian leaders approach the gate guards and start negotiating. Since the negotiation does not seem to lead anywhere, it results in a build-up of anger and frustration, which manifests in the aggressive behaviour of civilians towards the soldiers. The soldiers are expected to react according to the rules of engagement.

In our simulator, the interaction between civilians and soldiers is modelled as an exchange of events corresponding to performed elementary actions. These events are available to other (observing) agents subject to a filtering process through which we model their sensory perception. In our initial experiments, we had our simulated soldiers use simplified rules of engagement listed in Table 1.

Civilian actions in the table are listed in the order of increasing aggressiveness (starting with "provoke slightly") and the soldier responses in the order of increasing severity. The two are coupled through a numerical value called "Cognitive Evaluation". Each form of civilian aggression is assigned a numerical "Cognitive Evaluation" in the range <0, 100>, and each soldier response is assigned a Cognitive range consisting of min-max Cognitive Evaluations for which it is to be used.

All actions shown in the table are perceived by the observing civilians and cause discrete changes in the intensity of their motives. In this way they contribute to the overall scenario development over time. Moreover, civilians are endowed with a simple "proto-anticipation" mechanism: when a civilian agent decides to execute a

certain behaviour pattern, it first estimates the time it would normally take if the physical movement were in a straight line and without any physical obstacle along the way. If more than twice this time elapses without the agent being able to complete the chosen behaviour pattern, then the agent's anger and readiness for aggression increase, which forces the "frustrated" agent to switch to more aggressive behaviours. We term this mechanism "the civilian escalation principle".

Table 1. Simplified rules of engagement: coupling of soldier responses with civilian actions

Civilian action	Cognitive Evaluation	Standard soldier response	Cognitive Range
Negotiate (only civilian leaders)	10	Communicate calm	<0, 20)
Provoke slightly	35	Communicate warning	<20, 40)
Provoke aggressively	45	Gesticulate	<40, 50)
Threaten	55	Show of weapon	<50, 60)
Attack without weapon	70	Load gun	<60, 80)
Attack by throwing stone	90	Warning shot in the air	<80, 100>

Regarding soldiers, we treat their cognitive evaluation as a kind of motive with a few departures from the standard motive dynamics: First, in each simulation cycle each soldier reacts only to the event with the largest cognitive evaluation, and ignores all the other events. Secondly, in the preparation for a new simulation step, the Cognitive Evaluation variable is always reset to zero in order to prevent any cumulative or memory-effects (continuous dynamics) over time.

Our model for soldier behaviour stipulates a "soldier escalation principle" similar to the civilian one: if the soldiers perceive an aggressive civilian action and if the same or more aggressive civilian action has already occurred earlier in the scenario, then their response is to be one degree more severe than the standard response listed in the table. It means, for instance, that while a soldier agent responds to the first "provoke aggressively" with a "gesticulate" action, it would respond to all the repeated occurrences of "provoke aggressively" with a "show of weapon."

In order to implement the "soldier escalation principle", the soldier agents are equipped with a "proto-memory" that records the last time when each particular form of civilian aggression occurred in the scenario. This proto-memory is a kind of inverted index – a map with all the possible Cognitive Evaluations as keys. Each key points to a timestamp recording the most recent occurrence of each Cognitive Evaluation in the scenario.

In subsequent stages of the EUSAS project we plan to endow our agents with a full-fledged memory and anticipation mechanisms, as well as standard path-planning and collision-avoidance logic. However, even the simplified model presented above sufficed to validate the model of collective aggression in the relatively simple and obstacle-free environment of our validation scenario.

As we varied the parameters of civilian agents, we observed synchronised changes in their emotional (motivational) state as well as global changes in the

scenario evolution. Overall, the project modelling group felt that the observations confirmed the validity of our model. We intend to perform a deeper analysis through Data Farming (multi-parametric studies), but the theory underlying these techniques is beyond the scope of this chapter.

Multi-agent systems are known to give rise to complex patterns of global behaviour based on relatively simple rules governing individual agents. The resulting behaviour is often not derivable analytically and tends to strike observers as unusual.

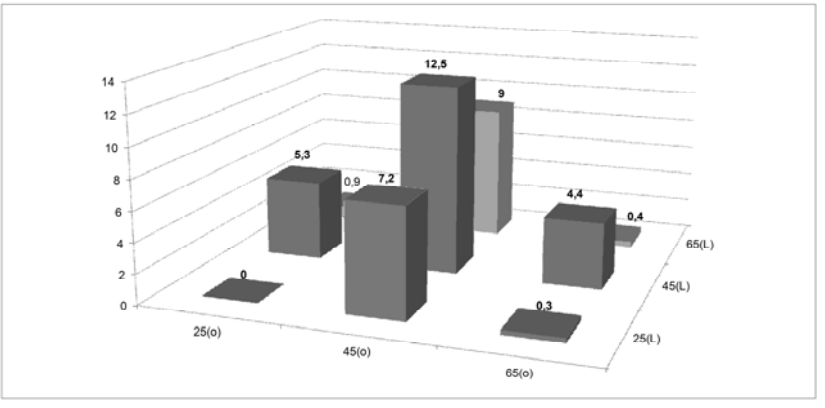


Fig. 6. Average number of panicking agents (z-axis) based on the initial readiness for aggression (RFA) of civilian leaders (L) and ordinary civilians (o).

We witnessed this as we tried to figure out which parameter settings would maximise the number of panicking agents. After an extensive experimentation we found out that the most relevant parameter was the "readiness for aggression" (RFA), but we had to differentiate between the RFA of the civilian leaders and that of the ordinary civilians. Fig. 6 shows the number of panicking agents as a function of the initial level of RFA of civilian leaders (L) and ordinary civilians (o). The number of panicking civilians is calculated as an average from ten independent simulation runs for each parameter setting. To a casual observer it is far from obvious, first, why RFA should be at all significantly correlated with the number of panicking civilians and, second, why the maximum of 12.5 panicking civilians (on average) should occur for the medium rather than the more extreme values of RFA. Due to space constraints, we can only briefly indicate the reasons here. First, panic tends to occur after severe soldier responses, such as warning shots in the air, which, by the rules of engagement, are only permitted after highly aggressive civilian actions. These, in turn, require high RFA, so RFA must not be too low. At the same time, our model stipulates that high emotional arousal (in terms of intense fear or anger) blocks the sensory perception of the agents. When RFA is high, it feeds back and fuels anger, so the agents become "blind with anger" and continue in their aggressive behaviour in spite of warning shots in the air. So, RFA must not be too high either. This is, of course, only a very brief and tentative outline of the dynamics of our model, not a comprehensive explanation. It should not be taken to imply that

panic depends only on RFA – there are in fact many other factors, which we simply kept at their default values recommended by the model developers. A more comprehensive exploration of our model in a more representative spectrum of influencing factors remains one of our key future tasks.

5 Conclusions and Future Work

We have demonstrated an approach to realistic human behaviour modelling leaning heavily towards semantic perspective and methods. Our modelling basis was the PECS reference model specifically designed to facilitate the uptake of sound sociological and psychological models of human behaviour. In Section 2 we have presented such a novel computational model of collective aggression. In Section 3 we have indicated how this model might be implemented in a modular fashion with the support of semantic elements, such as ontology, JAVA reflection and JAVA annotations. In Section 4 we have partly validated our approach experimentally on a simple, yet representative scenario in the context of the EUSAS project. In the subsequent stages of the project, we intend to validate our model and architecture on increasingly demanding scenarios and in a growing spectrum of functionalities. The most demanding of these is undoubtedly behaviour cloning – an attempt to learn and improve our models by imitating human experts. It will require a substantial enhancement of our present implementation, yet we believe our architecture has the needed flexibility and can accommodate these new demands.

Acknowledgments. This work is supported by the EDA project A-0938-RT-GC EUSAS and by Slovak Research and Development Agency under the contract No. APVV-0233-10.

References

[1] Hluchý, L., Kvassay, M., Dlugolinský, Š., Schneider, B., Bracker, H., Kryza, B., Kitowski, J.: Handling internal complexity in highly realistic agent-based models of human behaviour. In: Proc. 6th Int. Symp. on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, pp. 11–16 (2011)

[2] Urban, C.: PECS A reference model for the simulation of multiagent systems. In: Ramzi, S., Klaus, G.T., Gilbert, N. (eds.) Tools and techniques for social science simulation. Physica-Verlag, Heidelberg (2000)

[3] Schmidt, B.: Modelling of human behaviour: The PECS reference model. In: Verbraeck, A., Krug, W. (eds.) Proc.14th European Simulation Symp., SCS Europe (2002)

[4] Belavkin, R.V.: The role of emotion in problem solving. In: Proc. AISB 2001 Symp. on Emotion, Cognition & Affective Computing, Heslington, York, UK, pp. 49–57 (2001)

[5] Marinier, R., Laird, J.E., Lewis, R.L.: Computational unification of cognitive behaviour and emotion. J. Cogn. Syst. Res. 10, 48–69 (2008)

[6] Cochran, R.E., Lee, F.J., Chown, E.: Modeling emotion: Arousal's impact on memory. In: Proc. 28th Annual Conf. of Cognitive Science Society, pp. 1133–1138 (2006)

[7] Cañamero, D.: Modeling motivations and emotions as a basis for intelligent behaviour. In: Proc. 1st Int. Symp. on Autonomous Agents (Agents 1997), pp. 148–155 (1997)
[8] Booker, L.B., Goldberg, D.E., Holland, J.H.: Classifier systems and genetic algorithms. Artif. Intell. 40, 235–282 (1998)
[9] Summers, C., Markusen, E.: Collective violence. Harmful behaviour in groups and governments. Rowman & Littlefield Publishers, Lanham (1999)
[10] Berkowitz, L.: Aggression. A psychological analysis. McGraw-Hill, New York (1962)
[11] Prentice-Dunn, S., Rogers, R.W.: Deindividuation and the self-Regulation of Behaviour. In: Paulus, P.B. (ed.) Psychology of group influence. Lawrence Erlbaum Associates, Hillsdale (1989)
[12] Staub, E.: Predicting collective violence: The psychological and cultural roots of turning against others. In: Summers, C., Markusen, E. (eds.) Collective violence. Rowman & Littlefield Publishers, Lanham (1999)
[13] Latané, B.: Dynamic social impact. In: Hegselmann, R., Mueller, U., Troitsch, K.G. (eds.) Philosophy and methodology of the social sciences, vol. 23. Kluwer Academic Publishers, Dordrecht (1996)
[14] Schmidt, B.: The modelling of human behaviour. SCS-Europe BVBA, Ghent (2000)
[15] de Paiva, D.C., Vieira, R., Musse, S.R.: Ontology-based crowd simulation for normal life situations. In: Proc. Computer Graphics International (CGI 2005) Conf., Stony Brook, New York, USA, pp. 221–226 (2005)
[16] Bandini, S., Manzoni, S., Redaelli, S.: Towards an ontology for crowds description: A proposal based on description logic. In: Umeo, H., et al. (eds.) Proc. 8th Int. Conference on Cellular Automata for Research and Industry, Springer, Heidelberg (2008)
[17] Okuyama, F.Y., et al.: An ontology for defining environments within multi-agent simulations. In: Guizzardi, G., de Farias, C.R.G. (eds.) Proc. 1st Workshop on Ontologies and Metamodeling in Software and Data Engineering, Florianópolis, Brazil (2006)
[18] Laclavík, M., Dlugolinský, Š., Šeleng, M., Kvassay, M., Schneider, B., Bracker, H., Wrzeszcz, M., Kitowski, J., Hluchý, L.: Agent-based simulation platform evaluation in the context of human behavior modeling. In: Dechesne, F., Hattori, H., ter Mors, A., Such, J.M., Weyns, D., Dignum, F. (eds.) AAMAS 2011 Workshops. LNCS, vol. 7068, pp. 396–410. Springer, Heidelberg (2012)

2-DOF and Fuzzy Control Extensions of Symmetrical Optimum Design Method: Applications and Perspectives

Stefan Preitl, Alexandra-Iulia Stînean, Radu-Emil Precup,
Claudia-Adina Dragoş, and Mircea-Bogdan Rădac

"Politehnica" University of Timisoara, Department of Automation and Applied Informatics,
Bd. V. Parvan 2, RO-300223 Timisoara, Romania
{stefan.preitl,radu.precup,claudia.dragos,mircea.radac}@aut.upt.ro,
kassandra3107@yahoo.com

Abstract. This chapter treats theoretical results concerning the Symmetrical Optimum method (SO-m), linear 2-DOF and fuzzy control extensions, perspectives and applications. The theoretical results are related to the Extended SO-m (ESO-m) and the double parameterization of the SO-m (2p-SO-m) introduced previously by the authors. Digital implementation aspects are given. The applications deal with speed and position control of rapid plants in mechatronic systems with focus on electrical drives with BLDC motors and variable moment of inertia.

1 Introduction

The basic version of the Symmetrical Optimum method (SO-m) was elaborated by Kessler for an efficient (model-based) design of the controllers for servo systems. The method is characterized by the fact that in the open-loop transfer function (abbreviated with t.f.) $H_0(s)$, [1], [2]:

$$H_0(s) = C(s) \cdot P(s), \tag{1}$$

$C(s)$ – the controller's t.f. and $P(s)$ – the controlled plant t.f., the result is a double pole (by the SO-m) in the origin and the parameters of the controllers (PI(D)-type) can be computed (recalculated) online by means of compact formulas. Basically, the design situations correspond to benchmark-type model for the plant and typical controllers – of PI(D) type – eventually extended with reference filters.

This chapter treats two extensions of the SO-m introduced by the authors [3] and by their former colleague Zs. Preitl in [4] and [5] focused on obtaining better

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 19–37.
springerlink.com

dynamics of the control structure, enhancement of robustness, and enlarging of area of applications. The efficiency of controller tuning can be proved in the time domain and in the frequency domain as well. For the variation of the parameters in $P(s)$ (for example, the gain k_P in a given or approximated domain, $k_{P\min} \le k_P \le k_{P\max}$) these methods ensure the online computation of the controller parameters for a minimum guaranteed phase margin. The method found extensions in 2-DOF interpretation and fuzzy control solutions. In area of applications can be included remarkable mechatronic applications [6]. In this frame the use of brushless DC motors (BLDC-ms) is with great actuality due to its attractive driving properties [7]–[9]. For the case of servo applications characterized by Variable Moment of Inertia (VMI) the use of controllers adapted on the operating point is very attractive and necessary; for this purpose the extended versions are for a practical importance in design of robust speed controllers [10]–[12].

The chapter is structured as follows. A short overview on the SO-method is presented in section 2. Section 3 gives details on the extensions of the SO-m some implementation details. Section 4 deals with an extension for Takagi-Sugen-Fuzzy Control solution (TS-FC). Section 5 deals with a case study concerning a low-power servo application with BLDC-m. Section 6 is dedicated to the concluding remarks.

2 The Basic Version of the Symmetrical Optimum Method

In its practical form the method Symmetrical Optimum method (abbreviated SO-m) [1], [2] is applicable for plants having a pole in origin (the main case for the positioning systems) characterized by a t.f. expressed as (1) (a) or (b)

$$P(s)=\frac{k_P}{s(1+sT_\Sigma)} \quad \text{(a)}, \quad P(s)=\frac{k_P}{s(1+sT_\Sigma)(1+sT_1)} \quad \text{(b)} \tag{2}$$

(T_Σ includes the effects of small time constants of the plant). For this cases the use a PI or a PID controller, having the t.f. (2) (a) or (b):

$$C(s)=\frac{k_c}{s}(1+sT_c) \quad \text{(a)}, \quad C(s)=\frac{k_c}{s}(1+sT_c)(1+sT_c') \quad \text{(b)} \tag{3}$$

is recommended (in the case of PID controllers $T_c'=T_1$, the pole-zero cancellation). Accordingly, the closed-loop t.f. with respect to the reference input *r* $H_r(s)$, can expressed as

$$H_r(s)=\frac{b_0+b_1 s}{a_0+a_1 s+a_2 s^2+a_3 s^3}, \tag{4}$$

and the "optimal controller" is obtained imposing the conditions

$$2a_0a_2 = a_1^2,\ 2a_1a_3 = a_2^2, \tag{5}$$

with the result $a_0 = 1$, $a_1 = 4T_\Sigma$, $a_2 = 8T_\Sigma^2$, $a_3 = 8T_\Sigma^3$ and $b_0 = 1$, $b_1 = 4T_\Sigma$. The controller parameters can be computed using the compact relations

$$k_c = \frac{1}{8k_P T_\Sigma^2},\ T_c = 4\,T_\Sigma\ \ (T_c' = T_1)\cdot \tag{6}$$

The "optimal performance" guaranteed by the SO-m – viz. $\sigma_1 \approx 43\%$ (overshoot), $t_s \approx 16.5T_\Sigma$ (settling time), $t_1 \approx 3.1T_\Sigma$ (first settling time) and a small phase margin, $\varphi_r \approx 36^0$ (the main drawback of the SO-m) – are seldom acceptable, so a retuning of the controller or use of adequate designed reference filters are strongly recommended.

Efficient ways for performance enhancement are based on two extensions of the SO-m, the Extended Form of the Symmetrical Optimum Method (ESO-m), and the Double Parameterization of the Symmetrical Optimum Method (2p-SO-m).

Both methods (presented in [3] and [4] and later in [5]) are based on a generalized form of the equations (5) in the form of

$$\beta^{1/2}a_0a_2 = a_1^2,\ \beta^{1/2}a_1a_3 = a_2^2, \tag{7}$$

where β is the design parameter.

The methods are focused to fulfill an increased value for the phase margin, good (better) tracking performances and efficient disturbance rejection.

3 The Extensions of the Symmetrical Optimum Method

3.1 The Extended Symmetrical Optimum Method (ESO-m)

The Extended Symmetrical Optimum method (ESO-m) [3] is dedicated mainly to positioning systems (see the integral component in the t.f.s (1)), characterized by t.f.s in form of (1) (a) or (b). Applying the optimization relation (7) as result the characteristic t.f.s $H_0(s)$ and $H_r(s)$ will obtain the forms

$$H_0(s) = \frac{1+\beta T_\Sigma s}{\beta^{3/2}T_\Sigma^2 s^2(1+T_\Sigma s)},\quad H_r(s) = \frac{1+\beta T_\Sigma s}{\beta^{3/2}T_\Sigma^3 s^3 + \beta^{3/2}T_\Sigma^2 s^2 + \beta T_\Sigma s + 1}\cdot \tag{8}$$

The method ensures the compact design relations for the parameters, in the general form

$$k_c = \frac{1}{k_P \beta^{3/2} T_\Sigma^2},\ T_c = \beta T_\Sigma \ \ (T_c' = T_1), \tag{9}$$

which leads to significantly improved performance. Fig. 1 presents the main control system performance indices as function of the design parameter β.

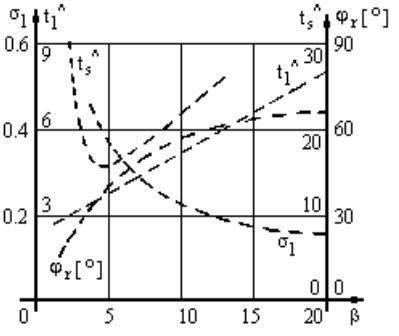


Fig. 1. Performance indices σ_1, $t_s\hat{} = t_s / T_\Sigma$, $t_1\hat{} = t_1 / T_\Sigma$ and $\varphi_r\,[^\circ]$ versus β.

The method offers also a good support in controller design for plants with variable parameters (for example, the variation of k_P in the domain $[k_{Pmin}, k_{Pmax}]$ and similarly for T_Σ) and the possibility for online computing the value of β which ensures a minimum guaranteed phase margin. This is the situation imposed by electrical drives with Variable Moment of Inertia (VMI). The recommended domain for β is $4 < \beta \le 9\,(16)$.

Useful connections of the results with those obtained by minimizing the integral quadratic performance indices are proved in [13].

3.2 The Double Parameterization of the SO-m (2p-SO-m)

The 2p-SO-m introduced in [3] and [5] is dedicated to driving systems (speed control, characterized by t.f-s in the following without integral (I) component:

$$P(s) = \frac{k_p}{(1+sT_\Sigma)(1+sT_1)}\ \text{(a)} \quad \text{or} \quad P(s) = \frac{k_p}{(1+sT_\Sigma)(1+sT_1)(1+sT_2)}\ \text{(b)}, \tag{10}$$

with $T_1 > T_2 >> T_\Sigma$. The method is based on the generalized optimization conditions (7) and on a supplementary defined parameter *m*:

$$m = T_\Sigma / T_1 \quad (T_\Sigma / T_1 << 1). \tag{11}$$

Finally the characteristic t.f.s $H_0(s)$ and $H_r(s)$ will obtain the optimized forms given in (12) and in (13), respectively:

$$H_{0opt}(s) = \frac{1 + \beta T_{\Sigma m} s}{\beta^{3/2} T_\Sigma^{'} \frac{m}{(1+m)^2} s(1+sT_1)(1+sT_\Sigma)} \quad \text{with} \quad T_\Sigma^{'} = \frac{T_\Sigma}{(1+m)}, \tag{12}$$

$$\begin{aligned} H_{r\,opt}(s) &= \frac{(1 + \beta T_{\Sigma m} s)}{\beta^{3/2} T_\Sigma^{'3} s^3 + \beta^{3/2} T_\Sigma^{'2} s^2 + \beta T_\Sigma^{'} s + 1} \\ &= \frac{(1 + \beta T_{\Sigma m} s)}{(1 + \beta^{1/2} T_\Sigma^{'} s)[1 + (\beta - \beta^{1/2}) T_\Sigma^{'} s + \beta T_\Sigma^{'2} s^2]}. \end{aligned} \tag{13}$$

In this case the compact design relations are

$$k_c = \frac{(1+m)^2}{\beta^{3/2} k_p T_1 \frac{T_\Sigma}{T_1} m} (1+m), \quad T_c = \beta T_\Sigma \frac{[1 + (2 - \beta^{1/2})m + m^2]}{(1+m)^3}. \tag{14}$$

Mainly, the 2p-SO-m ensures efficient disturbance-rejection for a special case of servo-system applications with "great and variable" moment of inertia. The system performances regarding the reference input are synthesized in Fig. 2 and extended and useful conclusions are available in [5].

In comparison with other classical benchmark-type model oriented design methods [1], the 2p-SO-m can be recommended for servo-systems (speed control) characterized with great differences between the large and the small time constants ($0.05 < m \le 0.2$) and high performance imposed regarding load disturbances. Comparing the control system performance with that ensured by the Modulus Optimum method (MO-m), (β in the of domain of $4 < \beta \le 9$ (12)) the effect of load disturbances is faster rejected. The method is easy applicable for online redesign of the controller parameters.

The quasi-continuous (QC) implementation of the PI(D) controller as digital control algorithm can be supported by the informational diagram presented in Fig. 3. The additional state variable x_k is associated to the I component, and an AWR measure is inserted. The equivalency between the parameters {K_{pid}, K_i, K_d, K_{arw}} and the continuous parameters of the controllers {k_r, T_r, T_r'} is easily calculable,

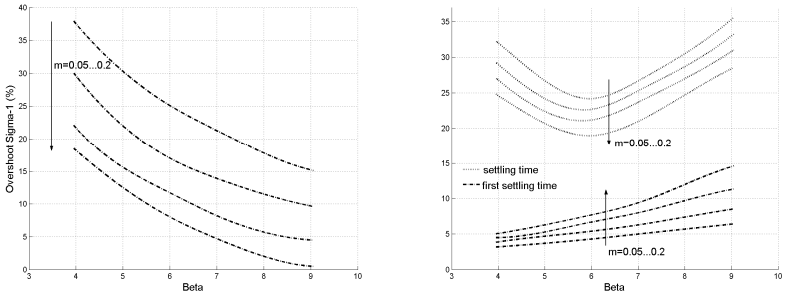


Fig.2. System performance regarding the reference input; $\sigma_{1,r}, t_{s,r}, t_{1,r} = f(\beta)$, m – parameter.

and it depends on the sampling period value, T_s. If the controllers are developed on the basis of linearized models the solution allows the bumpless commutation from one controller to another depending on the operating points [14].

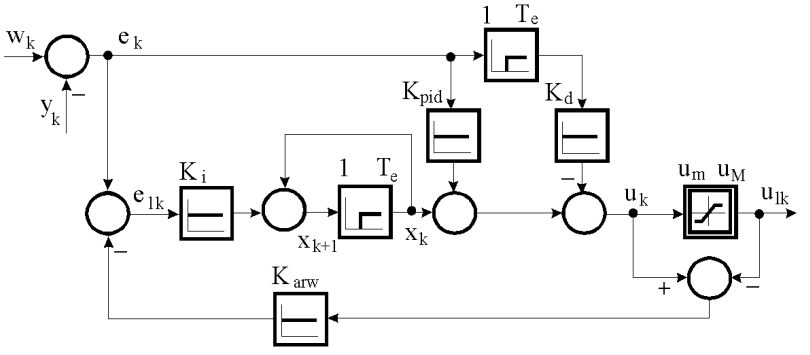


Fig. 3. A quasi-continuously operating PID digital control algorithm implementation.

3.3 Sensitivity Analysis in the Frequency Domain

The control system performance indices versus β can be represented in a graphical form in the frequency domain. An efficient measure of the sensitivity and of the robustness of our extensions is based on the relation of the sensitivity function $S_0(s)$ and of the complementary sensitivity function $T_0(s)$, the Nyquist diagrams and the $M_{s0} = f(\beta)$ circles, and the values of M_{s0}^{-1} versus β (in Fig. 4), the phase margin φ_r versus β and/or Bode diagrams for different values of β and m.

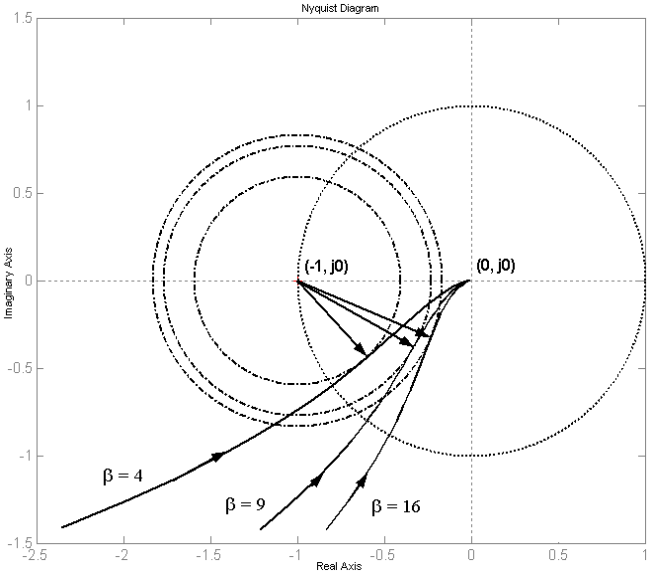


Fig. 4. Nyquist curves and M_{S0}^{-1} circles for $\beta \in \{4,9,16\}$ for ESO-m.

For example, Fig. 4 illustrates the Nyquist curves, the $M_{S0} = f(\beta)$ circles and the $M_{S0}^{-1} = f(\beta)$ circles are presented for the ESO-m for representative values of β, $\beta \in \{4,9,16\}$. Detailed diagrams are offered in [5] for the 2p-SO-m.

Fig. 5 illustrates these diagrams just for the extreme situation characterized by $m = 0.05$ and $\beta \in \{4,9,12\ (20)\}$. The curves point out the increase of robustness when the value of β is increased. Such diagrams can be useful for controller design (CAD) by allowing to fixing the value of β according to the desired performance.

3.4 Further Performance Enhancement: Alternative Controller Structures

Both extensions found applications incorporated in alternative control structures and algorithms; such applications are exemplified as follows. Starting with Fig. 6 a particular case of control structure (CS) containing controllers with non-homogenous dynamics with respect to the two inputs is presented in Fig. 7 [14].

The controller blocks are characterized by own t.f.s. Starting with the CS with a 1-DOF controller, given in Fig. 7 (a), the use of the reference filter, $F_r(s)$ can

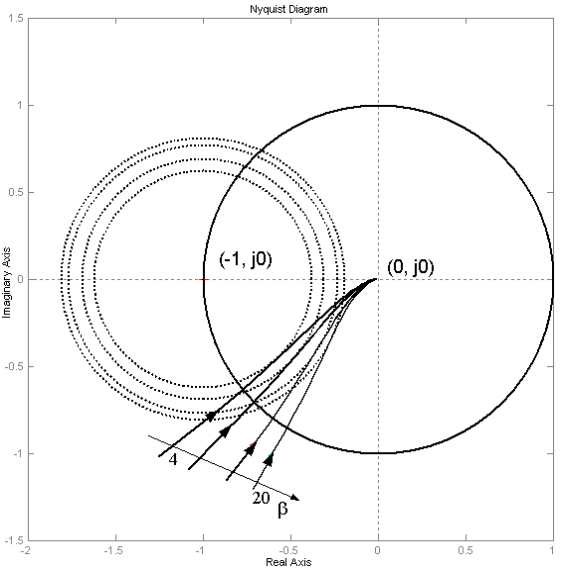


Fig. 5. Nyquist curves and M_{S0}^{-1} circles for $m = 0.05$ and $\beta \in \{4,9,12\,(20)\}$ for 2p-SO-m.

ensure an efficient pole-zero compensation. This approach allows a 2-DOF interpretation of the design [15].

The performance enhancement is exemplified here only for the ESO-m [3]. The methodology is the same for the 2p-SO-m [5].

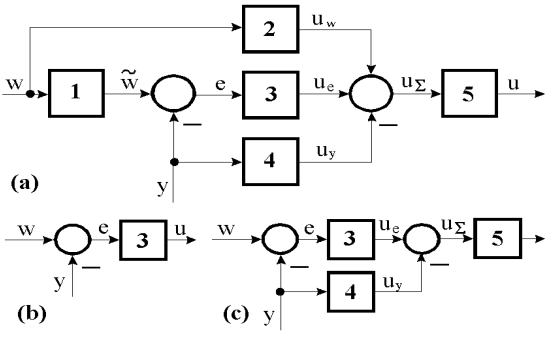


Fig. 6. Typical controller structures and particular forms of the modules [14].

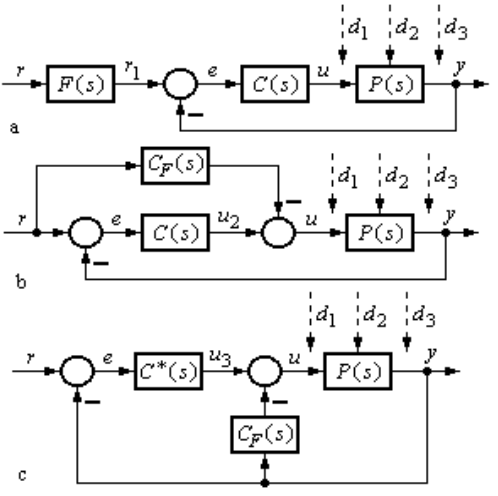


Fig. 7. Structures of 1-DOF and 2-DOF controller as extension of an 1-DOF controller.

A first version of reference filter *F(s)* is recommended to compensate the effect of the complex-conjugated poles in (8) (also in (13) and, together with this, the effect of the zero:

$$F_r(s) = \frac{1 + (\beta - \beta^{1/2})T_\Sigma s + \beta {T_\Sigma}^2 s^2}{(1 + \beta T_\Sigma s)(1 + sT_f s)}. \tag{15}$$

Consequently, the control system behavior in the relation $r \to r_1 \to y$ becomes aperiodically with the main performance indices $\sigma_1 = 0$ and $t_s \approx (3...5)(\beta - 1)T_\Sigma$ (and φ_r according to Fig. 1):

$$\tilde{H}_r(s) = \frac{1}{(1 + \beta T_\Sigma s)(1 + sT_f)}. \tag{16}$$

A second version of filter can be used to compensate only the effect of the zero in (13); accordingly:

$$F_r(s) = \frac{1}{1 + \beta T_\Sigma s}. \tag{17}$$

The control system behavior in the relation $r \to r_1 \to y$ is given by the following t.f. and the closed-loop system has an oscillatory behavior only for $\beta < 9$:

$$\tilde{H}_r(s) = \frac{1}{(1+\beta^{1/2}T_\Sigma s)[1+(\beta-\beta^{1/2})T_\Sigma s+\beta T_\Sigma{}^2 s^2]}. \quad (18)$$

Similar types of filters are used in case of the 2p-SO-m, having β and m as parameter [5].

3.5 Equivalency between 1-DOF (PID) and 2-DOF Controllers

Let us consider the block diagram given in Fig. 7 (a). Replacing the feedback controller $C(s)$ on the input channel and the forward loop, this CS can be transposed into a 2-DOF CS in its classical RST representation, where $R(\ )$, $S(\ )$ and $T(\ )$ are the specific polynomials [16]. The PI or PID controllers (with/without reference filters) can be restructured in form of the 2-DOF controller, and vice-versa, where the presence of a conventional controller can be highlighted. Two types of such rearranged PI(D) structure are detailed in Fig. 7 (b) and (c). The rearrangements allow take into account the design experience from case of PI and PID controllers.

For example, if the controller from Fig. 7 (b) is be characterized by a continuous t.f. with the "traditional" [1] tuning parameters $\{k_R, T_i, T_d, T_f\}$; the t.f.s of the equivalent blocs are

$$C(s) = \frac{u(s)}{e(s)} = k_R(1+\frac{1}{sT_i}+\frac{sT_d}{1+sT_f}),\ C_F(s) = \frac{u_f(s)}{r(s)} = k_R(\alpha_1+\alpha_2\frac{sT_d}{1+sT_f}). \quad (19)$$

For the CS given in Fig. 7 (c) (with the notation $C(s) = C^*(s)$) the t.f.s are

$$\begin{aligned} C^*(s) &= \frac{u(s)}{e(s)} = k_R[(1-\alpha_1)+\frac{1}{sT_i}+(1-\alpha_2)\frac{sT_d}{1+sT_f}], \\ C_P(s) &= \frac{u_f(s)}{r(s)} = k_R(\alpha_1+\alpha_2\frac{sT_d}{1+sT_f}). \end{aligned} \quad (20)$$

Depending on the values of α_1 and α_2 (parameters), for the presented blocks the behaviors from in Table 1 are obtained. The choice of a certain representation of the controller depends on:

- the structure of the available controller;
- the adopted algorithmic design method and the result of this design.

Table 1. Connections between 2-DOF controller and extended 1-DOF controller structure (P – proportional, D – derivative, I – integral, L1(2) – first (second) order lag filter)

Fig.7 (a)		F(s)	-	F(s)C(s)	C(s)	Remarks
Fig.7 (b)		-	C_F	C(s)–C_F(s)	C(s)	-
Fig.7 (c)		-	C_P	C*(s)	C*(s)+C_P(s)	-
α_1	α_2	-	-	(ref. channel)	(feedback)	
0	0	1	0	PID	PID	1-DOF controller
0	1	PDL2	DL1	PI	PID	1-DOF with non-homogenous behavior
1	0	PD2L2	P	PID-L1	PID	
1	1	PL2	PDL2	I	PID	
α_1	α_2	PID controller with pre-filtering (2-DOF controller)				

The comparisons between 2-DOF CS and 1-DOF CS can be performed on the basis of the structures presented in Fig. 7. Some results in this context are presented in [5] and [16].

4 Extensions to Fuzzy Controller Solutions

In [11] and [12] the authors presents some 2-DOF fuzzy controllers defined on the basis of the Mamdani or Takagi-Sugeno Fuzzy blocks FB-T_c, see Figs. 8 to 11. The linear blocks are presented in their continuous-time forms for the sake of simplifying the presentation. This hybrid treatment can lead to complicated problems in the systematic analysis of the fuzzy CS structures. It is largely accepted [12] that Takagi-Sugeno Fuzzy Controllers (TS-FCs) are more flexible to the operating point changes and can ensure better performance in the cases of plants with nonlinearities.

The TS-FC solution can be developed based on the presented PI or PID control solutions [13]. The set of linearized model of the plant used in the TS-FC design can be treated as a linear system with variable parameters.

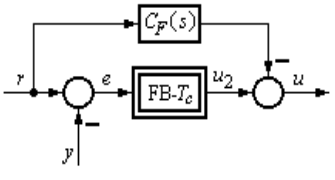


Fig. 8. Structure of feed-forward 2-DOF PI-fuzzy controller.

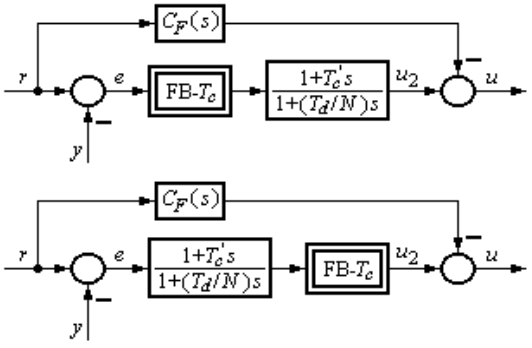


Fig. 9. Structures of feed-forward 2-DOF PID-fuzzy controllers.

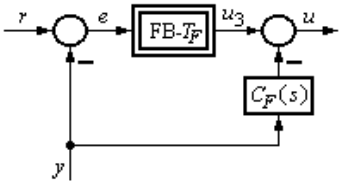


Fig. 10. Structure of feedback 2-DOF PI-fuzzy controller.

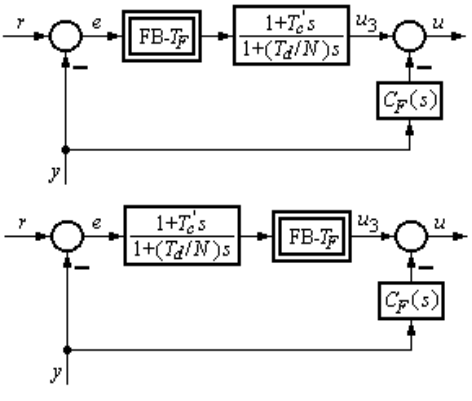


Fig. 11. Structure of feedback 2-DOF PI-fuzzy controller.

To develop the PI-FC (or more generally a PID-FC), Fig. 12, the continuous-time PI controllers must be discretized resulting in the incremental versions of the PI quasi-continuous digital controllers

$$\Delta u_{i,k} = K_{Pi}\,\Delta e_{i,k} + K_{Ii}\,e_{i,k} = K_{Pi}(\Delta e_{i,k} + \delta_i \cdot e_{i,k}),\ i = \overline{1,2}, \tag{21}$$

where $\{K_{Pi}, K_{Ii}, \delta_i\}$ are functions of $\{k_{ci}, T_{ci}\}$:

$$K_{Pi} = k_{ci}T_{ci}[1-(h/2T_{ci})],$$
$$K_{Ii} = k_{ci}\,h,\ \ \delta_i = K_{Ii}/K_{Pi} = 2h/(2T_{ci}-h),\ \ i=\overline{1,2}\ . \tag{22}$$

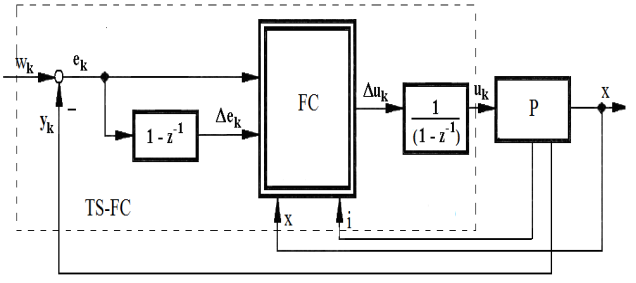


Fig. 12. Block diagram of control system with Takagi-Sugeno fuzzy controller.

Three input membership functions are defined initially to have in mind the low-cost implementation of the 2-DOF fuzzy controllers. More membership functions can be defined for nonlinear processes and high performance specifications. Fig. 13 points out the tuning parameters of the Takagi-Sugeno fuzzy block FC: B_e and $B_{\Delta e}$ (if the FC is a Mamdani type also the output membership function must be done, having as parameter $B_{\Delta u}$).

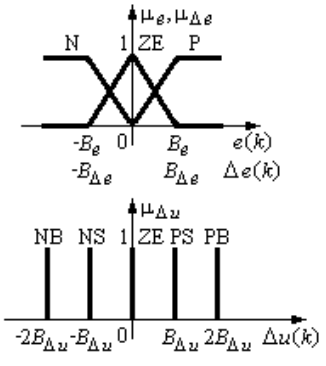


Fig. 13. Membership functions of FC.

This obtained control algorithm, permanently actualized to the plant parameters' modifications, is included in the TS-FC. The TS-FC use *if–then* fuzzy rules expressed in the following general form [17]:

$$
\begin{aligned}
&\text{Rule 1: IF } e(k) \text{ IS N AND } \Delta e(k) \text{ IS P}\\
&\qquad \text{THEN } \Delta u(k) = K_P^1[\Delta e(k) + \mu^1 e(k)],\\
&\text{Rule 2: IF } e(k) \text{ IS ZE AND } \Delta e(k) \text{ IS P}\\
&\qquad \text{THEN } \Delta u(k) = K_P^2[\Delta e(k) + \mu^2 e(k)],\\
&\text{Rule 3: IF } e(k) \text{ IS P AND } \Delta e(k) \text{ IS P}\\
&\qquad \text{THEN } \Delta u(k) = K_P^3[\Delta e(k) + \mu^3 e(k)],\\
&\text{Rule 4: IF } e(k) \text{ IS N AND } \Delta e(k) \text{ IS ZE}\\
&\qquad \text{THEN } \Delta u(k) = K_P^4[\Delta e(k) + \mu^4\, e(k)],\\
&\text{Rule 5: IF } e(k) \text{ IS ZE AND } \Delta e(k) \text{ IS ZE}\\
&\qquad \text{THEN } \Delta u(k) = K_P^5[\Delta e(k) + \mu^5 e(k)],\\
&\text{Rule 6: IF } e(k) \text{ IS P AND } \Delta e(k) \text{ IS ZE}\\
&\qquad \text{THEN } \Delta u(k) = K_P^6[\Delta e(k) + \mu^6 e(k)],\\
&\text{Rule 7: IF } e(k) \text{ IS N AND } \Delta e(k) \text{ IS N}\\
&\qquad \text{THEN } \Delta u(k) = K_P^7[\Delta e(k) + \mu^7 e(k)],\\
&\text{Rule 8: IF } e(k) \text{ IS ZE AND } \Delta e(k) \text{ IS N}\\
&\qquad \text{THEN } \Delta u(k) = K_P^8[\Delta e(k) + \mu^8 e(k)],\\
&\text{Rule 9: IF } e(k) \text{ IS P AND } \Delta e(k) \text{ IS N}\\
&\qquad \text{THEN } \Delta u(k) = K_P^9[\Delta e(k) + \mu^9 e(k)].
\end{aligned}
\tag{23}
$$

The TS-PI-FC uses the *max* and *min* operators in the inference engine, assisted by the a classical rule base. The strictly positive parameters of TS-PI-FC, $\{B_e, B_{\Delta e}, \mu^{(i)}\}$ must be determined by means of the development method given in [17]. The parameters $\mu^{(i)}$ can introduce additional nonlinearities that can be useful for control system performance enhancement especially when controlling complex plants, nonlinear or with variable parameters, where the stability analysis is necessary [18]–[22].

The basic values of the parameters B_e, $B_{\Delta e}$ and $\mu^{(i)}$ are chosen in accordance with the experience of an expert in control systems and are in connection with the domains of variation of the reference input. The nonlinear behavior which is adequate for the FC can be obtained by increasing the number of input linguistic terms.

5 Application: Driving System with BLDC Motor

The presented application is an electrical driving system with BLDC-m with VMI, characterized by the following parameters [23]: $p = 2$, $R_s = 1\Omega$, $L_s = 0.02H$,

$V_{DC} = 220V$, $J_{e0} = 0.005kgm^2$. The motor can operate at any desired speed within the range $\omega \in [0, 314] rad/s$.

The CS given in Fig. 14 (as a Simulink scheme) is basically a cascade control structure. The inner loop – with the on-off controller – can be characterized as a second-order system with delay [24].

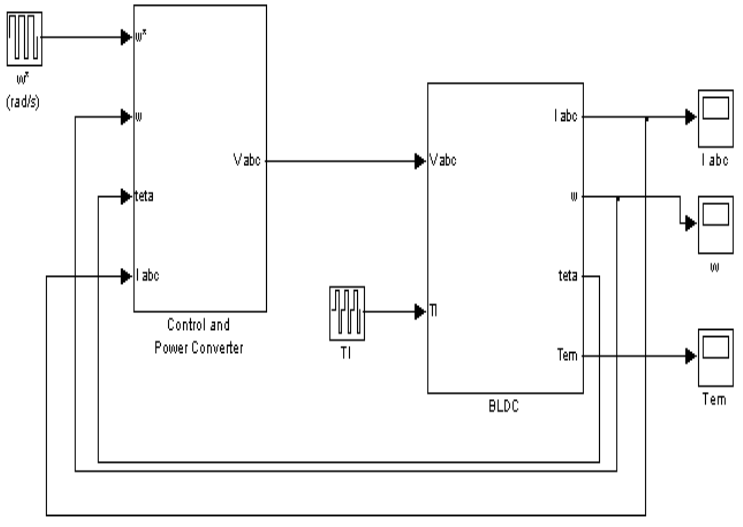


Fig. 14. Classical CS and controller block of BLDC-m.

On the basis of the 2p-SO-m the main (speed) controller (PI) has the parameters K_{Ci} = 1 and T_{ri} = 0.001 s, which are calculated for nominal value of the plant parameters, and they are recalculated periodically in accordance with parameter modifications. The controller (in its PI version) includes also a classical version of an Anti-Windup Reset (AWR) circuit, Fig. 14. The AWR operates only in the same operating conditions at high speed and high-value disturbance without disrupting stable operation of the system.

The CS presented in Fig. 14 allows the replacement of the controller by customizing the *s* (of the controller) and visualization of all measurements whose evolution is interesting, because the simulations gives instant values and their measured values must be filtered.

The application simulates a winding process with VMI as shown in Fig. 15, where the reference input must be correlated with the modification of the working roll radius. In this context, two basic aspects occur at the development of the

control structure: the modification of the reference input (ω), and the controller parameter tuning/retuning. To tackle the first aspect, the following condition must be fulfilled:

$$v_t(t) = \text{const} \rightarrow \omega_0(t) = k / r(t) \cdot \tag{24}$$

where a measurement of the radius $r(t)$ enables the continuous modification of the reference input $\omega_0(t)$; the variance of the moment of inertia, according to

$$J_e(t) = \frac{1}{2} \rho \pi l r_t^4(t) \tag{25}$$

requires much attention in the controller design, which parameters adapted in time varying; the solution based on an external speed control loop with linear PI or PI-TS-FC-s with parameter adaptation was presented in Section 2. Accepting a linear variation of $r(t)$ with 25% the value of $J(t)$ increases twice, and accordingly the controller parameters must be updated.

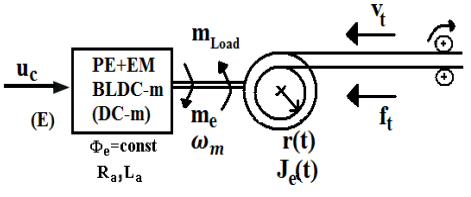


Fig. 15. Functional diagram of electrical driving system with VMI (a) and reference input correction system (b).

The BLDC-m drive was tested by simulation using the following scenario: for a given $\omega_{0,0}$ and a corresponding linear speed $v_t(t)$ followed by a reference changing, two linear speed changing and finally the stop ($v_{tf0}(t)$=0); the initial value the parameters of the speed controller were initialized regarding to the values of $\{J_{e0}, k_{P0}, T_{10}\}$. Due to the modification of $r(t)$ the reference input $\omega_0(t)$ was recalculated continuously according to (24), and timely the parameters of the controller.

Fig. 16 illustrates the simulation results concerning the output linear speed $v_t(t)$ in the given scenario.

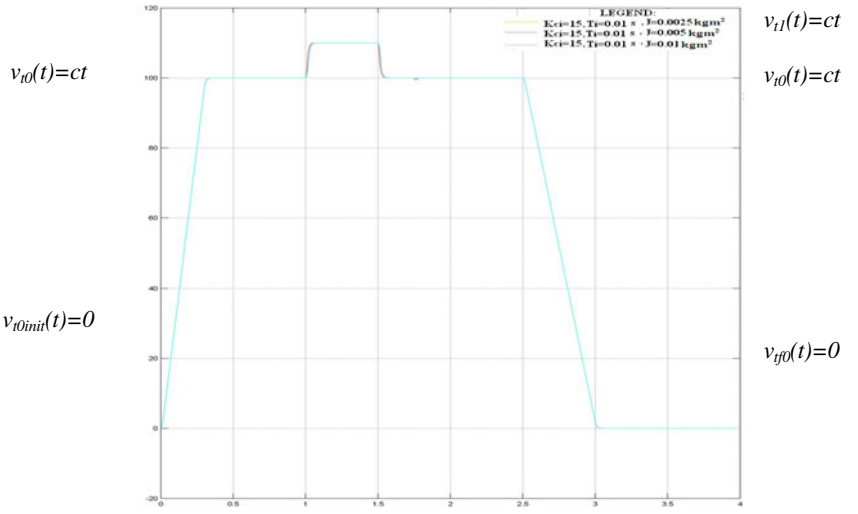


Fig. 16. Output speed (linear speed $v_t(t)$ for control system of electrical driving system with VMI with PI controller and second order reference filter.

No discontinuities in the variation of $v(t)$ are observed and during the winding regime the output of the controller remains within the limitations.

6 Conclusions

Based on the practical version of the SO-method, this chapter has presented extensions – the ESO-m and the 2p-SO-m – focused on benchmark-type models, which enable generalizations of the optimization conditions and compact design relations can be given. The presented extensions enlarge significantly the areas of application and usefulness of the SO method specific for mechatronic system applications, and they ensure better control system performance.

Basically, the control design is discussed in continuous time, but they can be relatively easy implemented in quasi-continuous digital version by using the presented approach (for example, [2]). Useful conclusions about the extended methods can be obtained based on sensitivity analysis of the controlled servo systems (applications) due in time-domain (for example, [15].

A Takagi-Sugeno fuzzy controller extension of the presented methods has also been investigated. The application has dealt with a BLDC driving application with VMI confirm the applicability of the methods.

Other applications, i.e., the non-homogenous variant of the controller, the model-based design extensions using Smith predictor structures, are possible research and application themes. They will be accompanied by several modeling and application-oriented approaches [25]–[33].

Acknowledgments. This work was supported by a grant of the Romanian National Authority for Scientific Research, CNCS – UEFISCDI, project number PN-II-ID-PCE-2011-3-0109. Also, the work was partially supported by the strategic grant POSDRU 6/1.5/S/13 (2008) of the Ministry of Labour, Family and Social Protection, Romania, co-financed by the European Social Fund – Investing in People.

References

[1] Åström, K.J., Hägglund, T.: PID controllers theory: Design and tuning. Instrument Society of America, Research Triangle Park (1995)
[2] Föllinger, O.: Regelungstechnik. Elitera Verlag, Berlin (1985)
[3] Preitl, S., Precup, R.E.: An extension of tuning relations after symmetrical optimum method for PI and PID controllers. Automatica 35, 1731–1736 (1999)
[4] Preitl, Z.: Improving disturbance rejection by means of a double parameterization of the symmetrical optimum method. Scientific Bulletin of the Politehnica University of Timişoara, Series Automation and Computer Science 50(64), 25–34 (2005)
[5] Preitl, Z.: Model based design methods for speed control applications. Editura Politehnica, Timisoara, Romania (2008)
[6] Isermann, R.: Mechatronic systems: Fundamentals. Springer, New York (2005)
[7] Baldursson, S.: BLDC motor modelling and control - A Matlab/Simulink implementation. M.Sc. Thesis, Institutionen för Energi och Miljö, Göteborg, Sweden (2005)
[8] Nasar, S.A., Boldea, I.: Electric drives, 2nd edn. CRC Press (2005)
[9] Boldea, I.: Advanced electric drives. PhD courses (2010-2011). "Politehnica" Univ. Timisoara, Timisoara (2011)
[10] Mink, F., Bahr, A.: Adaptive speed control for drives with variable moments of inertia and natural drequencies. LTi DRIVES GmbH Entwicklung Software. Lahnau, Germany (2011)
[11] Christian, J.A., Turbe, M.A., Kabo, E.M., Manno, L.C., Johnson, E.N.: Development of a variable inertia reaction wheel system for spacecraft attitude control. In: Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, Rhode Island, USA, p. 13 (2004)
[12] Akpolat, Z.H., Asher, G.M., Clare, J.C.: A practical approach to the design of robust speed controllers for machine drives. IEEE Trans. Ind. Electro. 47, 315–324 (2000)
[13] Preitl, S., Precup, R.E.: Cross optimization aspects concerning the extended symmetrical optimum method. In: Preprints of PID 2000 IFAC Workshop, Terrassa, Spain, pp. 254–126 (2000)
[14] Preitl, S., Precup, R.E., Preitl, Z.: Control structures and algorithms, vol. 1, 2. Editura Orizonturi Universitare, Timisoara (2009) (in Romanian)
[15] Preitl, S., Precup, R.E., Stînean, A.I., Dragoş, C.A., Rădac, M.B.: Extensions in symmetrical optimum design method. Advantages, applications and perspectives. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisoara, Romania, pp. 17–22 (2011)
[16] Araki, M., Taguchi, H.: Two-degree-of-freedom PID controllers. Int. J. Control Automat. Syst. 1, 401–411 (2003)
[17] Preitl, S., Precup, R.E., Preitl, Z.: Aspects concerning the tuning of 2-DOF fuzzy controllers. In: Proceedings of Xth Triennial International SAUM Conference on Systems, Automatic Control and Measurements (SAUM 2010), Nis, Serbia, pp. 210–219 (2010)

[18] Škrjanc, I., Blažič, S., Matko, D.: Direct fuzzy model-reference adaptive control. Int. J. Intell. Syst. 17, 943–963 (2002)
[19] Blažič, S., Škrjanc, I., Matko, D.: Globally stable direct fuzzy model reference adaptive control. Fuzzy Sets Syst. 139, 3–33 (2003)
[20] Guesmi, K., Essounbouli, N., Hamzaoui, A.: Systematic design approach of fuzzy PID stabilizer for DC-DC converters. Energ. Convers. Manage. 49, 2880–2889 (2008)
[21] Peng, C., Han, Q.L.: Delay-range-dependent robust stabilization for uncertain T-S fuzzy control systems with interval time-varying delays. Inf. Sci. 181, 4287–4299 (2011)
[22] Garcia, A., Luviano-Juarez, A., Chairez, I., Poznyak, A., Poznyak, T.: Projectional dynamic neural network identifier for chaotic systems: Application to Chua's circuit. Int. J. Artif. Intell. 6, 1–18 (2011)
[23] Stînean, A.I., Preitl, S., Precup, R.E., Pozna, C., Dragoş, C.A., Rădac, M.B.: Speed and position control of BLDC servo systems with low inertia. In: Proceedings of 2nd International Conference on Cognitive Infocommunications (CogInfoCom 2011), Budapest, Hungary, p. 10 (2011)
[24] Stînean, A.I.: Electric drives simulations - brushless DC motor drive. PhD project courses. "Politehnica" Univ. Timisoara, Timisoara (2010) (in Romanian)
[25] Baranyi, P., Gedeon, T.D.: Rule interpolation by spatial geometric representation. In: Proceedings of 6th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 1996), Granada, Spain, pp. 483–488 (1996)
[26] Baranyi, P., Yam, Y., Várkonyi-Kóczy, A.R., Patton, R.J., Michelberger, P., Sugiyama, M.: SVD based complexity reduction to TS fuzzy models. IEEE Trans. Ind. Electron 49, 433–443 (2002)
[27] Horváth, L., Rudas, I.J.: Modelling and solving methods for engineers. Elsevier, Academic Press, Burlington, MA (2004)
[28] Vaščák, J., Madarász, L.: Adaptation of fuzzy cognitive maps - a comparison study. Acta Polytechnica Hungarica 7, 109–122 (2010)
[29] Johanyák, Z.C.: Student evaluation based on fuzzy rule interpolation. Int. J. Artif. Intell. 5, 37–55 (2010)
[30] Zhang, J., Shi, P., Xia, Y.: Robust adaptive sliding-mode control for fuzzy systems with mismatched uncertainties. IEEE Trans. Fuzzy Syst. 18, 700–711 (2010)
[31] Iglesias, J.A., Angelov, P., Ledezma, A., Sanchis, A.: Evolving classification of agents' behaviors: a general approach. Evolving Syst. 1, 161–171 (2010)
[32] Linda, O., Manic, M.: Self-organizing fuzzy haptic teleoperation of mobile robot using sparse sonar data. IEEE Trans. Ind. Electron 58, 3187–3195 (2011)
[33] Kasabov, N., Abdull Hamed, N.H.: Quantum-inspired particle swarm optimisation for integrated feature and parameter optimisation of evolving spiking neural networks. Int. J. Artif. Intell. 7, 114–124 (2011)

Mixed Multidimensional Risk Aversion

Irina Georgescu[1] and Jani Kinnunen[2]

[1] Academy of Economic Studies, Department of Economic Cybernetics,
Calea Dorobantilor 15-17, Sector 1, Bucharest, Romania
irina.georgescu@csie.ase.ro
[2] Institute for Advanced Management Systems Research,
Åbo Akademi University, Jouhakaisenkatu 3-5A, 20520, Turku, Finland
jani.kinnunen@abo.fi

Abstract. The topic treated in this chapter is the risk aversion of an agent in front of a situation of uncertainty with many risk parameters. We will study a general model of risk aversion in which some parameters are *probabilistically* described (by random variables) and others are *possibilistically* described (by fuzzy numbers). For the construction of this model, firstly, mixed expected utility, a notion, which unifies probabilistic and possibilistic aspects of expected utility theory is introduced. The notion of mixed risk premium vector is introduced as a measure of risk aversion with mixed parameters. The main result of the chapter is an approximate calculation formula for mixed risk premium vector. Lastly, our model is applied in the evaluation of risk aversion in grid computing.

1 Introduction

Situations of uncertainty are often met in social and economic life. Traditionally, phenomena of uncertainty are mathematically treated by probability theory. There are cases when probabilistic modeling is not adequate, e.g., when we do not have sufficiently large databases for probabilistic inference. Zadeh's possibility theory [1] is an alternative to probability theory to treat of phenomena of uncertainty.

The study of the risk aversion of an agent faced with a risk situation is a central theme of risk theory [2]. The concepts, which appear in the treatment of risk aversion, are defined in terms of probabilistic indicators (expected value, variance, etc.). In several cases we deal with a situation of uncertainty with several risk components. Some of these risk parameters can be described probabilistically and others possibilistically. In the first case parameters will be considered as random variables and in the second case as possibilistic distributions (in particular, fuzzy numbers). From here the idea of mixed vector appears, in which some risk

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 39–50.
springerlink.com

components are random variables and others are fuzzy numbers. If all components are random variables, we deal with a probabilistic risk vector, and if all components are fuzzy numbers, we deal with a possibilistic risk vector. The aim of this chapter is to treat risk aversion in case of mixed vectors.

The chapter is organized as follows. In Section 2 notions related to fuzzy numbers and to principal possibilistic indicators of fuzzy numbers are recalled.

In Section 3 two approaches of multidimensional risk aversion are presented: one based on probability theory [3, 4] the other on possibility theory [5, 6, 7].

In Section 4 we introduce mixed vectors and the notion of mixed expected utility. The mixed expected utility generalizes both probabilistic expected utility from [3] and the possibilistic expected utility from [5, 6, 7].

Section 5 treats risk aversion in the context offered by mixed vectors. The mixed model of risk aversion is defined based on the mixed expected utility. Mixed risk premium vector is defined as a measure of risk aversion of an agent faced with a situation with several risk parameters (some possibilistic and some probabilistic). A formula, which evaluates mixed risk premium vector depending on probabilistic and possibilistic indicators associated with the mixed vector is proved (see Theorem 1). In particular, the treatment of multidimensional probabilistic risk aversion from [3] and the treatment of multidimensional possibilistic risk aversion from [5] are obtained from the model studied in this section.

Section 6 proposes a way of applying mixed risk premium in grid computing, and Section 7 concludes the chapter.

2 Possibilistic Indicators of Fuzzy Numbers

Fuzzy numbers represent the best-known class of possibilistic repartitions. In this section the definition of fuzzy number and its most important indicators are recalled. For basic material we refer to [8, 9, 10, 11].

Let X be a set of states. A fuzzy subset of X (= fuzzy set) is a function $A: X \to [0, 1]$. A fuzzy set A is *normal* if there exists $x \in X$ such that $A(x)=1$. The support of A is defined by $supp(A)=\{x \in X | A(x)>0\}$.

In the following we consider that X is the set $\mathbf{R}$ of real numbers. For any $\gamma \in [0, 1]$, the γ-level set of a fuzzy set A in $\mathbf{R}$ is defined by

$$[A]^{\gamma} = \begin{cases} \{x \in R \mid A(x) \geq \gamma\} \text{ if } \gamma > 0 \\ cl\{x \in R \mid A(x) > \gamma\} \text{ if } \gamma = 0 \end{cases} \quad (1)$$

($cl(supp(A))$ is the topological closure of the set $supp(A) \subseteq \mathbf{R}$). A fuzzy set A in $\mathbf{R}$ is called *fuzzy convex* if $[A]^{\gamma}$ is a convex subset of $\mathbf{R}$ for any $\gamma \in [0, 1]$. A *fuzzy*

number is a fuzzy set of **R** normal, fuzzy convex, continuous and with bounded support.

Let A be a fuzzy number and $\gamma \in [0, 1]$. $[A]^\gamma$ is a closed and convex subset of **R**. We denote $a_1(\gamma)=min[A]^\gamma$, $a_2(\gamma)=max[A]^\gamma$. Hence $[A]^\gamma=[a_1(\gamma),a_2(\gamma)]$ for all $\gamma \in [0,1]$.

A non-negative and monotone increasing function f: $[0,1]\rightarrow\mathbf{R}$ is a weighting function if it satisfies the normality condition $\int_0^1 f(\gamma)d\gamma = 1$.

We fix a fuzzy number A and a weighting function f. Assume that $[A]^\gamma=[a_1(\gamma),a_2(\gamma)]$ for all $\gamma \in [0,1]$. The *possibilistic expected value* of A w.r.t. f was defined in [11] by

$$E(f,A) = \frac{1}{2}\int_0^1 (a_1(\gamma) + a_2(\gamma))f(\gamma)d\gamma. \tag{2}$$

The *possibilistic variance* of A w.r.t. f is defined by

$$Var(f,A) = \frac{1}{2}\int_0^1 [(a_1(\gamma) - E(f,A))^2 + (a_2(\gamma) - E(f,A))^2]f(\gamma)d\gamma. \tag{3}$$

Let A and B be two fuzzy numbers and f a weighting function. Assume that $[A]^\gamma=[a_1(\gamma),a_2(\gamma)]$ and $[B]^\gamma=[b_1(\gamma),b_2(\gamma)]$ for all $\gamma \in [0,1]$. The possibilistic covariance $Cov(f,A,B)$ of A and B w.r.t. f is defined by

$$\begin{aligned} Cov(f,A,B) = \frac{1}{2}\int_0^1 &[(a_1(\gamma) - E(f,A))(b_1(\gamma) - E(f,B)) \\ &+ (a_2(\gamma) - E(f,A))(b_2(\gamma) - E(f,B))]f(\gamma)d\gamma. \end{aligned} \tag{4}$$

An *n-dimensional possibilistic vector* has the form $\vec{A} = (A_1,...,A_n)$, where each component A_i is a fuzzy number. In our case, A_i represents the i-th risk parameter and $\vec{A}$ describes the risk situation overall.

3 Approaches to Multidimensional Risk Aversion

In this section we present in parallel some elements of multidimensional probabilistic risk aversion [3, 4] and of multidimensional possibilistic risk aversion [5].

Let Ω be a set of states and (Ω, K, P) a probability space on Ω. A random vector has the form $\vec{X} = (X_1,...,X_n)$, where each X_i is a random variable. If u: $\mathbf{R}^n\rightarrow\mathbf{R}$ is a continuous function then the function $u(\vec{X})$: $\Omega\rightarrow\mathbf{R}$ defined by $u(\vec{X})(\omega) = u(X_1(\omega),...,X_n(\omega))$ for any $\omega\in\Omega$ is a random variable. In interpretation, u can be regarded as an n-dimensional utility function. The probabilistic

expected value $M(u(\vec{X}))$ of the random variable $u(\vec{X})$ is the *probabilistic expected utility* of $\vec{X}$ w.r.t. u. Probability theory of risk aversion with several parameters is developed in a framework with two components: a random vector $\vec{X} = (X_1,...,X_n)$ representing the risk situations with parameters $X_1,...,X_n$ and a utility function u: $\mathbf{R}^n \to \mathbf{R}$ expressing an agent's attitude to risk.

Throughout this paper we assume that utility functions have the class C^2 and are strictly increasing in each argument.

Definition 1 [3]**.** *A probabilistic risk premium vector* $\vec{\pi} = (\pi_1,...,\pi_n)$ (associated with the random vector $\vec{X}$ and the utility function u) is a solution of the equation:

$$M(u(\vec{X})) = u(M(X_1) - \pi_1,...,M(X_n) - \pi_n). \tag{5}$$

We denote $e_i = M(X_i), i = 1,...n$ and $\vec{e} = (e_1,...,e_n)$.

Proposition 1 [3]**.** An approximate solution of equation (5) is given by

$$\pi_i^0 = -\frac{1}{2}\sum_{j=1}^{n} \frac{1}{u_j(\vec{e})} Cov(X_i, X_j) u_{ij}(\vec{e}), \quad i = 1,...,n. \tag{6}$$

We present now by [5] a possibilistic approach to risk aversion with several parameters. We consider a setting with three elements: a possibilistic vector $\vec{A} = (A_1,...,A_n)$, a weighting function f: [0,1]$\to$**R,** and a utility function u: $\mathbf{R}^n \to \mathbf{R}$.

Assume that for any $\gamma \in [0,1]$ and $i \in \{1,\ldots,n\}$, $[A]^\gamma = [a_i(\gamma), b_i(\gamma)]$. Then the *possibilistic expected utility* $E(f, u(\vec{A}))$ of $\vec{A}$ w.r.t f and u is defined by

$$E(f, u(\vec{A})) = \frac{1}{2}\int_0^1 [u(a_1(\gamma),...,a_n(\gamma)) + u(b_1(\gamma),...,b_n(\gamma))] f(\gamma) d\gamma. \tag{7}$$

Definition 2 [5]**.** *A possibilistic risk premium vector* $\vec{\rho} = (\rho_1,...,\rho_n)$ (associated with $\vec{A}$, f, and u) is a solution of the equation:

$$E(f, u(\vec{A})) = u(E(f, A_i) - \rho_1,...,E(f, A_n) - \rho_n). \tag{8}$$

We denote $m_i = E(f, A_i), \quad i = 1,...,n$ and $\vec{m} = (m_1,...,m_n)$.

Proposition 2 [5]**.** An approximate solution of equation (8) is given by

$$\rho_i^0 = -\frac{1}{2}\sum_{j=1}^{n} \frac{1}{u_j(\vec{m})} Cov(f, A_i, A_j) u_{ij}(\vec{m}), \quad i = 1,...,n. \tag{9}$$

4 Mixed Expected Utility

In this section we treat the situations of uncertainty in which some parameters are probabilistic while others are possibilistic. A mixed vector has the form $(A_1,...,A_n,X_1,...,X_m)$, where $A_1,...,A_n$ are fuzzy numbers and $X_1,...,X_m$ are random variables. The possibilistic vector $\vec{A}=(A_1,...,A_n)$ is called the possibilistic component of $(A_1,...,A_n,X_1,...,X_m)$ and the random vector $\vec{X}=(X_1,...,X_m)$ is its probabilistic component. The mixed vector $(A_1,...,A_n,X_1,...,X_m)$ will be denoted $(\vec{A},\vec{X})$. In interpretation, a mixed vector $(\vec{A},\vec{X})$ can represent a situation of risk with several parameters. Some parameters are probabilistic and modeled by random variables $X_1,...,X_m$ while other parameters are possibilistic and modeled by fuzzy numbers $A_1,...,A_n$. A risk situation modeled by a mixed vector $(\vec{A},\vec{X})$ will be called *mixed risk situation*.

In Section 3 the notion of probabilistic expected utility was used to define probabilistic risk premium vector and the notion of possibilistic expected utility was used to define possibilistic risk premium vector. Therefore to define risk premium vector in case of a mixed risk situation we should have a notion of mixed expected utility, which should generalize the two of them.

We fix a mixed vector $(\vec{A},\vec{X})=(A_1,...,A_n,X_1,...,X_m)$, an $m+n$ dimensional utility function g: $\mathbf{R}^{n+m}\to\mathbf{R}$ and a weighting function f: $[0,1]\to\mathbf{R}$. We assume that $[A_i]^\gamma=[a_i(\gamma),b_i(\gamma)]$ for any $i=1,...,n$ and $\gamma\in[0,1]$. Let $\gamma\in[0,1]$. We denote $\vec{a}(\gamma)=(a_1(\gamma),...,a_n(\gamma))$ and $\vec{b}(\gamma)=(b_1(\gamma),...,b_n(\gamma))$. We consider the functions $g(\vec{a}(\gamma),\vec{X}):\ \Omega\to\mathbf{R}$ and $g(\vec{b}(\gamma),\vec{X}):\Omega\to\mathbf{R}$ defined by

$$\begin{aligned} g(\vec{a}(\gamma),\vec{X})(\omega)&=g(a_1(\gamma),...,a_n(\gamma),X_1(\omega),...,X_m(\omega));\\ g(\vec{b}(\gamma),\vec{X})(\omega)&=g(b_1(\gamma),...,b_n(\gamma),X_1(\omega),...,X_m(\omega)). \end{aligned} \tag{10}$$

for any $\omega\in\Omega$. It is obvious that $g(\vec{a}(\gamma),\vec{X})$ and $g(\vec{b}(\gamma),\vec{X})$ are random variables, and we can consider their expected values $M(g(\vec{a}(\gamma),\vec{X}))$ and $M(g(\vec{b}(\gamma),\vec{X}))$.

Definition 3. The *mixed expected utility* of the mixed vector $(\vec{A},\vec{X})$ w.r.t. the utility function g and a weighting function f is defined by

$$E(f,g(\vec{A},\vec{X}))=\frac{1}{2}\int_0^1[M(g(\vec{a}(\gamma),\vec{X}))+M(g(\vec{b}(\gamma),\vec{X}))]f(\gamma)d\gamma. \tag{11}$$

Remark 1. (i) If $n=0$ then $M(g(\vec{a}(\gamma),\vec{X}))=M(g(\vec{b}(\gamma),\vec{X}))=M(g(\vec{X}))$. Thus,

$$E(f,g(\vec{A},\vec{X}))=\frac{1}{2}\int_0^1[M(g(\vec{X}))+M(g(\vec{X}))]f(\gamma)d\gamma=M(g(\vec{X})). \tag{12}$$

(ii) If m=0 then $M(g(\vec{a}(\gamma),\vec{X})) = M(g(\vec{a}(\gamma)) = g(\vec{a}(\gamma))$ and $M(g(\vec{b}(\gamma),\vec{X}))$ $= M(g(\vec{b}(\gamma)) = g(\vec{b}(\gamma))$. Thus,

$$\begin{aligned} E(f,g(\vec{A},\vec{X})) &= \frac{1}{2}\int_0^1 [M(g(\vec{a}(\gamma)) + M(g(\vec{b}(\gamma))]f(\gamma)d\gamma \\ &= \frac{1}{2}\int_0^1 [g(\vec{a}(\gamma)) + g(\vec{b}(\gamma))]f(\gamma)d\gamma = E(f,g(\vec{A})). \end{aligned} \tag{13}$$

Example 1. We will take n=2, m=0, and $g(x,y) = (x - E(f,A_1))(y - E(f,A_2))$ for any $x,y \in \mathbf{R}$. Then according to Remark 1 (ii) we have

$$E(f,g(\vec{A},\vec{X})) = E(f,g(\vec{A})) = Cov(f,A_1,A_2). \tag{14}$$

Example 2. We will take n=0, m=2, and $g(x,y) = (x - M(X_1))(y - M(X_2))$ for any $x,y \in \mathbf{R}$. Then, according to Remark 1 (i) we have

$$E(f,g(\vec{A},\vec{X})) = M(g(X_1,X_2)) = Cov(X_1,X_2). \tag{15}$$

The two examples from above show that both possibilistic covariance and probabilistic covariance are particular cases of mixed expected utility.

Lemma 1. If n=m=1 and $g(x,y) = (x - E(f,A_1))(y - M(X_1))$, then

$$E(f,g(\vec{A},\vec{X})) = 0. \tag{16}$$

Lemma 2. Let g: $\mathbf{R}^n \to \mathbf{R}$ be a continuous function and g_1: $\mathbf{R}^{n+m} \to \mathbf{R}$ be a function defined by $g_1(\vec{x},\vec{y}) = g(\vec{x})$ for any $\vec{x} \in \mathbf{R}^n$ and $\vec{y} \in \mathbf{R}^m$. Then

$$E(f,g_1(\vec{A},\vec{X})) = E(f,g(\vec{A})). \tag{17}$$

Lemma 3. Let h: $\mathbf{R}^m \to \mathbf{R}$ be a continuous function and h_1: $\mathbf{R}^{n+m} \to \mathbf{R}$ be a function defined by $h_1(\vec{x},\vec{y}) = h(\vec{y})$ for any $\vec{x} \in \mathbf{R}^n$ and $\vec{y} \in \mathbf{R}^m$. Then

$$E(f,h_1(\vec{A},\vec{X})) = M(h(\vec{X})). \tag{18}$$

Proposition 3. Let g: $\mathbf{R}^{n+m} \to \mathbf{R}$, h: $\mathbf{R}^{n+m} \to \mathbf{R}$ be two continuous functions and $\alpha,\beta \in \mathbf{R}$. Consider the function u: $\mathbf{R}^{n+m} \to \mathbf{R}$ defined by $u(\vec{x},\vec{y}) = \alpha g(\vec{x},\vec{y}) + \beta h(\vec{x},\vec{y})$ for any $\vec{x} = (x_1,...,x_n) \in \mathbf{R}^n$ and $\vec{y} = (y_1,...,y_m) \in \mathbf{R}^m$. Then

$$E(f,u(\vec{A},\vec{X})) = \alpha E(f,g(\vec{A},\vec{X})) + \beta E(f,h(\vec{A},\vec{X})). \tag{19}$$

Corollary 1. Let g: $\mathbf{R}^n \to \mathbf{R}$, h: $\mathbf{R}^m \to \mathbf{R}$ be two continuous functions and $\alpha,\beta \in \mathbf{R}$. Consider the function u: $\mathbf{R}^{n+m} \to \mathbf{R}$ defined by $u(\vec{x},\vec{y}) = \alpha g(\vec{x}) + \beta h(\vec{y})$ for any $\vec{x} \in \mathbf{R}^n$ and $\vec{y} \in \mathbf{R}^m$. Then

$$E(f,u(\vec{A},\vec{X})) = \alpha E(f,g(\vec{A})) + \beta M(h(\vec{X})). \tag{20}$$

5 Risk Aversion with Mixed Parameters

In this section we study risk aversion of an agent w.r.t. a situation of uncertainty with several risk parameters. Some of these parameters can be probabilistically modeled and others possibilistically. We introduce mixed risk premium vector, the notion by which we evaluate risk aversion in the mentioned case. The main result is a formula for the calculation of mixed risk premium vector.

We fix a mixed vector $(\vec{A},\vec{X}) = (A_1,...,A_n,X_1,...,X_m)$, a utility function u: $\mathbf{R}^{n+m} \to \mathbf{R}$ of class C^2 and a weighting function f: $[0,1] \to \mathbf{R}$.

For any $i, j \in \{1,\ldots, n\}$ and $k, l \in \{1,\ldots, m\}$ we denote

$$\begin{aligned} v_i(\vec{x},\vec{y}) &= \frac{\partial u(\vec{x},\vec{y})}{\partial x_i};\ v_{ij}(\vec{x},\vec{y}) = \frac{\partial^2 u(\vec{x},\vec{y})}{\partial x_i \partial x_j};\\ w_k(\vec{x},\vec{y}) &= \frac{\partial u(\vec{x},\vec{y})}{\partial y_k};\ w_{kl}(\vec{x},\vec{y}) = \frac{\partial^2 u(\vec{x},\vec{y})}{\partial y_k \partial y_l}, \end{aligned} \tag{21}$$

where $\vec{x} = (x_1,...,x_n) \in \mathbf{R}^n$ and $\vec{y} = (y_1,...,y_m) \in \mathbf{R}^m$.

Also we denote $a_i = E(f,A_i)$, $i = 1,...,n$ and $b_k = M(X_k)$, $k = 1,...,m$.

Definition 4. A *mixed risk premium vector* $(\vec{\rho},\vec{\pi}) = (\rho_1,...,\rho_n,\pi_1,...,\pi_m)$ (associated with the mixed vector $(\vec{A},\vec{X})$, the utility function u and the weighting function f) is a solution of the equation:

$$\begin{aligned} E(f,u(\vec{A},\vec{X})) = u(&E(f,A_1) - \rho_1,...,E(f,A_n) - \rho_n,\\ &M(X_1) - \pi_1,...,M(X_m) - \pi_m). \end{aligned} \tag{22}$$

Equation (22) may have several solutions $(\vec{\rho},\vec{\pi})$.

By the notations before Definition 4, equation (22) can be written as

$$E(f,u(\vec{A},\vec{X})) = u(a_1 - \rho_1,...,a_n - \rho_n,b_1 - \pi_1,...,b_m - \pi_m). \tag{23}$$

Remark 2. (i) If n=0, then according to Remark 1 (i), equation (22) becomes

$$M(u(\vec{X})) = u(b_1 - \pi_1,...,b_m - \pi_m). \tag{24}$$

We find the notion of probabilistic risk premium vector (cf. Definition 1).
(ii) If m=0, then according to Remark 1 (ii), equality (22) becomes

$$E(f,u(\vec{A}))=u(a_1-\rho_1,...,a_n-\rho_n). \tag{25}$$

Theorem 1. An approximate solution $(\vec{\rho},\vec{\pi})$ of equation (22) is given by

$$\begin{aligned}\rho_i^0&=-\frac{1}{2}\sum_{j=1}^{n}\frac{1}{v_i(\vec{a},\vec{b})}Cov(f,A_i,A_j)v_{ij}(\vec{a},\vec{b}),\quad i=1,...,n;\\ \pi_k^0&=-\frac{1}{2}\sum_{l=1}^{m}\frac{1}{w_k(\vec{a},\vec{b})}Cov(X_k,X_l)w_{kl}(\vec{a},\vec{b}),\quad k=1,...,m.\end{aligned} \tag{26}$$

Proof. We recall that the utility function u: $\mathbf{R}^{n+m}\to\mathbf{R}$ has the class C^2. By Taylor's formula and neglecting the Taylor remainder of second order, we obtain:

$$\begin{aligned}u(\vec{x},\vec{y})\approx u(\vec{a},\vec{b})&-\sum_{i=1}^{n}(x_i-a_i)\frac{\partial u(\vec{a},\vec{b})}{\partial x_i}-\sum_{k=1}^{m}(y_k-b_k)\frac{\partial u(\vec{a},\vec{b})}{\partial y_k}\\ &+\frac{1}{2}\sum_{i,j=1}^{n}(x_i-a_i)(x_j-a_j)\frac{\partial^2 u(\vec{a},\vec{b})}{\partial x_i\partial x_j}\\ &+\frac{1}{2}\sum_{k,l=1}^{m}(y_k-b_k)(y_l-b_l)\frac{\partial^2 u(\vec{a},\vec{b})}{\partial y_k\partial y_l}\\ &+\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{m}(x_i-a_i)(y_k-b_k)\frac{\partial^2 u(\vec{a},\vec{b})}{\partial x_i\partial y_k}.\end{aligned} \tag{27}$$

With the notations from the beginning of this section this relation can be written:

$$\begin{aligned}u(\vec{x},\vec{y})\approx u(\vec{a},\vec{b})&-\sum_{i=1}^{n}(x_i-a_i)v_i(\vec{a},\vec{b})-\sum_{k=1}^{m}(y_k-b_k)w_k(\vec{a},\vec{b})\\ &+\frac{1}{2}\sum_{i,j=1}^{n}(x_i-a_i)(x_j-a_j)v_{ij}(\vec{a},\vec{b})\\ &+\frac{1}{2}\sum_{k,l=1}^{m}(y_k-b_k)(y_l-b_l)w_k(\vec{a},\vec{b})\\ &+\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{m}(x_i-a_i)(y_k-b_k)\frac{\partial^2 u(\vec{a},\vec{b})}{\partial x_i\partial y_k}.\end{aligned} \tag{28}$$

Consider g: $\mathbf{R}^n \to \mathbf{R}$, h: $\mathbf{R}^m \to \mathbf{R}$, r: $\mathbf{R}^n \to \mathbf{R}$, s: $\mathbf{R}^m \to \mathbf{R}$, and z: $\mathbf{R}^{n+m} \to \mathbf{R}$ defined by

$$\begin{aligned} g(\vec{x}) &= \sum_{i=1}^{n} v_i(\vec{a},\vec{b})(x_i - a_i); \\ h(\vec{y}) &= \sum_{k=1}^{m} w_k(\vec{a},\vec{b})(y_k - b_k); \\ r(\vec{x}) &= \sum_{i,j=1}^{n} v_{ij}(\vec{a},\vec{b})(x_i - a_i)(x_j - a_j); \\ s(\vec{y}) &= \sum_{k,l=1}^{m} w_{kl}(\vec{a},\vec{b})(y_k - b_k)(y_l - b_l); \\ z(\vec{x},\vec{y}) &= \sum_{i=1}^{n}\sum_{k=1}^{m} (x_i - a_i)(y_k - b_k)\frac{\partial^2 u(\vec{a},\vec{b})}{\partial x_i \partial y_k}. \end{aligned} \tag{29}$$

Then $u(\vec{x},\vec{y}) \approx u(\vec{a},\vec{b}) - g(\vec{x}) - h(\vec{y}) + \frac{1}{2}r(\vec{x}) + \frac{1}{2}s(\vec{y}) + \frac{1}{2}z(\vec{x},\vec{y})$.

By Proposition 3 and Corollary 1 it follows:

$$\begin{aligned} E(f,u(\vec{A},\vec{X})) \approx\ & u(\vec{a},\vec{b}) - E(f,g(\vec{A})) \\ & + \frac{1}{2}E(f,r(\vec{A})) + \frac{1}{2}M(s(\vec{X})) + \frac{1}{2}E(f,z(\vec{A},\vec{X})). \end{aligned} \tag{30}$$

We consider the functions: g_i: $\mathbf{R} \to \mathbf{R}$, $i=1,\ldots,n$; h_k: $\mathbf{R} \to \mathbf{R}$, $k=1,\ldots,m$ defined by $g_i(x) = x - a_i$ for any $x \in \mathbf{R}$; $h_k(y) = y - b_k$ for any $y \in \mathbf{R}$.

We notice that $E(f,g_i(A_i)) = E(f,A_i) - a_i = 0$ and $M(h_k(X_k)) = M(X_k) - b_k = 0$ for any $i=1,\ldots,n$ and $k=1,\ldots,m$. By Proposition 3:

$$\begin{aligned} E(f,g(\vec{A})) &= \sum_{i=1}^{n} v_i(\vec{a},\vec{b})E(f,g_i(A_i)) = 0; \\ M(h(\vec{X})) &= \sum_{j=1}^{m} w_k(\vec{a},\vec{b})M(h_k(X_k)) = 0; \\ E(f,r(\vec{A})) &= \sum_{i,j=1}^{n} v_{ij}(\vec{a},\vec{b})Cov(f,A_i,A_j); \\ M(s(\vec{X})) &= \sum_{k,l=1}^{m} w_{kl}(\vec{a},\vec{b})Cov(X_i,X_j). \end{aligned} \tag{31}$$

To compute $E(f,z(\vec{A},\vec{X}))$ we consider the functions z_{ik}: $\mathbf{R}^2 \to \mathbf{R}$, $i=1,\ldots,n$ and $k=1,\ldots,m$, defined by $z_{ik}(x,y) = (x - a_i)(y - b_k)$ for any $(x, y) \in \mathbf{R}^2$.

By Proposition 3 and Lemma 1 it follows:

$$E(f,z(\vec{A},\vec{X})) = \sum_{i=1}^{n}\sum_{k=1}^{m} \frac{\partial^2 u(\vec{a},\vec{b})}{\partial x_i \partial y_k} E(f,z_{ik}(A_i,X_k)) = 0. \tag{32}$$

Replacing in the approximate value of $E(f,u(\vec{A},\vec{X}))$ one obtains:

$$\begin{aligned} E(f,u(\vec{A},\vec{X})) \approx\ & u(\vec{a},\vec{b}) + \frac{1}{2}\sum_{i,j=1}^{n} v_{ij}(\vec{a},\vec{b})Cov(f,A_i,A_j) \\ & + \frac{1}{2}\sum_{k,l=1}^{m} w_{kl}(\vec{a},\vec{b})Cov(X_k,X_l). \end{aligned} \tag{33}$$

By Taylor formula and omitting the remainder of first order one obtains:

$$\begin{aligned} &u(a_1-\rho_1,...,a_n-\rho_n,b_1-\pi_1,...,b_m-\pi_m)\\ &\approx u(\vec{a},\vec{b})-\sum_{i=1}^{n}\rho_i v_i(\vec{a},\vec{b})-\sum_{k=1}^{m}\pi_k w_k(\vec{a},\vec{b}). \end{aligned} \tag{34}$$

If $\rho_1^0,...,\rho_n^0,\pi_1^0,...,\pi_m^0$ are real numbers

$$\begin{aligned} \rho_i^0 &= -\frac{1}{2}\sum_{j=1}^{n}\frac{1}{v_i(\vec{a},\vec{b})}Cov(f,A_i,A_j)v_{ij}(\vec{a},\vec{b}), \quad i=1,...,n;\\ \pi_k^0 &= -\frac{1}{2}\sum_{l=1}^{m}\frac{1}{w_k(\vec{a},\vec{b})}Cov(X_k,X_l)w_{kl}(\vec{a},\vec{b}), \quad k=1,...,m, \end{aligned} \tag{35}$$

then a simple calculation shows that

$$\sum_{i=1}^{n}\rho_i^0 v_i(\vec{a},\vec{b}) = -\frac{1}{2}\sum_{i,j=1}^{n}v_{ij}(\vec{a},\vec{b})Cov(f,A_i,A_j); \tag{36}$$

$$\sum_{k=1}^{m}\pi_k^0 w_k(\vec{a},\vec{b}) = -\frac{1}{2}\sum_{k.l=1}^{m}w_{kl}(\vec{a},\vec{b})Cov(X_i,X_j). \tag{37}$$

By applying the relations (33), (34), (36), and (37) it follows that

$$\begin{aligned} E(f,u(\vec{A},\vec{X})) &\approx u(\vec{a},\vec{b})-\sum_{i=1}^{n}\rho_i^0 v_i(\vec{a},\vec{b})-\sum_{k=1}^{m}\pi_k^0 w_k(\vec{a},\vec{b})\\ &\approx u(a_1-\rho_1^0,...,a_n-\rho_n^0,b_1-\pi_1^0,...,b_m-\pi_m^0). \end{aligned} \tag{38}$$

In conclusion, $(\rho_1^0,...,\rho_n^0,\pi_1^0,...,\pi_m^0)$ is an approximate solution of (22). ■

6 Application to Evaluate Risk Aversion in a Grid

In this section we present a way of evaluating the risk aversion of an agent w.r.t. the situations of uncertainty, which can appear in grid computing.

We consider a grid composed of several nodes. In the functioning of a node N_i situations of uncertainty can appear: they can be described possibilistically or probabilistically. In the first case the behaviour of the node N_i is mathematically described by a fuzzy number A_i and in the second case by a random variable X_i. We assume that the grid has *n*+*m* nodes: the first *n* nodes are represented by the

possibilistic vector $\vec{A} = (A_1,...,A_n)$, and the other *m* nodes are represented by the random vector $\vec{X} = (X_1,...,X_m)$. The uncertainty situation related to the functioning of the grid is described by the mixed vector $(\vec{A},\vec{X})$. A utility function *u*: $\mathbf{R}^{n+m} \to \mathbf{R}$ expresses the attitude of the agent w.r.t. the *n+m* risk parameters. The mathematical framework is completed by a weighting function, conveniently chosen. The mixed risk premium vector ρ associated with the mixed vector $(\vec{A},\vec{X})$, the utility function *u* and the weighting function *f* evaluates the risk aversion of the agent w.r.t. the functioning of the grid. Its calculation is done by the formula given by Theorem 1 of Section 5.

Example 3. We consider a grid with *n+m* nodes $N_1,...,N_n,N_{n+1},...,N_{n+m}$. In the functioning of the grid *n+m* risk parameters appear. Assume that the risk parameters associated with the nodes $N_1,...,N_n$ are represented by the fuzzy numbers $A_i = (r_i,\alpha_i,\beta_i),\quad i = 1,...,n$ (see [9, 10]). Also we assume that the risk parameters associated with the nodes $N_{n+1},...,N_{n+m}$ are represented by the random variables $X_1,...,X_m$ having the covariance matrix $(\sigma_{kl})_{kl=1,...,m}$. We consider that the utility function *u*: $\mathbf{R}^{n+m} \to \mathbf{R}$ and the weighting function *f*: $[0,1] \to \mathbf{R}$ have the form: $u(x_1,...,x_n,y_1,...,y_m) = -e^{-2(x_1+...+x_n+y_1,...,y_m)}$; $f(\gamma)=2\gamma$ for $\gamma \in [0,1]$.

By Theorem 1 a simple calculation shows that an approximate solution of equation (22) of Section 5 is given by

$$\rho_i^0 = \sum_{j=1}^{n} Cov(f,A_i,A_j),\quad i = 1,...,n; \tag{39}$$

$$\pi_k^0 = \sum_{l=1}^{m} Cov(f,X_k,X_l),\quad k = 1,...,m. \tag{40}$$

We have for *i, j*=1,…,*n,*

$$Cov(f,A_i,A_j) = \frac{1}{36}[\alpha_i\alpha_j + \beta_i\beta_j + (\alpha_i+\beta_i)(\alpha_j+\beta_j)]. \tag{41}$$

Then (39) and (40) can be written as (42) and (43), respectively:

$$\rho_i^0 = \frac{1}{36}[\alpha_i \sum_{j=1}^{n} \alpha_j + \beta_i \sum_{j=1}^{n} \beta_j + (\alpha_i+\beta_i)\sum_{j=1}^{n}(\alpha_j+\beta_j)]; \tag{42}$$

$$\pi_k^0 = \sum_{l=1}^{m} \sigma_{kl},\quad k = 1,...,m. \tag{43}$$

7 Conclusions

This chapter proposes a new model of risk aversion with several parameters. The novelty of the model consists in the fact that some parameters are probabilistically modeled (by random variables) and others are possibilistically modeled (by fuzzy numbers).

The notion of mixed risk premium vector is introduced as a measure of risk aversion of an agent confronted by a risk situation with mixed parameters (probabilistic and possibilistic). Thus, a unification of two parallel concepts (probabilistic risk premium vector and possibilistic risk premium vector) is obtained.

The main result of the chapter is an approximate calculation formula for mixed risk premium vector.

As an application of our model a way of evaluating risk aversion, which appears in the functioning of a grid in which some nodes have a probabilistic description and others have a possibilistic description is presented.

Acknowledgments. Irina Georgescu was supported by CNCSIS-UEFISCSU project number PN II-RU 651/2010.

References

[1] Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets Syst. 1, 3–28 (1978), doi:10.1016/0165-0114(78)90029-5
[2] Arrow, K.J.: Essays in the theory of risk bearing. North-Holland, Amsterdam (1970)
[3] Duncan, G.T.: A matrix measure of multivariate local risk aversion. Econometrica 45, 895–902 (1977)
[4] Karni, E.: On multivariate risk aversion. Econometrica 17, 1391–1401 (1979)
[5] Georgescu, I., Kinnunen, J.: Multidimensional possibilistic risk aversion. Math Comput. Model 54, 689–696 (2011), doi:10.1016/j.mcm.2011.03.011
[6] Georgescu, I., Kinnunen, J.: Multidimensional risk aversion with mixed parameters. In: 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisoara, Romania, pp. 63–68 (2011); doi:10.1109/SACI.2011.5872974
[7] Fullér, R.: Introduction to Neuro-Fuzzy Systems. AISC. Springer, Heidelberg (2000)
[8] Carlsson, C., Fullér, R.: On possibilistic mean value and variance of fuzzy numbers. Fuzzy Sets Syst. 122, 315–326 (2001); doi: 10.1016/S0165-0114(00)00043-9
[9] Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York (1980)
[10] Dubois, D., Prade, H.: Possibility Theory. Plenum Press, New York (1988)
[11] Fullér, R., Majlender, P.: On weighted possibilistic mean and variance of fuzzy numbers. Fuzzy Sets Syst. 136, 363–374 (2003); doi:10.1016/S0165-0114(02)00216-6

Strong Berge and Strong Berge Pareto Equilibrium Detection Using an Evolutionary Approach

Noémi Gaskó, Rodica Ioana Lung, and Dan Dumitrescu

Babes-Bolyai University of Cluj-Napoca,
Faculty of Mathematics and Computer Science,
Str. M. Kogalniceanu 1, 400084 Cluj-Napoca, Romania
{gaskonomi,ddumitr}@cs.ubbcluj.ro, rodica.lung@econ.ubbcluj.ro

Abstract. Nash equilibrium is an important solving concept in Game Theory. Playing in Nash sense means that no player can deviate from the equilibrium strategy in order to increase her/his payoff. Some games can have several Nash equilibria. Strong Berge and strong Berge Pareto equilibria are important refinements of the Nash equilibrium. An evolutionary technique based on non-domination is proposed in order to detect the strong Berge and strong Berge Pareto equilibria. Some numerical experiments are presented in order to illustrate the proposed method.

1 Introduction

In non-cooperative Game Theory [1, 2] the most important solving concept is the Nash equilibrium [3]. Playing in Nash sense means that no player can change his strategy in order to obtain a better payoff for him.

In some games, which have several Nash equilibria, can appear a selection problem. Several refinements are introduced in order to solve this selection problem. Such equilibria is the strong Berge equilibrium [4]. The strong Berge Pareto equilibrium is one refinement of the Nash equilibrium, which means that pit is a strong Berge equilibrium and it is Pareto efficient.

2 Game Theory Prerequisites

Formally the description of a non-cooperative finite game is the following:

A finite strategic non-cooperative game is a system:

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 51–58.
springerlink.com

$$G = (N, S_i, u_i, i = 1.,..,n), \tag{1}$$

where:

- N represents a set of players, and n is the number of players;
- for each player $i \in N, S_i$ is the set of actions available,

$$S = S_1 \times S_2 \times ... \times S_n \tag{2}$$

is the set of all possible situations of the game and $s \in S$ is a strategy (or strategy profile) of the game;

- for each player $i \in N, u_i : S \to R$ represents the payoff function (utility) of the player i.

2.1 Nash Equilibrium

Nash equilibrium is a state from which no player can deviate in order to increase her payoff.

Formally we can describe:

Definition. A strategy profile $s^* \in S$ is a Nash equilibrium if the inequality holds:

$$u_i(s_i, s^*_{-i}) \le u_i(s^*), \forall i = 1,...,n, \forall s_i \in S_i, s_i \ne s^*_i, \tag{3}$$

where (s_i, s^*_{-i}) denotes the strategy profile obtained from s^* by replacing the strategy of player i with s_i.

2.2 Strong Berge Equilibrium

Berge introduced the strong Berge equilibrium [4]. The strong Berge equilibrium is stable against deviation of all the players except one of them.

Definition. A strategy profile $s^* \in S$ is a strong Berge equilibrium (SBE) of the game, if:

$$u_j(s^*_i, s_{-i}) \le u_j(s^*), \forall i \in N, \forall j \in N - i, \forall s_{-i} \in S_{-i.} \tag{4}$$

Remark. The strong Berge equilibrium is a subset of the Nash equilibrium:

$$SBE \subseteq NE. \tag{5}$$

Remark. If the number of players is equal to 2, the strong Berge equilibrium and the Nash equilibrium coincide.

2.3 Pareto Equilibrium

Defintion. A strategy profile $s^* \in S$ is Pareto efficient, when it does not exist a strategy $s \in S$, such that $u_i(s) \geq u_i(s^*)$, $i \in N$ with at least one strict inequality.

2.4 Strong Berge Pareto Equilibrium

Strong Berge Pareto equilibrium [5] is a refinement of the strong Berge equilibrium.

Definition. A strategy profile $s^* \in S$ is a strong Berge and Pareto equilibrium of the game, if s^* is a strong Berge equilibrium, and it is also Pareto efficient.

3 Evolutionary Equilibrium Detection

An evolutionary detection method is described in order to detect different equilibria. Generative relations for strong Berge and strong Berge Pareto equilibria are introduced.

3.1 Generative Relations

First generative relation for Nash equilibrium was introduced in [6]. A generative relation for an other refinement (Aumann equilibrium) is introduced in [7].

3.1.1 Generative Relation for Strong Berge Equilibrium

Consider two strategy profiles s^* and s from S. Denote by $sb(s^*, s)$ the number of players who lose by remaining to the initial strategy s^*, while the other players are allowed to play the corresponding strategies from s and at least one player switches from s^* to s.

We may express $sb(s^*, s)$ as

$$sb(s^*, s) = card[j \in N - i, u_j(s_i^*, s_{-i}) \geq u_j(s^*), \forall s_{-i} \in S_{-i}], \tag{6}$$

where *card[M]* denotes the cardinality of the multiset M (an element i can appear several times in M and each occurrence is counted in *card[M]*).

Defintion. Let $s^*, s \in S$. We say the strategy s^* is better than strategy s with respect to strong Berge equilibrium, and we write $s^* \prec_{SB} s$, if and only if the inequality

$$sb(s^*, s) < sb(s, s^*) \tag{7}$$

holds.

Defintion. The strategy profile $s^* \in S$ is a strong Berge non-dominated strategy, if and only if there is no strategy $s \in S, s \neq s^*$, such that s dominates $s \in S$ with respect to $\prec_{SB}$ i.e.

$$s \prec_{SB} s^*. \tag{8}$$

We may consider relation $\prec_{SB}$ as a candidate for generative relation of the strong Berge equilibrium. This means the set of the non-dominant strategies with respect to the relation $\prec_{SB}$ approximates the set of strong Berge equilibria.

3.1.2 Generative Relation for Strong Berge Pareto Equilibrium

We may express $sbp(s^*, s)$ as [8]:

$$\begin{aligned} sbp(s, s^*) &= card[j \in N, u_j(s_i^*, s_{-i}) \geq u_j(s^*), \forall j \in N - i, \forall s_{-i} \in S_{-i}] + \\ &+ card[i \in N, u_i(s) + u_i(s^*)] \end{aligned} \tag{9}$$

Defintion. Let $s^*, s \in S$. We say the strategy s^* is better than strategy s with respect to strong Berge Pareto equilibrium, and we write $s^* \prec_{SBP} s$, if and only if the inequality

$$sbp(s^*, s) < sbp(s, s^*) \tag{10}$$

holds.

Defintion. The strategy profile $s^* \in S$ is a strong Berge Pareto non-dominated strategy, if and only if there is no strategy $s \in S, s \neq s^*$, such that s dominates $s \in S$ with respect to $\prec_{SBP}$ i.e.

$$s \prec_{SBP} s^*. \tag{11}$$

We may consider relation $\prec_{SBP}$ as a candidate for generative relation of the strong Berge Pareto equilibrium: the set of the non-dominant strategies with respect to the relation $\prec_{SBP}$ approximates the set of strong Berge Pareto equilibria.

Remark. The strong Berge Pareto equilibrium can be interpreted as a composed strong Berge and Pareto equilibrium. The suitable generative relation is the sum of the two generative relations.

3.2 Evolutionary Detection Method

The nondomination with respect to the generative relation is used in the evolutionary multiobjective optimization method in order to detect the certain type of equilibrium (strong Berge and strong Berge Pareto equilibrium).The main idea is that the above described generative relations induce a specific nondomination concept associated to each type of equilibrium.

Every individual from the population is represented as a *n*-dimensional vector, representing a strategy profile. A population of strategies is evolved. The first population is generated randomly. Strategy population at iteration *t* may be viewed as the current equilibrium approximation.

The used approach can be summarized in a technique called Relational Evolutionary Equilibria Detection (REED):

1. Set *t* = *0*;
2. Randomly initialize a population of strategies;
3. Binary tournament selection and recombination using the simulated binary crossover (SBX) operator for;
4. Mutation using real polynomial mutation;

5. Compute the rank of each population member with respect to the generative relation. Order by rank;
6. Rank based selection for survival;
7. Repeat steps 3 - 6 until the maximum generation number is reached.

4 Numerical Experiments

For the numerical experiments a modified version of NSGA-II [9] has been used. The generative relation is considered for the rank based fitness assignment. The population size is 200 and the number of generation is 100. There are considered ten different runs with different random seed generators. Some continuous non-cooperative games are presented in order to illustrate the effectiveness and usability of the proposed evolutionary method.

4.1 Experiment 1

Let us consider the game G_1 having the following payoff functions [5]:

$$\begin{aligned} &u_1(x_1,x_2) = -x_1^2 - 2x_1 + 2x_2, \\ &u_2(x_1,x_2) = x_1 - 2x_2^2 - x_2, x_1, x_2 \in [0,1]. \end{aligned} \tag{12}$$

Fig. 1 presents the detected payoffs in a single run. Detected Nash, strong Berge and strong Berge Pareto equilibria payoffs are the same strategy profile (0,0).

4.2 Experiment 2

Let us consider the game G_2 having the following payoff functions:

$$\begin{aligned} &u_1(x_1,x_2,x_3) = x_1(10 - \sin(x_1^2 + x_2^2 + x_3^2)), \\ &u_2(x_1,x_2,x_3) = x_2(10 - \sin(x_1^2 + x_2^2 + x_3^2)), \\ &u_3(x_1,x_2,x_3) = x_3(10 - \sin(x_1^2 + x_2^2 + x_3^2)), x_1, x_2, x_3 \in [0,10]. \end{aligned} \tag{13}$$

Fig. 2 presents the detected payoffs in a single run. In the case of the strong Berge and strong Berge Pareto equilibria these refinements are useful to solve the selection problem.

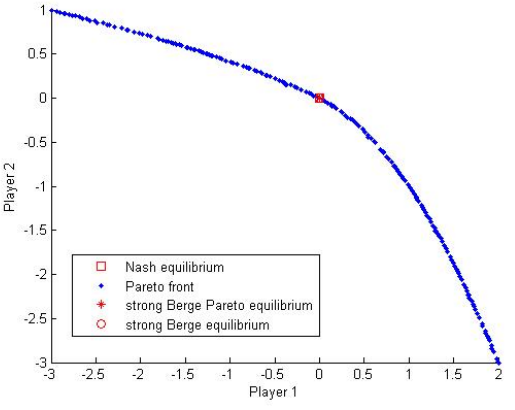


Fig. 1. Detected payoffs for the Nash equilibrium, Pareto front, strong Berge Pareto and strong Berge equilibria payoffs.

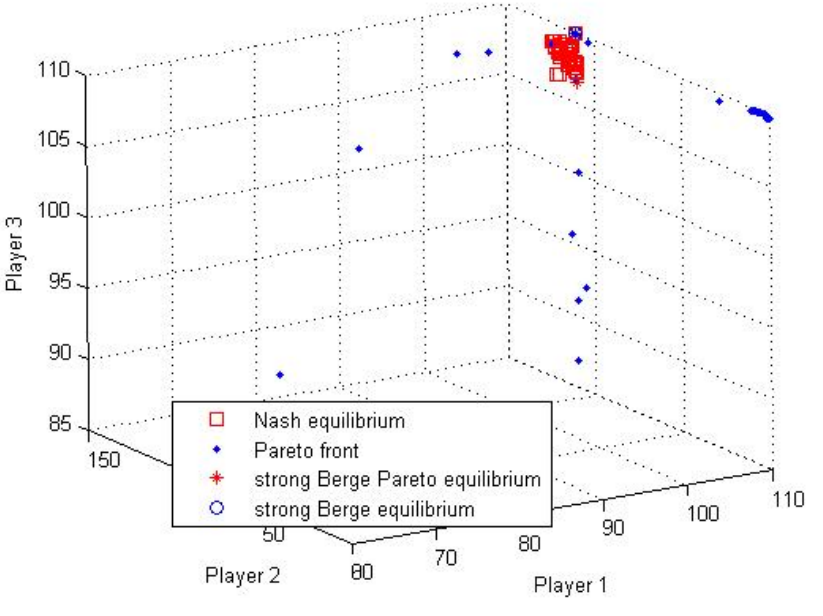


Fig. 2. Detected Nash equilibria, Pareto front, strong Berge Pareto, and strong Berge equilibria payoffs.

5 Conclusions

Nash equilibrium is an important solution concept in Game Theory. A disadvantage of this equilibrium is the selection problem which can appear by games having several Nash equilibria. Strong Berge and Strong Berge Pareto equilibria are some important refinements of the Nash equilibrium. Generative relations for these equilibria are described. An evolutionary method based on generative relations is described in order to detect several equilibria. Numerical experiments are presented in order to illustrate the potential of the proposed evolutionary technique.

Acknowledgments. The first author wishes to thank for the financial support provided from programs co-financed by The SECTORAL OPERATIONAL PROGRAMME HUMAN RESOURCES DEVELOPMENT, Contract POSDRU 6/1.5/S/3 - "Doctoral studies: through science towards society" and was made possible through the support of a grant from the John Templeton Foundation. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the John Templeton Foundation. This work was also supported by CNCSIS –UEFISCSU, project number PNII – IDEI 2366/2008.

References

[1] Gintis, H.: The bounds of reason, game theory and the unification of the behavioral sciences. Princeton University Press (2009)
[2] Osborne, M.: An introduction to game theory. Oxford University Press, New York (2004)
[3] Nash, J.F.: Non-cooperative games. Ann. Math. 54, 286–295 (1951)
[4] Berge, C.: Theorie generale des jeux a n-persones. Gauthier-Villars, Paris (1957)
[5] Nessah, R., Tazdait, T., Larbani, M.: Strong Berge and Pareto equilibrium existence for a non-cooperative game. Working Paper (2008)
[6] Lung, R.I., Dumitrescu, D.: Computing Nash equilibria by means of evolutionary computation. Int. J. Comp. Communic. Control III, 364–368 (2008)
[7] Dumitrescu, D., Lung, R.I., Gaskó, N., Mihoc, T.D.: Evolutionary detection of Aumann equilibrium. In: Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2010), Portland, OR, USA, pp. 827–828 (2010)
[8] Dumitrescu, D., Lung, R.I., Gaskó, N.: Detecting strong Berge Pareto equilibrium in a non-cooperative game using an evolutionary approach. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisoara, Romania, pp. 101–104 (2011)
[9] Deb, K., Agrawa, S., Pratab, A., Meyarivan, T.: A fast elitist Non-dominated Sorting Genetic Algorithm for multi-objective optimization: NSGA-II. In: Proceedings of the Parallel Problem Solving from Nature VI Conference, Paris, France, pp. 849–858 (2000)

A Proposal of the Information Retrieval System Based on the Generalized One-Sided Concept Lattices

Peter Butka[1], Jana Pócsová[2], and Jozef Pócs[3,4]

[1] Faculty of Economics, Technical University of Kosice,
Bozeny Nemcovej 32, 040 01 Kosice, Slovakia
peter.butka@tuke.sk
[2] Institute of Control and Informatization of Production Processes,
BERG Faculty, Technical University of Kosice,
Bozeny Necovej 3, 043 84 Kosice, Slovakia
jana.pocsova@tuke.sk
[3] Mathematical Institute, Slovak Academy of Sciences,
Gresakova 6, 040 01 Kosice, Slovakia
[4] Institute of Control and Informatization of Production Processes,
BERG Faculty, Technical University of Kosice,
Bozeny Nemcovej 3, 043 84 Kosice, Slovakia
pocs@saske.sk

Abstract. One of the important issues in information retrieval is to provide methods suitable for searching in large textual datasets. Some improvement of the retrieval process can be achieved by usage of conceptual models created automatically for analysed documents. One of the possibilities for creation of such models is to use well-established theory and methods from the area of Formal Concept Analysis. In this work we propose conceptual models based on the generalized one-sided concept lattices, which are locally created for subsets of documents represented by object-attribute table (document-term table in case of vector representation of text documents). Consequently, these local concept lattices are combined to one merged model using agglomerative clustering algorithm based on the descriptive (keyword-based) representation of particular lattices. Finally, we define basic details and methods of IR system that combines standard full-text search and conceptual search based on the extracted conceptual model.

1 Introduction

One of the possible and useful applications of knowledge modelling is to use created conceptual model from unknown (non-annotated) set of documents for

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 59–70.
springerlink.com

improvement of information retrieval results (against “classic” approaches). This process should include preprocessing of documents set, building of conceptual model (hierarchy of concepts, linkage of documents to hierarchy), and using of created model for improvement of IR, with or without the combination of conceptual and full-text search. This approach can be seen as problem decomposition method for building of conceptual model from text documents [2]. This conceptual model combines locally applied Formal Concept Analysis (FCA) and agglomerative clustering of particular models into one structure, which is suitable to support information retrieval process and can be easily combined with standard full-text search, as was described in [4].

Formal Concept Analysis (FCA, [7]) is a theory of data analysis that identifies conceptual structures among data sets. FCA produces concept lattice among the data that can be understand as knowledge-based model. Standard FCA works in two basic approaches - binary (crisp) and fuzzy case. In crisp case FCA is based on binary data tables (object has/has not attribute). Due to fact that data tables from textual documents usually contain real-valued attributes, some fuzzification of classic crisp method is needed. In our previous work we have used one approach to one-sided fuzzification presented in [8]. In this chapter we will provide practical usage of slightly different approach (theoretically presented in [3]) based on the generalized one-sided concept lattice (with the algorithm for its creation), which is able to support creation of concept lattices for contexts (object-attribute tables) with non-homogenous attributes, which are represented as complete lattices and the resulting set of model attributes can be viewed as a mix of qualitative (binary), quantitative (discrete, interval-based) or fuzzy logically based (truth values of various logical systems) ones. This is the main difference to our previous work and will provide several benefits. Since it is not necessary to concern only one type of complete lattice for possible values of attributes, we are able to support any type of attributes in the current domain of documents. Therefore we are able to use not only terms weights, but also any lattice-based truth value structure like binary, nominal, or ordinal attributes for qualitative rating of documents in dataset. This is really interesting specially for today's popular social-based rating of web documents and can be helpful for deep analysis of documents, where keyword-based similarity is high, but qualitative attributes can help in their correct distinction.

In order to provide searchable structure of the whole dataset from more concept lattices, merging of them is needed. Therefore, particular FCA models are labelled by characteristic attributes and simple agglomerative algorithm is used for clustering of local models, with the metric based on these characteristic attributes. This approach was also presented in [5], where grid computing was used to achieve better computing times. Then we will show the proposal for usage of merged

conceptual model in IR systems due to fact that it is natively keyword-based (with facet-based search by qualitative attributes), distributed and decomposed. Also linkage of documents to this conceptual structure is straightforward (thanks to attribute-object duality in FCA-based methods). Then we will specify details, methods, and characteristics of proposal for IR system, which is able to reuse conceptual model in combination with standard search.

In the second section of this chapter we will present idea of FCA-based merged conceptual model. Next, we will specify proposal for IR system suitable to reuse created conceptual model, which can be implemented in the future.

2 Hybrid FCA-Based Model of Documents Dataset

In this section we will describe a proposal of hybrid FCA-based model of text documents dataset. The presented approach is based on the decomposition of starting datasets (based on partitional clustering algorithm), creation of local models (based on FCA), description of such models in the form suitable for merging of models, and agglomerative clustering algorithm for their composition into final merged hierarchical model.

2.1 Problem Decomposition Approach for Model Creation

FCA can be used for creation of hierarchy of concepts and relations between these concepts. A problem with the use of FCA in domain of text documents with larger datasets is in time-consuming computation of concepts and hard interpretability of huge amount of concepts. Solution can be seen in combination with other methods like clustering algorithms used in the way known as problem decomposition approach:

- Clustering of input set of documents I can be viewed as a reduction step, where n clusters C_1, C_2,..., C_n of similar documents are found. The reduction step is based on the filtering of related terms of these objects (within the cluster). This step is top-down problem reduction approach to conceptual model extraction phase.
- Every cluster C_i (i=1..n) is independent training set (with the reduced cardinality of weight's vector), for each one a small FCA-based conceptual model O_i is built using our FCA algorithm.
- Small conceptual models O_1, O_2,..., O_n are merged together and global conceptual model M from tested collection is finally created.

Therefore, we can see this process as a transformation between several steps as follows: $S \rightarrow (C_1, C_2,..., C_n) \rightarrow (O_1, O_2,..., O_n) \rightarrow M$, where n is number of components of decomposition. As possible clustering method partitional algorithms can be used for identification of small groups (e.g., less than 10-15) of similar

documents. In our preliminary tests in previous work we have used Growing Hierarchical Self-Organizing Maps (GHSOM) algorithm for the first step in process. It is SOM-based (SOM - Self-Organizing Maps) hierarchical algorithm, where each layer is composed of independent SOM(s) that adjust their size according to the requirements of the input data. More information about this algorithm and its predecessors can be found in [6].

2.2 Generalized One-Sided Concept Lattice

In this section we provide a generalization of one-sided fuzzy concept lattices (independently described by Krajci [8] and by Yahia and Jaoua [1]). For the purposes of this chapter we provide only basic information needed for description of incremental algorithm, more theoretical details with proofs can be found in [3].

A 4-tuple $c = (B,A,L,R)$ is said to be a *generalized one-sided formal context* if the following conditions are fulfilled:

1. B is non-empty set of objects and A is non-empty set of attributes.
2. L: $A \rightarrow CL$ is a mapping from the set of all attributes to the class of all complete lattices CL. Hence, for any attribute a, $L(a)$ denotes a structure of truth values for attribute a.
3. R is generalized incidence relation, i.e., $R(b,a) \in L(a)$ for all $b \in B$ and $a \in A$. Thus, $R(b,a)$ represents a degree from the structure $L(a)$ in which the element $b \in B$ has the attribute a.

In our case relation R represents data table for analysis - document-term matrix of dataset vector model combined with other attributes for particular documents in one object-attribute model. The main difference to previous approaches is the possibility to create concept lattice for tables with different types of attributes (truth value structures).

If (B,A,L,R) is a generalized one-sided formal context, then we are able to define a pair of mappings $\uparrow: 2^B \rightarrow \prod_{a \in A} L(a)$ and $\downarrow: \prod_{a \in A} L(a) \rightarrow 2^B$ as follows:

$$\uparrow(X)(a) = \inf_{b \in X}(R(b,a)),$$

$$\downarrow(g) = \{\, b \in B : \ \forall a \in A, g(a) \leq R(b,a)\}.$$

A pair of mappings $(\uparrow,\downarrow)$ forms a Galois connection between 2^B and $\prod_{a \in A} L(a)$. Galois connections [9] are fundamental for FCA, therefore almost all known approaches of fuzzifying the classical FCA are based on such mappings and it is also true in our case. For formal context (B,A,L,R) denote $C(B,A,L,R)$ the set of all pairs (X,g), where $X \subseteq B$, $g \in \prod_{a \in A} L(a)$, satisfying $\uparrow(X) = g$ and $\downarrow(g) = X$. Set X is usually referred as *extent* and g as *intent* of the concept (X,g). If we define a partial order on $C(B,A,L,R)$ as $(X_1,g_1) \leq (X_2,g_2)$ iff $X_1 \subseteq X_2$ iff $g_1 \geq g_2$, then $C(B,A,L,R)$ with this partial order forms complete lattice, which is called *generalized one-sided concept lattice*.

Now we are able to provide an incremental algorithm for creation of generalized one-sided concept lattice. Let (B,A,L,R) be a generalized one-sided formal context. For $b \in B$ put $R(b)$ an element of $\prod_{a \in A} L(a)$ such that $R(b)(a) = R(b,a)$, i.e., $R(b)$ represents b-th row in data table R. Let 1_L denote the greatest element of $L = \prod_{a \in A} L(a)$, i.e., $1_L(a)=1_{L(a)}$ for all $a \in A$. Then pseudocode of algorithm for creation of generalized one-sided concept lattice can be written as follows.

Algorithm (Generalized One-Sided Concept Lattice)
Input: (B,A,L,R) – generalized formal context
begin
create lattice $L := \prod_{a \in A} L(a)$
$C := \{1_L\}, C \subseteq L$ is the set of all intents
while$(B \neq \emptyset)$
{
choose $b \in B$
$C^* := C$
for each $c \in C^*$
$C := C \cup \{c \wedge R(b)\}$
$B := B \setminus \{b\}$
}
for each $c \in C$
$C(B,A,L,R) := C(B,A,L,R) \cup \{(\downarrow(c),c)\}$
end
Output: $C(B,A,L,R)$ – set of all concepts

Let us remark that step for creation of the lattice $L := \prod_{a \in A} L(a)$ can be done in various ways and it is up to programmer. For example, it is not necessary to store all elements of $\prod_{a \in A} L(a)$, but it is sufficient to store only particular lattices $L(a)$, since lattice operations in L are calculated component-wise.

2.3 Description of Local Models

Use of the algorithm from previous subsection leads to creation of one generalized one-sided concept lattice for every subset from clustering phase within the presented problem decomposition approach. Particularly, if we will use hierarchical algorithm like GHSOM, then we will get one concept lattice for documents from every 'leaf' cluster (neuron without sub-map, in the end of expansion) in the hierarchy of GHSOM maps.

Every concept in particular lattice is characterized by extent and intent. Extent is set of objects (documents) and intent is set of corresponding attributes, which are weights of terms (words) and other attributes defined on documents. So, the concept can be viewed as a set of documents characterized by minimal value of its

attributes. Before the creation of lattices documents can be tested through attributes - if value of some attribute is lower then some threshold, new value of attribute is set to zero. This is inspired by work presented in [10] and helps in reduction of number of terms in concept lattice description.

If we have higher concept in hierarchy of lattice, the number of concept's attributes is smaller. Attributes with nonzero weights can be used as characteristics of actual concept (set of documents in concept). Every concept lattice then can be presented as hierarchy of concepts characterized by some attributes. Because we need some description of lattice for merging of lattice to one model, we need to extract attributes from particular lattices and create their representation based on these attributes. A weight of descriptive terms is based on level of terms in hierarchy (of course, important is highest occurrence of term). Other attributes (binary, ordinal, nominal) can be also characterized in similar way using some simple function for getting value (map value from its lattice structure to interval) from <0,1>. Then all attributes can be used for characterization of particular lattices and for clustering based on metric. Attributes from the input documents collection, which are not presented in concept lattice, have zero weight.

Function for assigning of weights can be different, e.g., $f(w,x) = w.x$, where w is value of base level of concept in hierarchy (this can be level of individual objects as concepts, or we can use some higher level as 'zero' base level), x is level of higher concept with term/attribute occurrence. It is possible to optimize weights and importance of higher level terms using selection of specific 'zero' level (this also helps to reduce number of attributes in local hierarchy). ´Height´ of concept in lattice is based on number of concept's documents (cardinality of concept).

Example of descriptive node based on one concept lattice (on some Times newspaper articles), attributes (terms) are described according to their highest occurrence within the concept lattice (first number in brackets) and by weight for this level (second number in bracket from interval <0,1>):

Documents: Times 60s/72 Times 60s/141 Times 60s/164 Times 60s/211
Terms: african(3, 1.0), southern(3, 1.0), prime(2, 0.75), britain'(2, 0.75), white(2, 0.75), feder(2,0.75), central(1, 0.5), black(1, 0.5), confer(1, 0.5), africa(1,0.5), field(1, 0.5), northern(1, 0.5), nationalist(1, 0.5), draft(1, 0.5)

Nodes can be combined into hierarchical agglomerative clustering structure by averaging of their weights (or any other operation suitable for their combination).

2.4 Clustering of Lattices Based on Descriptions

First step for clustering is to create one node for every local hierarchy (for every concept lattice), which contains: 1) list of documents; 2) list of characteristic attributes (sorted by value of weights); 3) vector of attributes weight's values (normalized). Particular nodes are then compared using vectors of attributes weights, vectors are normalized using interval <0,1>. After this step differences between

numbers of documents in particular nodes are respected. Comparison of lattices is used in process of agglomerative clustering of these nodes.

The similarity of two nodes is computed as follows. Let node U_i is represented using list of normalized weights of attributes $U_i = (u_{i1}, u_{i2}, ..., u_{in})$, where n is number of attributes in collection (with 0 implicitly used for attributes not included in the current node), and similarity of two nodes U_1 and U_2 is counted as:

$$sim(U_1, U_2) = \frac{\sum_{k \in K} \frac{\min(u_{1k}, u_{2k})}{\max(u_{1k}, u_{2k})}}{|K|}, \tag{1}$$

where K is a greatest subset of all attributes for which weight value is non-zero at least for one of the nodes U_1, U_2:

$$\max(u_{1k}, u_{2k}) > 0, \ \forall k \in K . \tag{2}$$

The proposed agglomerative clustering method is based on merging of most similar pairs of nodes in every step of the algorithm. Whole process can be described by the following procedure, where *MAXSIM* is maximal similarity found between all pairs in U for current step and *joint* is function for creation of new joint node from some pair of nodes p (e.g., by averaging of their weights):

Algorithm (Agglomerative clustering of description nodes)
Input: U – set of all nodes representing particular concept lattices
begin
  $A := \{(0,U)\}$
  while($|U|>1$)
  {
    $m := m + 1$
    $Pairs := \{(U_i,U_j): sim(U_i,U_j) = MAXSIM\}$
    $P := \emptyset$
    $New := \emptyset$
    foreach $p = (U_{p1},U_{p2}) \in Pairs$
      $P := P \cup \{U_{p1},U_{p2}\}$
      $New := New \cup joint(p)$
    $U := (U \backslash P) \cup New$
    $A := A \cup \{(m,U)\}$
  }
end
Output: A – agglomerative hierarchical structure of clustered nodes

Output of the algorithm is agglomerative hierarchical structure - simple hierarchy where leafs are nodes for particular concept lattices and nodes in the higher

levels are combination of their descriptions. Similarity of documents is based on their vectors of weights. Joint node from similar nodes is created by merging of set of their documents and by combination of their vector of weights. Final hierarchy contains nodes with the list of documents in it and the sorted list of characteristic attributes of nodes. Every node has link to its upper node and lower nodes. This structure and particular concept lattices can be reused in IR process.

3 A Proposal for Usage of Conceptual Model in IR System

The process of IR can be characterised by scheme represented in Fig.1. In control part it is important to connect blocks with control functionality like selecting of documents, choosing the method for indexing/analysis, and creation of index. The picture represents classic approach for IR system implementation. Only some of the blocks and edges are changed according to standard (full-text) search. In our case, we want to reuse standard architecture (process) with the minimum changes. Main differences to classic scheme are in five basic aspects, which are highlighted in Fig.1. It is possible due to fact that proposed conceptual model is keyword-based combined with some qualitative attributes (used for faceted search or as secondary ranking attribute) and is therefore compatible with standard textual operations and query format.

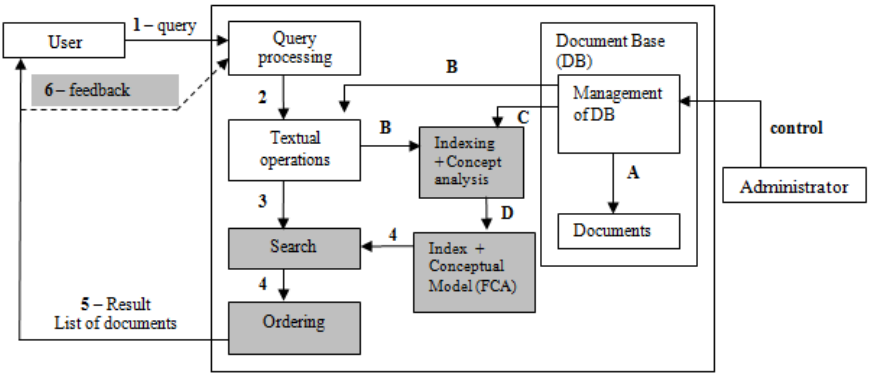


Fig. 1. Scheme of IR process with the combination of standard index and conceptual model.

Process of documents analysis (*Indexing + Concept analysis*) will be now combination of full-text analysis and proposed FCA-based method for text analysis. New generalized "index" (*Index + Conceptual model(FCA)*) is combination of full-text index and created hybrid FCA-based model (local FCA models and agglomerative clustering structure of their descriptions). For *Search* block, which returns unordered set of relevant documents to query, we need to define suitable method for returning the query results by full-text and conceptual search in specified combination. Therefore also *Ordering* block, which prepares ordered set of

documents from the previous step, needs to be revised in order to use the method for combination of full-text and conceptual ordering of results. Another place of possible change is within the process of query extension with automatic or manual (controlled by user) feedback based on the query expansion (step *6 - feedback*).

3.1 Use of Combined Index in Search

Use of our hybrid FCA-based model can be done in different ways, where difference is in application of query to the combined index. Classic approach is based on vector model of documents, while in our case we are able to combine *full-text search* (vector model based on standard part of index) and *conceptual search* (with the use of hybrid FCA-based model). By this combination we will get some specific modes of retrieval: 1) *Standard Full-text search* – query represented in vector model is used directly for search, only full-text part of combined index is used; 2) *Conceptual Model search* – vector representation of query is used for comparison with hybrid FCA-based model; 3) *Extended Full-text search* – full-text with automatic query expansion, where original query is used to search in conceptual model for expansion keywords; 4) *Combined Full-text and Conceptual search* – the result of search is provided in two separate lists (one for full-text and one for conceptual search), and lists are combined using ordering procedure in the next step of the process.

Full-text search is well-known method based on the analysis of similarity between vector representation of query and documents (query expansion do not change the method). More interesting is conceptual search in hybrid FCA-based model (conceptual part of combined index), where search follows three basic steps: 1) *Search for local models*; 2) *Search in selected local models*; 3) *Return of search result*. Details regarding possible options are summarized in next table.

In general, search within the descriptive nodes (search for local models) can be defined as some operator $H(q,A,V)$, which returns list of nodes from agglomerative structure A according to query q using search strategy V. Within the search in concept lattices we can define expansion operator $S(q,C,n, L)$, where C is most similar concept to query q in currently processed concept lattice L and n is specified size of neighbourhood, which is explored in order to return more concepts similar to C.

Use of particular approach for search can be switched manually or automatically. One problem with conceptual search is possibility that keyword is not available in FCA model due to preprocessing of documents dataset or pruning in concept lattice creation algorithm, but same term is (usually) still available in full-text part of combined index. Therefore, in case of automatic switch, we propose following procedure. If conceptual search found result, it is used for expansion of query in full-text. If not, then query is used in non-expanded form. Second choice is to manually switch modes (choosing the operators and their combinations) from the search approaches, e.g., using some user controlled switch in web browser application.

Table 1. Search possibilities for particular conceptual search steps in hybrid FCA-based model

Main Goal	Possible method	Details
1. Search for Local Models - use of agglomerative structure A	Search without hierarchical information	Only leaf nodes are compared to query
- representation of query q is compared to descriptive models of nodes - possible method = search strategy V	Strictly hierarchical search	Top-down approach where on every level most similar node(s) to query is(are) found and then is(are) used as a new "parent(s)" for search on lower level
Output: most similar leafs = concept lattices	Partially hierarchical search	Similarity is analysed to every node in structure - if node is higher in hierarchy, complete subtree under this node is used[1]
2. Search in Selected Local Models - use of particular concept lattices for search - representation of query q is compared to intent model of concepts - most similar concept to query q in current lattice L is C - possible method = use of expansion in search within the concept lattice; n is specified size of neighbourhood of concept C for expansion	Non-expanded search	$n = 0$, C is returned as result
	Expanded search	$n > 0$, result of search is concept C together with other concepts from the structure of current concept lattice L for which neighbourhood factor (path in concept lattice diagram) is less or equal to specified parameter n (conceptual expansion); first step of expansion is always up in hierarchy (if possible), because we want to find some new similar documents (from different subparts of diagram for current C)
3. Return of Search Results - concept(s) from conceptual search are returned as output from Search block	Return of results in conceptual search (used for strict version and for combined version)	Concepts with their extents (documents) and intents (attributes values) are returned
	Return of conceptual search results for full-text query expansion	If expansion of full-text search is expected, then intents are returned for automatic expansion of query

3.2 Ordering of Search Results

The output of ordering step is ordered list of documents according to some relevance function (to query). Difference in relevance evaluation comes from different modes of search. In case of full-text search relevance is directly based on vector

[1] The main idea here is to prefer leaf nodes (or lower level nodes) in order to avoid huge amount of concept lattices for exploring at the end. This can be achieved also by normalization of similarity measures using level of node in hierarchy (simply by the rule "lower is better").

model of IR (similarity of documents to query), for which we can define evaluation function E for full-text relevance („ranking“). If conceptual search is used only for full-text query expansion, function for ordering is still E.

For strict conceptual search we have list of concepts with their extents (lists of documents for every concept). Then conceptual ordering K is defined as follows: 1) documents from most similar concept (to query) are preferred, documents of other concepts are less preferred with increasing distance to most similar concept (according to neighbourhood), first occasion of document in concepts is used as its ordering base; 2) two documents with first occasion for same concept are ordered according to their vector-based similarity with query.

For combined search the input for ordering is in two lists, which can be provided separated (then ordering is based on previous approaches) or in one combined list. In second case we can define order differently, e.g., as weighted combination of full-text ordering E and conceptual ordering K using some parameters p and 1-p, where p is from <0,1>. If p is near 1, full-text search is dominant, if p is near 0, conceptual search is preferred. The whole ordering process can be then defined by operator $U(E,K,p)$.

3.3 Formal IR System and Possible Extensions

Standard formal description of IR System is defined as a tuple $(D, Q, F, R(q_i, d_j))$, where D is a set of documents representations, Q is a set of queries representations, F is a set of relations between representation of documents (index structure), and R is evaluation function for ordering of returned relevant documents to queries. In our case D and Q are not changed (vector-based model with some other attributes, if needed). F is structure of combined index (classic full-text index and hybrid FCA-based model). Evaluation function R is in our case combination of more parts, which are defined by particular operators: H for search in agglomerative hierarchical structure of nodes in conceptual model, S for extended search in concepts within the concept lattices, and U for ordering of searched results.

In general, system is designed and proposed with minimal change from the perspective of user. Ordering and presentation of results are provided in standard form as well as queries inputs (keywords with some facets and qualitative attribute entries). Also feedback is solved only using automatic query expansion, with possibility to have manual switch (if needed). We can still imagine many extensions to such system, which are basically of two types: 1) Improving the quality of retrieval results by updating the conceptual index and/or search algorithms without the involvement of users (includes extension of FCA-based model, improved quality of conceptual search, better model for combined index, etc.); 2) Extension based on involvement of users which include design and implementation of specific user interfaces with conceptual model in mind (choosing of concepts, use of GHSOM structure for search, etc.), manual feedback and lattice-based presentation of results.

4 Conclusions

In this chapter we have proposed method for creation of conceptual model based on the FCA (with the new method for creation of Generalized One-sided Concept Lattice) and agglomerative clustering and its usage within the system for information retrieval (IR). Automatic generation of such conceptual models can be strongly beneficial, especially for huge documents sets without semantic annotations. In such cases our proposal of conceptual analysis of documents and IR system can be simply applied and very useful, while it is not necessary to change keyword-based character of search and results (with conceptual information extracted directly from documents set).

Acknowledgments. This work was supported by the Slovak VEGA Grants No. 1/0685/12, No. 2/0194/10 and No. 1/0390/10, and also by the Slovak Research and Development Agency under contracts APVV-0208-10 and APVV-0035-10.

References

[1] Ben Yahia, S., Jaoua, A.: Discovering knowledge from fuzzy concept lattice. In: Kandel, A., Last, M., Bunke, H. (eds.) Data Mining and Computational Intelligence, pp. 167–190. Physica-Verlag, Heidelberg (2001)

[2] Butka, P.: Combination of problem reduction techniques and fuzzy FCA approach for building of conceptual models from textual documents. In: Paralic, J., Dvorsky, J., Kratky, M. (eds.) ZNALOSTI 2006, Proceedings of the 5th Annual Conference, Ostrava, Czech Republic, pp. 71–82 (2006)

[3] Butka, P., Pocs, J.: Generalization of one-sided concept lattices (2011) (Submitted)

[4] Butka, P., Pocsova, J.: Hierarchical FCA-based conceptual model of text documents used in information retrieval system. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisoara, Romania, pp. 199–204 (2011)

[5] Butka, P., Sarnovsky, M., Bednar, P.: One Approach to Combination of FCA-based Local Conceptual Models for Text Analysis – Grid-based approach. In: Proceedings of 6th International Symposium on Applied Machine Intelligence (SAMI 2008), Herlany, Slovakia, pp. 131–135 (2008)

[6] Dittenbach, M., Merkl, D., Rauber, A.: Using growing hierarchical self-organizing maps for document classification. In: Proceedings of European Symposium on Artificial Neural Networks (ESANN 2000), Bruges, Belgium, pp. 7–12 (2000)

[7] Ganter, B., Wille, R.: Formal Concept Analysis. Springer, Berlin (1997)

[8] Krajci, S.: Cluster based efficient generation of fuzzy concepts. Neural Netw. World 13, 521–530 (2003)

[9] Ore, O.: Galois Connexions. Trans. Am. Math. Soc. 55, 493–513 (1944)

[10] Quan, T.T., Hui, S.C., Cao, T.H.: A Fuzzy FCA-based Approach to Conceptual Clustering for Automatic Generation of Concept Hierarchy on Uncertainty Data. In: Proceedings of CLA conference (CLA 2004), Ostrava, Czech Republic, pp. 1–12 (2004)

Visualization and Simulation Tool for Analyzing P-Graph Based Workflow Systems

József Tick

Óbuda University, Becsi út 96/b, H-1034 Budapest, Hungary
tick@uni-obuda.hu

Abstract. The operation of business and social organizations – on their own and also as being part of a big network – is highly complex. Without the modeling of business processes the efficient and cost effective operation of such systems is impossible. For this purpose the workflow systems have been introduced. The optimization of the workflow of the administrative work-based systems means a great challenge. Elaborated method is available for the P-graph based workflow modeling of administrative processes. This chapter wishes to give some help to this optimization by presenting the concept of visualization and simulation tool supporting workflow analysis and examination. Such an analysis makes the determination of resource constraints, bottlenecks and redundancies possible and enables more efficient operation.

1 Introduction

Different organizations execute a series of related activities when fulfilling their tasks. The optimization of this activity series is a fundamental interest of the organization since it significantly determines the efficiency of its operation.

In this field it can also be stated that information technology incorporated in the business processes of organizations to its full extent in the last decades. This is true for the role players in the profit-oriented, as well as in the non-profit-oriented sphere. The automation of office processes meant the entire reorganization of the business processes with the appearance of the Business Process Management (BPM). BPM examines and models the processes from lots of aspects. Experts developed the model especially in the field of office automation that is mostly capable of modeling workflows.

Workflow is such a widespread modeling tool that enables the efficient modeling of business, within which office processes. Several representations of workflow models can be found in literature. The most widespread on is the Petri-based

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 71–82.
springerlink.com

approach, which possesses numerous favorable features. It models the execution and scheduling of the processes quite expressively. Tokens used in the model can be well assigned to documents, thus the model is suitable for modeling administrative processes as well. There several other advantages can be built in the Petri-Net-based modeling using various available extensions, for instance the model capable of time management as well as the colored Petri-Net. A different approach to be mentioned is the workflow modeling using UML diagrams. Its advantage is that UML is highly widespread and standardized, furthermore, its significant support from the CASE tools side.

A new possibility of workflow modeling, the P-graph based workflow model which is a proved, well performing model formulated on correct mathematical bases was presented in [1]. A big advantage of the P-graph based model is that its optimal structure can be synthetically generated. [2] The theory of P-graphs was developed to describe, model and not at last to optimize processes (e.g., chemical processes). Due to the mathematically precise and exact optimization method P-graphs play a significant role in the modeling of industrial processes. The application of P-graphs in workflow modeling was introduced in [3]. Instead of the raw materials, half-made product and end products used in industrial processes, the terms Input Document, Immediate Document and Product Document are use on P-graph-based modeling. The interpretation of activities and resources have also changed, it was adjusted to the concepts used in workflow. Fig. 1 displays the fundamental building blocks of a variant of this model capable of modeling special office administrative processes [14,15]. The workflow built on these blocks manages documents, one or more documents to be processed and resources that are necessary for the execution of the activity must be assigned to each activity.

Fig. 2 presents a P-graph based workflow model of a simple administrative process [3].

The present modern information technology systems are visual-based ones, which means that using a CASE tools complicated structures, schematics and models can be compiled from the building blocks using a mouse or working on a Touch screen. In most of the cases another subsystem gives the compiled model a breath of life, it rotates, makes it move, makes it operates as well as simulates its operation. At the formulation of the concept the individual visual elements of the simulation is presented on the individual parts of this model.

The analysis of the designed workflow's operation, the identification of the bottlenecks regarding processing, the effective examination of the workflow's dynamic changes by all means claims for the development of a visualization and simulation tool.

This chapter aims to elaborate the concept of a visualization and simulation tool and to determine the main components of the implementation. The following method is carried out to reach our goal:

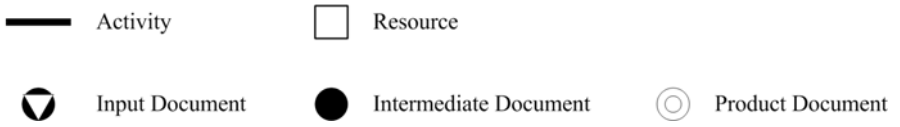


Fig. 1. The elements of the P-graph-based workflow used for modeling administrative processes.

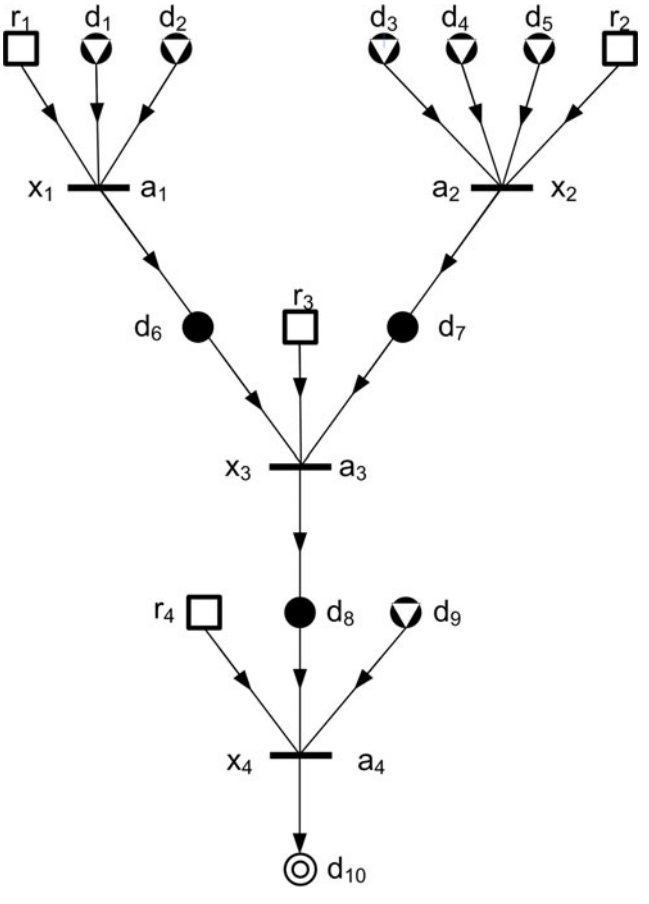


Fig. 2. P-graph-based workflow model of a simple administrative process.

First the types of services provided by tools serving to conduct similar tasks for workflows developed on different principles are examined. As a second step the simulation and visualization tools developed for non-workflow purposed p-graph usages are examined. As the next step the concept of a tool capable of the

visualization and simulation of p-graph based – especially for administrative purposes - workflow processes is determined.

2 Traditional Workflow Representation and Simulation Solutions

On the basis of the review of the corresponding literature on workflows it can be stated that modeling is interpreted quite loosely in this field. Solutions denoted as modeling are rather just graphical representations that are not justified by any correct mathematical theories. At the examination of modeling only the formal approaches that are the models that can be perfectly proved with adequate mathematical background are considered for our purposes. The systematic mathematical synthesis of the p-graph based workflow model is an advantage not only upon generating the optimal structure but at the proof of its validity as well. That is why the p-graph based workflow model has been introduced.

As it has already been mentioned several modeling concepts can be found in the professional literature. Among others the UML-based workflow modeling is a quite popular one but considering the above aspect only and exclusively the Petri-net based workflow modeling is to be considered. The introduction of the Petri net is not unintentional since it is a widespread descriptive tool for system modeling.

One of the most important advantages of the Petri-nets opposing other descriptive formalisms is that it defines a graphical and a mathematical representation simultaneously thus alloying transparency resulting from graphical representation and easy handling with the mathematical correctness of the formal models [6]. That is why Petri-nets can be well used for the precise and exact modeling of concurrent, asynchronous, distributed, parallel, non-deterministic and/or stochastic systems. Thus in several fields, e.g., in case of modeling operating systems [7, 14], to set up logistics models [8], the mathematical modeling method available due to the application of Petri-nets is successfully and pleasantly exploited to solve problems.

At the introduction of Petri-nets into workflow modeling Aalst and Hee [4] defined the fundamental structures using the elements of the Petri-net, where p (place) denotes one of the component types, the places, while t (transition) denotes the other component type, the transitions. To monitor the “processing” of the net tokens are used that describe state instance of control. Thus sequential-, parallel-, selective- and iterative- controlling patterns have been constructed. In case of the Petri-net talking about a bigraph, transition cannot be linked to transition and place cannot link to place. Fig. 3 displays the Petri net version of the already presented model in Fig. 2. To make the idea clear, the diverse type inputs, temporary storages and products are presented with places shaded differently. This is the way to differentiate resources defined in the previous model from documents.

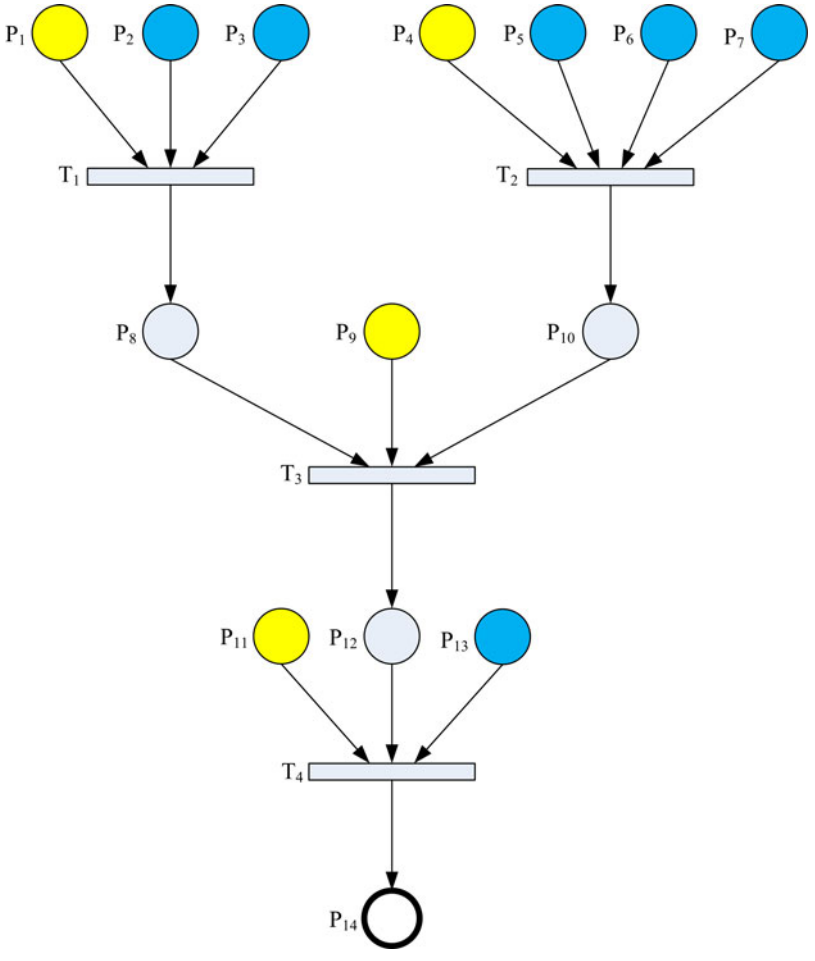


Fig. 3. Petri-Net based model of the problem modeled in Fig. 2.

In workflow modeling the application of higher level (extended) Petri-nets has been introduced to model more complex systems:

An important direction of the extension is the use of the colored Petri-net (CPN) formalism, which was elaborated by Jensen [9]. In case of workflow colorization [4, 10, 11] ensure the differentiability of the notions (pieces of information, documents) denoted by tokens from each other. Thus tokens can be typical, attributes can be joined to them, like the type of a document, etc. Therefore additional rules can be assigned to the transitions that prescribe the handling of various documents (different colors). This way of extension of the Petri-net makes the definition of the workflow model more exact and more general. It must be noted that the graphical representation in this case does not possess all the pieces of information as in the case of the basic Petri-net. In case of the colored extension a

pseudo coded specification must be paired to the description of transitions projecting the individual activities.

In case of simulation time management is always a corner point. Some changes in the volume of materials, the availability of resources are both time dependent factors. That is why in order to precisely determine the bottlenecks and the redundancies the appropriate time-factor management is necessary (resource peaks and downturns, etc.). The basic Petri-net does not handle time, that is why another direction of extension becomes important the so-called time dimension of the Petri-net. In this case tokens get a time stamp and a value that declares the time a token can be accessed during processing [5, 12, 13]. Thus the tokens waiting prior to a transition executing the process can be processed from diverse points of time. Process starts in FIFO system, always with the oldest token. Time can be assigned to the processing operation (transition) thus the time stamp of the arising token takes on the sum of the firing time and the processing time. With this extension the workflow model better fits real operations, enabling time management, the exploitation of critical time segments in the execution time, the prediction of expectations and/or the analysis of the emergence of the competitive conditions.

Different research directions of the modeling of Petri-net based workflow can be traced in professional literature. Majority of them deal with the application of the above mentioned extensions. Researches linked to the colored Petri-net are the dominant ones [4, 10, 11].

It is very useful to study the Petri-net solutions and their extensions for the formulation of the concept of a visualization and simulation tool supporting the application of p-graph based workflow in administrative model. Several solutions can be applied when designing the tool. It must, however, be noticed that the petri-net simulation has several efficiencies that hinder practical usability. These are for instance the individual assignment and differentiation of documents to and from each other, the life like management of the standby of resources.

3 Concept of the Visualization and Simulation CASE Tool

The traditional usage of p-graphs is primarily to model chemical processes. Here, the materials are homogenous, cannot be differentiated within itself, only the common qualitative and quantitative features are known (4 liters of water, 100 liters of hydrochloric acid, etc.). Consequently, the compliance and especially the simulation of these models are simple. These models are supported by several software products. PNS Draw software product that is accessible at http://www.p-graph.com must be mentioned, which is a Windows-based graph designing software. With the conventional drag and drop techniques it draws materials, operating units and connections between them, edits connection arrow position and define breakpoints and Bezier curves. The other tool to be mentioned is PNS Studio, which primarily manages traditional (e.g., chemical) processes. Its concept is far away from administrative processes; the more thorough investigation of this problem is not under the scope of this chapter and does not roll our research ahead.

The virtualization and simulation tool comprises such components that handle ordinary tasks such as GUI, communication component etc., however, there are three main components that form a mutually integrated and unified system:

- the editor component responsible for the input and modification of workflow
- the component checking the correctness of the graph and the compliance with workflow rules
- the components executing the actual simulation.

The information connection among the three components is secured by the internal data structure, while for the storage of the workflow an XML based descriptive file is at our disposal. Certainly the system manages more files that store the status of the simulation, the parameters needed for the operation, pre-defined test sequences etc. The components and their communication are shown in Fig. 4.

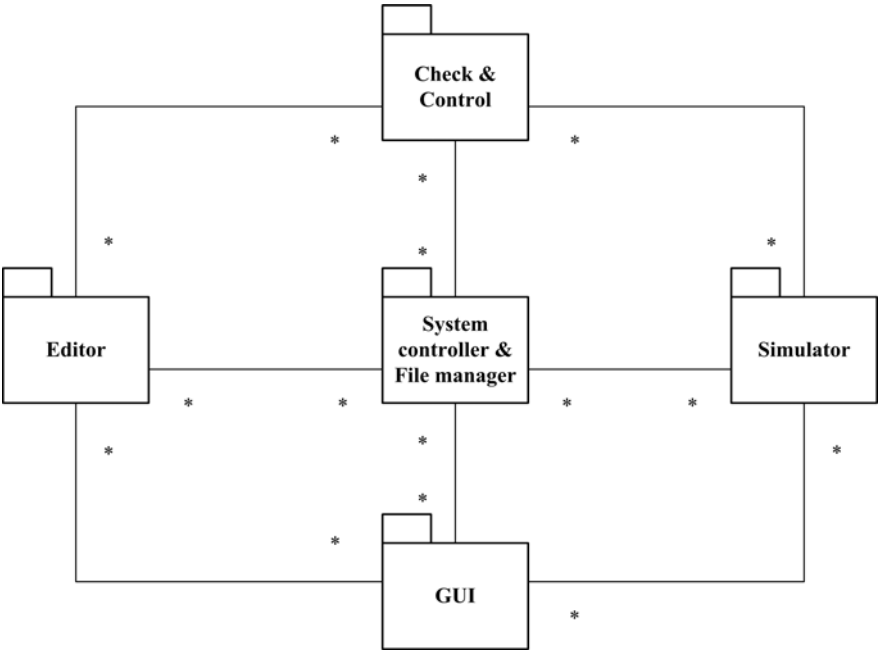


Fig 4. The components of the Visualization and Simulation CASE Tool.

The **editor** ***component*** does not differ basically from the traditional windows-based graphical editor that makes the basic building bricks available, where attributes can be defined and ensures the possibility of connection to shape the structure. The built-in intelligence and services of the editor makes the system well and easily usable. A well selected graphical menubar, an active case sensitive user help enable the easy handling of the whole system.

The ***checking and controlling component*** executes the rule-based checking of the workflow as graph essentially. From the aspect of simulation effectiveness, as

well as due to the active user support the concept of error-freeness is of high importance right at the compliance of the model. At the construction of the p-graph only the valid relations can be made, the non-legal operations can be hindered already when editing. For special requirements a posterior checking can be shot off either from the menu or it is automatically executed before the launch of the simulation.

The ***simulator component*** requires an entirely new conception, since such a type tool has not been developed so far. Simulation is an excellent tool for model examination and analysis. It enables the examination of the model that is optimized by mathematical methods in real-like environment and circumstances. Tracing the time duration is an important question, since the long-term behavior of the system can be analyzed only this way. Due to its novelty an especially great attention must be dedicated to the document-flow and the management of the documents in the workflow, that of resources, the management of time and the determination of the process of the simulation.

3.1 The Document Flow, the Management of Documents

Opposing the traditional usage fields of the workflow where the only problem is the resource management, in case of the workflows modeling administrative processes the management of the input and immediate documents needs distinct attention. In an administrative process documents are individually identified. In the course of the process the input documents used for the case are assigned to each other, which mean that for example the plan and the land register copy must be handled together in an activity when a building permit is issued. Should a document type is not available (e.g., if the land register copy has not arrived yet), then the building plan must be ordered back in the queue, since it cannot be processed. So the documents waiting to be processed are queuing in an input waiting line (FIFO) and wait for their turn to be processed.

In case of their turn should the other input documents linked to this case be available for the activity then the given document is processed, otherwise it is ordered to the rear of the waiting line. With this solution, however, asynchrony might occur, and might happen that the input documents of a case will never be all at once in ‘to be processed’ state. That is why such a solution must be chosen where the documents linked to a document of a case ready to be processed are looked for in the other waiting lines before the other inputs. If they are found, processing might start, otherwise the document is ordered to the rear of the queue. For the notation of the queuing and fowling of documents the book symbol is used as shown in Fig. 5.

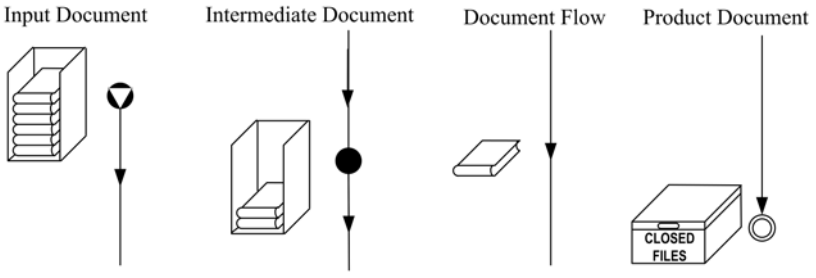


Fig. 5. Notation of the documents in the P-graph based workflow for administration modeling.

3.2 Management of Resources

In the course of the simulation of real processes one of the big problems is the finite nature of available resources. In this case it is true that in the case of workflow the resources are typically people who opposing machines need to have a break in work from time to time. That is why the time scheduling of the availability of resources must be defined before the launch of the simulation. Taking the eight hour work shift for granted the availability of resources can be presented with minute precision in a way similar to a bar chart, where dark areas mean availability, while light areas are the breaks.

Each activity can execute its task only if the documents needed for the case are available at each input and the resource is available for enough time before the launch of the activity. Otherwise the activity will turn into a standby condition which is denoted by the change in its color (from green to red). In such a case when more resources are linked to an activity then their intersection (AND operation) must be considered. The notation of resource is displayed in Fig. 6. (The triangle shows the actual time.)

3.3 Time Management

In the course of simulations the biggest problem is time management. It is especially true for real time simulations. During the simulation of administrative processes the simulated time must be transformed into a one with changeable value. An acceptable conventional ratio is one minute = one second, thus the simulation of

an eight hour shift takes eight minutes and the document flow speed can be easily tracked.

Execution time must be assigned to activities, which is possible also with minute precision. During simulation at the execution of an activity (active state) a continuously decreasing time displayed on a clock next to the symbol shows the state of the activity as illustrated in Fig. 6.

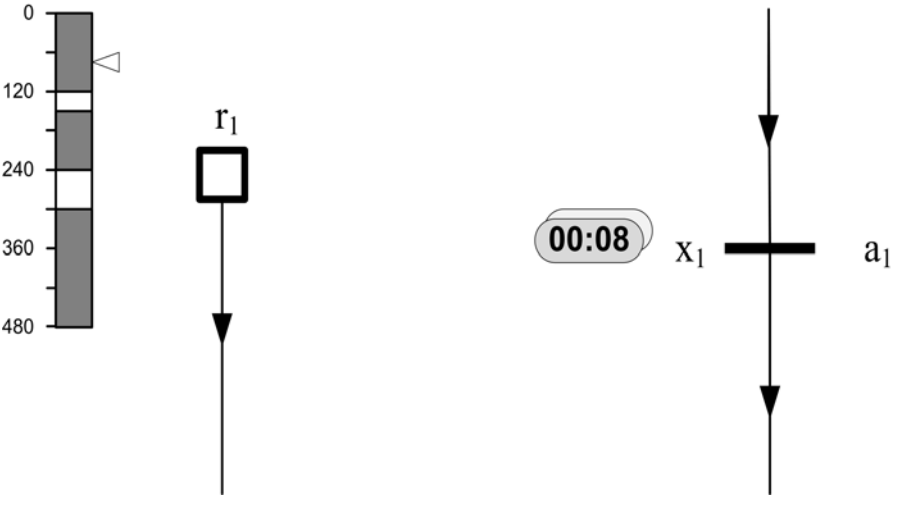


Fig. 6. Notation of the resources and the activities in the P-graph based workflow.

3.4 The Simulation Process

At the launch of the simulation a graph control is executed automatically which conducts the analysis of entirety and correctness. After the default time is set the order of activity executions can be determined in a recursive way by walking through the graph in a way that all the outputs are processed and the output corresponding to the its time shot appears. In the execution cycle each activity from inputs to outputs goes ahead with partial activity in the cycle corresponding to the respective time unit. This means that if a given activity is in standby position namely it is ready for processing and does not conduct execution at that time, then the first activity takes its next document in the queue and looks for the other documents linked to the relevant case at other inputs. This means a sequential search at each input. If the relevant documents are available at each input and the resource is available for the entire time claim of the activity, then the resource sets the time counter to the default state, takes the given documents from the queue and turns into execution state. In such a case when after having searched the entire

queue it does not find documents belonging to the case at the input or the resource is not available the activity stays in standby mode.

In such a case if an activity is in execution mode then the execution time of the activity is reduced proportionate to the time unit of the cycle. If it reaches 0, then a new document appears at the output and the activity turns into standby mode.

Going through each activity simulation reaches the end of a given cycle. After the cycle time of the simulation passes by the process is executed from the beginning.

During operation bottlenecks, the lack of resources, critically slow segments and redundant branches become visible. With the help of an editor the graph can be modified, the parameters of activities can be varied and resources can be assigned to individual activities.

4 Conclusions

Based on this chapter the realization of the system components has started. The editor component has been developed in total compliance with the concept. In the case of the component executing the simulation, however, a further problem to be solved is to develop an optimal strategy to match documents. In real life people involved in the process do not push such documents back in the list in the process systematically where matching cannot be conducted, but they "put them aside" and "keep in mind" and when the missing document/material arrives these documents are pulled back and processing goes on. In case of realization another problem has arisen, namely, that the administrators keep an eye on the legal deadline to be kept for the cases and just before the deadline they reorder the cases to finish processing by deadline.

Nevertheless, it can be argued that the outlined system concept has proven to be valid in the course of analyses, its realization is practical since it results in a well usable tool with the help of which the p-graph based workflow models describing administrative tasks can be effectively examined and analyzed.

References

[1] Friedler, F., Tarjan, K., Huang, Y.W., Fan, L.T.: Graph-theoretic approach to process syn-thesis: Axioms and theorems. Chem. Eng. Sci. 47, 1973–1988 (1992)

[2] Friedler, F., Fan, L.T., Imreh, B.: Process network synthesis. Problem Definition Networks 28, 119–124 (1998)

[3] Tick, J.: Application of P-graph-based workflow for administrative process modeling. In: Proceedings of the 9th IEEE International Symposium on Applied Machine Intelligence and Informatics, Smolenice, Slovakia, pp. 15–18 (2011)

[4] Van der Aalst, W.M.P., van Hee, K.M.: Workflow management – models, methods, and systems. The MIT Press, Cambridge (2002)

[5] Jablonski, S., Bussler, S.: Workflow management: Modeling concepts, architecture and implementation. International Thomson Computer Press (1996)

[6] Patericza, A. (ed.): Formalis modszerek az informatikaban. Typotex, Budapest (2006)

[7] Jeffry, J.M.: Using Petri nets to introduce operating system concepts. In: Proceedings of the SIGCSE Technical Symposium on Computer Science Education, San Antonio, TX, USA, pp. 324–329 (1991)
[8] Kemper, P.: Logistic process models go Petri nets. In: Proceedings of 7th Workshop Algorithmen und Werkzeuge für Petrinetze, pp. 69–74. Koblenz, Germany (2000)
[9] Jensen, K.: Coloured Petri nets and the invariant-method. Theor. Comput. Sci. 14, 317–336 (1981)
[10] Merz, M., Moldt, D., Müller-Jones, K., Lamersdorf, W.: Workflow modelling and execu-tion with coloured Petri nets in COSM. In: Proceedings of 16th International Conference on Application and Theory of Petri Nets, Torino, Italy, pp. 1–12 (1995)
[11] Liu, D., Wang, J., et al.: Modeling workflow processes with colored Petri nets. Comput. Ind. 49, 267–281 (2002)
[12] Qu, Y., Lin, C., Wang, J.: Linear temporal inference of workflow management systems based on timed petri nets models. In: Han, Y., Tai, S., Wikarski, D. (eds.) EDCIS 2002. LNCS, vol. 2480, pp. 30–44. Springer, Heidelberg (2002)
[13] van der Aalst, V.M.P.: Interval timed coloured Petri nets and their analysis. In: Ajmone, M.M. (ed.) Application and Theory of Petri Nets 1993, pp. 453–472. Springer, Heidelberg (1993)
[14] Tick, J.: Visualisation and simulation of P-graph based workflow systems. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, Romania, pp. 231–234 (2011)
[15] Tick, J.: P-graph-based workflow modelling. Acta Polytechnica Hungarica 4, 75–88 (2007)
[16] Shih-Yang, Y., Po-Zung, C., Chu-Hao, S.: Using Petri nets to enhance web usage mining. Acta Polytechnica Hungarica 4, 113–125 (2007)

Benchmark Based Comparison of Two Fuzzy Rule Base Optimization Methods

Zsolt Csaba Johanyák[1] and Olga Papp[2]

Department of Information Technology, Kecskemét College,
Izsáki út 10, H-6000 Kecskemét, Hungary
{johanyak.csaba,papp.olga}@gamf.kefo.hu

Abstract. Parameter optimization is a key step during the creation of a fuzzy rule based system. It also has a determining effect on the resulting system's performance. In this chapter, we examine the performance of several fuzzy systems obtained by applying two different optimization methods. In each case we start from an initial rule base that is created using fuzzy c-means clustering of a sample data set. The first examined optimization approach is the cross-entropy method while the second one is a hill-climbing based technique. We compare them in case of four benchmarking problems.

1 Introduction

Fuzzy inference based systems have been used successfully for several different purposes like fuzzy controllers (e.g. [31,32,34]), fuzzy expert systems (e.g. [13,27]), fuzzy modeling (e.g. [18]), management decision support (e.g. [30]) etc. Their performance is mainly determined by the rule base. In practice one of the most difficult aspects of the identification of a fuzzy system is the creation of the rule base. Having the necessary describing information in form of input-output data pairs this task can be a result of an automatic process as well.

In case of automatic rule base generation one of the following two approaches can be chosen. The first one divides the task into two separate steps, i.e. the structure definition and the parameter optimization (e.g. [6,17,31,34]).

The second approach works incrementally by simultaneously modifying the structure and the parameters, i.e. introducing or eventually eliminating rules and tuning the parameters of the membership functions (e.g. [20,36]).

In this chapter, following the first approach we set our focus on the second step. We examine two rule base parameter optimization solutions, i.e. the cross-entropy

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 83–94.
springerlink.com

method and a hill climbing approach based heuristic method. They are tried and compared in case of four benchmarking problems. In each case the initial rule base is created by a fuzzy clustering based method.

The rest of the chapter is organized as follows. Section 2 overviews the clustering based technique used for the generation of the initial rule base. Section 2 presents the two optimization methods being evaluated. Section 2 recalls the basic ideas of the FRISUV fuzzy inference method, which is applied owing to its low computational complexity and applicability in sparse rule bases. Section 5 presents the benchmark problems. The results of the benchmark tests are presented and discussed in Section 6.

2 Generation of the Initial Rule Base by ACP

The Automatic Fuzzy System Generation based on Fuzzy Clustering and Projection (ACP) [17] method covers the whole process of the automatic rule base generation from sample data. It consists of two steps, i.e. the generation of an initial rule applying fuzzy clustering and a parameter optimization based on a heuristic hill climbing type algorithm.

In this section, we recall the main points of the first step. The rule base generation starts with the determination of the optimal cluster number for a fuzzy c-mean (FCM) clustering of the output sample data. It is calculated by the help of a cluster validity index (CVI). One can find several CVIs in the literature. A survey and comparison of the available techniques was made by Wang and Zhang in [37]. ACP adapted the hybrid approach proposed by Chong, Gedeon and Kóczy [8] and enhanced it by the application of an upper limit for the cluster number.

After the determination of the optimal output cluster number one does a one-dimensional FCM clustering on the output data. The applied technique is based on the FCM proposed by Bezdek [5]. Then one estimates the output partition's fuzzy sets from the previously identified clusters. The Ruspini type partition with trapezoidal shaped fuzzy sets is produced by cutting the clusters at a specific α-level (usually α=0.85) and creating the set's cores from the width values of the cuts. One determines the supports of the sets automatically from the corresponding endpoints of the adjacent linguistic terms.

Next, the consequent linguistic terms are projected into the antecedent space. For each output fuzzy set one selects the corresponding data rows whose output falls into the support of the set. Then one does a one dimensional fuzzy clustering in each antecedent dimension on the current subset of the sample data. The cluster centers will be used later as reference points of the estimated antecedent fuzzy sets.

An identifier with three indices ($A_{i,j,k}$) is assigned to each cluster, where i is the ordinal number of the antecedent dimension, j is the ordinal number of the consequent fuzzy set in its partition, and k is the ordinal number of the cluster in the current clustering.

One can generate one or more rules for each output fuzzy set in the form

$$R_m : a_1 = A_{1,j,k_1} \; AND \ldots AND \; a_N = A_{N,j,k_N} \ldots \\ THEN \; b = B_j, \tag{1}$$

where m is the ordinal number of the rule, N is the number of antecedent dimensions. Thus one obtains several groups of cluster centers in each input dimension. Next, they are interleaved and renumbered in order to get a cluster partition in each dimension. Some of the cluster centers will be identical or nearly identical. They are united applying a prescribed distance threshold whose value is a parameter of the method. After a cluster union the related rules are also corrected or united. One estimates the antecedent partitions and linguistic terms using the same methods as in the case of the consequent partition.

3 Examined Rule Base Optimization Methods

3.1 Cross-Entropy Method

3.1.1 The Algorithm

The Cross-Entropy (CE) method, developed by Rubinstein in 1997 [1], has been devised to solve both continuous multi-extremal and discrete optimization problems. The basic idea of the algorithm is to solve a series of simple auxiliary smooth optimization problems iteratively using Kullback-Leibler cross-entropy, importance sampling, Markov chain and Boltzmann distribution.

Random search algorithms for global optimization include simulated annealing, tabu search and genetic algorithms. All of them use local search heuristics by applying the so-called local neighborhood structure. Compared to these algorithms, the CE method is quite different: it applies a global random search procedure. There is also a rising number of various applications using this algorithm: buffer allocation [3], static simulation models [4], queuing models of telecommunication

systems [5,6], control and navigation [7], DNA sequence alignment [8], reinforcement learning [9], project management [10] and others.

The CE method employs an iteration involving two main steps:

a. Approximation step: Generate random data samples.
b. Optimization step: Update the parameters of the random mechanism to get a better sample in the next iteration.

In the Approximation step importance sampling is used as a main requirement for the adaptive adjustment of the temperature in the Boltzmann distribution. The Kullback-Leibler cross-entropy is used as a measure of closeness between two samples. The Optimization step solves the optimization problem by converting it into a rare-event estimation problem, and then an adaptive algorithm is used to find the optimum.

We used the CE method for combinatorial optimization for our estimations in the benchmark problems. The algorithm works on a black-box basis: suppose we have a black box that returns an *S(x)* performance value for any *x* binary input data. We want to find the best *x* vector for which *S(x)* is minimal. The algorithm works as follows:

The CE method for combinatorial optimization

Initialize

Let *p(0)* be the initial probability

Let *t=1*.

Sampling

Generate samples *x(1), … , x(N)* according to probability *p(0)*

Calculate the performances *S(x(i))* for all *i* and order them increasingly *S(1) <= S(2) <=… <=S(N)*

Let *g(t)* be the *(1-r)* sample quantile of the performances:

g(t) = S([(1-r)](N))

Optimizing the probability parameters

Calculate *p(t)* using the original samples

$$p(t) = \frac{\sum_i I(S(x(i)) \geq g(t)) I(x(i) = 1)}{\sum_i I(S(x(i)) \geq g(t))} \tag{2}$$

Where *I* is an indicator function.

Stopping criteria

If the stopping criteria is met, then STOP, else *t=t+1* and

GOTO the sampling step.

The algorithm generates a series of levels *{g(t)}* and reference parameters *{p(t)}* such that *{g(t)}* tends to the optimal *g* = min (S(x))* and *{p(t)}* tends to the optimal reference parameter *p**.

The only initial parameters we need to fix are the sample size *N*, the initial probability parameter *p(0)*, the sampling probability density, and the sparsity parameter *r*. Apart from these fixed parameters, the algorithm is self-tuning.

3.1.2 Implementation Issues

Data *x* from our benchmark problems is given in the form of input-output pairs:

Input

- *SetsX*, implemented as cell array type, input values – each cell containing an *n(i)* x *m(i)* matrix that specifies the parameters of the *N(i)* fuzzy sets of the dimension *i* (initial values);
- *SetsY*, implemented as cell array type, output values – each cell containing an *n(i)* x *m(i)* matrix that specifies the parameters of the *N(i)* fuzzy sets of the dimension *i* (initial values).

Output

- *SetsX*, implemented as cell array type – estimated best input values;
- *SetsY*, implemented as cell array type – estimated best output values.

The algorithm was implemented using Matlab R2009a version on an Intel Xeon CPU 2.00 GHz 3.99 RAM PC. For sampling we used Matlab's Statistical Toolbox. For processing the fuzzy data we used the Fuzzy Rule Interpolation (FRIT) Toolbox [22].

Taking into consideration numerical errors, the indicator function has been altered to indicate nearness. The iteration stops if it reaches the maximum allowed iteration cycle number or when the updated parameter does not change from its previous value.

3.2 Parameter Optimization by ACP

ACP's algorithm [17] uses an iterative hill climbing approach for parameter optimization. After each step (each parameter modification) the resulting fuzzy system is evaluated by the means of a performance index (*PI*), which gives information about the difference between the prescribed output given by the sample data set and the calculated data set using the fuzzy system. Probably the most used and straightforward performance index is the relative value of the mean square error expressed in percentage of the output range (RMSEP).

The algorithm investigates the parameters one-by-one in course of each iteration cycle. First the system is evaluated with the current parameter set. Next one calculates two new values for the actual parameter, one by decreasing the original value and one by increasing the original value. The step is calculated by multiplying the range of the actual dimension by a constant (*C*). After the evaluation of the

system with the two new parameter values that value is kept from the available three ones (original and two new values), which results the best system performance. An iteration cycle contains the investigation of all parameters once.

At the end of the cycle one compares the actual performance with the performance of the system measured at the end of the previous cycle. If the amelioration of *PI* is greater than a threshold the value of *C* is doubled otherwise *C* is decreased to its half. When it becomes smaller than a threshold for the first time the process receives a second chance by increasing *C* to its original value. The algorithm stops when either *C* reaches its minimum for the second time or the prescribed number of iteration cycles is reached or the performance index of the system becomes better than a threshold value.

4 Fuzzy Inference

We chose a fuzzy rule interpolation based fuzzy inference (FRI) method due to the fact that it makes the development of fuzzy rule based systems possible applying a low complexity and compact rule base that contains only the most relevant rules. FRI methods (e.g. [4,7,15,19,24,25,26]) are able to infer even in those regions of the antecedent space where there are no applicable rules (sparse rule base). The selected FRI method was the Fuzzy Rule Interpolation based on Subsethood Values (FRISUV) [16]. It has a low complexity algorithm and its implementation exploits the vectorized computational capabilities of Matlab which ensures the fast determination of the output.

The basic idea of FRISUV is that it measures the closeness between the current fuzzy input and the rule antecedents by the means of an aggregative similarity measure which is calculated taking two components into consideration. The first one is the fuzzy subsethood value (*FSV*) that expresses the degree to which a fuzzy set is a subset of another set.

In case of fuzzy systems with a multidimensional input space *FSV* is interpreted as a subsethood value of two fuzzy relations, i.e. one defined by the rule antecedent sets ($\tilde{O}$) and one defined by the multidimensional observation ($\tilde{R}$). In order to keep low the computational complexity FRISUV does the calculations with the projections of the relations.

The second component of the similarity measure is the relative distance between the reference point of the fuzzy relation describing the observation and the relation corresponding to the rule antecedent. It is calculated by dividing the Euclidean distance by the maximum possible distance defined by the endpoints of the input ranges.

Finally, one calculates the similarity by

$$S\left(\tilde{O},\tilde{R}\right)=FSV\left(\tilde{O},\tilde{R}\right)\cdot 0.5+\left(1-d_{rel}\left(\tilde{O},\tilde{R}\right)\right)\cdot 0.5\,. \tag{3}$$

FRISUV determines the reference point of the conclusion from the position of the known rules' conclusion sets using the similarity between the current observation and the rule antecedents. The functional relationship is an extension of the Shepard crisp interpolation [35], which interpolates by calculating the weighted average of the known points' ordinates.

Supposing that the shapes of the consequent partitions' linguistic terms are identical and the sets differ only in their position the shape of the interpolated conclusion also has to adhere to this regularity. Thus its form will be identical with the shapes of the known sets.

5 Benchmark Problems

Ground level ozone (O_3) concentration is a very dangerous air pollutant. It is an irritating and reactive component in atmosphere that has negative impacts on human health, climate, vegetation and materials. Therefore it is important to develop models that are able to predict its concentration.

The experimental data were collected in an urban site of Northern Portugal [29]. In course of the experiments 10 characteristics were measured: the hourly average concentrations (in $\mu g/m^3$) of carbon monoxide (CO), nitrogen oxides (NO, NO2 and NOx) and O3; hourly averages of air temperature (T), solar radiation (RAD), relative humidity (RH) and wind speed (WS); the day of week (DW; the O_3 behavior is different on weekdays and on weekend).

Based on the results published in [29] we took only the most significant factors (NO, NO_2, NOx, T, RH, and O_3) into consideration in course of the modeling. We formed two groups of the experimental data: one containing 259 measurements for system training purposes and one with 84 measurements for testing purposes. The test data were selected randomly from the original sample.

Yield strength prediction aims the examination and modeling of the relation between the yield strength of thermoplastic composites and their composition. Thermoplastic composites are widely used for several different purposes. Their popularity origins from their advantageous mechanical properties, thermal stability, fire resistance, etc. that can be achieved by setting the proper proportion of the component materials (e.g. multiwall carbon nanotube, polycarbonate, ABS) [3].

The experimental results [1] were divided into two separate data sets, one for fuzzy system identification (training) and one for system validation (testing) purposes. The training data set contained the results of 31 experimental setups. The experiments were carried out with 10 replications, which results in total 310 experiments. The testing data set contained the results of 9 experimental setups. In this case the experiments were also carried out with 10 replications, which resulted in 90 experiments.

Prediction of petrophysical properties is a key issue in reservoir evaluation. Well log data analysis is an essential decision support tool in selecting the right

places for petroleum reservoir exploration. The collection of experimental data (borehole drilling, sampling and extensive laboratory analysis) is expensive. Therefore the fuzzy modeling of petroleum well data could be very advantageous [38].

In our case the main purpose was to create a fuzzy model that is applicable for the prediction of porosity (PHI) based on well log data described by three input variables. These are the gamma ray (GR), deep induction resistivity (ILD) and sonic travel time (DT).

We used the same training and testing data sets as in [38]. The training data set consisted of 71 data points and the testing data set consisted of 51 data points. The data were preprocessed and each variable was normalized to the unit interval.

The last benchmarking problem was based on a **synthetic data set** that was generated by the function

$$y = \left(1 + x_1^{-2} + x_2^{-1.5}\right)^2 \quad x \in [0.8, 1.8] \tag{4}$$

It is the same function as in [20]. The training data set consisted of 196 data points while the testing data set consisted of 82 data points.

6 Results

6.1 Performance of the Initial System

We used the relative value of the root mean square (quadratic mean) of the error (RMSEP) as performance index for the evaluation of the fuzzy systems. The performance of the initial system generated by the fuzzy clustering based approach overviewed in Section 2 is presented in Table 1 for both the case of the training and testing data sets.

Table 1. Performance of the initial systems

Data set	Train	Test
Ozone	49.7152	48.1454
Yield Strength	28.2649	18.8655
Well	14.5445	12.6675
Synthetic	32.6600	32.6729

6.2 Cross-Entropy Method

We performed repeated testing of the algorithm on the benchmarking problems presented in Section 5. The data sets have different sizes, ranges and number of dimensions.

We used two sets of parameters. In case of the first set the iteration number has been fixed to *25*, parameter *r* fixed to *0.1*, closeness indicator parameter has been set to *0.2*. In case of the second set the iteration number has been fixed to *25*, parameter *r* fixed to *0.05*, closeness indicator parameter has been set to *0.2*.

The sampling was performed using three different distribution functions:

a. *rand* - pseudorandom values drawn from the standard uniform distribution;
b. *betarnd* with parameters (*0.5,0.5*) - random numbers from the beta distribution;
c. *betarnd* with parameters (*2.0,2.0*) - random numbers from the beta distribution.

The test results for the cross entropy method applied to the benchmark problems can be seen in Table 2. Unfortunately the method did not improve the performance of the fuzzy system in case of all four benchmark problems. It performed best on the Ground Level Ozone Concentration data set where it achieved an improvement of over *70%* on the train set and over *72%* on the test set. The method also succeeded on the Synthetic data set, with a ratio of almost *40%* on the train set and over *44%* on the test set.

Table 2. Performance of the systems tuned by the CE method

Data set	Train	Test
Ozone	14.6531	13.2386
Yield Strength	38.2629	36.1852
Well	27.4533	28.5658
Synthetic	19.6862	18.2116

6.3 Parameter Optimization by ACP

Table 3 presents the results of the parameter optimization done with the help of the ACP method. Although ACP is only a local search method it ensured an at least 36.82% improvement of the system performance, i.e. in each benchmark case it ameliorated the RMSEP value. In order to avoid the overfitting of the system to the train data we kept that parameter set in each case, which ensured a quasi-optimal solution for both the train and test data.

Table 3. Performance of the systems tuned by the ACP method

Data set	Train	Test
Ozone	13.3393	11.8158
Yield Strength	6.2583	6.5763
Well	8.4876	8.2428
Synthetic	5.5484	6.7410

ACP performed best in case of the synthetic data set where it improved the system performance by 82.41% in case of the train data and by 78.62% in case of the test data. Most likely this result originates from the relatively smooth surface of the function and the uniform distributed sample points.

The worst results encountered in case of the petrophysical properties' prediction probably can be traced back to a local extreme where the method stuck.

7 Conclusions and Further Work

The CE method did not fulfill our expectations in case of all the data sets used for benchmarking purposes. It gave only in case of Ground Level Ozone Concentration prediction results that are close to the performance ensured by ACP. However, a clear advantage of CE was that it needed about 50-60% less time for finding the supposed solution.

Therefore our further research will concentrate on exploiting the global search capabilities and quickness of CE and the local search performance of ACP by developing an algorithm that combines the two approaches. We will also refine the CE method using the smoothing procedure for all of our test sets and will test the performance of the algorithm with other distribution functions.

Acknowledgments. This research was partly supported by the National Scientific Research Fund Grant OTKA K77809 and the Normative Application of R&D by Kecskemét College, GAMF Faculty.

References

[1] Ádámné, A.M., Belina, K.: Effect of multiwall nanotube on the properties of polypropylenes. Int. J. Mater. Form 1, 591–594 (2008)

[2] Alon, G., Kroese, D.P., Raviv, T., Rubinstein, R.Y.: Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment. Ann. Oper. Res. (2005); doi:10.1007/s10479-005-5728-8

[3] Ádámné, A.M., Belina, K.: Investigation of PP and PC carbon nanotube composites. In: Proceedings of 6th International Conference of PhD Students, Miskolc, Hungary, pp. 1–6 (2007)

[4] Baranyi, P., Kóczy, L.T., Gedeon, T.D.: A generalized concept for fuzzy rule interpolation. IEEE Trans. Fuzzy Syst. 12, 820–837 (2004)

[5] Bezdek, J.C.: Pattern reconition with fuzzy objective function algorithms. Plenum Press, New York (1981)
[6] Botzheim, J., Hámori, B., Kóczy, L.T.: Extracting trapezoidal membership functions of a fuzzy rule system by bacterial algorithm. In: Proceedings of 7th Fuzzy Days, Dortmund, Germany, pp. 218–227 (2001)
[7] Chen, S.M., Ko, Y.K.: Fuzzy interpolative reasoning for sparse fuzzy rule-based systems based on cuts and transformations techniques. IEEE Trans. Fuzzy Syst. 16, 1626–1648 (2008)
[8] Chong, A., Gedeon, T.D., Kóczy, L.T.: Projection based method for sparse fuzzy system generation. In: Proceedings of 2nd International Conference on Scientific Computation and Soft Computing, Crete, Greece, pp. 321–325 (2002)
[9] Cohen, I., Golany, B., Shtub, A.: Managing stochastic finite capacity multi-project systems through the cross-entropy methodology. Ann. Oper. Res. 134, 183–199 (2005)
[10] de Boer, P.T.: Analysis and efficient simulation of queueing models of telecommunication systems. Dissertation. University of Twente, Twente (2000)
[11] de Boer, P.T., Kroese, D.P., Rubinstein, R.Y.: A fast cross-entropy method for estimating buffer overflows in queueing networks. Manag. Sci. 50, 883–895 (2004)
[12] Helvik, B.E., Wittner, O.J.: Using the cross-entropy method to guide/Govern mobile agentïs path finding in networks. In: Pierre, S., Glitho, R.H. (eds.) MATA 2001. LNCS, vol. 2164, pp. 255–268. Springer, Heidelberg (2001)
[13] Hládek, D., Vaščák, J., Sinčák, P.: Hierarchical fuzzy inference system for robotic pursuit evasion task. In: Proceedings of 6th International Symposium on Applied Machine Intelligence and Informatics (SAMI 2008), Herľany, Slovakia, pp. 273–277 (2008)
[14] Homem-de-Mello, T., Rubinstein, R.Y.: Estimation of rare event probabilities using cross-entropy. In: WSC, vol. 1, pp. 310–319 (2002)
[15] Huang, Z.H., Shen, Q.: Fuzzy interpolation with generalized representative values. In: Proceedings of 2004 UK Workshop on Computational Intelligence, Loughborough, UK, pp. 161–171 (2004)
[16] Johanyák, Z.C.: Fuzzy rule interpolation based on subsethood values. In: Proceedings of 2010 IEEE Interenational Conference on Systems Man and Cybernetics (SMC 2010), Istanbul, Turkey, pp. 2387–2393 (2010)
[17] Johanyák, Z.C.: Sparse fuzzy model identification Matlab toolbox - RuleMaker toolbox. In: Proceedings of IEEE 6th International Conference on Computational Cybernetics (ICCC 2008), Stara Lesná, Slovakia, pp. 69–74 (2008)
[18] Johanyák, Z.C., Ádámné, A.M.: Mechanical properties prediction of thermoplastic composites using fuzzy models. Scientific Bulletin of "Politehnica" University of Timisoara. Romania. Transactions on Automatic Control and Computer Science 54(68), 185–190 (2009)
[19] Johanyák, Z.C., Kovács, S.: Fuzzy rule interpolation by the least squares method. In: Proceedings of 7th International Symposium of Hungarian Researchers on Computational Intelligence (HUCI 2006), Budapest, Hungary, pp. 495–506 (2006)
[20] Johanyák, Z.C., Kovács, S.: Sparse fuzzy system generation by rule base extension. In: Proceedings of 11th IEEE International Conference of Intelligent Engineering Systems (INES 2007), Budapest, Hungary, pp. 99–104 (2007)
[21] Johanyák, Z.C., Papp, O.: Comparative analysis of two fuzzy rule base optimization methods. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisoara, Romania, pp. 235–240 (2011)

[22] Johanyák, Z.C., Tikk, D., Kovács, S., Wong, K.K.: Fuzzy Rule Interpolation Matlab toolbox - FRI toolbox. In: Proceedings of IEEE World Congress on Computational Intelligence (WCCI 2006), 15th International Conference on Fuzzy Systems (FUZZ-IEEE 2006), Vancouver, BC, Canada, pp. 1427–1433 (2006)
[23] Keith, J., Kroese, D.P.: Sequence alignment by rare event simulation. In: WSC, vol. 1, pp. 320–327 (2002)
[24] Kóczy, L.T., Hirota, K.: Approximate reasoning by linear rule interpolation and general approximation. Int. J. Approx. Reason. 9, 197–225 (1993)
[25] Kovács, L.: Rule approximation in metric spaces. In: Proceedings of 8th IEEE International Symposium on Applied Machine Intelligence and Informatics (SAMI 2010), Herl'any, Slovakia, pp. 49–52 (2010)
[26] Kovács, S.: Extending the fuzzy rule interpolation "FIVE" by fuzzy observation. In: Reusch, B. (ed.) Computational Intelligence. Theory and Applications. AISC, pp. 485–497. Springer, Heidelberg (2006)
[27] Kovács, S., Kóczy, L.T.: Application of interpolation-based fuzzy logic reasoning in-behaviour-based control structures. In: Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2004), Budapest, Hungary, pp. 1543–1548 (2004)
[28] Menache, I., Mannor, S., Shimkin, N.: Basis function adaptation in temporal difference reinforcement learning. Ann. Oper. Res. 134, 215–238 (2005)
[29] Pires, J.C.M., Martins, F.G., Pereira, M.C., Alvim-Ferraz, M.C.M.: Prediction of ground-level ozone concentrations through statistical models. In: Proceedings of International Joint Conference on Computational Intelligence (IJCCI 2009), Funchal-Madeira, Portugal, pp. 551–554 (2009)
[30] Portik, T., Pokorádi, L.: Possibility of use of fuzzy logic in management. In: Proceedings of 16th International Conference on Building Services. Mechanical and Building Industry Days, Debrecen, Hungary, pp. 353–360 (2010)
[31] Precup, R.E., Doboli, S., Preitl, S.: Stability analysis and development of a class of fuzzy systems. Eng. Appl. Artif. Intell. 13, 237–247 (2000)
[32] Precup, R.E., Preitl, S., Tar, J.K., Tomescu, M.L., Takács, M., Korondi, P., Baranyi, P.: Fuzzy control system performance enhancement by Iterative Learning Control. IEEE Trans. Ind. Electron 9, 3461–3475 (2008)
[33] Rubinstein, R.Y.: The cross-entropy method for combinatorial and continuous optimization. Methodol. Comput. Appl. Prob. (1999); doi:10.1023/A:1010091220143
[34] Škrjanc, I., Blažič, S., Agamennoni, O.E.: Identification of dynamical systems with a robust interval fuzzy model. Automatica 41, 327–332 (2005)
[35] Shepard, D.: A two dimensional interpolation function for irregularly spaced data. In: Proceedings of 23rd ACM International Conference, New York, USA, pp. 517–524 (1968)
[36] Vincze, D., Kovács, S.: Incremental rule base creation with fuzzy rule interpolation-based Q-learning. In: Computational Intelligence in Engineering. SCI, vol. 313, pp. 191–203. Springer, Heidelberg (2010)
[37] Wang, W., Zhang, Y.: On cluster validity indices. Fuzzy Sets Syst. 158, 2095–2117 (2007)
[38] Wong, K.W., Gedeon, T.D.: Petrophysical properties prediction using self-generating fuzzy rules inference system with modified alpha-cut based fuzzy interpolation. In: Proceedings of 7th International Conference of Neural Information Processing (ICONIP 2000), Taejon, Korea, pp. 1088–1092 (2000)

Three Evolutionary Optimization Algorithms in PI Controller Tuning

Radu-Codruț David[1], Radu-Emil Precup[1], Stefan Preitl[1], József K. Tar[2], and János Fodor[2]

[1] "Politehnica" University of Timisoara, Department of Automation and Applied Informatics, Bd. V. Parvan 2, RO-300223 Timisoara, Romania
davidradu@gmail.com, {radu.precup,stefan.preitl}@aut.upt.ro,
[2] Óbuda University, Institute of Intelligent Engineering Systems, Bécsi út 96/B, H-1034 Budapest, Hungary
tar.jozsef@nik.uni-obuda.hu, fodor@uni-obuda.hu

Abstract. This chapter discusses three evolutionary optimization algorithms employed in the optimal tuning of PI controllers dedicated to a class of second-order processes with an integral component and variable parameters. The evolutionary algorithms used in this chapter are: Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA) and Charged System Search (CSS). The PI controllers are tuned such that to ensure a reduced sensitivity with respect to the parametric variations of the small time constant of the process. The application of the algorithms is illustrated in a case study.

1 Introduction

The optimal tuning of PI or PID controllers based on evolutionary algorithms is of great interest because these algorithms can offer solutions to optimization problems that involve complicated objective functions with possible multiple local minima and eventually complicated constraints. Some current applications of evolutionary algorithms to PI and PID controller tuning include real coded genetic algorithms, standard, modified and Niche Particle Swarm Optimization (PSO) algorithms applied to PID control of Single Input-Single Output (SISO) processes [1], Differential Evolution algorithms that solve multi-objective optimization problems specific to PI controller tuning [2], PSO and Gravitational Search Algorithms (GSAs) applied to optimal tuning of PI controllers such that to ensure a reduced parametric sensitivity with respect to process parametric variations [3], Cuckoo Optimization Algorithm applied to PID control of Multi Input-Multi Output processes [4], adaptive and fuzzy PSO applied to PID controller design [5, 6]. These algorithms prove to be effective in many applications.

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 95–106.
springerlink.com

In this chapter a comparison of three evolutionary algorithms dedicated to the tuning of PI controllers that belong to optimal control systems with a reduced parametric sensitivity with respect to the parametric variations of the process will be presented. The comparison is focused on a class of second-order processes with an integral component and with variable parameters. Starting with PSO and GSA evolutionary algorithms applied in [3] to the optimal tuning of PI controllers with a reduced parametric sensitivity with respect to the parametric variations of the gain and of the small time constant of the process, this chapter introduces a new algorithm with this respect, Charged System Search (CSS). CSS algorithms are based on Coulomb's and Gauss's laws from electrical physics and on the governing laws of motion from the Newtonian mechanics expressed in [7, 8] as a model in the framework of a multi-agent approach where each agent is a charged particle (CP). Other attractive civil engineering applications of CSS algorithms are given in [9, 10].

This chapter proposes twofold contributions. First, the application of a CSS algorithm to the optimal tuning of PI controllers is suggested. Second, the effectiveness of this algorithm is compared with PSO and GSA in a process control application.

This chapter is organized as follows. The process model and the optimization problems that ensure a reduced parametric sensitivity are defined in the next Section. Section 3 offers a short description of PSO, GSA and CSS focused on the optimal tuning of PI controllers. Section 4 applies PSO, GSA and CSS in the framework of a simple computational case study. Section 5 points out the conclusions.

2 Process Model and Optimization Problems

The process is modeled by the following transfer function:

$$P(s) = k_P / [s(1 + T_\Sigma s)], \tag{1}$$

where k_P is the process gain and T_Σ is the small time constant of the process. The transfer function in (1) includes the actuator and measuring element dynamics, it highlights the integral component, and it is a simplified model of processes with several complexities in servo system applications that belong to wide areas [11, 12, 13, 14, 15, 16, 17]. The simplification of the models leads to the fact that the parameters in (1) are variable, and the sensitivity analysis is justified.

Acceptable performance indices with respect to the modifications of reference input and of load-type disturbance inputs can be obtained if the process is controlled using the two-degree-of-freedom (2-DOF) control system structure presented in Fig. 1, where: r – reference input, y – controlled output, r_1 – filtered

reference input, $e = r_1 - y$ – control error, u – control signal, d– disturbance input, P – process, C – PI controller with the transfer function $C(s)$:

$$C(s) = k_c(1 + sT_i)/s = k_C[1 + 1/(sT_i)],\ k_C = T_i k_c, \tag{2}$$

k_C (k_c) – controller gain, T_i – controller integral time constant, and F – reference input filter with the transfer function $F(s)$:

$$F(s) = 1/(1 + T_i s). \tag{3}$$

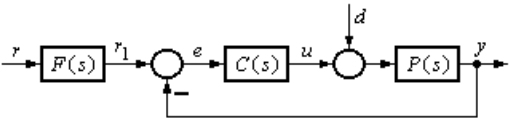


Fig. 1. Two-degree-of-freedom control system structure.

A state-space model of the process is [3]

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -(1/T_\Sigma)x_2 + (k_P/T_\Sigma)u + (k_P/T_\Sigma)d, \\ y &= x_1, \end{aligned} \tag{4}$$

where x_1 and x_2 are the state variables. The Extended Symmetrical Optimum (ESO) method [18] can be applied to tune the parameters of the PI controllers for such processes. The choice of the design parameter β specific to the ESO method within the domain $1 < \beta < 17$ guarantees a compromise to the imposed performance indices (overshoot, settling time, first settling time, phase margin, etc.) that can be set according to designer's option. The PI tuning conditions are

$$k_c = 1/(\beta\sqrt{\beta}T_\Sigma^2 k_P),\ T_i = \beta T_\Sigma. \tag{5}$$

The ESO method is advantageous as it reduces the number of the tuning parameters to only one, i.e., the design parameter β. The reference input filter can be itself tuned but, as shown in (3), it has a fixed parameter. An analysis of the 2-DOF control system structure that ensures flexibility in tuning both C and F is carried out in [19].

It is accepted that the state variable x_3 is the output of the integral component in the parallel construction of the PI controller, the state-space model of the PI controller is [3]

$$\begin{aligned} \dot{x}_3 &= (1/T_i)e, \\ u &= k_C(x_3 + e). \end{aligned} \tag{6}$$

A sensitivity model with respect to the parametric variations of the small time constant of the process T_Σ is derived as follows. In this context the controller is tuned according to (5) for the nominal set of values of the parameters of the process, $\{k_{P0}, T_{\Sigma 0}\}$, and the subscript 0 is used in the sequel to show the nominal values of variables and/or parameters. The state-space model of the PI controller is next expressed as

$$\begin{aligned} \dot{x}_3 &= [(1/\beta T_{\Sigma 0})]e, \\ u &= [1/(\sqrt{\beta} k_{P0} T_{\Sigma 0})](x_3 + e), \end{aligned} \tag{7}$$

and the models (4) and (7) lead to the following state-space model of the control system without F:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -[k_P/(\sqrt{\beta} k_{P0} T_{\Sigma 0} T_\Sigma)]x_1 - (1/T_\Sigma)x_2 + +[k_P/(\sqrt{\beta} k_{P0} T_{\Sigma 0} T_\Sigma)]x_3 \\ &\quad + [k_P/(\sqrt{\beta} k_{P0} T_{\Sigma 0} T_\Sigma)]r_1 + (k_P/T_\Sigma)d, \\ \dot{x}_3 &= -[1/(\beta T_{\Sigma 0})]x_1 + [1/(\beta T_{\Sigma 0})]r_1, \\ y &= x_1. \end{aligned} \tag{8}$$

The sensitivity model with respect to the modification of T_Σ is derived from (8), and its expression is

$$\begin{aligned} \dot{\lambda}_1 &= \lambda_2, \\ \dot{\lambda}_2 &= -[1/(\sqrt{\beta} T_{\Sigma 0}^2)]\lambda_1 - (1/T_{\Sigma 0})\lambda_2 + [1/(\sqrt{\beta} T_{\Sigma 0}^2)]\lambda_3 + [1/(\sqrt{\beta} T_{\Sigma 0}^3)]x_{10} \\ &\quad + (1/T_{\Sigma 0}^2)x_{20} - [1/(\sqrt{\beta} T_{\Sigma 0}^3)]x_{30} - [1/(\sqrt{\beta} T_{\Sigma 0}^3)]r_0, \\ \dot{\lambda}_3 &= -[1/(\beta T_{\Sigma 0})]\lambda_1, \\ \sigma_{T\Sigma} &= \lambda_1, \end{aligned} \tag{9}$$

where λ_1, λ_2 and λ_3 are the state sensitivity functions and $\sigma_{T\Sigma}$ is the output sensitivity function.

The following objective function is defined to achieve, when it is minimized, good performance indices and a reduced parametric sensitivity [3]:

$$I_{2e}^{T\Sigma}(\beta)=\int_0^{\infty}[e^2(t)+\gamma_{T\Sigma}^2\sigma_{T\Sigma}^2(t)]\mathrm{d}t\,, \tag{10}$$

where $\gamma_{T\Sigma}$ is the weighting parameter. The definition of the objective function in (10) allows the following definition of an optimization problem that ensures the optimal tuning of the PI controllers with a reduced sensitivity with respect to the parametric variations of the small time constant of the process:

$$\beta^*=\arg\min_{\beta>1} I_{2e}^{T\Sigma}(\beta)\,, \tag{11}$$

where β^* is the optimal value of the design parameter, and an inequality-type constraint is pointed out in (11) to guarantee the control system stability. PSO, GSA and CSS will be used as follows to obtain β^*.

3 Overview on PSO, GSA and CSS

The flowchart of a PSO algorithm is presented in Fig. 2 (a), where $j_{\max}$ is the maximum number of iterations. The equations that characterize the operating mechanism of PSO algorithms are given in [20, 21] and particularized in [3, 22] with focus on optimal tuning of controllers.

The process of validation of the obtained solution is unified for PSO, GSA and CSS. This process involves the fulfillment of the condition that guarantees the convergence of the algorithms, i.e., the convergence of the integral in (10)

$$|\,y(t_f)-r(t_f)\,|\le\varepsilon_y\,|\,r(t_f)-r(0)\,|\,, \tag{12}$$

where t_f is the final time moment and $\varepsilon_y=0.001$ for a 2% settling time. Theoretically $t_f\to\infty$ as shown in (10), but t_f takes practically a finite value to collect the information from all transients in the control system response. The condition (12) can be also viewed as a guarantee of the stability of the control system and a way to ensure zero steady-state control error of the control system as well.

The flowchart of a GSA is illustrated in Fig. 2 (b). The equations that characterize the operating mechanism of GSAs are defined in [23, 24] and presented in [3, 25] as particular versions applied to the optimal tuning of linear and fuzzy controllers.

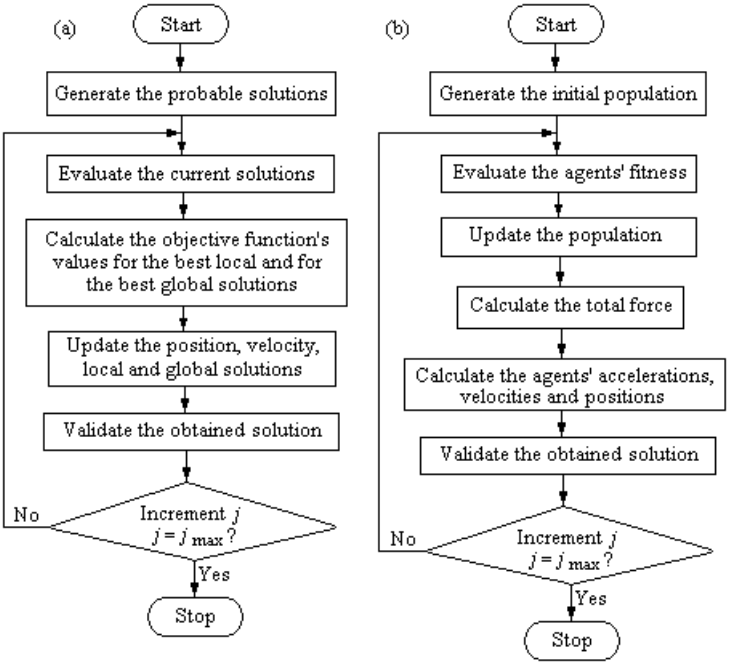


Fig. 2. Flowchart of PSO algorithm (a) and flowchart of GSA (b).

Each CP in a CSS algorithm has an associated magnitude of charge $q(i)$ and it creates an electrical field around its space. The magnitude of the charge is defined as follows in terms of the quality of its solution:

$$q(i) = (fit(i) - fit_{best}) / (fit_{best} - fit_{worst}),\ i = \overline{1, N}, \tag{13}$$

where fit_{best} and fit_{worst} are the so far best and the worst fitness of all CPs, $fit(i)$ is the objective function (i.e., the fitness) value of the i^{th} CP, and N is the total number of CPs. The separation distance r_{ij} between two CPs is

$$r_{ij} = (\| \mathbf{X}_i - \mathbf{X}_j \|) / (\| (\mathbf{X}_i + \mathbf{X}_j) / 2 - \mathbf{X}_{best} \| + \varepsilon), \tag{14}$$

where $\mathbf{X}_i$ and $\mathbf{X}_j$ are the positions of the i^{th} and j^{th} CPs, $\mathbf{X}_{best}$ is the position of the best current CP, ε is a small positive constant introduced to avoid singularities, and all positions in (14) are vectors in R^N.

Good CPs can attract the other agents and bad CPs repel the others, proportional to their rank c_{ij} calculated as follows in the context of using the attractiveness of all electric forces between any two CPs that may increase the exploitation ability of the CSS algorithm:

$$c_{ij} = \begin{cases} -1, & fit(i) < fit(j), \\ 1, & fit(i) \geq fit(j). \end{cases} \tag{15}$$

The value of the resultant electrical force F_j acting on the j^{th} CP is calculated as follows showing that each CP is considered as a charged sphere with radius a having a uniform volume charge density:

$$\begin{aligned} &F_j = q_j \sum_{i,i\neq j} (q_i r_{ij} i_1 / a^3 + q_i i_2 / r_{ij}^2) c_{ij} (X_i - X_j), \\ &i_1 = 0, i_2 = 1 \Leftrightarrow r_{ij} \geq a,\ i_1 = 1, i_2 = 0 \Leftrightarrow r_{ij} < a,\ j = \overline{1, N}. \end{aligned} \tag{16}$$

The new position $x_{j,new}$ and velocity $v_{j,new}$ of each CP are calculated according to the state-space equations [7, 8]

$$\begin{aligned} &x_{j,new} = rand_{j1} k_a (F_j / m_j)(\Delta t)^2 + rand_{j2} k_v v_{j,old} \Delta t + x_{j,old}, \\ &v_{j,new} = (x_{j,new} - x_{j,old}) / \Delta t, \end{aligned} \tag{17}$$

where k_a is the acceleration coefficient; k_v is the velocity coefficient to control the influence of the previous velocity, $rand_{j1}$ and $rand_{j2}$ are two random numbers uniformly distributed in the range of (0,1), m_j is the mass of the j^{th} CP which is set here to $q(j)$, and Δt is the time step set here to 1.

The following modifications of k_v and k_a with respect to the iteration index are used here as k_a is a control parameter of the exploitation and k_v controls the exploration process in order to ensure good performance of the CSS algorithm:

$$k_a = 1 - j / j_{\max},\ k_v = 1 + j / j_{\max}, \tag{18}$$

where j also indicates, for simplicity, the iteration index.

The flowchart of the CSS algorithm is presented in Fig. 3. The connection between the solution vector $\mathbf{X}_i$ and the variable in the optimization problem β is

$$\beta = \mathbf{X}_i, \tag{19}$$

and the connection between the fitness value and the objective function value is

$$fit(i) = I_{2e}^{T\Sigma}(\beta). \tag{20}$$

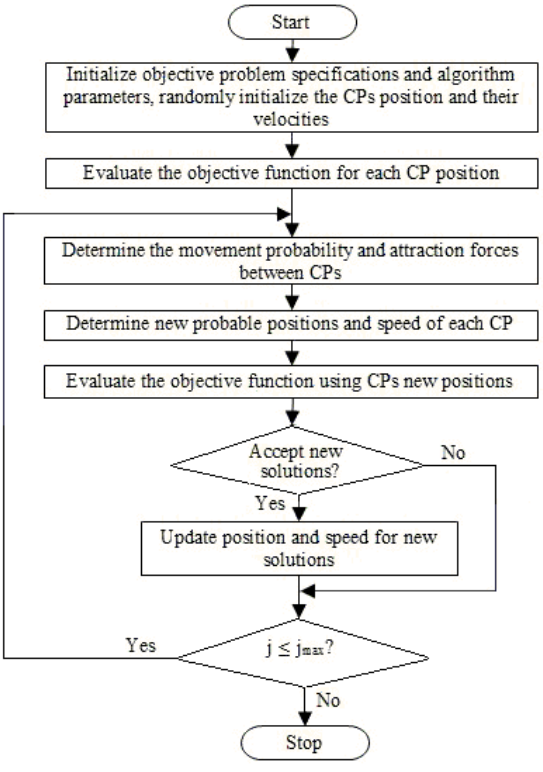


Fig. 3. Flowchart of CSS algorithm.

Equations (19) and (20) are presented in unified versions for all three evolutionary optimization algorithms. They map the algorithms onto the optimization problem defined in (11) with the objective function defined in (10).

4 Case Study

Some results concerning the three evolutionary optimization algorithms applied to the optimal tuning of PI controllers for the process with the transfer function specified in (1) and the parameters $k_P = 1$ and $T_\Sigma = 1\,\mathrm{s}$ are presented as follows. Different parameters and more complex models can be considered as shown in [26] for this process or in [27, 28, 29, 30, 31, 32, 33, 34, 35] for many similar but more convincing processes.

The three evolutionary algorithms were analyzed using the following values of the weighting parameter employed in solving the optimization problem defined in (11):

$$\gamma_{T\Sigma}^2 \in \{0, 0.001, 0.01, 0.1\}. \tag{21}$$

The acceleration weights in the PSO algorithm were set to $c_1 = c_2 = 0.3$. For GSA, the small constant was set to $\varepsilon = 10^{-4}$ and the initial value of gravitational constant was $g(k_0) = 100$. The CSS algorithm is characterized by the CPs considered as charged spheres with radius a having uniform volume charge density, $a = 1$ and $\varepsilon = 0.0001$. All three algorithms used an equal number of agents, $N = 20$, and $j_{\max} = 100$ iteration steps. The results for the PSO, GSA and CSS are synthesized in Table 1, Table 2 and Table 3, respectively.

Table 1. Results for PSO

$\gamma_{T\Sigma}^2$	β^*	k_C^*	T_i^*	$I_{2e\min}^{T\Sigma}$
0	6.128	0.403962	6.128	0.68278
0.001	5.55569	0.424259	5.55569	0.70662
0.01	4.17934	0.489155	4.17934	0.87171
0.1	3.21643	0.557587	3.21643	2.17727

Table 2. Results for GSA

$\gamma_{T\Sigma}^2$	β^*	k_C^*	T_i^*	$I_{2e\min}^{T\Sigma}$
0	6.126776	0.404002	6.126776	0.68278
0.001	5.557576	0.424187	5.557576	0.70662
0.01	4.181366	0.489036	4.181366	0.87171
0.1	3.214302	0.557772	3.214302	2.17727

Table 3. Results for CSS

$\gamma_{T\Sigma}^2$	β^*	k_C^*	T_i^*	$I_{2e\min}^{T\Sigma}$
0	6.12818	0.403956	6.12818	0.682779
0.001	5.55632	0.424235	5.55632	0.706624
0.01	4.17954	0.489143	4.17954	0.871706
0.1	3.22022	0.557259	3.22022	2.17727

The results presented in Tables 1 to 3 show algorithms' convergence to the optimal values of the controller tuning parameters and the minimum values of the objective function $I_{2e}^{T\Sigma}$. The results given in Tables 1 to 3 are illustrated for the minimum of the objective functions obtained after repeated simulations.

The results show that all three evolutionary algorithms give the same solution to the optimization problem (11). Therefore the evolutionary algorithms are validated for the given process and for the given optimization problem in the optimal tuning of PI controllers.

The behavior of the control systems without and after optimization is not presented here. However typical system responses obtained by digital simulation are presented for the considered process in [3] and for a process with different parameters and obtained by real-time experiments in [26].

5 Conclusions

This chapter has offered results concerning the application of three evolutionary optimization algorithms – PSO, GSA and CSS – to the optimal tuning of the parameters of PI controllers. A case study concerning the control of a process with a second-order dynamics and an integral component has been treated to exemplify the algorithms and to produce PI controllers with a reduced sensitivity with respect to the parametric variations of the small time constant of the process.

Future research will deal with the separate optimal tuning of the reference input filter parameters to increase the flexibility of the algorithms and to reduce the value of the objective function and with the reduction of the number of simulations in the implementations of the algorithms such that to be replaced by experiments on the real-world control systems. Applications to more complex processes will be aimed.

Acknowledgments. This work was supported by a grant of the Romanian National Authority for Scientific Research, CNCS – UEFISCDI, project number PN-II-ID-PCE-2011-3-0109, and by the cooperation between the Óbuda University, Budapest, Hungary, and the "Politehnica" University of Timisoara, Romania, in the framework of the Hungarian-Romanian Intergovernmental Scientific & Technological Cooperation Program.

References

[1] Li, J., Mao, J., Zhang, G.: Evolutionary algorithms based parameters tuning of PID controller. In: Proceedings of 2011 Chinese Control and Decision Conference (CCDC 2011), Mianyang, China, pp. 416–420 (2011)

[2] Reynoso-Meza, G., Sanchis, J., Blasco, X., Martinez, M.: An empirical study on parameter selection for multiobjective optimization algorithms using differential evolution. In: Proceedings of 2011 IEEE Symposium on Differential Evolution (SDE 2011), Paris, France, p. 7 (2011)

[3] David, R.C., Precup, R.E., Preitl, S., Tar, J.K., Fodor, J.: Parametric sensitivity reduction of PI-based control systems by means of evolutionary optimization algorithms. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisoara, Romania, pp. 241–246 (2011)

[4] Rajabioun, R.: Cuckoo optimization algorithm. Appl. Soft. Comp. 11, 5508–5518 (2011)

[5] Alfi, A., Modares, H.: System identification and control using adaptive particle swarm optimization. Appl. Math. Model 35, 1210–1221 (2011)

[6] Alfi, A., Fateh, M.M.: Intelligent identification and control using improved fuzzy particle swarm optimization. Expert Syst. Appl. 38, 12312–12317 (2011)

[7] Kaveh, A., Talatahari, S.: A novel heuristic optimization method: charged system search. Acta Mech. 213, 267–289 (2010)
[8] Kaveh, A., Talatahari, S.: Optimal design of truss structures via the charged system search algorithm. Struct. Multidisp. Optim. 37, 893–911 (2010)
[9] Kaveh, A., Talatahari, S.: Charged system search for optimum grillage systems design using the LRFD-AISC code. J. Constr. Steel Res. 66, 767–771 (2010)
[10] Kaveh, A., Safiri, P.: Charged system search algorithm for minimax and minisum facility layout problems. Asian J. Civil. Eng. 12, 703–718 (2011)
[11] Horváth, L., Rudas, I.J.: Modelling and solving methods for engineers. Elsevier, Academic Press, Burlington, MA (2004)
[12] Paláncz, B., Benyó, Z., Kovács, L.: Control system professional suite. IEEE Control Syst. Mag. 25, 67–75 (2005)
[13] Harmati, I., Lantos, B., Payandeh, S.: Fitted stratified manipulation with decomposed path planning on submanifolds. Int. J. Robot. Autom. 20, 135–144 (2005)
[14] Blažič, S., Matko, D., Škrjanc, I.: Adaptive law with a new leakage term. IET Control Theory Appl. 4, 1533–1542 (2010)
[15] Vaščák, J., Madarász, L.: Adaptation of fuzzy cognitive maps - a comparison study. Acta Polytechnica Hungarica 7, 109–122 (2010)
[16] Johanyák, Z.C.: Student evaluation based on fuzzy rule interpolation. Int. J. Artif. Intell. 5, 37–55 (2010)
[17] Garcia, A., Luviano-Juarez, A., Chairez, I., Poznyak, A., Poznyak, T.: Projectional dynamic neural network identifier for chaotic systems: Application to Chua's circuit. Int. J. Artif. Intell. 6, 1–18 (2011)
[18] Preitl, S., Precup, R.E.: An extension of tuning relations after symmetrical optimum method for PI and PID controllers. Automatica 35, 1731–1736 (1999)
[19] Precup, R.E., Preitl, S., Petriu, E.M., Tar, J.K., Tomescu, M.L., Pozna, C.: Generic two-degree-of-freedom linear and fuzzy controllers for integral processes. J. Franklin Inst. 346, 980–1003 (2009)
[20] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks (ICNN 1995), Perth, Australia, pp. 1942–1948 (1995)
[21] Kennedy, J., Eberhart, R.C.: A new optimizer using particle swarm theory. In: Proceedings of 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39–43 (1995)
[22] Precup, R.E., David, R.C., Preitl, S., Petriu, E.M., Tar, J.K.: Optimal control systems with reduced parametric sensitivity based on particle swarm optimization and simulated annealing. In: Köppen, M., Schaefer, G., Abraham, A. (eds.) Intelligent Computational Optimization in Engineering Techniques and Applications, pp. 177–207. Springer, Heidelberg (2011)
[23] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: A gravitational search algorithm. Inf. Sci. 179, 2232–2248 (2009)
[24] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: BGSA: binary gravitational search algorithm. Nat. Comput. 9, 727–745 (2010)
[25] Precup, R.-E., David, R.-C., Petriu, E.M., Preitl, S., Paul, A.S.: Gravitational search algorithm-based tuning of fuzzy control systems with a reduced parametric sensitivity. In: Gaspar-Cunha, A., Takahashi, R., Schaefer, G., Costa, L. (eds.) Soft Computing in Industrial Applications. AISC, vol. 96, pp. 141–150. Springer, Heidelberg (2011)

[26] Precup, R.E., David, R.C., Petriu, E.M., Preitl, S., Rădac, M.B.: Fuzzy control systems with reduced parametric sensitivity based on simulated annealing. IEEE Trans. Ind. Electron (2012); doi:10.1109/TIE.2011.2130493
[27] Baranyi, P., Yam, Y.: Singular value-based approximation with Takagi-Sugeno type fuzzy rule base. In: Proceedings of 6th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 1997), Barcelona, Spain, vol. 1, pp. 265–270 (1997)
[28] Baranyi, P., Korondi, P., Hashimoto, H., Wada, M.: Fuzzy inversion and rule base reduction. In: Proceedings of IEEE International Conference on Intelligent Engineering Systems (INES 1997), Budapest, Hungary, pp. 301–306 (1997)
[29] Chen, L., Bernard, O., Bastin, G., Angelov, P.: Hybrid modelling of biotechnological processes using neural networks. Control Eng. Pract. 8, 821–827 (2000)
[30] Abiyev, R.H., Kaynak, O.: Type 2 fuzzy neural structure for identification and control of time-varying plants. IEEE Trans. Ind. Electron 12, 4147–4159 (2010)
[31] Cotton, N.J., Wilamowski, B.M.: Compensation of nonlinearities using neural networks implemented on inexpensive microcontrollers. IEEE Trans. Ind. Electron 58, 733–740 (2011)
[32] Fang, Z., Xu, D., Tan, M.: A vision-based self-tuning fuzzy controller for fillet weld seam tracking. IEEE/ASME Trans. Mechatronics 16, 540–550 (2011)
[33] Linda, O., Manic, M.: Self-organizing fuzzy haptic teleoperation of mobile robot using sparse sonar data. IEEE Trans. Ind. Electron 58, 3187–3195 (2011)
[34] Peng, C., Han, Q.L.: Delay-range-dependent robust stabilization for uncertain T-S fuzzy control systems with interval time-varying delays. Inf. Sci. 181, 4287–4299 (2011)
[35] Wu, H.M., Hwang, C.L.: Trajectory-based control under ZMP constraint for the 3D biped walking via fuzzy control. In: Proceedings of 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), Taipei, Taiwan, pp. 706–712 (2011)

Statistical Analysis of Next Generation Network Traffics Based on Wavelets and Transformation ON/(ON+OFF)

Zoltan Gal and Gyorgy Terdik

University of Debrecen, Egyetem ter 1, H-4032 Debrecen, Hungary
ZGal@unideb.hu, Terdik.Gyorgy@inf.unideb.hu

Abstract. The significant increase of trunk channel bandwidth makes much easier to integrate different types of traffics on the tier links without activating high processing power consuming QoS (Quality of Service) mechanisms in the intermediate nodes. Self-similarity, long range dependence and fractal characteristics of packet flows are strongly influenced by the QoS parameters in congested network environment. Several models are proposed for the qualitative and quantitative evaluation of physical phenomenon supervened on different OSI layers at the routers and switches. Most of these claims relatively long traces for evaluating both scale independence and fractal characteristics. The highlights of common usage of wavelet and ON/(ON+OFF) transformations in network traffic analysis are evaluated in this chapter. We take into consideration the channel load and the channel intensity as complex time series for evaluation the statistical characteristics of changes in time of the flows nature in packet switched networks. UDP and TCP traffics in tier and LAN networks are considered and statistically analyzed based on MRA (Multi Resolution Analysis) wavelets method. A fast detection algorithm of data and real time traffic burstiness is presented for a QoS based packet switched network environment with congestion.

1 Introduction

Services of the Future Internet and Next Generation Networks have been universally and intensively explored in the communication area. The packet switched technology compounded with the multi gigabit capacity of the current transmission channels creates new perspectives for the converged data and real time network services. High speed tier links are rarely congested and the aggregation of high number of different traffics is likewise one best effort data flow transmitted with high speed. The parameters of IP data traffic networks are strongly influenced by the type of applied communication mechanisms. Compressing the

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 107–121.
springerlink.com

physical space and the communication delay between entities assists to improve the yearly doubling traffic on the Next Generation Network. In spite of most popular character of the Internet no global guarantees exits for communication services. Serious efforts are set for development of QoS methods worldwide involving analyzing of bandwidth, delay variation, blocking rate of services, packet lose rate, reliability and other parameters [1]. The QoS mechanisms for IP data networks are based on two approaches: i.) High channel bandwidth conception represented by the IETF (Internet Engineering Task Force) assuming enough and large capacity of cheap channel capacity for all network applications working on those links; ii.) Managed bandwidth conception having origin from the BISDN services of the ATM Forum and assuming finite channel capacities with optimum set of resource parameters.

The first method suppose higher evolution rhythm of the communication technology than the demand of users, sustaining low price of bit transmission, and on the other side the switching delay time of switches/routers can be decreased continuously. Such an over dimensioned worldwide network could resolve lot of current problems, but the economic aspects of network resource wastage prevents this solution. The managed bandwidth conception allows increase of network resources for applications running on limited channel capacities. Several technologies offer such solutions. First data link layer technology with QoS features was Frame Relay, allowing traffic prioritization with the bit DE (Discard Eligible) of FR frame and FECN/BECN (Forward Explicit Congestion Notification / Backward Explicit Congestion Notification) mechanisms. Explicit QoS mechanisms were introduced at the ATM (Asynchronous Transfer Mode) technology by the set-up phase of connection oriented (CO) L2 link. The MPLS (Multiprotocol Label Switching) technology uses L2.5 label switching CO mechanism to provide service guarantees approximate to the ATM [2].

The introduction and dispersion of QoS mechanism were generated by the VoIP (Voice over IP) services running on the Internet [3]. The converged network services like data, video and voice on the same common infrastructure has strong influence in QoS demand composition. The majority of current Internet applications are accessed from Ethernet end nodes. This aspect implies cost reduction if the intermediate nodes of LAN/MAN networks have similar L2 technology because the inexistence of frame type modification during the transmission simplifies the amount of switching task at the core and distribution level [1]. The different types of network applications put up different transfer guarantees [4]. The network resource demand of applications is varying in time even the applications are differently tolerant to the transmission delay and the delay variation. Several applications can support data loss while others cannot.

Two QoS mechanisms are proposed for popular network applications today. Intserv is based on RSVP (Resource Reservation Protocol) and assures quantitative metric and latency time for the guaranteed services. The connection allowance control process decides of the successful of the new connection request. This control mechanism supposes necessary RSVP intelligence at the intermediate

nodes. Based on this strict assumption, Intserv mechanism is only useful in autonomous system environment. The second, Diffserv QoS mechanism utilizes special fields of the PDUs (Protocol Data Unit) and the QoS priority is transmitted as color code inside of each PDU for a given OSI layer. Global usage of both, v4 and v6 versions of the IP protocol in LAN/MAN environment makes crucial the QoS mechanisms like DiffServ and IntServ for providing qualified integration of data, voice and video traffics. This classification of each PDU implies different treatment possibility inside of the routers and switches [5].

2 Wavelet Analysis Fundamentals

The wavelet based Multi Resolution Analysis (MRA) proposes fast pyramidal algorithm requiring ~O(n) computation steps for determining the self-similarity measure of the traces with n samples, consisting serious interest in the low delay aspect of the burst detection. Several models are proposed for the qualitative and quantitative evaluation of physical phenomenon supervened on different OSI layers at the intermediate nodes. Most of these claims relatively long traces for evaluating the scale independence and fractal characteristics. Let $X_t, t \in Z$ be a discrete time stochastic process. A stochastic process in continuous time $Y(t), t \in R$ with stationary increment is called a cumulative (background) process of X_t and X_t is the increment process of $Y(t)$, if the following relation holds [6]:

$$X_t = Y(t) - Y(t-1), \forall t \in Z. \tag{1}$$

In the case of network traffic modeling, X_t is stationary in the sense that the behavior or structure is invariant to the time offset. In a different sense, the time interval [0, t] has an absolute frame reference feature. The real value, continuous time process $Y(t), t \in R$ is H-ss (H-self similar with Hurst parameter), if the following holds:

$$Y(t) =_d a^H Y(t), \forall a > 0, t \geq 0 \tag{2}$$

and (0<H<1). This relation means that processes $Y(t), t \in R$ and $Y(at), t \in R$ are the same in the sense of finite dimensional distribution except scaling. In traffic modeling $Y(t)$ represents the quantity of data transferred in time interval [0,t]. The time series generated form X_t by disjoint blocks of length m is named $X^{(m)}$, aggregated process of order m, if

$$X^{(m)} = \frac{1}{m} \sum_{t=m(i-1)+1}^{mi} X_t, \forall t \in Z \tag{3}$$

holds [7, 8]. The second order self-similarity implies the existence of a particular correlated and asymptotically correlated structure during time aggregation. If process $Y(t), t \in R$ has stationary increments (H-sssi), then it is H-ss with stationary increment. If $Y(t)$ is H-sssi with finite variance, then 0<H<1 [6]. Process $X_t, t \in \mathrm{Z}$ is H-self similar and stationary (H-sss) is called SRD (Short Range Dependent) if 0<H<1/2 and is LRD (Long Range Dependent) if 1/2<H<1. The autocorrelation function $r_X(k), k \geq 1$ of second order self-similar process $X_t, t \in \mathrm{Z}$ with Hurst parameter 0<H<1, H≠1/2 can be estimated by following relation:

$$r_X(k) \sim H(2H-1)k^{2H-2}, k \to \infty. \tag{4}$$

The wavelets based Multi Resolution Analysis (MRA) proposes fast pyramidal algorithm requiring ~O(n) computation steps for determining the self-similarity measure of the traces with n samples, consisting serious interest in the low delay aspect of the burst detection [6]. Decomposition

$$\{T_Y(a,t) = \langle Y, \psi_{a,t} \rangle, a \in R^+, t \in R\} \tag{5}$$

is called CWT (Continuous Wavelet Transform), where the coefficients are given by the inner products of analyzing function set and signal $Y(t), t \in R$ [6]. The set of functions $\psi_{a,t}$ is generated from the ψ_0 reference pattern, called mother wavelet, in the following manner:

$$\{\psi_{a,t}(u) = \frac{1}{\sqrt{a}}\psi_0\left(\frac{u-t}{a}\right), a \in R^+, t \in R\}. \tag{6}$$

Any function element $\psi_{a,t}(u)$ can be constructed from ψ_0 mother wavelet by operator D_a -dilatation (scaling) and then T_t -time shifting:

$$\left.\begin{array}{l}(T_\tau \psi_0) \equiv \psi_0(t-\tau) \\ (D_a \psi_0) = 1/\sqrt{a}\,\psi_0(t/a)\end{array}\right\}. \tag{7}$$

The ψ_0 mother wavelet (e.g. DaubechiesN, CoifletN, HaarN, SymmletN, etc.) spread in a narrow space in time and frequency domains and has most of its energy within a limited frequency (scale) band [6]. It satisfies the admissibility condition:

$$\int u^k \psi_0(u)\,du = 0, k = 0,1,\ldots, N-1, \tag{8}$$

that is it should be a bandpass or oscillating function, hence the name "wavelet". Quantity $|T_Y(a,t)|^2$ is called scalogram and gives the energy level of signal Y controlled by dilatation parameter a around time moment t.

The MRA (Multi Resolution Analysis) proves that there exists a special sampling of the time-frequency plane, which contains all the information of signal Y as a subset of $\{T_Y(a,t) = \langle Y, \psi_{a,t} \rangle, a \in R^+, t \in R\}$. The construction method of this discrete subset is called DWT (Discrete Wavelet Transform). The basic idea of MRA is that signal Y is analyzed sequentially with rougher and rougher approximations. In this way, higher and higher frequency components are being left out from consideration. Detailed coefficients $d_Y(j,k)$ of the wavelet transformation are samples of set $\{T_Y(a,t)\}$ that place these elements in dyadic grid points:

$$d_Y(j,k) = T_Y(2^j, 2^j t), \tag{9}$$

where the base 2 logarithm of scale $\log_2(a = 2^j) = j$, is called octave$_j$. The ψ_0, mother wavelet utilized for analyzing second order processes decays at least exponentially in the time space, ensuring in this way second order statistical characteristics of the wavelet decomposition.

Wavelet coefficients $d_Y(j,k)$ of the process $Y(t), t \in R$, with H-ss feature demonstrate self-similarity. For second order processes, this aspect implies the following relation:

$$E[d_Y(j,k)^2] = 2^{j(2H+1)} E[d_Y(0,k)^2]. \tag{0.10}$$

If process $Y(t), t \in R$ is H-sssi, then the process of fixed scale index wavelet coefficients $\{d_Y(j,k), k \in \mathrm{Z}\}$ is a stationary process. In this way for $\forall k \in \mathrm{Z}$ relation (10) simplifies [6]:

$$E[d_Y(j,k)^2] = 2^{j(2H+1)} C(H, \psi_0) \sigma^2, \tag{11}$$

where

$$C(H, \psi_0) = \int |t|^{2H} \left(\int \psi_0(u) \psi_0(u-t) du \right) dt, \tag{12}$$

and $\sigma^2 = E[Y(1)^2]$. This relation of the wavelet coefficients does not assume the Gauss distribution, and can be utilized in the analysis of scale and frequency dependence of LRD processes [10]. In the wavelet domain, we only operate with $d_Y(j,k)$ belonging to octave$_j$ and we manipulate SRD processes during the

transformation. Based on (8) the wavelet coefficients of mother wavelet ψ_0 all average at zero. The stationary character makes averaging possible to process $d_Y(j,k)$, decreasing the variance in this way. The average statistics have low variance because of the SRD effect.

The random variable μ_j obtained from $d_Y(j,k)$, wavelet coefficients of second order stationary process Y, is called energy function of process $d_Y(j,k)$, which is the estimation of the standard deviation, too:

$$Energy_j = \mu_j = \frac{1}{n_j}\sum_{k=1}^{n_j}\left|d_Y(j,k)\right|^2 \tag{13}$$

where n_j, is the number of existing coefficients of octave$_j$ ($n_j = \lfloor 2^{-j} n \rfloor$) and n is the number of elements of signal Y. The analysis of second order dependence of signal Y is reduced in this way to the analysis of dependence of the energy μ_j of octave$_j$. The scale exponent can be obtained from the bias of the $\log_2(\mu_j)$ curve represented in function of octave$_j$.

$$y_j = \log_2(\mu_j) = \log_2(Energy_j) \approx (2H-1)\,j + c\,. \tag{14}$$

The plot of quantity y_j against octave$_j$ inside the confidence interval is called second order log-scale diagram (2-LD). A linear segment or segments can be utilized for estimating the Hurst parameter or parameters. If more than one linear segment can be detected, then the process is monofractal, otherwise it is multifractal. Based on the weighted-least-square (WLS) method, the estimated Hurst parameter inside of confidence interval $[j_1, j_2]$ of the octaves is given by:

$$\hat{H}(j_1, j_2) = \frac{1}{2}\left[\frac{\sum_{j=j_1}^{j_2} S_j j y_j - \sum_{j=j_1}^{j_2} S_j j \sum_{j=j_1}^{j_2} S_j y_j}{\sum_{j=j_1}^{j_2} S_j \sum_{j=j_1}^{j_2} S_j j^2 - \left(\sum_{j=j_1}^{j_2} S_j j\right)^2} + 1\right] \tag{15}$$

where $S_j = \frac{n \ln^2 2}{2^{j+1}}$ is weight. The scale independent time domain for the time series with sampling interval T can be calculated from interval $[j_1, j_2]$ of the octaves with the following formula:

$$[j_1, j_2] \rightarrow \left[2^{j_1} T, 2^{j_2} T\right]. \tag{16}$$

3 Transformation ON/(ON+OFF)

We developed our own analysis method for two simultaneous time series of the network traffic: frame sizes and interarrival times. Using the methods described above, we evaluate the behavior of packet switched network traffic more deeply.

The received frame size (in number of bytes) can easily be transferred to a time space. The transmission delay of impulses increases linearly with the physical size of the medium, which is ~1μs for each 300 meters [2]. Because our method analyses the time series of frames only at the receiving point of the channel, we do not deal with the effect of the finite spread character of electromagnetic signals. The time interval between the ending moment of actual frame reception and starting moment of the next frame reception is not only dependent on the waiting traffic, but also on the MAC (Medium Access Control) mechanism, too.

On the channel with Bw[bit/sec] transfer rate, the actual frame transmission time interval is called $L=T_{ON}$, and the inactivity time caused by an absence of frame and other technology dependent timing (MAC algorithm, etc.) is called T_{OFF}. There are (OFF, ON) states pairs repeating during the sampling time period.

The number of frames captured during the actual sampling (evaluation) time period T is called receiving rate (M) or channel intensity:

$$T = \sum_{j=1}^{M} (T_{OFF,j} + L_j) = \sum_{j=1}^{M} T_{OFF,j} + M \cdot L, \tag{17}$$

where L_j=1,2,…,M is the transmission time of frame, and L is the actual average transmission time interval.

We sample at equidistant time intervals (T=constant). In contrast with time series analysis where events during time period T are supposed to be very short, in the case of the ON/(ON+OFF) method these events are considered to have non-zero length and be uniformly distributed event series during time interval T [1, 2]. We take that in the k-th sampling period, in uniformly distributed time period L_k, average length frames are received with $M_k \in Z^+$ intensity (Fig. 1). There is a clear relation among these variables:

$$\left.\begin{aligned} T_k &= T_{OFF,k} + L_k \\ T &= M_k T_k \end{aligned}\right\}, k = 1,2,..., \lfloor t/T \rfloor. \tag{18}$$

We introduce the terms channel phase angle (φ_k), and complex traffic (Z_k), during the k-th sampling period by following ways, respectively:

$$\left.\begin{aligned} \varphi_k &= \tan^{-1}\left(\frac{L_k}{T_k}\right)_k \\ Z_k &= M_k + i \cdot \tan \varphi_k \end{aligned}\right\}, k = 1,2,..., \lfloor t/T \rfloor \tag{19}$$

where i is the complex unity root $(i^2 = -1)$. The average channel load during the k-th sampling period is given by:

$$Average\ \ channel\ \ load_k = \frac{L_k}{T_k} = \tan(\varphi_k)\,. \tag{20}$$

We generate one time series pair with transformation ON/(ON+OFF) (Fig. 1):

$$\left.\begin{array}{l}\{\tan\ \varphi_k, k = 1,2,..., \lfloor t/T \rfloor\}, Channel \quad Load \\ \{M_k \in \mathrm{Z}, k = 1,2,..., \lfloor t/T \rfloor\}, Channel \quad Int\ ensity\end{array}\right\}. \tag{21}$$

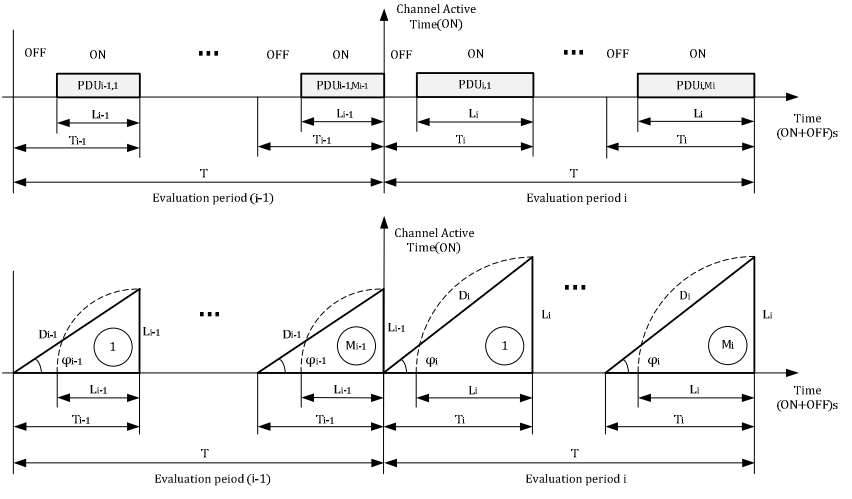


Fig. 1. Parameters of the transformation ON/(ON+OFF).

These series are well interpretable measures with physical meaning, and based on relations (19), the complex time series $\{Z_k, k = 1,2,..., \lfloor t/T \rfloor\}$ is capable of describing all characteristics of traffics transferred on the point-to-point channel. For special cases, the intensity time series $\{M_k \in \mathrm{Z}, k = 1,2,..., \lfloor t/T \rfloor\}$ can be coupled with channel phase time series $\{\varphi_k \in \mathrm{Z}, k = 1,2,..., \lfloor t/T \rfloor\}$. In this way we get another complex series:

$$\{W_k = M_k e^{i\varphi_k}, k = 1,2,..., \lfloor t/T \rfloor\}, \text{ where } i^2 = -1\,. \tag{0.22}$$

The physical parameters of the channel (transfer rate, MAC algorithm, etc.) and the sampling period T are well known parameters and the complex time series Z_k, or W_k can be used to calculate any conventional characteristics (interarrival time, received

bytes, etc.). The complex time series $\{W_k, k = 1,2,...,\lfloor t/T \rfloor\}$ occupy a closed area of the first 1/8 sector of the (M, φ) complex plane and touch the axes only in the origin.

The J_k amplitude of the Z_k complex process is called squared average intensity of the channel, and the phase of the Z_k is called specific channel phase (Fig. 2):

$$\left.\begin{array}{c} Z_k = J_k e^{i\theta_k} \\ J_k = \sqrt{M_k^2 + (\tan\varphi_k)^2} = \dfrac{M_k}{\cos\varphi_k} \\ \tan\theta_k = \dfrac{\tan\varphi_k}{M_k} \end{array}\right\}, k = 1,2,..., \lfloor t/T \rfloor. \tag{23}$$

The tangent of the specific channel phase, $\tan\theta_k$ is the specific channel load, namely average active time interval on unit of time. The phase and the intensity can be understood as increment processes given by the following formula:

$$\left.\begin{array}{c} \varphi_k = (\Phi_k - \Phi_{k-1}) - 2\pi, \Phi_0 = 0 \\ M_k = F_k - F_{k-1}, F_0 = 0 \end{array}\right\}, k = 1,2,..., \lfloor t/T \rfloor, \tag{0.24}$$

where $\left\{\Phi_k = \sum_{j=1}^{k} (\varphi_j + 2\pi), k = 1,2,...\right\}$ is strictly increasing series, called global channel phase, and $\left\{F_k = \sum_{j=1}^{k} M_k, M_k \in Z, k = 1,2,...\right\}$ increasing series, called arrived PDU number.

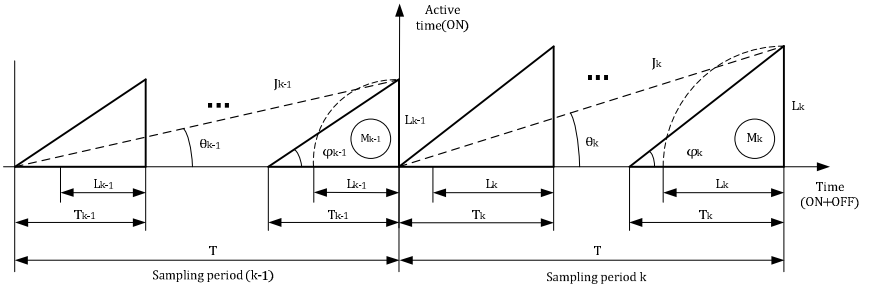


Fig. 2. Squared average intensity J_k and specific phase θ_k of the channel.

4 Characteristics of Transformation ON/(ON+OFF)

For evaluating the characteristics of transformation ON/(ON+OFF) we analyzed three data traces: MAWI Wide-Transit (F200903311500[1]) data traffic from 2009 and other two captured streams at the University of Debrecen in 2008 and 2007, during time intervals of 30 seconds, 3600 seconds and 60 seconds, respectively [11]. Basic characteristics of these traces are presented in Table 1.

Table 1. Basic characteristics of the analyzed traffics

Name of the stream	**F200903311500**	**VoIP-Trunk**	**VBR-video-congested**
Sampling place	Samplepoint-F, WIDE-TRANSIT link, Tokyo, Japan (MAWI)	University of Debrecen, Debrecen, Hungary	University of Debrecen, Debrecen, Hungary
Sampling date	2009.03.31. 15:00:00-15:00:30	26-Mar-2008 10:00-11:00	2-Apr-2007 20:01-20:02
No. of sampled frames	622.944	739,829	6,186
Type of data stream	IP in WAN network (94% TCP, 6% UDP, 73% HTTP)	VoIP trunk (100% IP voice, 0% data)	TCP+UDP (TCP: data, UDP: video)
Type of the channel	IEEE 802.3, 1000 Mbps	IEEE 802.3, 100 Mbps	IEEE 802.3, 1 Mbps
Type of QoS	"Best-effort", no congestion	"Best-effort", no congestion	QoS, congested channel UDP (DSCP=56) TCP (DSCP=0)
Sampling period (T) [ms]	1	100	100
Sampling accuracy (τ) [μs]	10	10	10
Average of channel intensity M []	20.76	20.54	9.51
Variation of channel intensity M []	10.72	9.21	2.01
Average of channel load tan(φ) [%]	12.31	0.0025	53.94
Variation of channel load tan(φ) [%]	10.43	0.0013	5.51

[1] http://mawi.wide.ad.jp/mawi/samplepoint-F/20090330/200903311500.html

These traces were selected in such way to have comparison possibility of Best-effort and QoS data traffics with and without congestion. First two traffics were running in Best-effort network conditions. The VoIP trunk traffic analyzed secondly was captured in congestionless environment. The aggregated voice traffic of IP/PBX gateway generated by a population of 1500 IP phones. The voice trunk link was 100 Mbps Ethernet and the capturing task was effectuated with $\tau = 10\ \mu$ sec accuracy in university environment on a working day for a one hour time interval. We did not use special QoS mechanism because the trunk link load was less than 1%. The aggregated voice streams were transmitted with DSCP=0. In the third case only TCP traffic was transferred with Best-effort (DSCP=0) mechanism in the QoS domain. The packet of the UDP based video conference fits the Ethernet frame and in this way the priority of the IP packets transmitting UDP video segments could be increased by setting DSCP=56 for these packets.

Utilizing transformation ON/(ON+OFF) we generated Z_k complex process for each trace, we analyzed its real and imaginary parts and we determined their basic characteristics. The components of Z_k complex process generated with ON/(ON+OFF) are expressive quantities with suggestive physical meanings. Both components generated with the ON/(ON+OFF) method are necessary for network traffic analysis, because both time series contain information for describing the physical process of content transmission.

The transformation ON/(ON+OFF) is useful to analyze not only Best-effort processes, but other QoS driven processes, as well. Based on wavelet transformation is able to estimate Hurst parameter using relatively short traces. This aspect is advantageous in practice, where detection of fractal characteristics of the actual traffic should be estimated asymptotically in real time.

We analyzed the scale dependence of the real and imaginary LRD parts of Z_k complex series with wavelet method, and we estimated the Hurst parameters, based on the energy plots (Fig. 3, Fig. 4). In Best-effort with low channel load cases strong correlation exists between the channel intensity and channel load, the scalograms are similar, but the Hurst parameters are slowly different (Table 2).

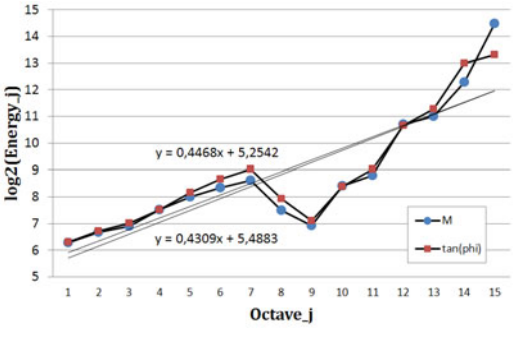


Fig. 3. Wavelet energy functions, F200903311500 trace.

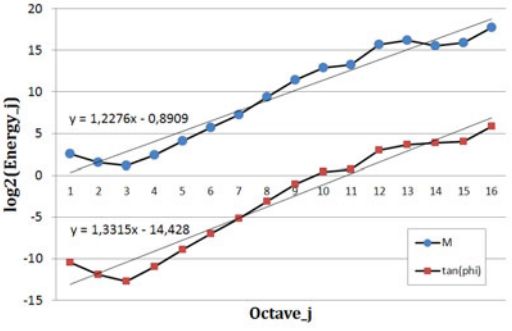


Fig. 4. Wavelet energy functions, VoIP-Trunk trace.

The time series component transmitting higher energy shows less LRD. The explanation of this phenomenon is that 94% of MAWI WAN traffic was generated by TCP-based HTTP, SMTP and FTP protocols in IP packets on a considerable loaded Ethernet channel with Best-effort (without QoS) method, and an important part (73%) of the channel traffic was provided by HTTP messages. In case of congested channel with QoS mechanism, the Hurst parameters are very different.

Table 2. Estimated Hurst parameters with wavelet method (H)

	F200903311500	**VoIP-Trunk**	**VBR-Video-Congested**
Channel intensity, M_k	0.736	0.618	0.608
Channel load, $\tan(\varphi_k)$	0.738	0.643	0.864

If the basic traffic process in QoS free network environment is LRD, then the real and imaginary parts of Z_k complex process generated with ON/(ON+OFF) transformation are LRD and there exists a strong correlation between them. The histogram of the channel load produces clustering, meaning that some load values appear with very low frequency. This effect is produced by the bursty character of the packet flow. The log diagram of the channel intensity histogram indicates no Gauss distribution, nor for the other two time series. The complex time series Z_k is situated inside of compact region, determined by lines (d_1) and (d_2):

$$\left.\begin{aligned} \frac{\tan\ \varphi_k}{M_k} &\le \frac{a}{Max\ \{M_k\}}, \quad if\ M_k \in [0, Max\ \{M_k\}] \\ \frac{\tan\ \varphi_k}{M_k} &\ge 0, \quad if\ M_k \in (0, Max\ \{M_k\}] \\ \frac{\tan\ \varphi_k}{M_k} &\ge \frac{a}{Max\ \{M_k\}}\left(1 - \frac{Max\ \{M_k\}}{2\,M_k}\right), \quad if\ M_k \in \left(\frac{Max\ \{M_k\}}{2}, Max\ \{M_k\}\right] \end{aligned}\right\}. \tag{25}$$

Based on relations (23) and (25) the rate of the relative load and the intensity of the channel ($\tan\theta_k$, specific channel load) is limited at the both sides:

$$\left.\begin{array}{c} 0 \le \tan\ \theta_k = \dfrac{\tan\ \varphi_k}{M_k}, \quad and \\ a\left(\dfrac{1}{Max\ \{M_k\}} - \dfrac{1}{2M_k}\right) \le \tan\ \theta_k \le a\ \dfrac{1}{Max\ \{M_k\}} \end{array}\right\}. \tag{26}$$

Parameter a is independent of the time and is a characteristic of the trace. For traces F200903311500, VoIP-Trunk, VBR-Video-Congested parameter a is 1.082, 1.354, and 2.521 respectively.

The aggregated traffic of several voice sessions running on the same Ethernet voice trunk channel without congestion are LRD processes and have estimated Hurst parameters of channel load and channel intensity very closed $\left(\hat{H}_{\tan(\varphi)} = 0.64, \hat{H}_M = 0.62\right)$. This phenomenon origins from the constant size of voice frames generated by the codecs implying approximately linear relation between channel load and channel intensity parameters. These values are appropriate but different because there were running more than one voice codec (G.711, G.729, etc) simultaneously between the IP phones and the voice gateway.

Diffserv QoS mechanism in service provider network radically influences the traffic behavior of classical network applications. These changes can be detected in the third traffic trace and can be measured with the complex time series generated by the ON/(ON+OFF) method. The Hurst parameters of the channel intensity and channel load for different QoS conditions are uncorrelated. Mixed Best-Effort TCP traffic with Diffserv based QoS controlled UDP traffic is LRD at the channel intensity time series, and the stationary character of the channel load is cancelled out for both transport layer protocols. The VBR streaming video traffic controlled by the Diffserv QoS mechanism produces more smooth traffic in congested environment. The congested Best-effort based TCP traffic is able to use only a part of the remaining network resources. If the video stream is not treated by QoS and is congested, the video service becomes inoperable. The explanation of this phenomenon is that remarkable part of Bellcore LAN/WAN traffic was generated by TCP-based SMTP and FTP protocols in short packets on lightly loaded 10 Mbps Ethernet channel with Best-effort (without QoS) method, and an important part of the channel was used by control signaling.

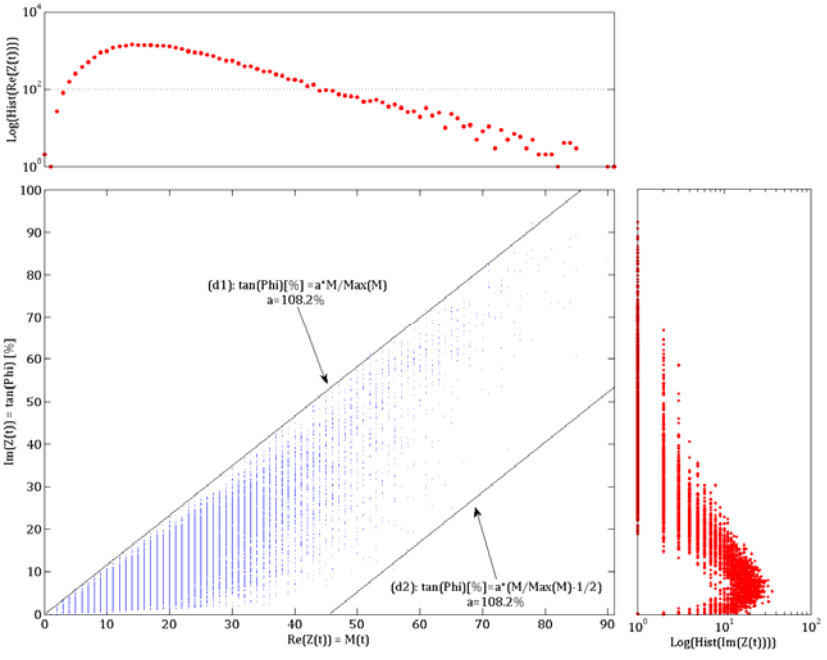


Fig. 5. Histograms of the real (M-channel intensity) and imaginary (tan(φ)-channel load) parts, F200903311500 trace.

The energy functions M and tan(φ) of a given trace determined by wavelet method are different and the linear interval location differs, as well. Similar types of traffic (e.q. voice trunk) traversing channel without congestion implies similar but different energy plots.

5 Conclusions

Both components generated with the ON/(ON+OFF) method are necessary for analyzing the network traffic, making possible to model Best-effort and QoS controlled processes, as well. This wavelets based method is able to estimate with low computation capacity the Hurst parameter of relatively short traces. This aspect is advantageous in practice, where detection of fractal characteristics of the actual traffic should be estimated asymptotically in real time. Based on the estimated Hurst parameter of the channel intensity and channel load eventual congestion occurrence and significant compound change of the aggregated traffic can be detected in semi-real time.

Acknowledgments. This work is supported by TÁMOP 4.2.1./B-09/1/KONV-2010-0007/IK/IT project. The project is implemented through the New Hungary Development Plan co-financed by the European Social Fund, and the European Regional Development Fund.

References

[1] Xiao, X., Ni, L.M.: Internet QoS: A big picture. IEEE Netw. Mag. 13, 8–18 (1999)
[2] ITU-T Study Group 2, Teletraffic engineering handbook (2006)
[3] Ortiz, L.F.: Solving QoS in VoIP: A formula for explosive growth? Brooktrout Technology (2001)
[4] Rezaul, K.M., Grout, V.: An overview of long-range dependent network traffic engineering and analysis: Characteristics, simulation, modelling and control. In: Proceedings of 2nd International Conference on Performance Evaluation Methodologies and Tools, Nantes, France (2007)
[5] Gal, Z.: VoIP LAN/MAN traffic analysis for NGN QoS management. Infocom J. LXIVL, 22–29 (2009)
[6] Park, K., Willinger, W.: Self-similar network traffic and performance evaluation. John Wiley & Sons, Inc. (2000)
[7] Terdik, G., Gyires, T.: Internet traffic modeling with Lévy flights. IEEE/ACM Trans. Netw. 17, 120–129 (2009)
[8] Terdik, G., Gyires, T.: Does the Internet still demonstrate fractal nature? In: Proceedings of Eighth International Conference on Networks (ICN 2009), Gosier, Guadeloupe, France, pp. 30–34 (2009)
[9] Daubechies, I.: Ten lectures on wavelets. Society for Industrial and Applied Mathematics, Philadelphia (1992)
[10] Jin, X., Min, G.: QoS analysis of queuing systems with self-similar traffic and heavy-tailed packet sizes. In: Proceedings of IEEE International Conference on Communications (ICC 2008), Beijing, China, pp. 100–104 (2008)
[11] Gal, Z., Terdik, G.: Wavelet analysis of QoS based network traffic. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, pp. 275–280 (2011)

Data Cleaning and Anomaly Detection for an Intelligent Greenhouse

Peter Eredics and Tadeusz P. Dobrowiecki

Budapest University of Technology and Economics,
Department of Measurement and Information Systems,
Magyar tudosok krt. 2, H-1117 Budapest, Hungary
{eredics,dobrowiecki}@mit.bme.hu

Abstract. The effectiveness of greenhouse control can be improved by the application of model based intelligent control. Such control requires a good model of the greenhouse. For a large variety of industrial or recreational greenhouses the derivation of an analytical model is not feasible therefore black-box modeling has to be applied. Identification of black-box models requires large amount of data from real greenhouse environments. Measurement errors or missing values are common and must be eliminated to use the collected data efficiently as training samples. Rare weather conditions can temporally lead to unusual thermal behavior around and within the greenhouse. Anomaly detection run on the measurement data can identify such unusual samples and by excluding those from the model building better models and higher validation accuracy can be achieved. This chapter discusses problems of cleaning the measurement data collected in a well instrumented greenhouse, and introduces solutions for various kinds of missing data and anomaly detection problems.

1 The Concept and the Requirements of Intelligent Control

Greenhouses are built in various sizes and types all around the world to house plants needing special environmental conditions. Greenhouses are widely used both for vegetable and ornamental plant production.

The basic operation of a greenhouse is as follows: The transparent walls and roofs allow the solar radiation to pass through, but the warmed up air is kept inside. To prevent extreme high or low temperatures inside the house several actuators can be utilized. Shading curtains, automatic windows and active cooling systems can slow down the temperature rise in the summer while heating appliances are used in the cold season.

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 123–134.
springerlink.com

Most greenhouses are provided with some kind of simple automated control [2]. Such traditional control systems are based on setting operating levels which are decided by the owner, e.g. the owner has to set a window opening temperature limit along with a window closing temperature limit. Such rules have to be created for all different actuators and control situations. Main advantages of this control scheme are its simplicity (yielding high reliability) and simple working logic (the owner of the greenhouse always knows exactly why particular actions are happening). On the other hand traditional control solutions have some major disadvantages:

- The owner has to adjust operating rules relying only on his or her expertise about the facility, and is not supported by the control system to do it optimally or even efficiently.
- The control is reactive. It means that the actuators are operated only after the set limit is reached. Unwanted situations cannot be avoided in advance, even if they could be predicted, or the limits must be set much more conservatively.
- The actuators are not synchronized. All actuators work independently based on their rules yielding suboptimal at the best total operation of the greenhouse.

The concept of an intelligent greenhouse is conceived to overcome these limitations [3]. In place of simple operating rules, the greenhouse owner specifies goals for the control (e.g. in the form of target parameter zones). The system should then build a model of the greenhouse and predict its future states to avoid unwanted circumstances in advance. Using the predictions from the greenhouse model AI planning could be used to create plans for all actuators jointly [4]. This novel approach is expected to help to overcome the limitations present in current greenhouse control systems.

The prospects of intelligent greenhouse control depend strongly on the accuracy of the modeling. Considering that greenhouses come in different sizes and are designed for different purposes, analytical models are usually not applicable (feasible) to this problem [5]. Solely a black box-model might be able to adapt to any greenhouse it is installed in. The main drawback of black-box modeling, however, is the large number of training samples needed to construct and tune the model [6]. The training samples must be derived from the training data recorded as time series characteristic to the evolution of the greenhouse where the whole control system is installed.

The accuracy of greenhouse models depends also on the time and space resolution of the measurements. While measuring e.g. every 5 minutes seems to be acceptable (due to the slow dynamics of the thermal processes in the greenhouse), the usual single location measurement used in traditional control systems is much too limited option. To build high precision models several measurement locations have to be set up at strategically selected locations within the greenhouse.

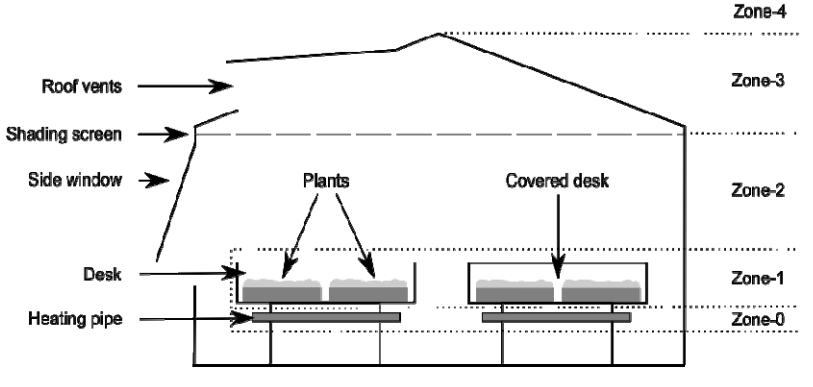
Data recorded in a real environment can be disturbed with several external influences: faulty operation of the data acquisition hardware (missing values or outliers) or rare, local weather phenomena can produce invalid data. The reparation or exclusion of such invalid data is essential to provide a reliable control for the greenhouse, and this is the responsibility of the data cleaning process: data records have to be classified as correct (can be stored without processing), as repairable (the data cleaning process is able to repair it) or as invalid (the record cannot be repaired, and has to be dropped). The problem of data cleaning was dealt with in the conference paper [1]. The main result of the present chapter is the management of the anomalous records. Records after the data cleaning process can all be classified as valid, but their usefulness for the model building can vary. High quality data records can speed up the model building and validation while low quality records should be omitted from the modeling process. The quality of data is determined by the anomaly detection process.

The structure of the chapter is as follows. Section 2 reviews the problem of the black-box modeling of the greenhouse. Section 3 provides information about the data acquisition system and the character of the measured data. Sections 4, 5, and 6 treat the problem of repairing the data coming from various sources. Section 7 introduces the problem of anomaly detection, presents interested cases and the proposed solution. Finally the experience is summarized and further research delineated.

2 Modeling the Experimental Greenhouse

A measurement and traditional control systems have been installed in a 100 m^2 greenhouse to collect real world measurement data. The greenhouse has 18 desks holding most of the time very young and sensitive ornamental plants.

The measurement system records temperatures from all desks and also from thermally quasi-homogeneous larger parts of the greenhouse called zones. Fig. 1



Fig, 1. Simplified thermal zone structure of an industrial greenhouse.

shows the zone partitioning of the greenhouse: Zone-0 is the heating pipe; Zone-1 contains the desks (some covered with foil for humidity protection); Zone-2 means the interior air under the shading screen; Zone-3 means the air above the shading screen while Zone-4 represents the environment immediately outside the house. Temperature data is collected every 5 minutes from all zones with 0.5 degree accuracy. In addition online weather data for the region of the greenhouse is read with hourly resolution. Regional weather data and forecasts are also recorded.

3 The Data Records

The data acquisition system is already installed and running since early 2008. Since then 297037 records were stored in the central database representing 23250 hours of measurement. Unfortunately the recording was not without breaks (the system was sometimes turned off for maintenance and power outages occurred also), but this amount of rough data is what is available for the modeling, nothing else. The structure of the records is presented in Table 1. The recorded attributes are grouped into physical quantities (global zone measurements; desk measurements; data from online sources) and actuator states. These different groups of collected data might contain different types of errors and consequently call for different handling. The system also records regional temperature forecasts from an online source for 6 hours ahead.

4 Processing Physical Quantities

Records of physical quantities can contain three kinds of invalid values. The data acquisition system might enter an error code instead of the valid data (in case of errors at the lower system levels, e.g. error in the communication with the sensors). These error codes are represented by numbers out of measurement range

Table 1. The structure of output data records of the greenhouse data acquisition system

Field Name	*Zone*	*Unit*
Heating pipe temp.	0	C
Desk 1 temperature	1	C
…		
Desk 18 temperature		
Under shading temp. 1	2	C
Under shading temp. 2		
Under shading radiation		%
Above shading temp.	3	C
External local temp.	4	C
External local radiation		%
External online temp.		C
Heating state	-	code
Shading state	-	
Upper windows state	-	
Side windows state	-	
Misting usage time	-	sec

for a given sensor, thus they can be processed together with the out of range measurements caused sometimes by sensor malfunctioning. Both cases can be differentiated by checking the recorded values against the upper and lower bounds of the operating range of the given sensor.

Some physical processes, driving particular quantities can exhibit large time constants, thus the change rate of these values will be limited. Comparing the actual value to the one read in the last recording can indicate such measurement errors with extreme high dynamics. Unfortunately identifying problematic values in the data alone is not enough to support the modeling process. Most black-box methods used for model building cannot handle missing input values and such attributes must be repaired in some way.

Copy method: Due to the slow process dynamics copying the previous valid recording (if it is not older than 10 minutes) is an acceptable solution in case of most physical quantities. If the last available record is too old, time-series prediction methods might be considered, but in case of many values missing this would be too resource intensive. Taking into account the high computational needs of the intelligent control such expensive data restoration methods cannot be afforded, and in this rare case it is better to drop the whole record.

Regression method: There is a specific physical parameter, namely the regional temperature value recorded from the internet that can be handled differently. This value is special in two main aspects: it is updated only once in an hour (thus copying the former value is acceptable in an hour range) and it contains somewhat redundant information as its value cannot be independent from the local temperature measurements. Considering it a simple regression model can be built based on local measurements to approximate the regional temperature value in case of internet black-out. This model uses the local temperature and radiation measurements as regressor values. A simple model of (1) was chosen for approximation as its accuracy is close to the sensitivity of the approximated data.

$$\tilde{T}_{regional} = a_0 + a_1 * T_{local} + a_2 * R_{local} + a_3 * R_{local}^2 \quad (1)$$

In (1) $T_{regional}$ is the approximated regional temperature value, T_{local} is the local temperature measurement while R_{local} is the local radiation data. Higher order regression models were also tested but their accuracy did not prove to be significantly better. The weighting parameters a_i must be calculated periodically to follow the changes in the relationship between the two measurement locations (i.e. the weather station and the greenhouse), but this calculation (using least squares method) is necessary only once a week to keep the approximation accurate.

Spatial interpolation: Temperature values recorded at the desks holding the plants call for a special handling too. Desks close to each other in position have closely related temperatures, thus restoring a missing desk value can be accomplished by a spatial interpolation. Unfortunately some of the desks used to be covered with transparent foil to keep the humidity high while other desks are left uncovered based on the plants' actual needs. This fact makes it impossible to simply use the data from the closest desks for the reconstruction process. The

desks for the interpolation must be selected dynamically for every restoring operation. On the other hand the state of the desks (i.e. foil covering) does not change frequently therefore these relationships can be cached and calculated only once a day.

In the current implementation the desks are sorted by the similarity (calculated based on the mean square difference of temperature values measured on the desks) of the previous 3 days. The weighted average of the 3 most similar desks is used as a replacement. The weighting factor is the normalized similarity value and the factors are daily recalculated. This method makes the system able to operate with some permanently malfunctioning desk sensor, which is very important considering the increasing unreliability of the large number of desk sensor used in the greenhouse.

5 Processing Actuator States

Two kind of actuator states are recorded in the given greenhouse: some actuators are characterized by their states (such as the windows with open/half-open/closed) while other actuators by the operation times (e.g. the operating time of the misting system) associated with them. The first group can be considered as a simpler case of the physical quantities as only error codes and out of the range state codes have to be identified and replaced. In the current implementation the only replace procedure available here is the copy method – later on with the greenhouse model available the system might be able to approximate the unknown state of the actuators based on the thermal behavior of the house.

For the second group of state descriptors, the only possible way to replace the missing data is to assume that the actuator was turned on, as it is the case most of the time.

6 Processing Regional Temperature Forecasts

The regional temperature forecasts are obtained from a free online source. Sometimes this source becomes inaccessible (because of site maintenance or other reasons) resulting in missing predictions. As local measurement data is more likely to be available in the system, the regional forecast restoration process can rely on them. A time-series mining method is implemented to produce local weather forecasts. Thus (1) can be used (with the weight factors recalculated if necessary) to present the missing regional prediction from this local forecast. Of course if there is a former regional forecast at hand it might be used in the first few hours (e.g. a 4 hours old, 6 hours long forecast has yet valid data for the first 2 hours, thus only the last 4 hours must be calculated).

7 Anomaly Detection

Anomaly detection is the last step of the data preparation process. Data cleaning methods introduced so far were intended to make a crisp decision about the quality of data (whether it is trustworthy of not), and replace the attributes qualified as unacceptable in one way or the other. This way data cleaning ensures a constant flow of acceptable data for the higher levels of the system, but lacks the ability to classify this data any further. In the case of the intelligent greenhouse the concept of anomaly detection refers to the process of examining the output data of the data cleaning process for its further usefulness.

Data quality measures the usefulness of data in the greenhouse modeling process. The data is high quality if it represents well the natural thermal dynamics of the greenhouse and is suitable to run modeling methods efficiently. Data quality will be low, if the data comes from some exceptional phenomena of the greenhouse or if it is disturbed with notable measurement noise. Running model building algorithms on high quality data can be much more efficient, as the low quality records will only introduce additional noise components into the learning process.

Data recorded in the greenhouse can be low quality for many reasons. A malfunctioning actuator can generate unwanted effects inside the house, e.g. if the windows would be stacked closed the whole day, the internal thermal processes would develop differently than on normal days. The weather is also strongly affecting the greenhouse, and rapid weather changes or fluctuations in the solar radiation can have unexpected effects inside the greenhouse. The data quality can be transformed in practice into the qualification as ordinary and extraordinary situations. Ordinary situations are good examples of the normal life of the greenhouse, while extraordinary situations are rarely occurring special events.

With this practical consideration about the data quality the test results of the greenhouse control can be analyzed with more insight as well. The greenhouse control can execute bad decisions and produce suboptimal operation in case of extraordinary circumstances, as these situations are rare and preparing for them is hard. On the other hand the greenhouse control, trained on high quality data, intended to cover the majority of ordinary situations well, has to be able to produce good control performance in case of ordinary circumstances.

7.1 Extending Data Records

The goal of anomaly detection is to identify data records with unusual internal structure (the attributes are messed up by some external disturbance) or unusual dynamics (the thermal processes of the greenhouse evolve abnormally). The first case requires a detailed analysis of the interrelationship between the attributes, while the second case can be handled by using previous data records to identify

tendencies. The method of extending data records detailed here is aimed to gather all necessary attributes into a single extended data record to be analyzed. The data record extender is a function with all temperature and radiation and control inputs listed in Table 1. This means 32 real value inputs along with some administrational data, such as record IDs and timestamps.

Different data types are processed in different ways. Each temperature record is extended into 2+24 values. The first value is the mean of the attribute over the last 2 measurements while the second value is the difference of the attribute and its last measured value. The next 24 values are generated as the difference of the given value and all other 24 temperature measurements of the system.

Each radiation input is extended to 2 values. The first value is the mean of the last 2 measurements while the second value is the difference exactly as in case of temperature data. The low number of radiation measurement (only 2) and the strong effect of the shading screen control onto their relationship makes it hard to express their connection in any simple way, thus radiation relations are not handled by this method.

The actuator control signals are handled universally as if all actuators would have 3 possible states. This assumption is true for the upper windows and the shading screen, while the heating and the side windows use only 2 of the 3 available states. For each actuator the time distribution among the states is calculated based on the last 12 measurement records (60 minutes). Along with the state distributions the number of state changes is also given for each actuator separately as the number of actuator commands issued also holds important information about the state of the control system.

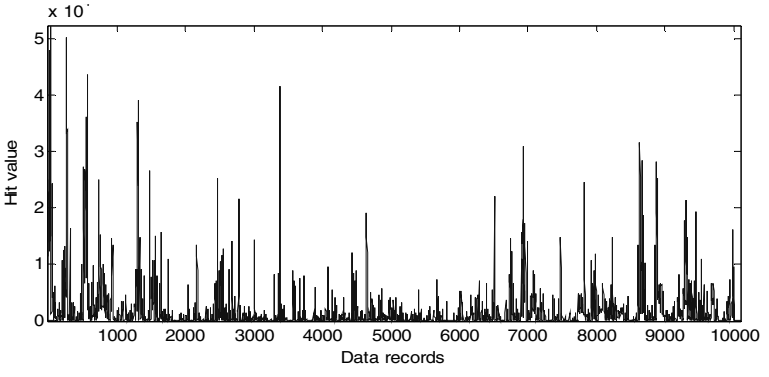
After extending all input parameters, the extended data record contains 644 numeric values. Most of these values represent the inter relation between the different temperature values, and this data is somewhat redundant, but the extended data record describes both the dynamics of the system and the relationships of the attributes together in a simple form.

7.2 Detecting Anomalies in Cleaned Data

A simple model is built from the extended data records at the beginning of the anomaly detection operation, and this model is updated with every new data record analyzed. The model consists of 24 averaged data records for each hour of the day, and 24 data records representing the standard deviations of each element in the model at the given hour. The first 5000 data records (approx. 17 days of measurement) are used to build this model and to calculate the standard deviations. Other data records can be tested against the model. After processing the new records the model is iteratively updated, taking into account the new records with the 0.01 discount factor, and standard deviation values are also updated. This way

the system model can keep up with the structural changes of the greenhouse, while its structure is kept simple.

After building the initial system model data records can be run through the anomaly detection. As the first step the data record has to be extended using the data record extending function. The second step is to select the appropriate hour from the system model based on the measurement time of the input record. After that the two extended data records (the one from the system model, built from several previous averaged measurements; and the one generated from the input record) can be compared attribute by attribute. Every time when the two records differ on any value more than 2 times the standard deviation, the input record gets 1 hit point. These hit points are summed after calculating all differences, and the sum is squared, producing a relative measure of how extraordinary the given input record is, called a hit value. The higher the hit value is, the more unusual the given data record is. Fig. 2 shows an example of this measure on tests run on 10 000 data records from the summer of 2008.



Fig, 2. Anomaly detection run on 10000 data records from the summer of 2008.

7.3 Anomalies Detected

The anomaly detection method produces a hit value to all input data records. The majority of data records have a uniformly low hit value as most of the time the thermal system of the greenhouse is operating normally. The hit value function shows occasional spikes, and this is where the unusual situations take place. After careful investigation of the spikes the following three types of anomalies can be detected in the data records.

At sample 3372 in Fig. 2 the sensors placed on desks 11-13 were malfunctioning, and produced a temperature drop of 5 degrees. The next measurement was correct, and this error was not corrected by the data cleaning system, as similar temperature changes can happen under normal circumstances. The anomaly

detection gave a high hit value to this data record as several temperature values were behaving strange compared to their previous values and to the other measurements around them. Identifying such temporal and rare sensor malfunctioning makes it possible to eliminate such measurements from the model building and validating process yielding higher precision thermal models for the greenhouse.

Around sample 6923 several spikes can be seen in Fig. 2. These measurements were made on 07-07-2008, a summer day with a high cloud activity. The radiation measurements revealed that the solar radiation was rapidly changing during the day. This lead to unusual thermal oscillations inside the greenhouse and this special situation (caused indirectly by the clouds) was indicated by the anomaly detection. Such days should be eliminated from model building and testing as the cloud coverage prediction is out of scope for any complexity of greenhouse models, thus these changing circumstances are impossible to forecast.

Sample 8629 was recorded on 14-07-2008 at 11:01 am. This was the time, when a strong storm reached the environment of the greenhouse causing a sudden end to the usual morning warming process and dropping temperature with more than 5 degrees for several hours. At the middle of the day the parameters of the greenhouse were changing just if it was late evening, and this disturbance was indicated by the anomaly detection. Measurement data from such special weather phenomena are very useful for testing control systems in extreme weather conditions when quick reaction and safe operation has the priority.

Many other spikes were also investigated in Fig. 2, and a lot of these anomalies could be classified in one of the three events detailed before, namely the malfunctioning sensor, the unpredictable weather or the rare (unusual) weather phenomena. Unfortunately it also has to be noted, that the number of false alarms is significant, but the large measurement database lowers the importance of this problem.

8 Results

The data cleaning system has been implemented in the experimental greenhouse. All measurements (both historical and new measurements are handled the same way) are stored in the raw measurement data table in the central database. The data cleaner application reads this data along with the metadata (holding information about the sensors validity range, the actuators type and legal state count, etc.). If the data record can be repaired then it is processed accordingly in this application. The regional weather data is the only exception, as it is repaired in a separate module shared with the regional forecast related modules. All data vectors are stored in the output data table along with indication of valid measurement values or repaired approximations.

The weather forecasts are recorded in a raw weather forecast data table. The forecast cleaner application looks for raw weather data in every hour. If the

forecast is present it is simply copied to the output table. In case of missing forecast the local weather forecast is needed: due to the 1 hour time resolution of the weather forecasts, hourly averages are calculated for all important attributes. These values are stored in a separate data table, because computing them every time would mean useless overhead for the system. Based on hourly averages the local temperature and radiation forecasts are generated. These forecasts are used by the shared regional temperature approximation module and are stored in the output weather data table.

The data cleaning system was used to process all 297037 data records (containing more than 9.8 million measurements) collected in the greenhouse. The copy method was used 32109 times to restore missing values. The spatial interpolation between desks has been executed 781150 times – this large number is caused by some permanently malfunctioning desk sensors. During spatial interpolation the weighting factors have been recalculated 535 times. The regional temperature approximation module was used 50362 times to restore actual regional temperature and 11412 times to rebuild regional forecast values based on locally generated forecasts. This way the system was able to raise the number of full data vectors from 184809 to 276760 available for the model building (50% gain). The number of multiple step training time series examples (especially important for training and testing the models prediction capabilities) has also been notably increased.

In conventional modeling problems measurement records can be used for modeling right after data cleaning. The case of the greenhouse is special, as rare external influences can drive the physical system into unusual states. In such states the collected measurements represent only the temporal state of the greenhouse well, therefore omitting them from modeling seems beneficial. For this aim after cleaning the data anomaly detection took place. This processing was run to identify rare weather phenomena or special sensor malfunctions and to indicate them in the database. During the anomaly detection process the data records were greatly extended to represent all relationships between the measured values. These extended records were compared than element by element to an hourly model of the system. Data records notably different from the system model got a high hit value indicating that these measurements should be handled with care later in the thermal modeling and validation process.

9 Conclusions

The efficiency of greenhouse control systems can be improved by implementing model based intelligent control methods. Thermal modeling applicable to a whole variety of the greenhouses is only feasible with black-box models because of their ability to adapt to any greenhouse the system is installed in. Black box modeling requires large amount of measurement data from the greenhouse with high resolution both in time and space. A sophisticated measurement and control system

working round the clock and for a long time, might have malfunctioning sensors and any kind of internal errors causing incomplete measurement records. These records have to be corrected before using them with black box modeling techniques.

This chapter introduced various recovery methods for different types of measurement data collected in the greenhouse. Randomly missing attributes were always replaced by their pervious know value. In case of consequently missing attributes either spatial interpolation or regression methods were used. This way the number of useable data records has been increased by 50%. This means that the time needed to collect data before training the black box models can be decreased by 1/3. Data cleaning presented valid data vectors, but did not say anything about the usefulness of these records. That is why the data went through anomaly detection to measure how useful single records are for model building. In this process the data records got a hit value indicating their similarity to the normal operation of the greenhouse. High hit value records must be handled separately, as these high values indicate unusual situations inside the greenhouse.

Acknowledgments. The authors gratefully acknowledge the support of the Hungarian Fund for Scientific Research (OTKA), Grant #73496. This work is supported by the grant TÁMOP - 4.2.2.B-10/1--2010-0009 and is partly supported by National Office of Research and Technology (NKTH), NAP-1-2005-0010 (BelAmI_H) project.

References

[1] Eredics, P., Dobrowiecki, T.P.: Data cleaning for an intelligent greenhouse. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, pp. 293–297 (2011)

[2] Zazueta, F.S., Bucklin, R., Jones, P.H., Smajstrla, A.G.: Basic concepts in environmental computer control of agricultural systems. Technical report, Agricultural and Biological Engineering Dept, Institute of Food and Agricultural Sciences, University of Florida, USA (2008)

[3] Drummond, M., Bresina, J.: Planning for control. In: Proceedings of 5th IEEE International Symposium on Intelligent Control, Philadelphia, PA, USA, pp. 657–662 (1990)

[4] Blasco, X., Martineza, M., Herreroa, J.M., Ramosa, C., Sanchisa, J.: Model-based predictive control of greenhouse climate for reducing energy and water consumption. Comput. Electron Agric. 55, 49–70 (2007)

[5] Bot, G.P.A.: Physical modeling of greenhouse climate. In: Proceedings of IFAC/ISHS Workshop, Matsuyama, Japan, pp. 7–12 (1991)

[6] Cunha, J.B.: Greenhouse climate models: An overview. In: Proceedings of European Federation for Information Technology in Agriculture Conference, Debrecen, Hungary, pp. 823–829 (2003)

Clustering of Interval Data Using Self-Organizing Maps – Application to Meteorological Data

Chantal Hajjar[1,2] and Hani Hamdan[1]

[1] SUPELEC, Department of Signal Processing and Electronic Systems, Plateau de Moulon, rue Joliot-Curie 3, 91192 Gif-sur-Yvette cedex, France
[2] Université Libanaise, Beirut, Lebanon
{Chantal.Hajjar,Hani.Hamdan}@supelec.fr

Abstract. The self-organizing map is a kind of artificial neural network used to map high dimensional data into a low dimensional space. This chapter presents a self-organizing map to do unsupervised clustering for interval data. This map uses an extension of the Euclidian distance to compute the proximity between two vectors of intervals where each neuron represents a cluster. The performance of this approach is then illustrated and discussed while applied to temperature interval data coming from Chinese meteorological stations. The bounds of each interval are the measured minimal and maximal values of the temperature. In the presented experiments, stations of similar climate regions are assigned to the same neuron or to a neighbor neuron on the map.

1 Introduction

In real world applications, data may not be formatted as single values, but are represented by lists, intervals, distributions, etc. This type of data is called symbolic data. Interval data are a kind of symbolic data that typically reflect the variability and uncertainty in the observed measurements. Many data analysis tools have been already extended to handle in a natural way interval data: principal component analysis (see for example [1]), factor analysis [2], regression analysis [3], multilayer perceptron [4], etc. Within the clustering framework, several authors presented clustering algorithms for interval data. Chavent and Lechevallier [5] proposed a dynamic clustering algorithm for interval data where the prototypes are elements of the representation space of objects to classify, that is to say vectors whose components are intervals. In this approach, prototypes are defined by the optimization of an adequacy criterion based on Hausdorff distance [6, 7]. Bock [8]

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 135–146.
springerlink.com

constructed a self-organizing map (SOM) based on the vertex-type distance for visualizing interval data. Hamdan and Govaert developed a theory on mixture model-based clustering for interval data. In this context, they proposed two interval data-based maximum likelihood approaches: the mixture approach [9, 10, 11] and the classification approach [12, 13]. In the mixture approach, a partition of interval data can be directly derived from the interval data-based maximum likelihood estimates of the mixture model parameters by assigning each individual (i.e. a vector of intervals) to the component which provides the greatest conditional probability that this individual arises from it. In the classification approach, a partition is derived by maximizing the interval data-based likelihood over the mixture model parameters and over the identifying labels for mixture component origin of each vector of intervals. Chavent [14] presented an algorithm similar to that presented in [5] by providing the L_∞ Hausdorff distance between two vectors of intervals. De Souza and De Carvalho [15] proposed two dynamic clustering methods for interval data. The first method uses an extension for interval data of the city-block distance. The second method is adaptive and has been proposed in two variants. In the first variant, the adaptive distance has a single component, whereas it has two components in the second variant. De Souza et al. [16] proposed two dynamic clustering methods, based on Mahalanobis distance, for interval data. In both methods, the prototypes are defined by optimizing an adequacy criterion based on an extension for interval data of the Mahalanobis distance. In the first method, the used distance is adaptive and common to all classes, and prototypes are vectors of intervals. In the second method, each class has its own adaptive distance, and the prototype of each class is then composed of an interval vector and an adaptive distance. El Golli et al. [17] proposed an adaptation of the self-organizing map to interval-valued dissimilarity data by implementing the SOM algorithm on interval-valued dissimilarity measures rather than on individuals-variables interval data.

In this chapter, we propose a self-organizing map based on an extension of L_2 distance for interval data. In Section 2, we give a definition of the self-organizing maps and their training algorithms. In Section 3, we present the SOM algorithm for interval data. In Section 4, we show the results of implementation of our approach on real interval data observed in Chinese meteorological stations. Finally, in Section 5, we give our conclusion.

2 Self-Organizing Maps

The Self-Organizing Maps (SOM) are invented by Kohonen [18, 19]. A SOM is a kind of artificial neural network that uses unsupervised learning to map high dimensional data consisting of n vectors $\mathbf{x}_i \in \Re^p (i = 1,\ldots,n)$, into a low dimensional space, usually two, called a map grid. The map consists of K neurons that can be arranged either on a rectangular or on a hexagonal lattice. Fig. 1 represents

a rectangular map grid of 7 rows and 7 columns. To each neuron $k,(k=1,...,K)$ is associated a prototype vector $\mathbf{w}_k$ in the input space $\Re^p$, and a location $\mathbf{r}_k$ in the output space formed by the line number and the column number of the neuron on the map grid. In the following, we present two variants of the SOM training algorithm: the incremental training algorithm and the batch training algorithm.

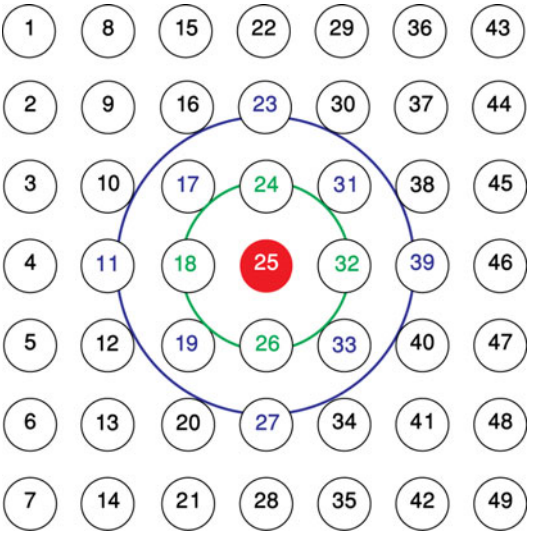


Fig. 1. Square map grid of 49 neurons.

2.1 Incremental Learning Algorithm

In the map training, each training step of the map consists of presenting randomly an input vector **x**, chosen from the data set, to the network and then finding the Best Matching Unit (BMU) of this vector. The BMU is the neuron whose prototype vector is closest to the input vector **x** in term of Euclidian distance. If we denote c the BMU of input vector **x** and $\mathbf{w}_c$ the prototype vector of this BMU, c is defined as the neuron for which:

$$d(\mathbf{x},\mathbf{w}_c) = \underset{k=1,...,K}{min} d(\mathbf{x},\mathbf{w}_k) \tag{1}$$

where d is the Euclidian distance.

After finding the BMU, the prototype vectors of the map are updated toward the input vector using a neighborhood function that determines how much the neurons are close to the BMU (Fig. 2).

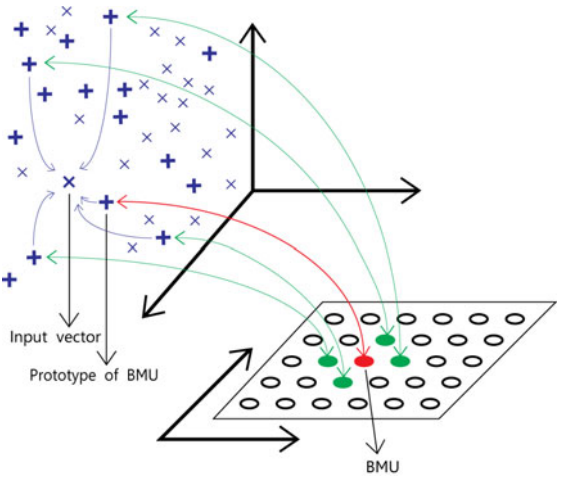


Fig. 2. Mapping the input space to the output space.

Let $h_{ck}(t)$ be the neighborhood function between neuron c and neuron k at time t, and let $N_c(t)$ be the set of neurons that lie within a certain radius around neuron c at time t. This radius, called neighborhood radius, takes large values at the beginning of the training and then decreases with time. In its simplest form, the value of the neighborhood function $h_{ck}(t)$ is 1 if neuron k belongs to the set $N_c(t)$ and 0 otherwise. For example, in Figure 1, if the BMU is neuron 25 and if the neighborhood radius is equal to 1 (inner circle), then the set N_{25} contains the neurons 25, 18, 24, 26 and 32, and only the prototype vectors of these neurons are updated. A more flexible neighborhood function is the Gaussian neighborhood function:

$$h_{ck}(\sigma(t)) = \exp\left(-\frac{d^2(\mathbf{r}_c, \mathbf{r}_k)}{2\sigma^2(t)}\right) = \exp\left(-\frac{\|\mathbf{r}_c - \mathbf{r}_k\|^2}{2\sigma^2(t)}\right), \tag{2}$$

where $\mathbf{r}_c$ and $\mathbf{r}_k$ are respectively the location of neuron c and neuron k on the grid and $\sigma(t)$ is the neighborhood radius at time t. The width of the Gaussian function is defined by $\sigma(t)$. Equation (3) shows the updating of the prototype vectors at iteration $t+1$:

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \alpha(t) h_{ck}(\sigma(t))[\mathbf{x}(t) - \mathbf{w}_k(t)] \ \ (k = 1, \ldots, K), \tag{3}$$

where $\mathbf{x}(t)$ is the input vector, $h_{ck}(\sigma(t))$ is the neighborhood function, c is the BMU of $\mathbf{x}$, $\sigma(t)$ is the neighborhood radius and $\alpha(t)$ is the learning rate. The

learning rate takes large values at the beginning of the training and decreases with time.

2.2 Batch Training Algorithm

In the batch algorithm, the prototype vectors are updated after presenting the whole data set to the map according to Equation (4). Each prototype vector is replaced by a weighted mean over the data vectors. The weights are the neighborhood function values:

$$\mathbf{w}_k(t+1) = \frac{\sum_{i=1}^{n} h_{kc(i)}(\sigma(t))\mathbf{x}_i}{\sum_{i=1}^{n} h_{kc(i)}(\sigma(t))} \quad (k = 1, \ldots, K), \tag{4}$$

where $h_{kc(i)}(\sigma(t))$ is the neighborhood function between the neuron k and the BMU $c(i)$ of the input vector $\mathbf{x}_i$, $\sigma(t)$ is the neighborhood radius and $c(i)$ is the neuron whose prototype vector is closest to $\mathbf{x}_i$ in term of Euclidian distance. All the input vectors associated to the same BMU have the same value of the neighborhood function.

In the last few iterations of the algorithm, when the neighborhood radius tends to zero, the neighborhood function $h_{kc(i)}(\sigma(t))$ will be equal to 1 only if $k = c(i)$ (k is the BMU of input vector $\mathbf{x}_i$) and 0 otherwise. The input data set is then clustered into K classes. The center of each class C_k is the neuron k whose prototype vector $\mathbf{w}_k$ is a mean of the data vectors belonging to that class. This implies that the updating formula of Equation (4) will minimize, at convergence of the algorithm, the L_2 distance clustering criterion:

$$G = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} d(\mathbf{x}_i, \mathbf{w}_k) \cdot \tag{5}$$

In addition, using the values of the neighborhood function as weights in the weighted mean defined in Equation (4) will preserve the topology of the map. We notice that this method is similar to the dynamical clustering method (in this case K-means) with the advantage that clusters that are close to each other are mapped to neighboring neurons on the map grid.

3 Self-Organizing Maps for Interval Data

Let $\mathbf{R} = \{\mathbf{R}_1, ..., \mathbf{R}_n\}$ be a set of n symbolic data objects described by p interval variables. Each object $\mathbf{R}_i$ is represented by a vector of intervals $\mathbf{R}_i = ([a_i^1, b_i^1], \ldots, [a_i^p, b_i^p])^T$, where $[a_i^j, b_i^j] \in I = \{[a,b]; a,b \in \Re, a \leq b\}$.

3.1 L_2 Distance between Two Vectors of Intervals

The distance between two vectors of intervals $\mathbf{R}_i = ([a_i^1, b_i^1], \ldots, [a_i^p, b_i^p])^T$ and $\mathbf{R}_{i'} = ([a_{i'}^1, b_{i'}^1], \ldots, [a_{i'}^p, b_{i'}^p])^T$ is defined by the Minkowski-like distance of type L_2 given by:

$$d = \sum_{j=1}^{p} \left[\left| a_i^j - a_{i'}^j \right|^2 + \left| b_i^j - b_{i'}^j \right|^2 \right]. \tag{6}$$

3.2 The Algorithm

The set R is used to train a self-organizing rectangular map of K neurons. Each neuron k of the map has an associated p-dimensional prototype consisting of a vector of intervals $\mathbf{w}_k = ([u_k^1, v_k^1], \ldots, [u_k^p, v_k^p])^T$. The proposed algorithm is based on the batch training algorithm exposed in Section 2.2. The L_2 distance defined in Equation (6) is used to measure the proximity between two vectors of intervals. The Gaussian neighborhood function presented in Equation (2) is used to determine the neighborship between neurons.

The algorithm is listed as follows:

1. Initialization: $t = 0$.
   - Choose the map dimensions ($lines, cols$). The number of neurons is $K = lines \cdot cols$.
   - Choose the initial value (σ_i) and final value (σ_f) of the neighborhood radius.
   - Choose the total number of iterations ($totalIter$).
   - Choose the first K input vectors as the first initial prototype vectors:
2. Allocation:

- For $i = 1$ to n, compute the Best Matching Unit $c(i)$ of the input vector $\mathbf{R}_i = ([a_i^1, b_i^1], \ldots, [a_i^p, b_i^p])^T$. $c(i)$ is the neuron whose prototype vector is closest to data vector $\mathbf{R}_i$ in term of L_2 distance:

$$d(\mathbf{R}_i, \mathbf{w}_{c(i)}) = \underbrace{min}_{k=1,\ldots,K} d(\mathbf{R}_i, \mathbf{w}_k), \tag{7}$$

  where d is the L_2 distance defined in Equation (6).

- For $k = 1$ to K, compute the values of the neighborhood function $h_{kc(i)}(\sigma(t)), (i = 1,..., n)$.

3. Training:
   - Update the prototype vectors of the map:
     For $k = 1$ to K, compute the prototype vectors as follows:

$$u_k^j(t+1) = \frac{\sum_{i=1}^{n} h_{kc(i)}(\sigma(t)) a_i^j}{\sum_{i=1}^{n} h_{kc(i)}(\sigma(t))}; (j = 1, \ldots, p),$$

$$v_k^j(t+1) = \frac{\sum_{i=1}^{n} h_{kc(i)}(\sigma(t)) b_i^j}{\sum_{i=1}^{n} h_{kc(i)}(\sigma(t))}; (j = 1, \ldots, p).$$

4. Increment t and reduce the neighborhood radius $\sigma(t)$ according to Equation (8). Repeat step 2 until t reaches the maximum number of iterations ($totalIter$):

$$\sigma(t) = \sigma_i + \left(\frac{t}{totalIter}\right) \cdot (\sigma_f - \sigma_i). \tag{8}$$

4 Experimental Results

The data set used as experiment concerns monthly minimal and maximal temperatures observed in 60 meteorological stations mounted all over China. These data are provided by the Institute of Atmospheric Physics of the Chinese Academy of Sciences in Beijing, China and can be downloaded at http://dss.ucar.edu/datasets/ds578.5/.

Table 1 shows an example of the interval data set. The lower bound and the upper bound of each interval are respectively the monthly minimal and maximal temperature recorded by a station for the year 1998.

Table 1. Chinese stations minimal and maximal monthly temperatures

Num.	Station	January	February	...	December
1	AnQing	[1.8,7.1]	[2.1,7.2]	...	[4.3,11.8]
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
30	NenJiang	[-27.9,-16]	[-27.7,-12.9]	...	[-26.1,-13.8]
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
60	ZhiJiang	[2.7,8.4]	[2.7,8.7]	...	[5.1,13.3]

The data set consists of $n = 60$ vectors of intervals each one of dimension $p = 12$. The self-organizing map is a square grid composed of $K = 16$ neurons. The training of the map is performed according to the algorithm described in Section 3.2. The initial neighborhood radius is $\sigma_i = 3$ and the final neighborhood radius is $\sigma_f = 0.1$. The total number of iterations is $totalIter = 200$. The initial prototype vectors are equal to the first K input vectors.

4.1 Visualization of the Map and the Data in Two-Dimensional Subspace

In order to be able to visualize the map and the data, we used interval principal component analysis (Method of centers) [1] to project the prototype vectors and the data vectors on a subspace spanned by two eigenvectors of the data with greatest eigenvalues. Fig. 3 shows the results of PCA projection of the data vectors and the prototype vectors connected with their centers. We notice a good degree of map deployment over the data and a good degree of map topology preserving.

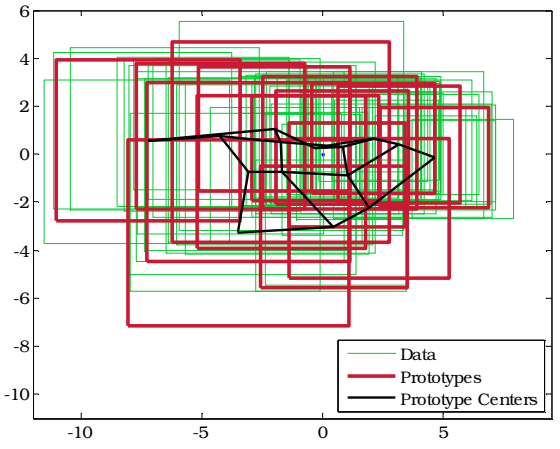


Fig. 3. PCA Projection of the prototype vectors and the data.

4.2 Clustering Results and Interpretation

The trained network leads us to a partition of 16 clusters. Fig. 4 shows the repartition of the data vectors over the 16 neurons. Fig. 5 represents the China map containing the 60 stations.

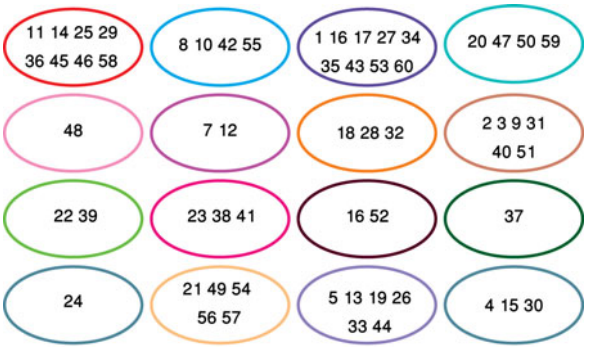


Fig. 4. SOM grid and stations clustering results.

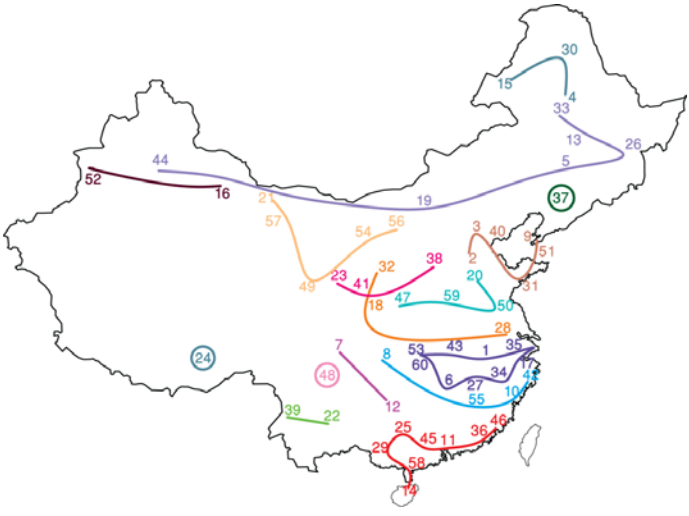


Fig. 5. Clusters distribution on the geographical map of China.

All stations of the same cluster are drawn with the same color. We can conclude that the stations located near each other geographically or situated approximately at the same latitude tend to be assigned to the same cluster or to a neighbor cluster. The first neuron (first line, first column) of the grid contains stations installed in the warmest regions of China (south coast). The last neuron of the grid (fourth line, fourth column) contains stations installed in the coldest regions of China. We can conclude that colder regions are found as we move down and to the right on the SOM grid. Table 2 gives the list of the 60 stations.

Table 2. Chinese meteorological stations

Number	Station	Number	Station
1	AnQing	31	QingDao
2	BaoDing	32	QingJiang
3	BeiJing	33	QiQiHaEr
4	BoKeTu	34	QuZhou
5	ChangChun	35	ShangHai
6	ChangSha	36	ShanTou
7	ChengDu	37	ShenYang
8	ChongQing	38	TaiYuan
9	DaLian	39	TengChong
10	FuZhou	40	TianJin
11	GuangZhou	41	TianShui
12	GuiYang	42	WenZhou
13	HaErBin	43	WuHan

Table 2. *(continued)*

14	HaiKou	44	WuLuMuQi
15	Hailaer	45	WuZhou
16	HaMi	46	XiaMen
17	HangZhou	47	XiAn
18	HanZhong	48	XiChang
19	HuHeHaoTe	49	XiNing
20	JiNan	50	XuZhou
21	Jiu Quan	51	YanTai
22	KunMing	52	Yi Ning
23	LanZhou	53	YiChang
24	LaSa	54	YinChuan
25	LiuZhou	55	YongAn
26	MuDanJiang	56	YuLin
27	NanChang	57	ZhangYe
28	NanJing	58	ZhanJiang
29	NanNing	59	ZhengZhou
30	NenJiang	60	ZhiJiang

5 Conclusions

We proposed an algorithm to train the self-organizing map for interval data. We used the L_2 distance to compare two vectors of intervals. We applied our method on real interval temperature data provided by Chinese meteorological stations. We obtained good clustering results while preserving the topology of the data. Stations installed in regions with similar climate were allocated to the same neuron or to a neighbor neuron on the SOM grid.

A prospect of this work might be to develop a self-organizing map based on Mahalanobis distance for interval data which allows the recognition of clusters of different shapes and sizes. Another prospect would be to cluster the self-organizing map itself in order to reduce the number of clusters.

References

[1] Cazes, P., Chouakria, A., Diday, E., Schektman, Y.: Extension de l'analyse en composantes principales à des données de type intervalle. Rev. Stat. Appl. 14, 5–24 (1997)

[2] Chouakria, A.: Extension des méthodes d'analyse factorielle à des données de type intervalle. PhD dissertation, Université Paris-Dauphine (1998)

[3] Billard, L., Diday, E.: Regression analysis for interval-valued data. In: Kiers, H., Rasson, J.-P., Groenen, P., Schader, M. (eds.) Data Analysis, Classification, and Related Methods, Proceedings of 7th Conference of the International Federation of Classification Societies (IFCS 2000), pp. 369–374. Springer, Heidelberg (2000)

[4] Rossi, F., Conan-Guez, B.: Multi-layer perceptron on interval data. In: Jajuga, K., Sokolowski, A., Bock, H.-H. (eds.) Classification, Clustering, and Data Analysis, pp. 427–436. Springer, Berlin (2002)
[5] Chavent, M., Lechevallier, Y.: Dynamical clustering of interval data: optimization of an adequacy criterion based on Hausdorff distance. In: Jajuga, K., Sokolowski, A., Bock, H.-H. (eds.) Classification, Clustering, and Data Analysis, pp. 53–60. Springer, Berlin (2002)
[6] Barnsley, M.: Fractals everywhere, 2nd edn. Academic Press (1993)
[7] Chavent, M.: Analyse des données symboliques. Une méthode divisive de classification. PhD dissertation, Université Paris-Dauphine (1997)
[8] Bock, H.-H.: Clustering methods and Kohonen maps for symbolic data. J. Jpn. Soc. Comput. Stat. 15, 217–229 (2003)
[9] Hamdan, H., Govaert, G.: Classification de données de type intervalle via l'algorithme EM. In: Proceedings of XXXV^{ème} Journées de Statistique (SFdS), Lyon, France, pp. 549–552 (2003)
[10] Hamdan, H., Govaert, G.: Modélisation et classification de données imprécises. In: Proceedings of Premier Congrès International sur les Modélisations Numériques Appliquées (CIMNA 2003), Beyrouth, Liban, pp. 16–19 (2003)
[11] Hamdan, H., Govaert, G.: Mixture model clustering of uncertain data. In: Proceeding of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2005), Reno, Nevada, USA, pp. 879–884 (2005)
[12] Hamdan, H., Govaert, G.: CEM algorithm for imprecise data. Application to flaw diagnosis using acoustic emission. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics (IEEE SMC 2004), The Hague, The Netherlands, pp. 4774–4779 (2004)
[13] Hamdan, H., Govaert, G.: Int-EM-CEM algorithm for imprecise data. Comparison with the CEM algorithm using Monte Carlo simulations. In: Proceedings of IEEE International Conference on Cybernetics and Intelligent Systems (IEEE CIS 2004), pp. 410–415 (2004)
[14] Chavent, M.: An Hausdorff distance between hyper-rectangles for clustering interval data. In: Banks, D., House, L., McMorris, F.-R., Arabie, P., Gaul, W. (eds.) Classification, Clustering, and Data Mining Applications, pp. 333–340. Springer, Heidelberg (2004)
[15] De Souza, R.M.C.R., De Carvalho, F.A.T.: Clustering of interval data based on city-block distances. Pattern Recognit. Lett. 25, 353–365 (2004)
[16] De Souza, R.M.C.R., De Carvalho, F.A.T., Tenrio, C.P., Lechevallier, Y.: Dynamic cluster methods for interval data based on Mahalanobis distances. In: Proceedings of 9th Conference of the International Federation of Classification Societies (IFCS 2004), pp. 351–360. Springer, Heidelberg (2004)
[17] El Golli, A., Conan-Guez, B., Rossi, F.: Self-organizing maps and symbolic data. J. Symb. Data Anal. (JSDA) 2 (2004)
[18] Kohonen, T.: Self organization and associative memory, 2nd edn. Springer, Heidelberg (1984)
[19] Kohonen, T.: Self-organizing maps, 3rd edn. Springer, Heidelberg (2001)
[20] De Carvalho, F.A.T., Brito, P., Bock, H.-H.: Dynamic clustering for interval data based on L_2 distance. Comput. Stat. 21, 231–250 (2006)
[21] Hajjar, C., Hamdan, H.: Self-organizing map based on L_2 distance for interval-valued data. In: Proceedings of IEEE 6th International Symposium on Applied Computational Intelligence and Informatics (IEEE SACI 2011), Timisoara, Romania, pp. 317–322 (2011)

A Set of Java Metrics for Software Quality Tree Based on Static Code Analyzers

Ciprian-Bogdan Chirilă and Vladimir Crețu

"Politehnica" University Timişoara,
Faculty of Automation and Computers,
Bd. V. Parvan 2, 300223 Timisoara, Romania
{chirila,vcretu}@cs.upt.ro

Abstract. Assessing software quality allows cost cuts from the early development stages. Software quality information helps taking development decisions, checking fault corrections effect, estimating maintenance effort. Our fault density based quality model relies on static source code analyzers and on a set of language specific metrics. We compute the fault ratio for each static analyzer rule. Giving user defined weights to fault ratios we can quantify quality as a number. We identified, described informally and implemented in a prototype a set of Java metrics in order to fulfill our model and to accomplish our quality assessment goal.

1 Introduction

Quality assessment has a great impact in the software development process. Quality measurement during the development of a software system can cut significantly future maintenance costs. In this chapter we propose a software quality assessment model which is based on static code analyzers designed for Java [1] object-oriented systems. Static code analyzers are software tools which analyze the software systems source code and detect programming rules and principles violations.

In Fig. 1 we present the main elements of our approach. We start with the Java source code of the object-oriented system. Next, we feed the source code to static code analyzers in order to generate lists of issues. On the other hand, we parse the source code with our prototype called "ProStarter" in order to apply a set of base metrics and to obtain their measurement results. The list of issues and the computed results are used by our quality assessment model to measure the quality of the analyzed object-oriented system. The goal of our work is to identify and to define a suite of Java specific metrics in order to implement our quality assessment model.

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 147–162.
springerlink.com

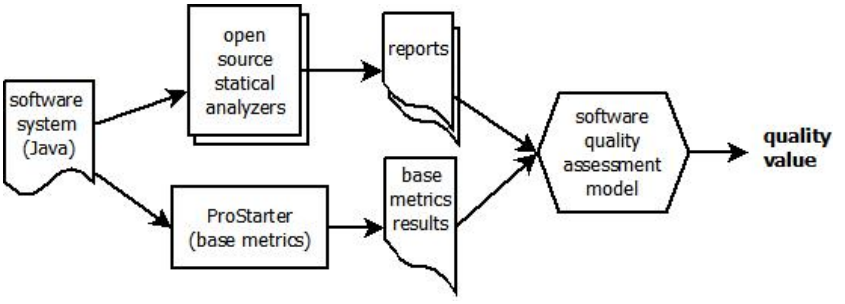


Fig. 1. Quality assessment approach.

The chapter is structured as follows. In Section 2 we present our quality assessment model in a nutshell. In Section 3 we identify a set of class related metrics in the context of Java programming language. Section 4 identifies metrics for counting Java statements. Section 5 identifies metrics related to Java expressions, operators (e.g. cast operator), identifiers. In Section 6 we identify a suite of Java Server Pages and Java Server Faces related metrics. Section 7 presents the XML model representation. Section 8 shows the factor coverage determined by the rule to factor mappings. In Section 9 we present a few implementation elements for the defined Java metrics. Section 10 presents metrics related works. Section 11 concludes and sets the future work.

2 The Quality Assessment Model in a Nutshell

For Java object-oriented systems there are several free source code analyzers like: PMD [2], FindBugs [3], Metrics [4], CheckStyle [5], inFusion [6], ucDetector [7] which are used in the production of industrial strength projects. In our work we will rely on two source code verification and validation tools namely PMD [2] and FindBugs [3], since they are very commonly used in industry.

Our quality assessment methodology is based on three main principles [8]:

i) maping static analyzer rules to general purpose quality factors;
ii) computing a score in a given range [min,max] for each rule based on the density of rule violating entities (classes, methods, statements, expressions) using formula (1). Parameter *NoOfViolatingEntities* represents the number of entities violating the programming rules, while parameter *TotalNoOfSameKindEntities* denotes the number of all entities of the same kind;
iii) computing the quality value as a weighted mean of the previously computed scores.

$$R = (\max - \min) \cdot (1 - \frac{NoOfViolatingEntities}{TotalNoOfSameKindEntities}) \quad (1)$$

Firstly, each rule is mapped to a quality factor according to their description. The selected factors are: understandability, completeness, conciseness, portability, consistency, maintainability, testability, usability, reliability, efficiency and they are inspired from the ISO/IEC 9126 standard [9]. The mapping process is highly subjective and depends on the mapers code reviewing experience.

For example, the *EmptyCatchBlock* rule detects all empty catch blocks from try-catch statements. This means that such empty catch blocks do not treat exceptions which may be thrown at the runtime. We consider that such an aspect will affect mostly the reliability quality factor so the mapping is made to this quality factor.

Secondly, the penalty induced by a rule is proportional with the number of erroneous entities detected by that rule divided by the total number of same kind entities from the project. Entities mean: packages, classes, methods, fields, statements (e.g. while, for), operators etc. Coming back to our empty catch block example, the *EmptyCatchBlock* rule will imply a score which will be computed using the formula (2), which is a particularization of the general formula (1):

$$R = (10-0)\cdot(1-\frac{NoOfEmptyCatchBlocks}{NoOfCatchBlocks}), \tag{2}$$

where:

i) *NoOfEmptyCatchBlocks* is the number of all empty catch blocks from the analyzed project;
ii) *NoOfCatchBlocks* is the total number of catch blocks from the project.
In a software system having 1000 try-catch statements with 100 empty catchblocks the penalty fraction computed according to formula (2) will be 100/1000=0.10 and the score according to the same formula having min=0 and max=10 will be 10*(1-0.1)=9.00.

In order to assess software quality using our model and in particular for the empty catch block example we need two metrics for the penalty fraction nominator and denominator:

i) *NoOfEmptyCatchBlocks* representing the count of all issues which are automatically detected and reported by the *EmptyCatchBlock* PMD rule;
ii) *NoOfCatchBlocks* is a Java specific base metric whose values must be obtained by other code analysis tools. Metrics [4] static code analyzer represents a partial solution in this sense because it computes some of the needed base metrics, but not all of them, so we consider building metrics module in our "ProStarter" prototype.

Thirdly, all computed scores are weighted and added altogether to compute the value of the software quality. In formulas (3), (4), (5), (6) we define the mathematical model through which the software quality value is computed:

$$Q = \sum_{i=1}^{N} f_i \cdot F_i \cdot \tag{3}$$

In formula (3) the terms have the following meanings: Q represents the numerical value of the software quality, N is the number of factors from the model, F_i is the score of the quality factor, f_i is the weight factor for F_i. In order to preserve the score values in a predefined range of [min,max] we have impose the restriction given in equation (4):

$$\sum_{i=1}^{N} f_i = 1 \cdot \tag{4}$$

The sum of all weight factors f_i must be equal to 1:

$$F_i = \sum_{j=1}^{M_i} r_j \cdot R_j \cdot \tag{5}$$

In (5) we define the formula for computing the score values for each F_i factor where: M_i represents the number of rules mapped to quality factor F_i, R_j is the computed score according to rule R_j, r_j is the weighted factor for R_j. In equation (6) we impose the same restriction as earlier that the sum of all r_j weight factors is 1:

$$\sum_{j=1}^{M_i} r_j = 1, i = 0..N \cdot \tag{6}$$

Theoretically, quality factors can be arbitrarily chosen but we consider that in practice quality factors should be computed using formulas (7) and (8):

$$f_i = \frac{M_i}{\sum_{k=1}^{N} M_k}, \tag{7}$$

$$r_i = \frac{1}{M_i} \cdot \tag{8}$$

Our quality assessment model is represented using the human-readable XML standard [10] and is described in Section 7.

3 Class Metrics

In this section we will identify a set of class related metrics needed in our quality assessment model.

Classes in Java are grouped in subsystems or packages corresponding to folders on the disk. The *NoOfPackages* metric counts the total number of packages from the analyzed object-oriented system. It is used in the context of rule *PackageCase* detecting packages having uppercase names. Thus, we can compute the percentage of packages violating this rule.

Metric *NoOfClasses* denotes number of classes in a project. It is used in our quality assessment model by rules like: *ExcessiveClassLength* which identifies all classes over a given threshold length; *TooManyMethods* counting classes with too many methods; *AtLeastOneConstructor* counting classes which have no constructors; *CouplingBetweenObjects* counting unique attributes, local variables and return types within an object. Objects with a higher degree of coupling than a specified threshold will be reported as violating the rule.

Regarding to this metric we are aware of the fact that Java has several kinds of classes like: nested classes (static and non-static), inner classes, anonymous classes.

Metric *NoOfInterfaces* counts number of interfaces in a system. The rule entitled *AvoidConstantInterface* detects interfaces containing constants and uses this metric in our quality assessment model to compute penalty and scoring.

Metric *NoOfClassifiers* counts the number of both classes and interfaces and is defined by formula (9):

$$NoOfClassifiers=NoOfClasses+NoOfInterfaces. \quad (9)$$

This metric is used by rule *NcssTypeCount* which counts the number of non-commenting source statements for a given class or interface. If the count is not in a predefined range then the violating classifiers (classes and interfaces) are reported as flaws. The percentage of these flaws is computed with the help of *NoOfClassifiers* metric.

In relation to *NoOfClasses* and *NoOfInterfaces* metrics we have to define a metric named *NoOfCompilationUnits* which counts the number of the .java files. We remind that such a compilation unit may contain multiple classes and interfaces and at least one of them is public having the same name as the .java file. In our model the *ExcessiveImports* rule uses the *NoOfCompilationUnits* metric in order to count the percentage of excessive import units.

Metric *NoOfFields* counts the number of fields from the analyzed project. The rule named *SingularField* counts all fields which are used in only one method for storing some values, and not reading them somewhere else. Such fields can be transformed into local variables and in order to compute their percentage and their rule scoring we use the *NoOfFields* metric.

The *NoOfMethods* metric is used by several rules following our approach. The *ProperCloneImplementation* rule detect clone methods which do not call *super.clone()*. Such violations count should be divided by the *NoOfMethods* metric in order to compute the penalty percentage and the rule scoring. Rule *CloneThrowsCloneNotSupportedException* detects clone methods which do not

throw the *CloneNotSupportedException* exception. In this case we consider also using the *NoOfMethods* metric for penalty and score computation.

Rule *NPathComplexity* detects the number of acyclic execution paths through a method. Methods with more than 200 paths are considered as being too complex. We consider that a suitable denominator for this rule is the *NoOfMethods* base metric.

Rule *ExcessiveMethodLength* is another example which uses the *NoOfMethods* metric. In this context the rule count the number of methods which exceeds a given line code count limit. The count of oversized methods is divided to the number of all methods in the project, thus computing the percentage of oversized methods in the project and calculating the score.

The *NoOfFinalizers* metric counts the number of *finalizer()* methods. These methods act like C++ destructors and they are meant to perform finalization tasks for an object. In our model this metric is used by several rules:

i) *EmptyFinalizer* rule detects such empty methods which should not exist in the code. The identified metric will help us compute the percentage of such senseless methods.
ii) *FinalizeOnlyCallsSuperFinalize* rule detects finalizer methods calling their superclass versions without taking any other actions. Such implementations are not semantically correct. The identified metric helps detecting the percentage of such flaws.
iii) *FinalizeOverloaded* rule detects finalize methods having arguments. This means that the finalizer method is overloaded. There are two possibilities for the code and both are wrong: a) the method contains no finalizing actions but it was named by chance with a dedicated name; b) the method do contain finalizing actions and because of the wrong signature it will never be called by the garbage collector. Both such situations are detected by this rule and the percentage of such abnormal situations can be computed with the help of the identified metric.
iv) Rule *FinalizeDoesNotCallSuperFinalize* detects another semantical anomaly in the dynamic of the object destruction mechanism.
v) *FinalizeShouldBeProtected* rule was written in order to check an encapsulation aspect: finalizer rules must be protected, thus they are available only in the class hierarchy. The rule detects all deviations from this principle. The base metrics, again, facilitates the computation of the violation percentage and implicitly the scoring.

The *NoOfModifiers* metrics counts the total number of modifiers. This metric is used in quality assessment in the context of rules like:

i) *UnnecessaryFinalModifier* detects all explicitly final declared methods within a final class;
ii) *UnusedModifier* detects superfluous modifiers attached to interface fields and methods. Implicitly, fields in interfaces are public static and final, while methods are public and abstract.

4 Statement Metrics

In this section we present metrics identified at the statement level.

The *NoOfCalls* metric counts the frequency of calls within the analyzed project. It could be equally categorized as a statement or as an expression related metric. This metric is used in our model by several rules like:

i) *CheckResultSet* which states that the return values of database navigation methods must be checked before using them. The score computed for this rule with our approach will be the percentage of unchecked calls.
ii) *AvoidAccessibilityAlteration* rule detects calls which break encapsulation by exporting private constructor objects, thus allowing instance creation. This rule will help us computing the percentage of encapsulation breaking calls.
iii) *DoNotCallGarbageCollectionExplicitly* rule detects garbage collector unnecessary calls. According to this rule the percentage of such calls is computed and a score is issued.
iv) *AvoidCallingFinalize* rule detects unnecessary "destructor" calls, like the previous rule.
v) *DoNotCallSystemExit* rule is dedicated to web applications. It is natural choice disallowing calls to *System.exit()* in this context. Using the base metric *NoOfCalls* we can compute the percentage and the penalty score for such abnormal calls.

The *NoOfIfs* metric counts the number of *if* instructions. It will be used in our model in the context of the following rules:

i) Rule *EmptyIfStmt* detects the empty conditional statements.
ii) Rule *UnconditionalIfStatement* detects if statements which are always true or false.
iii) *ColapsibleIfStatements* detects conditional statements which can be consolidated into one conditional by separating their condition with a short circuit boolean operator.
iv) Rules *IfStmtsMustUseBraces* and *IfElseStmtsMustUseBraces* detect badly written *if* and *else* branches using no braces for their blocks.
v) *AvoidDeeplyNestedIfStmts* detects nested conditionals which are hard to read and understand.

All the previously listed rules are based on the *NoOfIfs* base metric for the score computation.

Metric *NoOfSwitches* calculates the number of switch instructions from the project. It is used in the context of the following rules:

i) Rule *EmptySwitchStatements* detects *switch* statements which are void. Such statements should be avoided.
ii) Rule *SwitchStmtsShouldHaveDefault* detects *switch* statements which have no default label. This means that the switch statement is incomplete and some values are not taken into account explicitly.
iii) Rule *DefaultLabelNotLastInSwitchStmt* detects *default* labels which are not last in the choice list. Such a practice invalidates the choices after the default label making them unreachable.

iv) Rule *NonCaseLabelInSwitchStatement* detects *switch* statements having legal, but confusing names for their labels. Such names should be changed for increased understandability.
v) Rule *MissingBreakInSwitch* detects *switch* statements which have no breaks. Breaks are needed in order to exit the *switch* statement after the execution of one label instructions. Sometimes one can execute the instructions from several switch labels if the logic of the application needs it. In practice such cases are rare so they must be detected as possible bugs.
vi) Rule *TooFewBranchesForASwitchStatement* detects *switch* statements with just a few branches easily replaceable by a conditional statement for increased understandability.

For all deviations from the listed rules our model uses the *NoOfSwitches* base metric for the score computation.

Base metric *NoOfLoops* computes number of Java loops: *for*, *while*, *do while*. We mention that in the latest releases of Java the *for* statement has two forms: the old form inherited from C and the new one dedicated to iterators.
Metric *NoOfFors* counts the number of for statements. This base metric stands as denominator for the following PMD rules of our model:

i) Rule *ForLoopsMustUseBraces* counts the number of *for* loops which use no braces.
ii) Rule *ForLoopShouldBeWhileLoop* counts the number of *for* loops which could be simplified to *while* loops, thus increasing understandability.

For the implementation of this metric we consider counting the number of *for* keywords encountered in the AST.

In the same family of loops we consider the metric *NoOfWhiles* counting number of *while* instructions. This base metric is used by the rules detecting anomalies in this context.

i) Rule *EmptyWhileStmt* detects void *while* statements.
ii) Rule *WhileLoopsMustUseBraces* detects *while* statements with no braces.
For the sake of completeness we define also the base metric *NoOfDoWhiles* denoting the number of *do while* instructions. The relation between the last discussed metrics is given by formula (10):

$$NoOfLoops=NoOfFors+NoOfWhile+NoOfDoWhiles. \quad (10)$$

Base metric *NoOfTries* counts the number of *try-catch* instructions. Rule *EmptyTryBlock* detects empty *try* blocks.

The *NoOfCatches* metric counts the number of *catch* instructions used in the project. In the presented example relying on formula (1) we explained the aspect assessed by our quality model.

In the context of exceptions we identified a metric which we name *NoOfFinallyBlocks* denoting the number of finally blocks. Rule *EmptyFinallyBlock* detects the empty finally blocks. Combining the two of them we can compute the percentage of the empty finally blocks within the entire project and the corresponding scoring.

5 Expression Metrics

In this section we will present expression related metrics.

Metric *NoOfInstantiations* counts the number of instantiations found in the project source code. Instantiating some particular classes are considered issues by some rules.

i) Rule *AvoidThreadGroup* detects instantiations of *ThreadGroup* class which are not recommended because contain methods which are not thread safe.
ii) Rule *BigIntegerInstantiation* detects *BigInteger* and *BigDecimal* class instantiations for values 0,1,10. These classes have public static members with those values already instantiated.
iii) Rule *AccessorClassGeneration* detects code which instantiates private constructors from outside of the constructor class.
iv) Rule *InstantiationToGetClass* detects object instantiations for class interrogation. The efficient way to get such information is to use the *.class* public member instead.
v) Rule *SimpleDateFormatNeedsLocale* detects instantiations of *SimpleDate* class without selecting a *Locale* object before.
vi) Rules *IntegerInstantiation*, *ByteInstantiation*, *ShortInstantiation*, *LongInstantiation* detect several numeric type instantiations for String to value conversions and they are considered to be memory consuming. The alternative is to use the wrapper method *valueOf* from each class.
vii) Rule *StringInstantiation* detects String instantiations which usually are not necessary.
viii) Rule *StringBufferInstantiationWithChar* detects cases when a *StringBuffer* class is instantiated with a character literal argument. In such cases the code value of the character is taken into account determining the size and not the content as maybe intended.

Metric *NoOfComparisons* counts the number of comparisons with the == or != operators. Rule *UnusedNullCheckInEquals* detects abnormal situations when after checking a reference for null, the object is passed to another objects's *equal*s method instead of invoking *equals* on the first object.

Metric *NoOfConstants* counts the number of constant literals from the project. Rule *AvoidUsingOctalValues* detects uses of octal values which are forbidden because of readability reasons. Rule *AvoidUsingHardCodedIP* detects uses of hard coded IP addresses in the project. *The NoOfConstants* metric is useful to compute the scoring for the previously presented two rules.

6 Script Metrics

In this section we will present a set of metrics identified in the process of quality assessment for Java web applications. The PMD rules dedicated to Java web applications detect erroneous entities like:

i) Java Server Pages (JSP);
ii) Java Server Faces (JSF);
iii) JSP scriptlets;
iv) iframes from HTML code.

We will start with the *NoOfScripts* metric which counts the scripts from the web application. It is a good practice to put scripts in tag libraries rather than in Java server pages. The rule entitled *NoLongScripts* will detect the scripts over a specified threshold and together with the earlier presented metric will help computing the violation percentage.

Metric *NoOfIFrames* counts the number of HTML iframes. Rule *IframeMissingSrcAttribute* counts iframes which have empty src attributes. When displayed, such iframes have no content at all. With the two iframe counts we can compute the percentage of empty iframes.

Metric *NoOfJSFs* counts the number of pages containing Java Server Faces mechanisms. Rule *DontNestJsfInJstlIteration* detects incorrect uses of this mechanism. The base metric will help us in computing the violation ratio.

Metric *NoOfJSPs* counts the Java Server Pages from a Java web project. This metric is used by several PMD rules:

i) Rule *NoScriptlets* detects JSP pages which contain scriptlets. They should be factored into tag libraries or JSP declarations. In order to compute the violation percentage of these aspects we rely on the *NoOfJSPs* base metric.
ii) Rule *NoJSPForward* locates JSP pages using forwarding directives. Such a behavior is not semantically correct, so it is detected as an issue by the PMD static analyzer. The percentage of violations in this sense is calculated on using the *NoOfJSPs* metric.
iii) Rule *NoHtmlComments* detects Java Server Pages which contain HTML comments increasing the transfer load between the server and the client at runtime. Such comments should be switched to JSP comments. The percentage of violations in this case is computed with the help of NoOfJSPs base metric.
iv) Rule *DuplicateJspImports* detects JSPs which have duplicate imports.
v) Rule JspEncoding detects JSPs which miss the *meta* encoding tag.

Metric *NoOfScriptAttributes* counts the number of HTML attributes used in scripts and in combination with the *NoClassAttribute* PMD rule which detects attributes using the dedicated name *class*. For this kind of situations using the rule result and the base metric value we can compute the violation percentage and the scoring.

7 Model Representation

Our quality model is represented as a hierarchical set of XML files. On the top of the hierarchy is the "quality factors.xml" file which contains meta-information like: model quality factors, their descriptions and concrete project information like: project name, weight factors and computed scores.

In Fig. 2 we present the structure of the XML representation of the proposed quality assessment model. The main element of the XML file is <quality> which contains a <description> child element and a list of child <factor> elements. Each <factor> element has several attributes like name, weight factor and computed score. Each factor element has its own <description> child element embedding a descriptive text.

```
<?xml version="1.0" encoding="utf-8"?>
<quality project="ProjectName" computedscore="8.03">
 <description>Software quality ...</description>
 <factor name="Understandability" weightfactor="1" computedscore="8.55">
  <description>Understandability is ...</description>
 </factor>
 <factor name="Completeness" weightfactor="2" computedscore="7.66">
  <description>Completeness is ...</description>
 </factor>
 ...
</quality>
```

Fig. 2. Quality factors XML representation fragment.

Next, for each analyzer we have an XML file where we set the mapping between the rules and factors: "pmd-rules.xml", "fb-rules.xml".

In Fig. 3 we present a fragment from the XML representation of rules to factors mappings. The main element of the XML file is the <tool> element. This element has a name property. The first child of this element is a <description> element containing a descriptive text. Next, all the <ruleset> elements are listed having properties like name, weight factor and computed score. Each rule set contains a list of <rule> elements. The rule element is the most complex element having the following attributes. Name denotes the name of the rule. *Factor1* denotes the most affected quality factor when the rule is violated. *Factor2* denotes the secondly most affected quality factor when the rule is violated. *Profile* is an attribute which may get the following values:

i) *all* - stands for the fact that the current rule is suitable to all kind of Java projects;
ii) *embedded* - means that the current rule fits to Java embedded systems;
iii) *web* - specifies that the rule is dedicated to Java web projects;
iv) *enterprise* - denotes that the current rule is used for Java enterprise applications.

```
<?xml version="1.0" encoding="utf-8"?>
<tool name="pmd">
<ruleset name="AndroidRules" weightfactor="1" computedscore="7.00">
 <description>These rules deal with the Android SDK ...</description>
 <rule name="CallSuperFirst" factor1="Reliability" profile="embedded"
  denominator="NoOfCalls" weightfactor="1" computedscore="7.00">
  <description>
   Super should be called at the start of the method.
  </description>
 </rule>

 <rule name="CallSuperLast" factor1="Reliability" profile="embedded"
  denominator="NoOfCalls" weightfactor="1" computedscore="8.50">
  <description> Super should be called at the end of the method.</description>
 </rule>

 <rule name="ProtectLogD" factor1="Reliability" profile="embedded"
  denominator="NoOfCalls" weightfactor="1" computedscore="6.33">
  <description>
   Log.d calls should be protected by checking Config.LOGD first.
  </description>
 </rule>
...
</ruleset>
...
</tool>
```

Fig. 3. PMD rules XML representation fragment.

The weight factor and computed score attributes are useful in the numerical quality assessment process.

8 Quality Model Factors Coverage

In this section we will analyze how the quality factors are covered by the analyzer rule sets. We are interested how much the quality factors are covered by each source code analyzers separately and then globally.

In Fig. 4 we present the number of rules from each static code analyzer mapped to each quality factor and also the coverage percentages. We notice that each tool covers in an acceptable manner the quality factors.

The most covered quality factor is reliability. This factor is covered by almost 50% of the rules for both considered source code analyzers.

The least covered quality factor is usability: by one rule in PMD and by no rule in FindBugs. This is normal since source code analysis can not tell much about the programs usability.

Great differences between coverage factors provided by the two source code analyzers resulted for factors like: completeness, security and conciseness.

	Factor/Analyzer	PMD	FB	Total	PMD%	FB%
1	Reliability	60	83	143	41.96	58.04
2	Completeness	4	111	115	3.48	96.52
3	Efficiency	48	42	90	53.33	46.67
4	Consistency	27	39	66	40.91	59.09
5	Understandability	30	17	47	63.83	36.17
6	Conciseness	35	8	43	81.40	18.60
7	Testability	28	13	41	68.29	31.71
8	Security	2	32	34	5.88	94.12
9	Maintainability	8	17	25	32.00	68.00
10	Portability	8	7	15	53.33	46.67
11	Usability	1	0	1	100.00	0.00
	Total	251	369	620	40.48	59.52

Fig. 4. Rule sets covering quality factors.

9 Prototype Implementation

The ongoing implementation is based on a Java source code modeling library, namely Recoder [11] based on JavaCC [12, 13]. The base structure for the implementation of our metrics is the AST (Abstract Syntax Tree) of the analyzed object-oriented system. The metrics implementation is based on counting AST nodes of a given type. For example, the *NoOfFors* metric implementation is presented in Fig. 5. In Fig. 5 we present a code fragment capable of computing the value for the *NoOfFors* metric. Firstly, a cross reference service configuration object is created. Secondly, the input path attribute is set for this object. Thirdly, the project files are checked to be in the same path. Fourthly, the list of compilations units is retrieved from the service configuration object. Fifthly, each compilation unit is iterated and a tree walker inspects all program elements. When a “for” statement is encountered the metric value is incremented.

```
CrossReferenceServiceConfiguration crsc =
new CrossReferenceServiceConfiguration();
crsc.getProjectSettings().setProperty(PropertyNames.INPUT_PATH,
      "./input-project/src");
crsc.getProjectSettings().ensureSystemClassesAreInPath();
SourceFileRepository sfr = crsc.getSourceFileRepository();
List<CompilationUnit> cul = sfr.getAllCompilationUnitsFromPath();
crsc.getChangeHistory().updateModel();
int NoOfFors=0;
for(CompilationUnit cu : cul)
{
  TreeWalker tw = new TreeWalker(cu);
  while (tw.next())
  {
    if(tw.getProgramElement() instanceof For)
    {
      NoOfFors++;
    }
  }
}
```

Fig. 5. NoOfFors Metric implementation.

In the "ProStarter" prototype each metric is modeled as a class embedding a code sequence like the one presented earlier. The script metrics need different parsers in order to parse HTML, JSP, JSF pages.

10 Related Works

In literature there are lots of metric definitions and several open source projects implementing them. In our model we will reuse some classic metrics which are already implemented in existing source code analyzers.

In [14] a suite of object oriented metrics is defined and implemented in the *ckjm* static source code analyzer [15]. The implemented metrics are:

i) WMC: Weighted methods per class;
ii) DIT: Depth of Inheritance Tree;
iii) NOC: Number of Children;
iv) CBO: Coupling between object classes;
v) RFC: Response for a Class;
vi) LCOM: Lack of cohesion in methods.

In our approach we do not use these metrics directly. Static code analyzers may use them in the faults detection. The *Metrics* static code analyzer [4] implements a series of Java metrics:

i) Number of Classes;
ii) Number of Children;
iii) Number of Interfaces;
iv) Depth of Inheritance Tree (DIT);
v) Number of Overridden Methods (NORM);
vi) Number of Methods (NOM);
vii) Number of Fields;
viii) Lines of Code;
ix) Specialization Index (NORM*DIT/NOM);
x)McCabe Cyclomatic Complexity [16];
xi) Weighted Methods per Class (WMC);
xii) Lack of Cohesion of Methods (LCOM*).

The metrics used by our model are: Number of Classes, Number of Methods, Number of Overridden Methods, Number of Fields.

Robert Martin in [17] defines a suite of metrics for measuring the interdependence of subsystems which are also implemented in *Metrics* static code analyzer: i) Afferent Coupling (Ca) ii) Efferent Coupling (Ce) iii) Instability Ce/(Ca+Ce) iv) Abstractness (A) v) Normalized Distance from Main Sequence (Dn) |A+I-1|.

The *Metrics* source code analyzer with other analyzers measure software systems in a structural manner. Some metrics are computed at level of packages, classes, methods. Our model uses a more global approach. For example, we do not need to count the number of classes in each package, but the total number of classes from the entire project.

11 Conclusions and Future Work

In this book chapter we identified a suite of metrics (class, statement, expression, script related) needed for the quality assessment based on static code analyzers. Classical metrics are generally suitable to almost all programming languages. Language specific metrics are needed in order to implement our fault density quality assessment model.

Analyzing the issues detected by this static code analyzer we can draw the conclusion that classical metrics are not enough and that language specific metrics are needed in this matter. We identified four categories of metrics related to: classes, statements, expressions and web scripts.

On the other hand the effort to implement language specific metrics is not very great thanks to the Recoder library, which is based on the JavaCC parser generator. The majority of the metrics can be implemented by counting the lexical entities of the analyzed object-oriented system.

As future work we intend to complete the implementation of the identified metrics in special module of the “ProStarter” software tool. Next, we intend to

incorporate other static code analyzers like: *CheckStyle*, *inFusion*, *ucDetector* and others. Further, we intend to experiment the entire methodology on industrial strength projects.

Acknowledgments. The authors thank Dacian Tudor for the inception of this chapter idea and Dana Juratoni for her commitment in the implementation of the prototype.

References

[1] Gosling, J., Joy, B., Steele, G., Bracha, G.: The Java Language Specification, 3rd edn. Addison-Wesley Professional (2005)
[2] Copeland, T., et al.: PMD, http://pmd.sourceforge.org
[3] Pugh, B., Hovemeyer, D., Langmead, B., et al.: Findbugs - FindBugs in Java programs (2011), http://pmd.sourceforge.org
[4] Metrics 1.3.6 (2011), http://metrics.sourceforge.net
[5] CheckStyle 5.3 (2011), http://checkstyle.sourceforge.net
[6] Loose Research Group, inFusion (2011), http://loose.cs.upt.ro
[7] Spieler, J.: Unnecessary code detector (2011), http://www.ucdetector.org
[8] Chirila, C.B., Cretu, V.: A suite of Java specific metrics for software quality assessment based on statical code analyzers. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisoara, Romania, pp. 347–352 (2011)
[9] International Standard Organization, ISO/IEC 9126 information technology software product evaluation - quality characteristics and their guidelines for their use (2005)
[10] World Wide Web Consortium, Xml extensible markup language technology (2011), http://www.w3.org/standards/xml/
[11] Gutzmann, T.: Recoder (2011), http://recoder.sourceforge.net
[12] Sun Microsystems, Java Compiler Compiler (JavaCC) - the Java Parser Generator (2011), https://javacc.dev.java.net
[13] Copeland, T.: Generating parser with JavaCC (2011), http://generatingparserswithjavacc.com/
[14] Chidamber, S.R., Kemerer, C.F.: Towards a metrics suite for object-oriented design. In: Proceedings of OOPSLA 1991, pp. 197–211 (1991)
[15] Spinellis, D.: Chidamber and Kemerer Java metrics, ckjm (2007), http://www.spinellis.gr/sw/ckjm/
[16] McCabe, T.J.: A complexity measure. In: Proceedings of 2nd International Conference on Software Engineering (ICSE 1976), Los Alamitos, CA, USA, pp. 308–320 (1976)
[17] Martin, R.C.: OO design quality metrics an analysis of dependencies (1994), http://www.objectmentor.com/resources/articles/oodmetrc.pdf

VLearning, a New Direction for eLearning Challenges

Andrea Tick

Budapest Business School, Faculty of International Business and Management,
Diosy L. u.22-24, 1165 Budapest, Hungary
tick.andrea@kkfk.bgf.hu

Abstract. In the information society the exponentially growing knowledge means a serious challenge to all the participants in the teaching-learning process. The methodology of education and its technical background have developed and improved drastically, which after the proliferation of the information technology and the internet got a new spur. It has been especially true since the wider and wider spreading of the web 2.0 and the 3D based virtual environments. Compared to former eLearning solutions this new virtual learning environment based on 3D information technology encourages more vividly the improvement of the efficiency of the learning management systems, the intaking of a wider scope of students in the learning process and fosters study motivation slightly subconsciously. This chapter outlines the development of this technology starting from eLearning to vLearning, as well as analyzes the most significant components of the mentioned technology, with the use of which education strives to make pace with the challenges of the present era.

1 Introduction

In the information society of the 20th and 21st century the speeding-up development of information and telecommunications technologies primarily internet-technologies influences and partly changes our whole life style. The absorption of telecommunication and mass-media technologies enhanced the birth of the Global village as proposed by Marshall McLuhan in the 1960s. [1] According to him, with the help of electric technology connections can be built from any point of the world thus contracted the world into a small village. He presupposed the internet as an "extension of consciousness" [1] thirty years before its invention. As he stated, a computer as a research and communication instrument could enhance retrieval, obsolesce mass library organization, retrieve the individual's encyclopedic function and flip into a private line to speedily tailored data of a saleable kind [1]. The instantaneous movement of information from every quarter of the world to

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 163–174.
springerlink.com

every point of the world at the same time calls forth a radical change in the learning process as well as in the visualization of the learning environment. A new, more effective form of knowledge acquisition becomes a central role player in the evolving knowledge-based society, which is characterized by constructivist pedagogy, emphasizing the training of competences and life-long learning. These new knowledge acquisition processes claim that the roles of both students and teachers change. The era of lecturing in schools has passed by, the frontal method of educating becomes out-of-date, where a closed learning environment is created, set groups are formed and the emphasis is on teaching a previously determined set of data. Students need open, active, creative, collaborative, and primarily motivating learning environments. From being purely passive message receivers students turn into active participants of the learning process who strive to gain knowledge so are eager to actively participate in the process. The role of teachers also changes; their tasks are modified and become more fine-tuned, although the fundamental tasks do not disappear. The traditional approach to student-teacher personal distance changes, since it shrinks and the 'disciplined' form turns into a more 'democratic' one. Teachers have already gone through such a catharsis at the introduction of teaching computer sciences, when it was not rare that some students knew more in certain fields than the teacher teaching him/her [8].

As time passes by not only the roles change, not only the process but the learning environment changes as well, consequently learner-centered actual virtual classrooms come to existence in the global village in which the student-student as well as the student-teacher relationships transform radically. In this process that is in developing virtual learning environments and classrooms the fundamental transformation of communications technologies, the usages of web 2.0 as well as the proliferation of eLearning technologies are the key role players.

2 The Characteristics of Internet-Based Learning Environment

At the beginning of the spreading of the internet simple tools already appeared to help information gaining, to provide an online study support. With the rapid proliferation of the World Wide Web these solutions have become more and more user friendly, their usage has become more and more comfortable for users. As a result, the first applications that primarily supported distance learning came to light. Several authors have already analyzed the features of the internet environment. Dahl claims that internet "makes the communication activity active where the learner chooses what s/he is prepared to see and with whom to interact". "The internet", she proposes, "offers the possibility of a far more interactive experience, with a feedback directly provided" [2]. In reality the wider and wider use of the internet primarily founded the possibility of interactivity in the field of study

supporting environment as opposed to previous systems. Interactivity, community forming opportunities created a new layer in the study environment. The creation of self-organizing groups of students enables the formation of classes and study groups even at a very simply technical level.

Teaching and learning methods have already been classified according to the layers of information technology used. Ranked according to the technology development Computer-based Learning (CBL) appeared first, where the learning process centered around the use of the computer. These computers were not necessarily connected into a network, students worked individually, exploring the autonomous way of learning. The spreading of internet and the World Wide Web, the appearance of computer networks gave birth to online learning enabling student to step out of the local learning environment to a virtual one. Online learning strengthens the constructivist way of learning, trains competences and faces learners with popping up challenges. The appearance of Learning Management Systems (LMS) and Learning Content Management Systems (LCMS) required the compliance of such learning materials that enable individual and autonomous learning. At this point the question of how to adjust teacher pre– and in-service training to meet the market demand arises. The booming of web 2.0 applications and social networking, web-based learning (WBL) is becoming more and more popular, enabling on the spot training, permanent accessibility to knowledge, the feeling of belonging to a society, a coven. Personalized Learning environments (PLE), 3D virtual learning environments help students to make learning more personal, more comfortable and more motivating (Fig. 1).

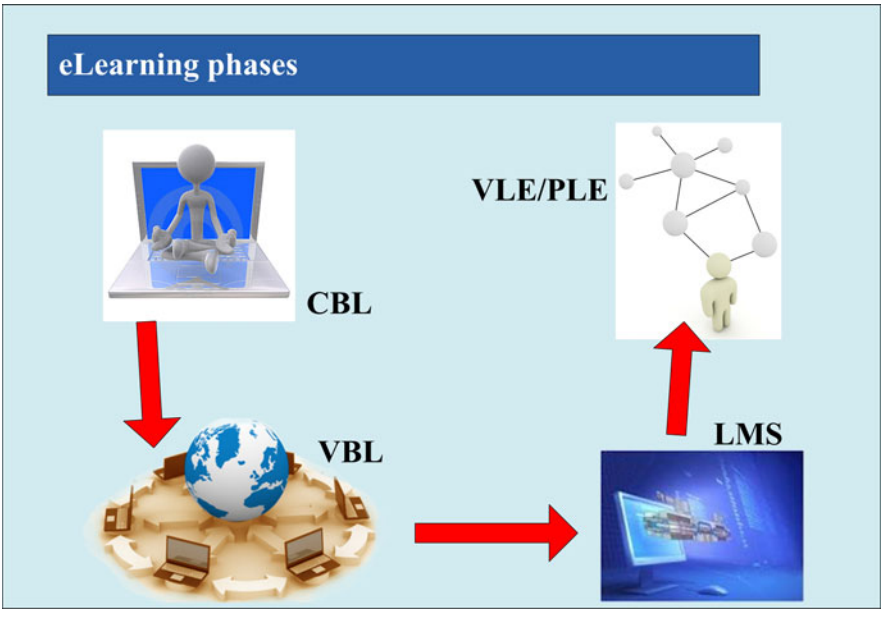


Fig. 1. eLearning phases.

In parallel to a change in the student-student relationship the student-teacher relationship changes as well. When using the internet both teachers and students take on the roles and characteristics of their society in an entirely natural way. They do this without being conscious about it, namely the process is not consciously controlled. Web 2.0. applications, the booming of Facebook, Second Life and other virtual communities and social networking applications force both teachers and students to change their attitude and approach to teaching and learning.

With the help of the above mentioned technologies an entirely new, interactive authentic study environment can be created that to a great extent supports an autonomous, individual type study form, nevertheless enables the development of group and class level courses, and for students it gives the feeling of belonging to a coven. This community regarding its aims and objectives is an entirely self-organizing, self-cleaning, homogenous community that to a great extent might back the study process through the evolving positive team spirit.

3 eLearning Management Tasks

"The generic term "e-learning" or "eLearning" describes any form of electronically supported learning, ranging from the use of learning software on a personal computer to the use of an intranet or the Internet for interaction within networks ("Web-based training")" [3].

There are some activities needed for a successful eLearning course. On the one hand adequate hardware supply including multimedia capable terminals and workstations connected to the internet or an intranet must be installed, a suitable eLearning environment, either applying a Learning Management System, developing or joining a virtual learning environment are needed. Furthermore, a teacher staff trained to develop learning material suitable for web-based learning. The central component of eLearning services and of the creation of virtual classrooms is management. For eLearning-based learning it is crucially necessary to apply such an adequate and well selected integrated Learning Management System that fulfills the requirements of students, teachers as well as administrators simultaneously. This system that is the so-called Learning Management System (LMS) in a traditional sense, however, is only a tool whose main attractiveness for students is not its complexity or simplicity but the fact that they can follow and complete a course in an eLearning form. The services of such learning management systems are also of great importance since these make the system attractive to users; however, the content accessible in the system is at least as important as the offered services.

A good learning management system is capable of managing not just the learning material but the knowledge base and the student administration as well, meets educational, administrative and application requirements. An LMS is such an

information system that is capable of educational process automation, namely it is capable of delivering educational content and source material and is capable of the management of the entire learning process. If we actually analyze the relationships between two participants, namely students and teachers/tutors, these can be summarized as in Table 1.

Table 1. Relations between student and teacher in Learning Management System

Student-Teacher interaction in LMS		
Roles	Teacher	Student
Teacher	professional cooperation, communication	eLearning administration, communication (registration, course-registration, learning, test, exams, consultation, instant communication)
Student	social networking (chat, blog, email, mailing list)	

The services of advanced learning management systems are quite differentiated. The basic functions can be summarized as follows below. A Learning Management System provides the following possibilities (Fig. 2):

- *to students*: registration, enrolment to e-learning courses to have access to e-learning contents, to fill in self-tests, to "sit" for exams, to search contents
- *to teachers/tutors*: uploading eLearning teaching material, audio and visual teaching material, monitoring study progress, making statistics, developing tests and exams, evaluation of results, online consultation,
- *to each system user*: use of blogs, emailing, creation of mailing lists, operation of message boards, creation of self-profile.

Competence management, knowledge level analysis, progress plan, the possibility of virtual classes and resource assignment (place, room, book, teacher/tutor, etc) are all among the services of the most comprehensive learning management systems providing the highest number of services [4]. Most of the systems are self-service ones at the student interface, they enable self-registration thus ensures course accessibility. On the one hand a learning management system is capable of managing registered students, on the other hand it can handle contents and information, furthermore, it is capable of assigning these two to each other, thus it can display learning materials and provide courses.

We must differentiate between learning management systems and content management systems (CMS) since the latter utilizes and enables content editing, classification and display in the learning management system. This means that CMSs are equipped with such editing tools that LMSs do not support. At present such integrated LMSs are, of course, preferred and deployed that support learning content editing in and transfer from and to other systems.

Fig. 2. Typical login-page of Learning Management System.

4 Personalized Learning Environment – A Virtual Option

The development of the next level of study support software systems was triggered by two radical– one social and one technical - changes in the last couple of years.

4.1 The Birth of the "Digital-Generation"

Since the boom of the information technology a new generation has grown up. Different generations have diverse attitude to internet use and lifelong learning. The oldest one, the so-called veterans born before the baby boom regard the internet and the use of computer as a challenge since they met the internet late in the life. The baby boom generation born up to the middle of the 60s' uses the computer and the internet for work and mainly at their workplace. The next generations called XYZ are the children of the digital generation. They gain information they want quickly medium-independently and share and re-interpret this information with their virtual peers. The internet is present in the private life of the Generation X (born between the middle 60s' and the early 80s') but it does not influence and change their lifestyle to a great extent. The Millennium Generation or Net Generation as the Generation Y is also called was born from early 80s' to the early years of 2000 is the first wave of the digital generation. They are the children of the baby boomers They saw the rise of instant communication technologies and the use web 2.0 based websites is part of their life. They already look for

such applications and eager to join social networking and study in peer-oriented environments. This generation knows what it wants, when, where and how it wants it. The youngest generation, Generation Z called also as Generation I or Internet generation was born within the use of the internet. They are exposed to the world wide web as early as on their birthday. The use of instant and text messages, mobile internet devices, cloud computing justifies their nickname "digital natives". The children of this generation usually have an online identity.

Consequently, the generations grown up in the last two decades is not the "Nintendo-generation" that grew up on small manual games, but the "PlayStation-generation" grown up on a much more developed HD quality technology. This is the generation that learnt to use the computer well before they learnt to write, who handles the DVD player better than their parents. This generation lives "on-line", handles the virtual worlds as the extension of the real one. This generation is the one that if goes home from school, logs in the social network and chats with the classmates, while in another window plays a 3D videogame with someone from the other end of the world. Consequently, these students browse much more comfortably in the virtual space, since it is entirely natural to them [5].

4.2 Development of Hardware and Software Resources

In parallel to serious software developments the hardware part of computers developed significantly in the last decade. Multi-core processors and the large clock frequency have multiplied processing speed. The perfectness of audio solutions as well as a drastic change in video cards must be mentioned out of the special hardware solutions, which enable the use of totally real-time life-like graphical effects.

At the same time both system and application software products have gone through a significant change. The designers of online 3D games have developed such technologies with the help of which virtual worlds can be quite realistically realized.

4.3 Emergence of 3D Virtual Learning Environment

Interestingly a further development in study support software systems started with game software products. Developments in computer hardware and software resources as well as the developed graphical software of computer games have enabled the development of 3D Virtual Learning Environments (VLE) thus creating the possibility for the emergence of 3D Personalized Learning Environment.

VLE is a three-dimensional online space where a virtual person, a so-called avatar represents a student. The term virtual refers to an online, internet, or web-based component, the learning component differentiates VLE for educational purposes from other virtual environments and the term environment is simply a location in which users can gather together in a social context. A VLE can range from web sites through virtual classrooms to 3D immersive worlds. A set of web

pages, however, does not constitute a virtual learning environment unless there is social interaction about or around the information [6]. A few examples of virtual social interactions may be instant messaging, discussion boards, emails, blogs, and podcasts.

The avatar that can be parameterized determines the personality, the presence, the place and the interaction of a student in the virtual space within the VLE: students can communicate with each other, with the teacher/tutor and with Computer-based Agents in real time. They can use the digital documents, videos and audio files loaded in the system.

In the 3D VLE students can travel around 3D worlds, walk on the campus, enter the library, look around, and enter a classroom, or even follow a physical experiment.

The first such type highly successful initiative was the development of the system at the University of Texas using Second Life. On the virtual area bought by the university all the 16 campuses of the university were realized. These "private islands are a great way to maintain a safe and controlled learning environment within a world with almost infinite possibilities" as shown in Fig. 3 [6].

Fig. 3. 3D Virtual Education Centre at Texas University in Second Life.

The system of the University of Texas occupies over fifty Second Life islands including 9 campuses and 6 science and medical research campuses. With the realization of the system the university strives to do research in the ways and methods of learning. At the development phase it was an important factor that this virtual solution is much cheaper than similar type developments in the real world. Audiovisual solutions are highlighted, e.g., the voice of the lecturer in a virtual lecture room is actually a real lecture recorded in the real world. The technical background gives space to the development of various virtual experiments.

Other universities and institutes like Open University (Great Britain), the Goethe Institute, Princeton (Fig. 4, Fig. 5) and lots of other educational institutes joined Second Life as well to offer its courses in 3D Virtual Learning Environment. All of these institutes represent themselves on an island creating such learning environment where students find learning comfortable as well as informative.

Fig. 4. Island Map of Open University in Second Life.

Fig. 5. 3D Virtual Education Centre of Open University in Second Life.

5 The Added Value of Virtual Learning Environments

After getting to know some operating VLEs let us examine whether these Virtual Learning Environments have added value and if they do what kind of added value they have compared to earlier developed Learning Management Systems. Certainly, in order to conduct exact examinations, long time data collection and control groups are needed based on which valid and reliable statistical analyses can be pursued. These facts are still not available due to the young age of the initiatives; however, some conclusions can be drawn from experience.

5.1 3D VLE Characteristics

It is without any doubts that a 3DVLE is in compliancy with those primarily game environments which surround the target group, namely students. Consequently, user comfort provided by 3DVLE strengthens the feeling of safety while decreases the threshold at the start of usage, which users must overcome until they become skilled users of the system.

The familiar environment decreases user inhibition and hesitation, as a result of which users navigate more bravely in the system, try out new ways, discover new sections in the system that further motivate them in the further creative usage of such VLEs.

A special characteristic of 3D virtual systems is the use of the avatar, which helps users represent themselves or be creative and hide behind a character to remain unknown, but which helps to project users' wishes, so we can create and form such a character that we would like to be, and we can define and decide what kind of environment and circumstances we would like to live in. Each avatar has its own personal account that maintains personal information for assessment purposes while students' personal identification is not shown to everyone in the game world. The colorization of the everyday life, the access of an arbitrary virtual environment gives the success of Second Life. Thus the idea comes naturally that if either mountain climbing without any danger or even going to different places is possible then why couldn't we go to university and explore the world of knowledge and gain new and interesting knowledge?

In the case of arbitrarily selected software products the software's appearance, its offered effects and its display interface are all outstandingly attractive features that especially in the beginning can grasp user attention and can induce users to start playing and getting acquainted with it. Later some other features of the given product become dominant from the user's point of view. Ease of use, reliable operation and the feature of customization, etc. are all features of interest but at first the product "must be sold" and interest must be raised in the target group. These aspects also strengthen that the 3D VLE solution is advantageous.

5.2 Analyzing 3D VLE from the Aspect of Usability

Comparing VLEs, LMSs and CMSs from the aspect of knowledge material delivery it must be stated that in quite a lot of cases the use of a 3D environment is not necessary to make students understand the material to be learnt. The "traditional" study support systems also enable the display of text-based learning material, to play audio recordings, to view video shots or to incorporate animation in the material to be learnt. Primarily in higher education, where studying and research fields are much more concentrated than in public education, it is not necessary to use 3D technology. In order to present and dress up the learning material some earlier solutions can be accepted and deployed, the application of the 3D technology at all costs is not justified neither by pedagogical nor by didactical reasons.

The situation is different in public education especially in elementary education. Here the attractive appearance, the interestingness, the appealing interface of the teaching material is an important motivating factor; to catch students' attention with spectacular elements is a pedagogical factor. An effective tool of raising interest can be the world of 3D. Especially in the case of the younger generation with the use of the 3D world's fairy elements knowledge can be delivered to youngsters almost unnoticed, knowledge appetite can be induced in students to progress in the tale through the content and reach "the end", thus learning the lesson. The personalized creatures can guide students through the 3D virtual world on the road of "cognition" to the 'spring of knowledge'.

Especially for younger people who spend a substantial portion of their media time in front of a computer it is useful and practical to offer virtual education and gaming systems that strive to deliver the 'lessons' in the form of interesting games. These purpose-built gaming environments can result in maximizing student engagement in learning and can generate rich social interactions. However, we must differentiate between games for education and games for learning in a 3D VLE. Experience shows [6] that in case of public education the ability to control the learning environment has proved to be a critical part of using 3D VLEs. That is why 3D VLEs specifically designed for younger students are closely monitored and controlled. On the other hand a close look at the amount of content and assessments is also necessary expressed as "A meaningful and relevant context provides a springboard for inquiry, information-gathering and sharing, and reflection of theoretical concepts and relationships, and norms and practices" [6].

6 Conclusions

Considering higher education as the target group of our interest it must be stated that "at least 300 universities around the world teach courses and conduct research in Second Life." as Claudia L'Amoreaux at Linden Lab states in [6]. Should we consider higher education institutions as sites of knowledge management and as knowledge intensive service providers in parallel with the fact that the role of knowledge-based services is growing [7] the appearance of these institutions seem reasonable in 3D virtual learning environment where apart from knowledge transfer from the higher education institutions and students, students themselves can share and distribute the knowledge gained. This implies that developing 3D VLEs in higher education seems successful, however its usage in public education provides almost infinite opportunities since the motivating factor is more focused on the younger generation. It must not be forgotten, however, that the learning environment and progress must be much more closely and strictly monitored and controlled in public education than in higher education. Applying virtual game environments students gain useful, practical and actual information, knowledge and competences instead of playing endless games. Virtual environments compared to

traditional learning support environments have a much deeper impact on youngsters. This means that these systems bring much higher added value and benefit for the future generation in the learning process. To sum it up it can be concluded that "the 3D VLE experience is designed to serve as a reinforcement of the concepts learned in the traditional lessons" [6] and not instead.

References

[1] McLuhan, M.: The Gutenberg galaxy, the making of typographic man. University of Toronto Press, Toronto (1962)
[2] Dahl, S.: Communication and culture transformation. European University, Barcelona (2001), http://www.stephweb.com/capstone/capstone.pdf (accessed April 29, 2011)
[3] Anetta, L.A., Folta, E., Klesath, M.: V-Learning – distance education in the 21st century through 3D Virtual Learning Environments. Springer, Berlin (2010)
[4] Gábor, A.: e-learning oktatasmenedzsment keretrendszerek (Learning Management Systems) (2003), http://www.oktopusz.hu/domain9/files/modules/module15/271CFD97067E893.pdf#search=%22ny%C3%ADlt%20forr%C3%A1sk%C3%B3d%C3%BA%20tanulm%C3%A1nyi%20keretrendszer%22 (accessed September 22, 2006)
[5] Längin, D.W., Ackerman, P.A., Lewark, S.: Internet-based learning in higher forestry education. Unasylva 55, 39–46 (2004)
[6] Barkand, J., Kush, J.: GEARS a 3D virtual learning environment and virtual social and educational world used in online secondary schools. Electron. J. e-Learning 7, 215–224 (2009)
[7] Farkas, F., Király, Á.: What makes higher education knowledge-compatible? Acta Polytechnica Hungarica 6, 93–104 (2009)
[8] Tick, A.: A new direction in the learning processes, the road from eLearning to vLearning. In: Proccedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, pp. 359–362 (2011)
[9] Jones, S.: "The Internet goes to College" Pew Internet and American Life Project 2002, Washington DC (2007), http://www.pewinternet.org/pdfs/PIP_College_Report.pdf (accessed September 20, 2007)
[10] Gáti, J., Kártyás, G.: Virtual classrooms for robotics and other engineering applications. In: Proceedings of 19th International Workshop on Robotics in Alpe-Adria-Danube Region, Budapest, Hungary, pp. 481–486 (2010)
[11] Gáti, J., Kártyás, G.: New trend in teaching and learning environments for distance higher education. In: Proceedings of 5th International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, pp. 459–464 (2009)
[12] Lim, C.P.: Spirit of the game: Empowering students as designers in schools? Br. J. Educ. Technol. 39, 996–1003 (2008)
[13] Dillenbourg, P.: Virtual Learning Environments. In: EUN Conference 2000 (2000), http://tecfa.unige.ch/tecfa/publicat/dil-papers-2/Dil.7.5.18.pdf (accessed February 10, 2009)

A Receding Horizon Control Approach to Navigation in Virtual Corridors

Domokos Kiss and Gábor Tevesz

Budapest University of Technology and Economics, Department of Automation and Applied Informatics, Magyar Tudósok krt. 2., 1117 Budapest, Hungary
{domokos.kiss,tevesz}@aut.bme.hu

Abstract. Applications in mobile robotics require safe and goal-oriented motion while navigating in an environment obstructed by obstacles. The dynamic window approach (DWA) to collision avoidance and its different variants provide safe motion among obstacles, although they have the same limitation, namely using an objective function consisting of weighted terms. Different situations require different weights; however, there is no algorithm for choosing them. The Global Dynamic Window Approach with Receding Horizon Control (GDWA/RHC) presented in this chapter is similar to DWA but it uses a global navigation function (NF) and a receding horizon control scheme for guiding the robot. In order to make the calculation of the navigation function computationally tractable it is constructed by interpolation from a discrete function. In addition to that the domain of the navigation function is restricted to a virtual corridor between the start and goal positions of the robot.

1 Introduction

The problem of feasible and safe navigation is one of the most important areas of mobile robotics research. For autonomous robots reaching a goal position while avoiding collisions with obstacles is a crucial task. Many solutions to this problem were proposed in the last decades [1, 2, 3]. There are on the one hand global methods for motion planning that rely on a priori environment information, on the other hand local obstacle avoidance methods based on sensory data collected during operation. Global planning methods guarantee feasible or even optimal paths to the goal, but are computationally complex. Local (or reactive) obstacle avoidance methods utilize sensory data collected during operation and are in most cases computationally more efficient hence well-suited for real-time applications. However, in most cases there is no guarantee for actually reaching the goal. This chapter deals with the dynamic window approach (DWA) of obstacle avoidance which is a reactive method by nature. Keeping the basic idea, we reformulate the original method by taking global information about free space connectivity into account and applying a receding horizon control (RHC) scheme to eliminate shortcomings

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 175–186.
springerlink.com

of the original DWA and its variants. Information regarding free space connectivity is managed by a global navigation function (NF) which is evaluated using a recursive algorithm. To reduce the computational cost of NF calculation a virtual corridor is defined leading to the goal and the function is evaluated only on the domain of this corridor.

The organization of this chapter is as follows. In Section 2 we give a short survey of collision avoidance methods based on the idea of dynamic window approach. Section 3 describes our approach using a global navigation function and receding horizon control (RHC). In Section 4 we present the concept of virtual corridors which has the role of reducing the computational cost of navigation function evaluation. The results are summarized in Section 5.

2 Dynamic Window Approach to Obstacle Avoidance

The dynamic window approach, proposed by Fox et al. [4] has advantages compared to former local obstacle avoidance methods including Virtual Force Field [5] and Vector Field Histogram [6], which are based on potential fields. The dynamic window approach has the difference in comparison to these that control commands for the robot are carried out directly in the space of velocities. The word "velocity" means here a two-dimensional vector (v,ω) which consists of the translational velocity v and the angular velocity ω of a nonholonomic robot equipped with synchro drive or differential drive. These robot models have the same motion equation which is shown in Section 3.1. This approach takes the dynamics of the robot into account by reducing the search space to velocities reachable in a short time interval. The set of these velocities is the so called *dynamic window*. In addition to that only velocities are considered that are safe with respect to the obstacles (admissible velocities). To choose from admissible velocities an objective function is evaluated and maximized. This objective function is a weighted sum of three terms:

$$G(v,\omega)=\sigma\left(\alpha\cdot\mathrm{heading}(v,\omega)+\beta\cdot\mathrm{dist}(v,\omega)+\gamma\cdot\mathrm{vel}(v,\omega)\right) \tag{1}$$

where heading(v,ω) is a measure of going into the direction of the goal, dist(v,ω) is the smallest distance to the next obstacle along the circular path segment belonging to (v,ω) and vel(v,ω) is simply the projection of (v,ω) on the translational velocity v to favor high motion speeds. The ranges of these functions are normalized between 0 and 1. The corresponding weights are α, β and γ. σ is a smoothing function.

Experimental results presented in [4] show that the dynamic window approach to collision avoidance yields a fast and safe robot motion. However, since the DWA and the above mentioned other methods are based on local decisions without taking connectivity information of the free space into account, the robot can

get trapped in local minima (i.e. it can stop afar off the goal point) or enter a limit cycle that prevents reaching the goal. Another problem of the DWA is that there is no algorithm for choosing the objective function weights. Different situations require different weights to ensure successful motion. There are also scenes where DWA fails independent of weighting.

Brock and Khatib [7] extended the dynamic window approach to the case of holonomic robots and address the problem of local minima by introducing a navigation function (NF) which is a local minima-free function defined on the discretized configuration space, having a unique minimum at the goal. The modified objective function in [7] has the following form:

$$\Omega_g(\mathbf{p},\mathbf{v},\mathbf{a}) = \alpha \cdot \mathrm{nf1}(\mathbf{p},\mathbf{v}) + \beta \cdot \mathrm{vel}(\mathbf{v}) + \gamma \cdot \mathrm{goal}(\mathbf{p},\mathbf{v}) + \delta \cdot \Delta\mathrm{nf1}(\mathbf{p},\mathbf{v},\mathbf{a}) \qquad (2)$$

where **p** is the current position, **v** and **a** are the desired velocity and acceleration, respectively. nf1(**p**,**v**) increases if the velocity is aligned with the NF gradient, vel(**v**) favors high velocities (if far from the goal), goal(**p**,**v**) is a binary function (one if the trajectory is expected to pass through the goal area), and Δnf1(**p**,**v**,**a**) is the decrease of NF value. The problem of local minima is eliminated in many cases through the global distance information represented by the NF terms, but not at all times. It is shown by Ogren and Leonard [8] that limit cycles can evolve if the velocity term outweighs the NF terms. They reformulate the dynamic window approach as a model predictive control (MPC) problem (also referred to as receding horizon control, RHC). They assume a holonomic robot model that can be considered as a double integrator in the plane $\ddot{r} = u$, where $r \in R^2$ is the position of the robot. Thus the control u is the acceleration in contrast with [4] (where the velocity is the control signal). The state vector of the holonomic robot is chosen to $x = (r, \dot{r}) = (r_x, r_y, \dot{r}_x, \dot{r}_y)$. The acceleration u and the velocity $\dot{r}$ are bounded. The objective function in [8] has a different form than in [4] and [7]:

$$V(x) = \frac{1}{2}\dot{r}^T\dot{r} + \frac{k}{\sqrt{2}}NF(r) \qquad (3)$$

where k is a positive constant. The first term corresponds to the kinetic energy, the second term is proportional to a local minima free navigation function, similar to the one in [7]. The objective function has to be minimized – in contrast with [4] and [7], where the maximum had to be obtained. Ogren and Leonard show that for a given set of u the system is stable in the Lyapunov sense (but not asymptotically). However, this property is not enough to ensure convergence, since the case of $\|\dot{r}\| = 0$ afar off the goal cannot be excluded. This fact is worked around by a timeout condition in their proposed algorithm.

3 The GDWA/RHC Method

An obstacle avoidance approach is presented in this section that assumes a non-holonomic robot model similar to [4] and applies a receding horizon control scheme to obtain translational and angular velocities as controls. An objective function similar to the one in [8] is minimized with the difference that it does not contain a kinetic energy term, hence the problem of zero velocity afar off the goal disappears. The method is called Global Dynamic Window Approach with Receding Horizon control.

3.1 Robot Model

We consider a nonholonomic robot model that moves on a two-dimensional plane (nonholonomicity means that it has less control inputs than degrees of freedom). The shape of the robot is assumed to be circular. The state vector is given by

$$\mathbf{x}(t) = \begin{bmatrix} r_x(t) & r_y(t) & \varphi(t) \end{bmatrix}^T \tag{4}$$

where $r_x(t)$ and $r_y(t)$ represent the instantaneous position and $\varphi(t)$ the orientation of the robot. We use velocity motion model [9] hence the control vector is as follows:

$$\mathbf{u}(t) = \begin{bmatrix} v(t) & \omega(t) \end{bmatrix}^T \tag{5}$$

where $v(t)$ and $\omega(t)$ stand for the magnitude of translational and angular velocity, respectively. The system has 3 degrees of freedom (dimension of $\mathbf{x}(t)$) and two control inputs (dimension of $\mathbf{u}(t)$), thus it is underactuated. The motion equation is given by

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t)\right) \cdot \mathbf{u}(t) \tag{6}$$

where $\mathbf{f}()$ is a nonlinear function. In matrix form:

$$\underbrace{\begin{bmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{\varphi} \end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} \cos(\varphi(t)) & 0 \\ \sin(\varphi(t)) & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{f}(\mathbf{x}(t))} \cdot \underbrace{\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}}_{\mathbf{u}(t)} \tag{7}$$

The kinematic constraint, namely the translational velocity points always in the heading direction $\varphi(t)$, can be easily seen. The translational and angular velocities are bounded:

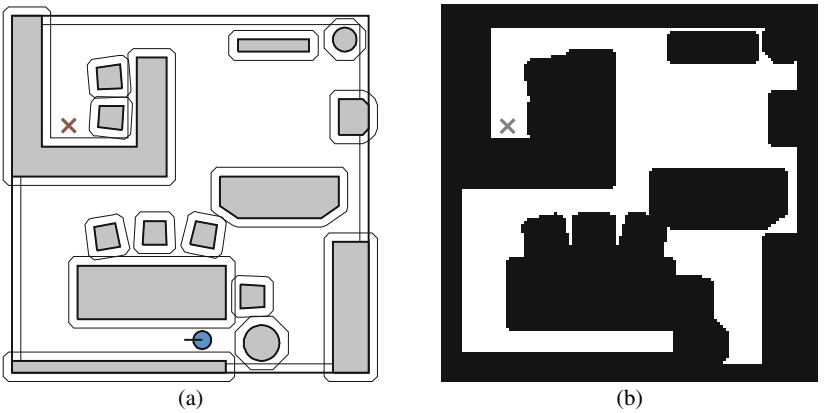
$$\begin{aligned} v_- \le v(t) \le v_+ \\ \omega_- \le \omega(t) \le \omega_+ \end{aligned} \tag{8}$$

The dynamic properties of the robot are described by maximal translational and angular accelerations and decelerations:

$$\begin{aligned} a_- \le \dot{v}(t) \le a_+ \\ \beta_- \le \dot{\omega}(t) \le \beta_+ \end{aligned} \tag{9}$$

3.2 World Model

The environment of the robot is represented by a hybrid model. The obstacles are given by simple geometric shapes (polygons and circles) in the continuous two-dimensional Euclidean space. The model of a simple indoor environment is shown in Fig. 1a. Since we assume a circle-shaped robot, the workspace can be transformed into a configuration space by augmenting obstacles by the robot radius. In the resulting reduced free space model the robot is represented by a single point. This geometric representation is used for collision detection in the algorithm. In order to easily define navigation function (described in Section 3.3) an occupancy grid representation is also introduced. The 2D-plane is quantized equidistant and a binary occupancy function is defined on the discrete points (see Fig. 1b).



(a) (b)

Fig. 1. Hybrid world model. (a) Geometric representation (b) Occupancy grid

Note that the above mentioned hybrid world model is used only for simplicity. There are many robotic applications where no a priori geometric representation of the environment is available. In these cases the occupancy grid has to be built first based on observations (e.g. using range sensors) and the geometric model of obstacles has to be derived from it. The problem of map building goes beyond the scope of this chapter. For further information about mapping methods see [9].

3.3 Navigation Function

The global property of the proposed approach is ensured by the utilization of a navigation function (NF). The NF is a scalar-valued function representing the distance from the goal point. It is defined as follows.

Definition (Navigation function): Let the continuous configuration space of a mobile robot moving on a plane be denoted by $C \subset R^2$. The set of obstacle points is denoted by $O \subset C$, the set of free space points by $F = C \setminus O$. The goal position to be reached is $\mathbf{r}_g \in F$. Let F be a connected set. A real-valued function defined on F with exactly one minimum at $\mathbf{r}_g$ is called a *navigation function* (NF).

Proposition (Construction of NF): A navigation function fulfilling the above definition can be constructed by the following procedure:

1. Replace the continuous configuration space with a square grid of discrete points such that the goal coincides with a grid point. Let the grid constant be denoted by ℓ. The set of grid points is $C_d \subset R^2$.
2. Denote the set of grid points belonging to obstacles by $O_d \subset C_d$. Denote the set of free grid points by $F_d = C_d \setminus O_d$ (discrete free space).
3. Let the navigation function value be zero at the goal point. Starting from there, the function value of every free adjacent point is to be increased by the corresponding value shown in Fig. 2. This has to be done only if the adjacent point is unexplored or its former value is greater than the one to be actually given. Continue this step until all points of F_d have been discovered or the starting position of the robot has been reached. The resulting function values represent the lengths of the shortest paths leading to the goal point, taking sideward and diagonal steps into account. Thus a monotonically increasing discrete function has been produced.

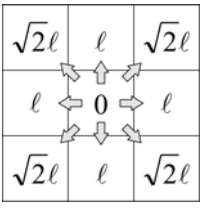
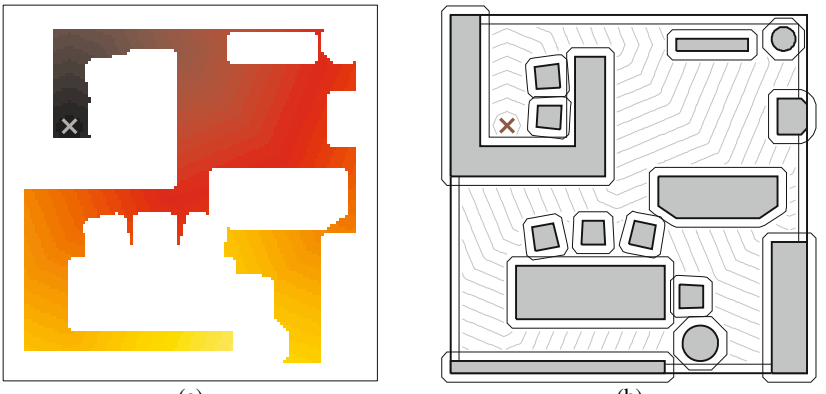


Fig. 2. NF calculation scheme



(a) (b)

Fig. 3. Example of NF, illustrated by (a) gradient and (b) level curves

4. The grid points belonging to F_d divide the continuous free space F into squares, and the function values have already been determined at the corners of these squares. It can be stated that all inner points of the squares belong to F. Although there can be free space parts outside of these squares (semi-occupied squares). These free space parts are simply excluded from F (we consider the semi-occupied squares as fully occupied) and the resulting modified continuous free space is denoted by F'.
5. To be tractable for continuous applications, the function values in all points of F' should be determined. In the simplest case a zero-order interpolation could be applied. This simple method is unsuitable for our purpose because the resulting piecewise constant function has "plateaus" thus there can be motions that do not lower the function value while approaching the target. Therefore we use bilinear interpolation inside the squares.

The above described algorithm produces a navigation function fulfilling the given definition. A proof of this proposition is given in [10]. In Fig. 3 an example is depicted showing the NF values by shades (Fig. 3a) as well as plotted as level curves onto the geometric representation of the environment (Fig. 3b).

3.4 Receding Horizon Control Scheme

The task of obstacle avoidance is considered as a constrained optimization problem. The objective function which has to be minimized is a navigation function (NF) constructed by the algorithm described above. At time instant t we shall choose the control $\mathbf{u}(\cdot)$ which minimizes the NF value at the end of a time horizon of length T, and apply $\mathbf{u}(t)$ as control input:

$$\mathbf{u}(\cdot) = \arg\min_{\mathbf{u}(\cdot)} \mathrm{NF}\big(r_x(t+T), r_y(t+T)\big) \tag{11}$$

where $r_x(t + T)$ and $r_y(t + T)$ can be derived from the motion equation (7) by integration:

$$\begin{aligned} r_x(t+T) &= r_x(t) + \int_t^{t+T} v(\tau)\cdot\cos\left(\varphi(t) + \int_t^{\tau}\omega(\vartheta)d\vartheta\right)d\tau \\ r_y(t+T) &= r_y(t) + \int_t^{t+T} v(\tau)\cdot\sin\left(\varphi(t) + \int_t^{\tau}\omega(\vartheta)d\vartheta\right)d\tau \end{aligned} \tag{12}$$

The NF values on the grid points are computed and stored off-line and the actual $\mathrm{NF}\left(r_x(t+T), r_y(t+T)\right)$ values are interpolated during execution according to (10). Other constraints concerning the controls arise from the velocity (8) and acceleration (9) bounds:

$$\begin{aligned} v_- \le\ & v(\tau) \le v_+ \\ \omega_- \le\ & \omega(\tau) \le \omega_+ \\ v(t) + \tau\cdot a_- \le\ & v(\tau) \le v(t) + \tau\cdot a_+ \\ \omega(t) + \tau\cdot \beta_- \le\ & \omega(\tau) \le \omega(t) + \tau\cdot \beta_+, \quad \forall\tau\in[t, t+T] \end{aligned} \tag{13}$$

Safety is ensured by a constraint of admissible controls. A control $\mathbf{u}(\cdot)$ is admissible, if

$$\begin{bmatrix} r_x(\tau) \\ r_y(\tau) \end{bmatrix} \notin O, \quad \forall\tau\in[t, t+T]\cup[t+T, t+T+T_{brake}] \tag{14}$$

where O is the set of obstacle points in the configuration space. We assume that a maximal braking control $\mathbf{u}^*(\cdot)$ is applied in the time interval $[t + T, t + T + T_{brake}]$ where T_{brake} is the time needed to halt the robot from the velocity at $t + T$. This means that only those controls are admissible those

- do not cause a collision inside the horizon and
- allow the robot to stop safely beyond the horizon.

To make the problem more tractable and computationally efficient, we assume discrete time and piecewise constant velocities. The horizon length is $T = kT_s$, where T_s denotes sampling time. Piecewise constant velocities mean that $v(\tau) = v(t)$ and $\omega(\tau) = \omega(t)$ for all $\tau \in [t, t + T_s]$. In this case (12) and (13) are simplified:

$$\begin{aligned} r_x(t+kT_s) &= r_x(t) + \frac{v(t)}{\omega(t)}\left[\sin\left(\varphi(t)+\omega(t)\cdot kT_s\right) - \sin\left(\varphi(t)\right)\right] \\ r_y(t+kT_s) &= r_y(t) - \frac{v(t)}{\omega(t)}\left[\cos\left(\varphi(t)+\omega(t)\cdot kT_s\right) - \cos\left(\varphi(t)\right)\right] \end{aligned} \tag{15}$$

$$\begin{aligned} v_- &\le v(t+T_s) \le v_+ \\ \omega_- &\le \omega(t+T_s) \le \omega_+ \\ v(t)+T_s\cdot a_- &\le v(t+T_s) \le v(t)+T_s\cdot a_+ \\ \omega(t)+T_s\cdot \beta_- &\le \omega(t+T_s) \le \omega(t)+T_s\cdot \beta_+ \end{aligned} \tag{16}$$

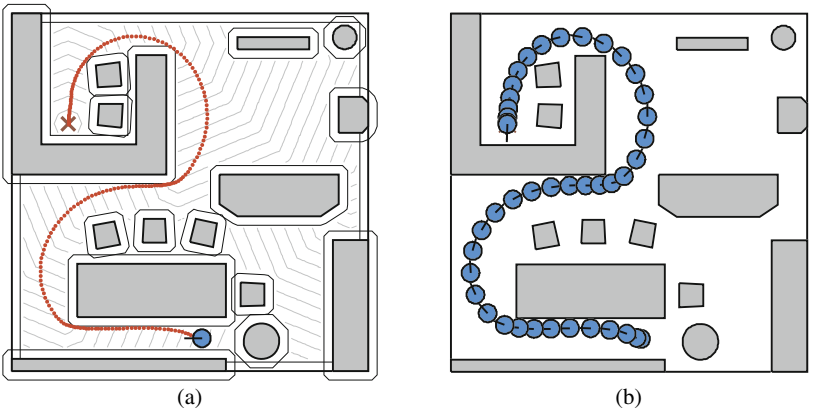
Note that the last two inequality of (16) define the actual *dynamic window* at time instant t.

During the optimization process we have to choose a control value at every discrete time instant which satisfies (14) and (16) and minimizes (11). To do this, the velocity space is also quantized, which results in a countable set of controls. The velocity configurations (v,ω) being admissible and inside the actual dynamic window are called *candidate controls*. We take each candidate control value from the dynamic window and apply it virtually to the robot for the duration of kT_s using (15). The control resulting in the smallest NF value at $t + kT_s$ will be chosen and applied to the robot in the time interval $[t, t + T_s]$.

Note that to ensure a smooth halt at the target point the horizon T must be chosen long enough to allow stopping inside the horizon.

3.5 Simulation Results

The effectiveness of the proposed method was tested by simulations. The resulting path of a robot in the example environment is depicted in Fig. 4. The trajectory of the center point and the reduced free space is illustrated in Fig. 4a together with NF levels. It can be seen that the robot always strives to move in directions of NF descent. Fig. 4b shows the original workspace (non-augmented obstacles) and the robot itself. The smoothness of the resulting path is ensured by taking the dynamic limitations of the robot into account.



(a) (b)

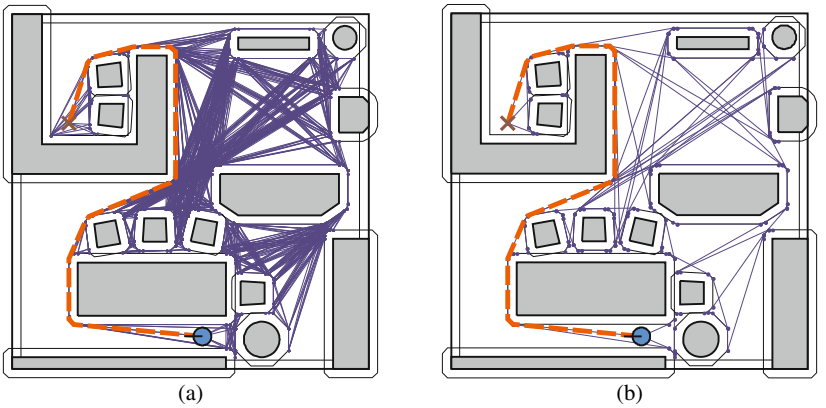
Fig. 4. Simulated path of a robot using GDWA/RHC

4 Virtual Corridors

The presented method has a notable shortcoming, namely the NF is calculated for almost the whole configuration space, including regions those will never be crossed by the robot. If the algorithm described in Section 3.3 is implemented as breadth-first search which works like "wave propagation", the NF building process stops when the start position of the robot is reached by the "wave front". The number of visited grid cells is square proportional to the length D of the shortest path between start and goal points thus the running time can be expressed as $O(D^2)$ [13]. This prevents real-time application in case of larger and changing environments where the NF should be recalculated on-line.

A better solution would be to calculate NF values only in a given neighborhood of the ideal path. In other words the NF calculation should be restricted to a virtual corridor along the ideal path. If the width of this corridor is chosen to a constant value then the number of cells belonging to it depends on the path length. Thus the running time of NF calculation becomes $O(D)$.

In order to define a virtual corridor the shortest path leading from the start position to the goal point has to be determined first. We use the geometric environment representation and a *visibility graph* [1] technique to obtain the path. By definition, the set of vertices of the visibility graph comprises all the obstacle polygon vertices as well as the start and goal points. Edges of the graph are connecting vertices those are "visible" to each other, in other words the line segment connecting them does not intersect any obstacle. After building this graph the problem of determining the shortest possible path from the start to the goal point can be reduced to a minimum-cost-path finding problem in the visibility graph [1, 11]. Since the geometric environment representation is used, the computational overhead of graph building and shortest path search is independent of environment scaling. Fig. 5a illustrates the visibility graph for the example environment.



(a) (b)

Fig. 5. Visibility graph (a) and tangent graph (b) for determining the shortest path

It can be seen that the visibility graph has many edges that can never form part of the shortest path. An improvement of the visibility graph algorithm is known as the tangent graph that excludes many unnecessary edges from the graph. Edges those are not "co-tangent" with respect to obstacles are excluded. To determine whether an edge is "co-tangent" the line joining the two vertices has to be examined. If an internal angle of any of the two respecting polygons is divided by the line, the corresponding edge is not "co-tangent" to those polygons. For an arbitrary pair of polygons it is shown in [12] that there exist only four co-tangent lines. The tangent graph representation is considerably more efficient than the visibility graph for the same obstacle distribution (see Fig. 5b). Assuming two obstacle polygons with n and m vertices, the visibility graph can have in worst case nm edges between them, while the tangent graph has at most four [11].

After determining the shortest path, it should be dilated to a corridor in order to be able to define a navigation function on it. The border of the virtual corridor is treated as an obstacle border in the GDWA/RHC algorithm. The breadth of the corridor should be chosen carefully. If it is chosen too narrow, it can result in slow motion since in this case the robot cannot move according to its dynamic abilities. An example of the effect of using a virtual corridor is shown in Fig. 6. The example scene already seen in Fig. 4 is depicted without (Fig. 6a) and with (Fig. 6b) the virtual corridor. The NF is illustrated by its level curves. The GDWA/RHC algorithm has been applied to both scenes and it can be seen that the resulting trajectories are almost the same.

The use of virtual corridors considerably reduces the computational cost of navigation function building. For example, if the environment is scaled up by a factor $r > 1$ (which means the ideal distance D also grows to rD), then the cost without virtual corridor becomes r^2 times bigger. At the same time, with virtual corridors the cost grows only by the factor r while having a constant (scale independent) overhead of tangent graph building and shortest path search. A numerical example is given in [13].

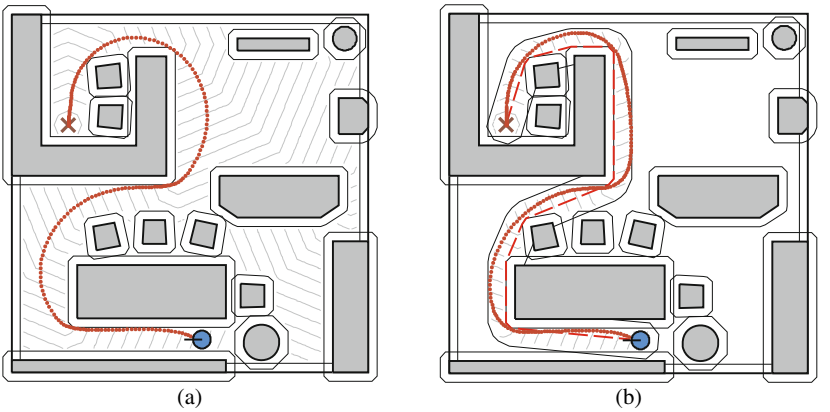


Fig. 6. Reduction of NF domain by using a virtual corridor

5 Conclusions

A receding horizon control based collision avoidance method is proposed for nonholonomic mobile robots, based on the dynamic window approach. To avoid local minima traps and limit cycles during motion, a navigation function is used that incorporates free space connectivity information into the optimization process. There are no weights in the objective function hence the problem of tuning them disappeared. Addressing the problem of NF calculation cost in wide environments the concept of virtual corridors is introduced. A linear relationship between NF building cost and start-goal distance is obtained. The price for that has to be paid by an additional tangent graph building and searching process based on the polygonal model of obstacles. In case of a changing environment the virtual corridor and the NF on it has to be calculated repeatedly.

Acknowledgments. This work is connected to the scientific program of the "Development of quality-oriented and cooperative R+D+I strategy and functional model at BUTE" project. This project is supported by the New Hungary Development Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-000).

References

[1] Latombe, J.: Robot Motion Planning. Kluwer Academic Publishers, Boston (1991)
[2] Laumond, J.P.: Robot Motion Planning and Control. LNCIS, vol. 229. Springer, Heidelberg (1998)
[3] LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
[4] Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. IEEE Robot. Autom. Mag. 4, 23–33 (1997)
[5] Koren, Y., Borenstein, J.: Potential field methods and their inherent limitations for mobile robot navigation. In: Proceedings of IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, pp. 1398–1404 (1991)
[6] Borenstein, J., Koren, Y.: The vector field histogram – fast obstacle avoidance for mobile robots. IEEE Trans. Robot. Autom. 7, 278–288 (1991)
[7] Brock, O., Khatib, O.: High-speed navigation using the global dynamic window approach. In: Proceedings of IEEE International Conference on Robotics and Automation, Detroit, MI, USA, pp. 341–346 (1999)
[8] Ogren, P., Leonard, N.E.: A convergent dynamic window approach to obstacle avoidance. IEEE Trans. Robot. 21, 188–195 (2005)
[9] Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2005)
[10] Kiss, D., Tevesz, G.: A receding horizon control approach to obstacle avoidance. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, pp. 397–402 (2011), doi:10.1109/SACI.2011.5873035
[11] Dudek, G., Jenkin, M.: Computational Principles of Mobile Robotics. Cambridge University Press (2010)
[12] Liu, Y.H., Arimoto, S.: Path planning using a tangent graph for mobile robots among polygonal and curved obstacles. Int. J. Robot. Res. 11, 376–382 (1992)
[13] Kiss, D., Tevesz, G.: Efficient calculation of navigation functions for obstacle avoidance. In: Proceedings of Automation and Applied Computer Science Workshop, Budapest, Hungary, pp.128–139 (2011)

High Speed Stereo Vision Based Automotive Collision Warning System

Adrian Leu[1], Dorin Aiteanu[2], and Axel Gräser[1]

[1] Institute of Automation, University of Bremen, Otto-Hahn-Allee 1, Bremen, Germany
{aleu,ag}@iat.uni-bremen.de
[2] Signum Bildtechnik GmbH, Rüdesheimer Str. 21, München, Germany
aiteanu@signumbt.de

Abstract. This chapter presents a high speed, low latency stereo vision based collision warning system for automotive applications. The system uses two high speed cameras running at 100 fps and achieves latency below 0.1s by using an Nvidia Tesla C1060 GPU for accelerating computational expensive algorithms. From each pair of captured stereo images a disparity map is computed using the block matching algorithm, which is afterwards segmented in order to detect different objects in the scene. This segmentation is performed using a novel segmentation method based on the pixels' intensity value and their connectivity. For each detected object its distance to the front of the vehicle is computed and the degree of danger is estimated by the collision warning module. Extensive experiments show that the presented system delivers reliable results for object detection as well as precise results in terms of estimated distance to the detected objects.

1 Introduction

Advanced driving assistance systems (ADAS) are systems which help the driver in its driving process and are present in most personal vehicles. Their purpose is to increase car safety and implicitly road safety as well. Some examples are adaptive cruise control (ACC), lane departure assistant, lane change assistant, intelligent speed adaptation (ISA) collision warning and the well known in-vehicle navigation system, which is typically a global positioning system (GPS).

Among advanced driving assistance systems, collision warning is one of the most discussed topics in the automotive industry. Various systems have been developed and tested, which make use of different types of sensors. The approaches can be divided by the types of sensors they use. Some approaches use passive sensors, like monocular cameras [1] and stereo cameras [2] while others use active

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 187–199.
springerlink.com

sensors like laser scanners [3, 4] and radars [5]. The main difference between these types of sensors is that passive sensors sense naturally available energy, like the reflected sunlight, while active sensors provide their own source for illumination. Since each sensor has advantages and disadvantages, some sensor fusion approaches try to make use of the benefits brought by merging the data obtained from different sensors, for example by combining a stereo camera with a radar [6].

The sensor used in this chapter is a stereo camera. The advantage of using a passive sensor is that no special illumination source is required. Stereo cameras also bring the advantage that the complete scene is imaged and three-dimensional (3D) data can be obtained, compared to non-visual tracking systems like radar, which give 2D coordinates. This additional data can be used to classify objects.

This chapter is organized as follows: Section 2 presents the layout of the system and system specifications; Section 3 introduces the stereo camera system and its calibration; Section 4 explains the disparity map computation; the novel disparity map segmentation method is presented in Section 5; Section 6 introduces the collision warning module; Section 7 presents parallelization of presented algorithms, together with the GPU implementation using CUDA; in Section 8 the results are presented and discussed and Section 9 contains the conclusion and outlook.

2 System Layout and Specifications

The system presented in this chapter is based on the system introduced by the authors in [7]. The main difference is replacing the Bumblebee XB3 camera with two pco.1200 s high speed cameras [8]. Fig. 1 a) shows the pco.1200 s camera, courtesy of PCO AG, Kelheim, Germany. Both cameras are set to capture 100fps at a resolution of 1280x480 pixels and a pixel depth of 10 bit, monochrome.

The reason of using these new cameras is that a higher frame rate (100fps compared to 15fps of the Bumblebee XB3) allows detecting dangerous objects quickly enough for a vehicle speed of up to 50 km/h. It also opens the possibility of a reliable tracking of objects among different consecutive frames, which enables the computation of the objects' path and allows a more accurate prediction to be made of whether a collision is likely to occur in the near future.

The 10 bit dynamics of the cameras offers, besides the exposure time, an additional parameter which can be controlled in order to obtain images of good contrast in different illumination conditions. The exposure time is automatically adapted by analyzing the image histogram in each frame in order to ensure a good exposure. The 10 bit images are afterwards scaled to 8 bit for further processing by selecting the relevant histogram region and fitting it to an 8 bit domain.

Rain and dirt can influence the performance of this system, so a clean view of the scene has to be ensured. Assuming the driver uses the windscreen wipers and other tools to ensure a good visibility for him, the ideal position for the stereo

camera system is behind the windscreen, since it automatically gets a clean view as well. For this purpose a special support has been constructed, allowing easy mounting and unmounting of the cameras inside the vehicle. Fig. 1 b) shows this special construction with cameras installed behind the test vehicle's windscreen.

a) b)

Fig. 1. Camera pco.1200 s (a) and Camera placement inside the vehicle (b).

The goal of this system is to warn the driver if a high probability arises for a collision to occur. A frontal collision is defined by a contact between the front bumper of the car and another object. Therefore the position of the camera system with respect to the front bumper is very important in order to allow the computed distances from the camera to the objects to be adjusted.

As can be seen in Fig. 2, the camera coordinate system has its origin in the center of the left camera while the vehicle coordinate system has its origin below the center of the front bumper, on the road, as usual for automotive applications.\

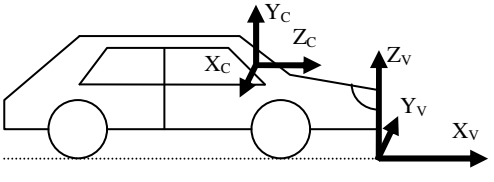


Fig. 2. Relationship between the camera coordinate system and the vehicle coordinate system.

The translations between the camera coordinate system and the vehicle coordinate system are: T_x = -0.30m, T_y = -1.20m and T_z =1.65m , where T_x, T_y and T_z are translations along the x, y and z axes of the camera coordinate system. The axes orientations are: X_C is parallel to Y_V, Y_C is parallel to Z_V and Z_C is parallel to Y_V.

3 Stereo Camera System Properties and Calibration

The stereo camera system presented in this chapter contains two single cameras, which are arranged so that their optical axes are parallel, as can be seen in Fig. 3.The

distance between the two cameras is 0.456m. For each camera two coordinate systems can be defined: an image coordinate system *(u,v)* and a camera coordinate system *(x, y, z)*. These two types of coordinate systems group the camera parameters into intrinsic (focal length *f* and the principal point *c*) and extrinsic parameters (rotation and translation of the cameras). The camera parameters determine the way a 3D point is projected onto the image plane.

A 3D point cannot be obtained from a single image just by knowing the image coordinates of its projection. However, if two cameras are looking at the point *A*, reconstruction of the 3D point can be performed using epipolar geometry [9].

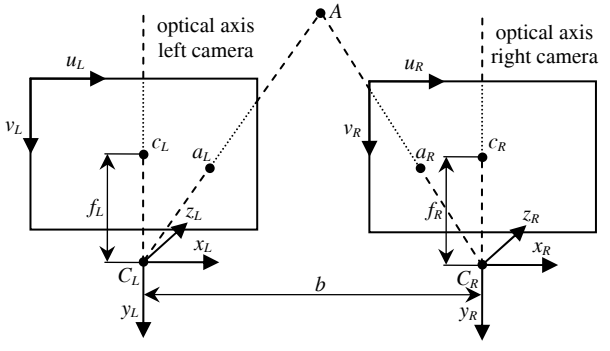


Fig. 3. Stereo camera system made up of left and right camera and the twoimage and camera coordinate systems.

Retrieving the original 3D point is possible if the camera intrinsic and extrinsic parameters are known. The procedure of determining the camera parameters, also known as camera calibration, generally requires a big set of 3D points together with their corresponding projections on the image. Tsai's camera calibration method requires a planar pattern of well known shape and size to be imaged, e.g. a chessboard [10]. With enough images of the pattern, in different poses, the camera parameters can be uniquely identified. In this project the calibration of the stereo camera system has been performed using 20 image pairs of a chessboard.

An important aspect is measuring the calibration accuracy, which is performed in two steps. The first test is used to measure the accuracy of each camera's computed intrinsic and extrinsic parameters. The test consists in projecting the 3D coordinates of all corner points on the image plane and summing the distances between them and the detected corners. The second test uses epipolar geometry and the fact that correspondence points must lie on the epipolar line. The sum of distances between each of the detected corners in the right image and the corresponding epipolar line is computed. Ideally both sums are zero if the calibration was successful.

4 Stereo Rectification and Disparity Map Computation

For a general stereo camera setup the correspondent point in the right image for an image point in the left image lies along the correspondent epipolar line [9]. If the optical axes of the two cameras are parallel, the epipolar lines will be parallel to the image lines, therefore simplifying the process of finding the correspondent point. If additionally the image planes of the two cameras lie in the same plane (the focal lengths of the two cameras are equal), the image lines, on which a point in the left image and its correspondent point on the right image lie, coincide.

In this system, the cameras have the same focal length and parallel optical axes, but due to mechanical misalignments stereo calibration has to be performed in order to meet above condition. This allows dense disparity maps to be computed, which are very useful for 3D reconstruction of the scene. After calibrating the cameras, each image pair is stereo rectified, so that for each point in the left image, its correspondence point in the right image lies on the same image line. An illustration of the effect of stereo rectification can be seen in Fig. 4.

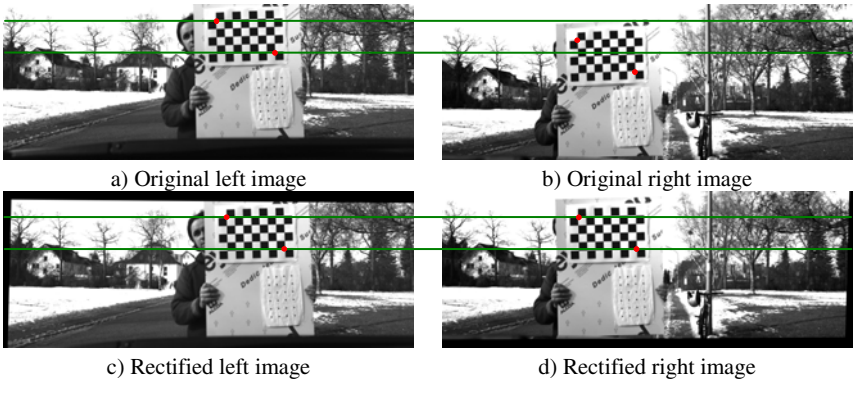


Fig. 4. Effect of stereo rectification: original image pair (a, b) and rectified image pair (c, d).

For constructing the disparity map, usually the left image from a stereo camera is considered to be the reference image and the right image is the correspondent image. The disparity map computation is based on solving the stereo correspondence problem and on calculating the differences in coordinates of corresponding points in stereo images [9]. For a given scene point $A(X, Y, Z)$, the corresponding points in the left and right image are respectively $a_L(u_L, v_L)$ and $a_R(u_R, v_R)$. With known coordinates of the corresponding points, the disparity is defined as:

$$d = u_L - u_R \tag{1}$$

where d is the disparity and u_L and u_R are coordinates of the corresponding points.

Disparity maps are represented as gray-scale images, where pixel intensity represents the disparity of a point in the left image having the same coordinates. In a disparity map, dark pixels belong to objects that are further away from the camera, while brighter pixels belong to objects that are closer to the camera.

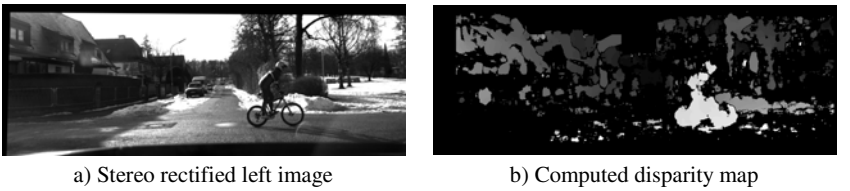
If the disparity value for a point $a_L(u_L,v_L)$ is known, its corresponding 3D location $A(X, Y, Z)$ with respect to the stereo camera coordinate system is given by:

$$X = \frac{b \cdot (u_L - u_{cL})}{d} \quad Y = \frac{b \cdot (v_{cL} - v_L)}{d} \quad Z = \frac{b \cdot f}{d} \tag{2}$$

where f is the focal length of the left camera, b is the base line of the stereo camera, $c(u_{cL}, v_{cL})$ is the principal point of the left camera and d is the disparity value.

Various methods exist for solving the stereo correspondence needed for disparity map computation, differing mainly in computation speed and performance. For this project the block matching algorithm was selected, due to its good performance at relatively low computational expense [11]. This algorithm uses blocks of pixels for solving the correspondence problem. The blocks are always odd sized, having the currently processed pixel in the center of the block. The usage of blocks for finding the correspondent pixel is intended to overcome the problem of ambiguity. This problem occurs when for one pixel in the reference image there is more than one pixel in the correspondent image, on the corresponding image line, which has the same pixel value as the considered pixel in the reference image.

Fig. 5 illustrates disparity map computation using block matching.



a) Stereo rectified left image b) Computed disparity map

Fig. 5. Disparity map computation using block matching algorithm.

The downside of using blocks is that all pixels in one block are considered to have the same disparity value. Therefore the selection of the correct block size is very important since a too small block might not solve the ambiguity problem, while a too large block might lead to not finding a match at all and also needs a higher processing time. As can be seen in Fig. 5 b), no correspondence could be found for the regions with low texture (e.g. the sky), and therefore the corresponding pixels in the disparity map are black.

5 Disparity Map Segmentation

In order to understand the scene in front of the vehicle, the disparity map has to be reliably segmented, that is the objects have to be separated from each other and from the background.

For automotive applications there are mainly two directions of disparity map segmentation. One direction is to use the so called bird perspective [12]. This representation presents a top view of the scene, as known from radar applications and, after removing the ground plane, the pixels are grouped into objects. The second main direction is the so called U/V Disparity, which consists in two projection images of the disparity map. In these projections, the U respectively the V image axis is the axis on which the projection is applied, the other axis is the disparity value and the pixel intensity represents the number of pixels having a specific coordinate and a specific disparity value. The V Disparity is used to detect the road surface and obstacles lying on it since a surface in the disparity map is mapped to a line in the V Disparity [13, 14]. The object detection however is not always reliable enough, so another approach helps segmenting by using preprocessed radar signals [6] in order to generate "masterpoints" that validate the result. These methods output a rectangular region of interest (ROI) for each object of the scene.

In contrast to state-of-the-art methods, the proposed segmentation algorithm uses directly the disparity map, and works independently of the image content. The method is based on the two-pass algorithm for blob detection introduced by Saphiro and Stockman in [15], but has been adapted to segment gray-scale disparity maps. The output of the algorithm is a labeled image containing separated objects, as it gives a different label for each object and also a ROI of irregular shape that fits the object better as a classical, rectangular ROI. Fig. 6 illustrates blob detection and disparity map segmentation, where, for better visualization, each label is represented by a color.

Fig. 6 c) shows an example of disparity map which should be segmented, meaning to group pixels belonging to the same object. As can be seen in Fig. 6 c) and e), in disparity maps neighboring pixels belonging to the surface of an object have close disparity values, while on the edges of the object the difference in disparity value between the pixels of the object and of the background is large. These transitions in disparity values are used for segmentation.

It can be observed in Fig. 6 f) that the vertical object is dividing the object behind it in two pieces. However, by analyzing the disparity values on the edge of the two pieces it is possible to conclude that they should be merged. The result can be seen in Fig. 6 g).

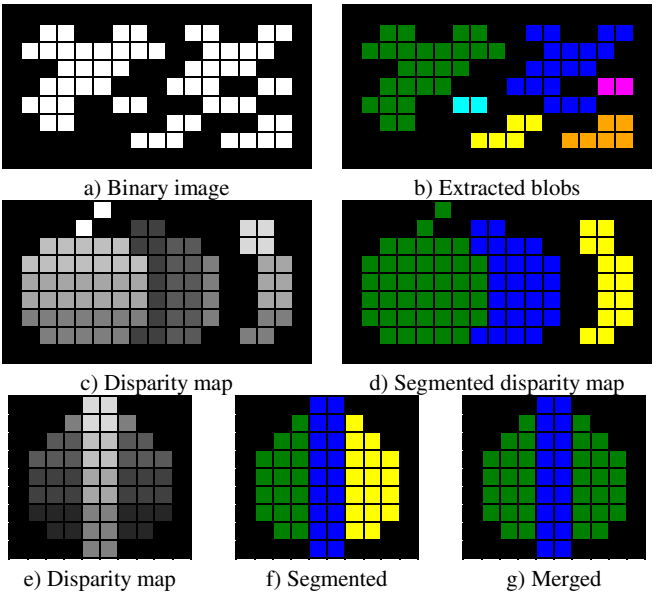


Fig. 6. Illustration of disparity map segmentation and merging.

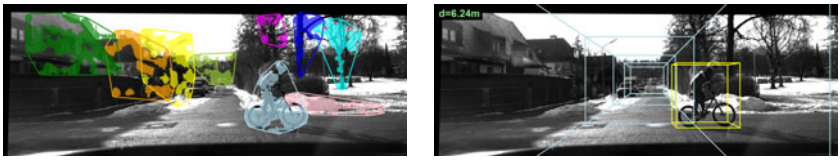
6 Collision Warning

The final step of the algorithm is collision warning. This is done using the distance to each of the objects detected in the scene, which is computed using:

$$D = \sqrt{X^2 + Y^2 + Z^2} \tag{3}$$

where D is the distance from the camera center to the object's center of mass and X, Y, Z are the 3D coordinates of the object's center of mass, computed using (2), adjusted for the translation between the camera coordinate system and the vehicle coordinate system. Assuming the car is heading forward, a driving tunnel can be defined as the space which the car needs in front of it in order to safely pass. This tunnel is 50% wider and 50% higher than the car width and height respectively.

Fig. 7 a) shows the result of the disparity map segmentation overlaid on the rectified left image. The segmented objects are bounded by a convex hull [16]. Fig. 7 b) shows the computed driving tunnel overlaid on the image. The driving tunnel contains markings every 5 meters from the bumper, until a distance of 20 meters. Also a cuboid is displayed, which bounds the closest object: the bike.



a) b)

Fig. 7. Segmented disparity map (a) and bounding cuboid for closest object (b).

If an object is inside the driving tunnel or even close to it, the driver should be warned. For this, the distance between the front bumper of the car and the center of mass of the object is used. The computed distance is shown in Fig. 7 b) in the upper left corner. If an object is inside the driving tunnel, based on its distance to the vehicle, the collision warning returns a severity level W_L according to:

$$W_L = \begin{cases} low, if\ D \geq 20m \\ medium, if\ D < 20m\ AND\ D > 10m \\ high, if\ D \leq 10m \end{cases} \quad (4)$$

where D is the distance defined in (3).

In Fig. 7 b) the bike is located at 7.89m with respect to the camera. The shift between the camera coordinate system C and the vehicle coordinate system V is 1.65m on the Z axis, so the distance from the front bumper to the bike is 6.24m. Accordingly, the returned severity level W_L is *high*. If more than one object is in the driving tunnel at a time, the severity level is given by the closest object.

7 GPU Implementation Using CUDA

In order to achieve a high rate of processed frames, the stereo rectification, the disparity map computation and the segmentation have been hardware accelerated by using a Graphical Processing Unit (GPU). The used GPU was an Nvidia Tesla C1060, programmed using the Nvidia CUDA API.

The big difference between CPU and GPU programming is the number of cores. A regular CPU has 2-4 cores and a good CPU has up to 16 cores, while the Nvidia Tesla C1060 GPU has 30 multiprocessors (MP) of 8 cores each, summing up to a total of 240 scalar processor cores (SP), on which a maximum of 65536 threads can run almost parallel [17]. The only constraint is that all threads will execute the same operation at the same time according to the SIMD principle [18]. Therefore the algorithms must be parallelized in order to efficiently run on a GPU. The main condition for parallelizing an image processing algorithm is that the nature of the algorithm allows the image to be divided into small regions that are processed independently and that the results can be merged into a resulting image.

Stereo rectification, disparity map computation using block matching as well as the proposed disparity map segmentation method have all been implemented on the GPU. The required processing time can be seen in Table 1.

Table 1. Processing time comparison between CPU and GPU

	1280x480 px		640x240 px	
Operation	CPU	GPU	CPU	GPU
Memory transfer CPU•GPU	–	0.7 ms	–	0.2 ms
Stereo rectification	12.2 ms	0.9 ms	3.5 ms	0.2 ms
Disparity map computation	68.6 ms	26.4 ms	16.7 ms	5.1 ms
Disparity map segmentation	48.1 ms	16.7 ms	14.1 ms	4.3 ms
Total processing time	128.9 ms	44.7 ms	34.3 ms	9.8 ms

It can be observed that the processing time in case of 1280x480 images is around 45 ms per frame, while for 640x240 it is below 10ms, so the desired frame rate of 100fps is achieved for this resolution. The system latency is below 0.1s.

8 Results

In this section the segmentation results from two videos are shown followed by a performance evaluation using a Bosch PLR 50 laser distance measuring device.

Fig. 8 a)-d) presents four frames from a video taken at a crossroad, where a bicycle is coming from the left side. Fig. 8 e)-h) presents four frames of a video taken at a crossing. The video starts with a pedestrian crossing followed by a bicycle. It can be seen that the presented system detects multiple objects in the scene.

In Fig. 8 a) and b) a pedestrian can be seen on the left side and a bike on the right side. Both objects are detected after segmenting the disparity map. Additionally the street poles containing traffic signs are detected in these frames as well.

In order to evaluate the performance of the system in terms of computed distance between the car and the detected objects, a test has been performed using a laser distance measuring device. In this setup, a person was standing in front of the car at a certain distance, measured using the laser distance measurer. At the same time the camera system measured the distance to the person using the presented algorithm. This measurement has been repeated for 49 different distances ranging from 6m to 30m, with a step of 0.5m. Fig. 9 a) shows the results of this test and Fig. 9 b) shows the error between the distance measured with the laser measurement device and the data measured using the proposed stereo vision system.

The laser distance measuring device has been chosen due to its measurement precision of typically 2mm according to the manufacturer. It can be observed in Fig. 9 a) that the measuring results obtained from the proposed stereo vision system follow the values obtained by the laser distance measuring device. After 20m

steps become visible in the distance obtained by the vision system. This is due to the fact that disparity values are small for that distance and a difference of one unit in the disparity value is characterized by the step size. This means that distance values between two steps are not noticed. It is planned to add subpixel accuracy to the calculation in order to make the steps smaller even at distances above 20m.

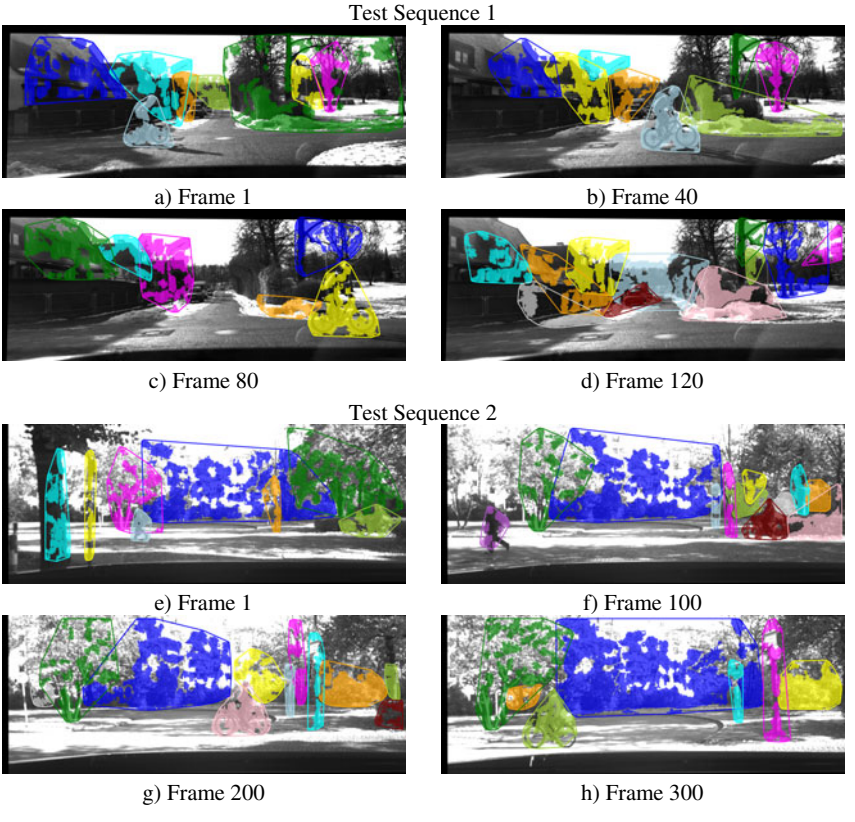
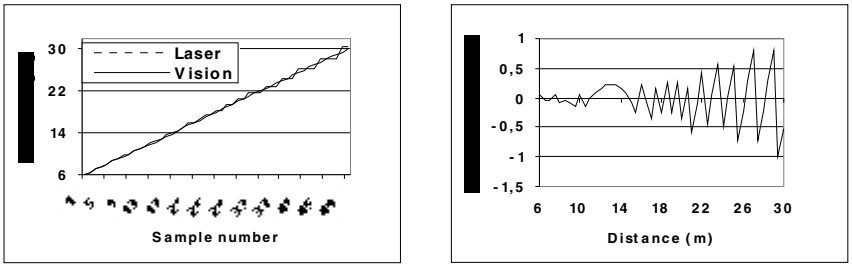


Fig. 8. Segmented disparity map overlaid on rectified left image – first test video (a-d) and second test video (e-h)

Fig. 9 b) shows the distance error as difference between the distances measured with the laser distance measuring device and the corresponding values obtained from the vision system. The absolute error increases with the distance, but the relative error is not higher than 5% for the entire tested range of 6m to 30m.



a) b)

Fig. 9. Comparison between data obtained from the laser distance measuring device and data obtained using the proposed stereo vision system (a) and distance error (b)

9 Conclusions and Outlook

In this chapter a novel stereo vision based collision warning system has been presented. The system uses two high speed cameras delivering frames at 100fps with a resolution of 1280x480 pixels and is able to process the data stream at full speed at a resolution of 640x240 by making use of an Nvidia Tesla C1060 GPU, programmed using the Nvidia CUDA API.

The presented system performs stereo rectification on the input images for allowing dense disparity maps to be computed using the block matching algorithm. The resulting disparity map is segmented using a novel disparity map segmentation method based on pixels' intensities. After segmentation, the objects present in the scene are separated from each other and from the background and the distance to each of them is computed. This distance is used for collision warning, where a driving tunnel is defined for allowing safe driving of the vehicle. If an object crosses the driving tunnel, the system warns the driver of a possible collision.

The presented results show that objects are well separated from each other and the test using a laser distance measurer shows that the computed distance using the stereo vision system is reliable, having an error below 5% in the range 6-30m.

It is planned in future to add subpixel accuracy when computing the disparity map in order to allow the distance to far away objects to be estimated in finer steps. Also a classification of the objects as well as tracking will be implemented in order to predict a possible collision more accurate.

Acknowledgments. The research leading to these results has been funded by the German Federal Ministry of Education and Research (Ger. BMBF) in the project HiSpe3D-Vision (BMBF Grant 01IS09021C). The authors would also like to thank pco. for their big help in setting up the camera system.

References

[1] Cui, J., Liu, F., Li, Z., Jia, Z.: Vehicle localisation using a single camera. In: Proceedings of IEEE Intelligent Vehicles Symposium (IV), San Diego, CA, USA, pp. 871–876 (2010)

[2] Nedevschi, S., Vatavu, A., Oniga, F., Meinecke, M.M.: Forward collision detection using a Stereo Vision System. In: Proceedings of International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, pp. 115–122 (2008)

[3] Fuerstenberg, K.C., Dietmayer, K.C.J., Willhoeft, V.: Pedestrian recognition in urban traffic using a vehicle based multilayer laserscanner. In: Proceedings of IEEE Intelligent Vehicles Symposium (IV), Versailles, France, vol. 1, pp. 31–35 (2002)

[4] Ewald, A., Willhoeft, V.: Laser scanners for obstacle detection in automotive applications. In: Proceedings of IEEE Intelligent Vehicles Symposium (IV), Dearborn, MI, USA, pp. 682–687 (2000)

[5] Wenger, J., Hahn, S.: Long range and ultra-wideband short range automotive radar. In: Proceedings of IEEE International Conference on Ultra-Wideband (ICUWB), Singapore, pp. 518–522 (2007)

[6] Teutsch, M., Heger, T., Schamm, T., Zöllner, J.M.: 3D-segmentation of traffic environments with U/V-disparity supported by radar-given masterpoints. In: Proceedings of IEEE Intelligent Vehicles Symposium (IV), San Diego, CA, USA, pp. 787–792 (2010)

[7] Leu, A., Aiteanu, D., Gräser, A.: A novel stereo camera based collision warning system for automotive applications. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, pp. 409–414 (2011)

[8] PCO 1200 s product description, `http://www.pco.de/high-speed-cameras/pco1200-s/` (accessed August 2011)

[9] Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press (2003)

[10] Tsai, R.Y.: An efficient and accurate camera calibration technique for 3D machine vision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, USA, pp. 364–374 (1987)

[11] Tao, T., Koo, J.C., Choi, H.R.: A fast block matching algorithm for stereo correspondence. In: Proceedings of IEEE Conference on Cybernetics and Intelligent Systems, Chengdu, China, pp. 38–41 (2008)

[12] Huang, Y., Fu, S., Thompson, C.: Stereovision-based object segmentation for automotive applications. EURASIP J. Appl. Signal Proc. 14, 2322–2329 (2005)

[13] Soquet, N., Aubert, D., Hautiere, N.: Road segmentation supervised by an extended V-disparity algorithm for autonomous navigation. In: Proceedings of IEEE Intelligent Vehicles Symposium (IV), Istanbul, Turkey, pp. 160–165 (2007)

[14] Labayrade, R., Aubert, D., Tarel, J.: Real time obstacle detection in stereovision on non flat road geometry through V-disparity representation. In: Proceedings of IEEE Intelligent Vehicles Symposium (IV), Versailles, France, pp. 646–651 (2002)

[15] Shapiro, L.G., Stockman, G.C.: Computer vision. Prentice-Hall (2001)

[16] Preparata, F.P., Hong, S.J.: Convex hulls of finite sets of points in two and three dimensions. Communic. ACM 20, 87–93 (1977)

[17] Nvidia Tesla C1060 product description, `http://www.nvidia.com/object/product_tesla_c1060_us.html` (accessed March 2011)

[18] Cockshott, W.P., Renfrew, K.: SIMD programming manual for Linux and Windows. Springer (2004)

Multithreaded Peripheral Processor for a Multicore Embedded System

Horia V. Caprita[1] and Mircea Popa[2]

[1] "Lucian Blaga" University of Sibiu, Faculty of Engineering, Str. Emil Cioran 4, 550025 Sibiu, Romania
horia.caprita@ulbsibiu.ro
[2] "Politehnica" University of Timisoara, Faculty of Automation and Computers, Bd. V. Parvan 2, 300223 Timisoara, Romania
mircea.popa@ac.upt.ro

Abstract. Multithreaded and multicore architectures represent a good solution used for increasing of parallelism degree exploited in modern computing systems and they can reduce the power dissipated in the chip by using low-frequency clock signals. These advantages recommend these parallel architectures for integration in the embedded systems, with restrictions imposed by the relatively small integration area. Particularities of embedded applications require hardware support able to handling in real time the peripheral interrupt requests. The performance of this hardware influences the performance of the entire parallel system. The current trend is to integrate in one microcontroller more processors used for general purpose application and a processor used for peripherals related applications (like interrupt services). In this chapter we present a peripheral processor architecture implemented in multithreaded technology, processor able to handle more tasks concurrently. Parallel execution of these peripheral tasks, in conjunction with a multicore processor used in processing of the general application (e.g. operating system), will lead to an increase in overall performance of the embedded system.

1 Introduction

The development of electronic technology today has allowed the implementation of complex architectures, which led to the emergence of multicore processors technology [8]. Multicore architectures are built from superscalar and multithreaded processors. Integrating new technologies in embedded applications requires the development of multicore processors that can be integrated into a smaller area like a classic microcontroller. These processors must manage fewer resources and be able to manage multiple tasks simultaneously [4]. More so simple processors (cores) will

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 201–212.
springerlink.com

compete for processing a single application or multiple processing tasks; overall system performance will depend on the current performance of each processor in the system [8].

Nowadays, in order to achieve higher performances, the processors can exploit two forms of parallelism: instruction-level parallelism (ILP) and thread-level parallelism (TLP) [10]. Multithreading technique permits the parallel processing of multiple independent threads (TLP). A multithread processor is able to concurrently execute instructions coming from multiple threads. The "thread" means either explicit thread defined by programmer and managed by the operating system, or an implicit thread, which means a contiguous instructions sequence (independent in comparison with other sequences), statically generated (by compiler) or dynamically generated (run-time by hardware). The processors which interleave instructions from threads defined by user exploit the explicit multithreading concept [3, 10].

Two techniques can be used to implement the explicit multithreading [12]:

- Interleaved Multithreading (IMT): the thread switching (context switching) is done by each pipeline cycle (each instruction). That is the reason for often defining this technique as fine grain multithreading;
- Blocked Multithreading (BMT): the context switching is done by events that have long latencies (Load/Store instructions, RAW dependencies etc). That is the reason for often defining this technique as coarse grain multithreading.

The superscalar architectures permit more than one instruction to be issued by a cycle. In these architectures the processor controller tries to feed all the functional units, issuing, each cycle, a larger as possible number of instructions (Instruction Level Parallelism approach). This strategy must be extended to the Thread Level Parallelism (TLP) concept. That is, the concurrent issued instructions come either from the same thread or from different threads. This technique is named Simultaneous Multithreading (SMT) [10, 12]. The implicit multithreading is based on speculative execution of statically or dynamically generated threads, derived from a sequential single thread program. The generated threads (microthreads) are executed in the same context. The main objective of this technique is the execution speed-up of a sequential program [3].

In this chapter we propose a peripheral processor architecture implemented in multithreaded technology, able to handle more tasks concurrently: SimpleScalar Peripheral Multithreaded architecture (SS-PMT) that is derived from SimpleScalar simulator [1]. This architecture is fed with instructions belonging to the peripheral interrupt handlers, and is derived by:

- Resources multiplication (Program Counter, Stack Pointer, Register File etc.);
- attaching of a thread tag (id_thread) to each instruction, integrating of a proper tag identification algorithm, in each block of the simulator;
- Implementing of the IMT and BMT techniques in the Fetch Instruction Unit.

We will evaluate the performances of this architecture compared to the performance of standard SimpleScalar architecture.

2 Related Work

D.M. Tullsen [9] proposes a SMT architecture derived from a superscalar out-of-order architecture. The proposed processor fetches up to eight instructions/cycle. The WAR and WAW dependencies are solved by register renaming technique and two instruction queues are used for instruction issuing. After execution the instructions are in-order retired, in such a way releasing the allocated resources (registers). The following resources are included for SMT model implementation:

- Multiple PC registers (program counters);
- Multiple stacks (one stack/thread) for procedures return addresses prediction;
- Instruction Retire Unit (a instruction queues release unit) and a trap mechanism attached to each thread;
- Thread identifier in each BTB (branch target buffer) entry for dynamically allocation of BTB entries for the threads;
- Register file multiplication to hold multiple contexts and to provide a proper support for register renaming technique.

All threads shared the instruction queues. For instructions fetch, a Round-Robin (RR) algorithm applied to the threads in which a cache miss didn't yet occurs, selects the appropriate PC register.

A speculative SMT architecture is presented in Chappell et al. 1999 [3]. The authors start from the idea according to the SMT processors have poor efficiency when execute programs which are not split in threads. For processor performance enhancement a SSMT (Simultaneous Subordinate Microthreading) is proposed; the SSMT processor spawns subordinated threads, implemented in microcode (microthreads). The microthreads are concurrently executed together with the main thread, thus enhancing the branch prediction accuracy, the prefetch process accuracy and the cache hit rate. When a latency generator event occurs (i.e. wrong branch prediction, miss accesses in cache etc.) the appropriate microthread is activated by injecting its instructions in decoding/renaming unit. The microthread and the main thread register renaming processes are separately done by using Register Alias Tables (RATs). Instruction issue unit can select instructions either from the main thread, or from microthreads. The SSMT architecture claims the compiler and operating system support. The compiler must analyze the main thread behavior and correlate the microthreads length according to the delay slots generated by the main thread. The microthreads must be loaded in microRAM memory before the program running by inserting them in executable program file. The microthreads loading process is done by special loading instructions inserted by compiler in the program startup area.

In Redstone et al. 2003 [6] the authors start from the idea that the large scale SMT processors implementation is difficult due to the extremely large number of registers, respectively contexts. In order to reduce the hardware complexity the authors propose an architecture which increases the TLP level without excessively increasing of the physical registers number (register file dimension). The proposed architecture uses minithreads. A program which runs in its own context can create minithreads which will share the program context (the same register file). The mtSMT abbreviation specifies a SMT processor which is executing microthreads and the mtSMTi,j abbreviation specifies a SMT processor which can support i hardware contexts with j minithreads/context. For example, a mtSMT4,2 processor can exploit the same TLP quantity like an 8 contexts SMT processor using only a half number of registers/context. The key concept of the mtSMT architectures is the fact that the minithreads share the logical registers belonging to the same context, that is, the same physical registers. Two instructions, belonging to different minithreads which share the same physical register, will share the same variable. Notice that the register renaming hardware must not be modified (is the same like in SMT architecture). Only the mapping process of physical registers addresses to renaming table entries will be modified.

3 Previous Research

In [2] we have introduced two scalar multithreaded architectures derived from SimpleScalar simout-order simulator [1]:

- SimpleScalar Interleaved Multithreading architecture (SS-IMT)
- SimpleScalar Blocked Multithreading architecture (SS-BMT).

Interleaved Multithreading technique (IMT) is often called fine-grain multithreading [12]. IMT eliminates hazards control and data dependencies between instructions. Memory latencies are hidden by scheduling of instructions belonging to other threads until the current transaction with memory is completed. This model requires a number of threads at least equal to the number of pipeline levels. Table 1 depicts the resources shared by threads and the multiplied resources which will be allocated to each thread of SS-IMT.

Table 1. Shared/Multiplied resources in SS_IMT

Shared resources	Multiplied resources
Cache memory	Register File
Functional units	PC (Program Counter)
Register Update Unit (RUU) / Load Store Queue (LSQ)	SP (Stack Pointer)
Ready, Event and Fetch Queues	Create Vector [regs] (destination registers)

The SS-IMT architecture switches to another thread after each instruction fetch (Fig. 1).

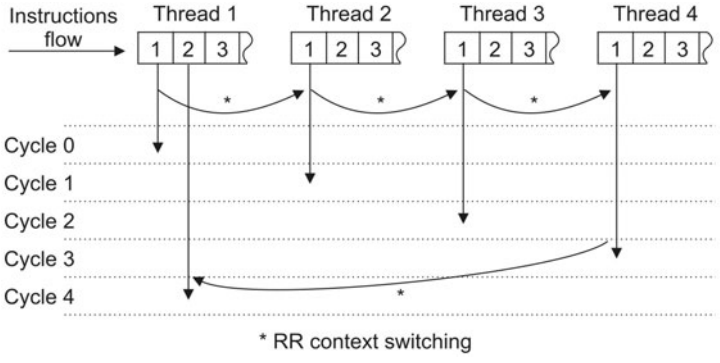


Fig. 1. SS-IMT context switching.

As in the previous case, the performed changes led to Blocked Multithreading architecture (coarse-grain-multithreading) with implicit threads switch. The context switch is triggered by events like "switch on load" and "switch on store" [10]. The "block interleaving" multithreading architecture assumes that the instructions are fetched from the same thread until a switching context event occurs (Fig. 2).

Contrary to the "Cycle by Cycle Interleaving" model, the SimpleScalar Blocked Multithreading architecture (SS-BMT) works very well with a small number of threads. Between two consecutive switches the active thread will be processed with maximum speed. If the SS-BMT processor executes only one thread, it works like a pipeline processor. The threads handling was obtained by adapting some units of SimpleScalar simout-order simulator.

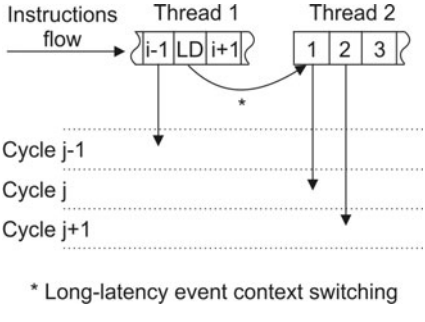


Fig. 2. SS-BMT context switching.

These architectures were evaluated by running 100 millions of dynamic instructions/benchmark. For each simulation we defined some groups of benchmarks named "thread sets". We have three categories of thread sets: 2 benchmarks thread sets (18 sets), 4 benchmarks thread sets (22 sets) and 8 benchmarks thread set (1 set). For example, each 2 benchmarks thread set (named Gr.-2Th-*) contains a combination of two benchmarks selected from SPEC95 suite, each 4 benchmarks

thread set (named Gr.-4Th-*) contains a combination of four benchmarks selected from SPEC95 suite and the 8 benchmarks thread set (named Gr.-8Th-*) contains the all threads of SPEC95 suite used in our simulation (Table 2).

The results presented in [2] were very good and conduct us to continue our research in this area. Moreover, these results guiding us in the development process of the new simulators, which should contain simultaneous multithreading technology in order to exploit more thread/instruction level parallelism.

Table 2. Some examples of the threads sets defined by authors

2 benchmarks thread set	code	4 benchmarks thread set	code	8 benchmarks thread set	code
applu.ss, m88ksim.ss	Gr.-2Th-1	applu.ss, swim.ss, turb3d.ss, m88ksim.ss	Gr.-4Th-1	apsi.ss, applu.ss, swim.ss, turb3d.ss, wave5.ss, tomcatv.ss, mgrid.ss, m88ksim.ss	Gr.-8Th-1
applu.ss, tomcatv.ss	Gr.-2Th-2	applu.ss, swim.ss, turb3d.ss, mgrid.ss	Gr.-4Th-2	-	-
…	…	…	…	…	…
wave5.ss, mgrid.ss	Gr.-2Th-18	swim.ss, wave5.ss, tomcatv.ss, mgrid.ss	Gr.-4Th-22	-	-

4 Multithreaded Peripheral Processor SS-PMT

In this chapter we propose peripheral processor architecture implemented in multithreaded technology, processor able to handle more tasks concurrently. This processor is used for peripheral interrupt handling and will work concurrently with the main processor (Fig. 3).

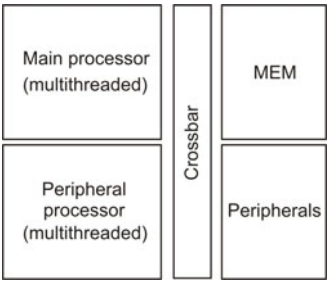


Fig. 3. SS-PMT - Multithreaded embedded architecture with peripheral processor.

Parallel execution of these peripheral tasks (interrupt service routines), in conjunction with multicore processor used in processing of the general application

(e.g. operating system), will lead to an increase in overall performance of the embedded system. This architecture contains four main blocks: the main processor used for general purpose applications or embedded operating systems, peripheral processor used for peripheral interrupt handling, peripheral modules (i.e. asynchronous and synchronous serial interfaces, CAN controllers, analog-to-digital converters etc.) and shared (common) memory unit. Peripheral processor SS-PMT uses multithreaded technology. The context switching (thread switch) is done using the Round-Robin algorithm. The thread term specify in this case an interrupt handler, activated by a peripheral interrupt request. The main role of this processor is to response in real-time to hardware interrupt requests generated by the peripheral modules. This processor communicates with the main processor using the crossbar network and data variables stored in common memory.

In this chapter we evaluate the performance of the peripheral processor from architectural point of view. The overall performance of the entire embedded system represents a goal of our future research.

4.1 SS-PMT Architecture

Basically, the structure of SS-PMT embedded architecture (Fig. 4) is based on SS-IMT general architecture [2].

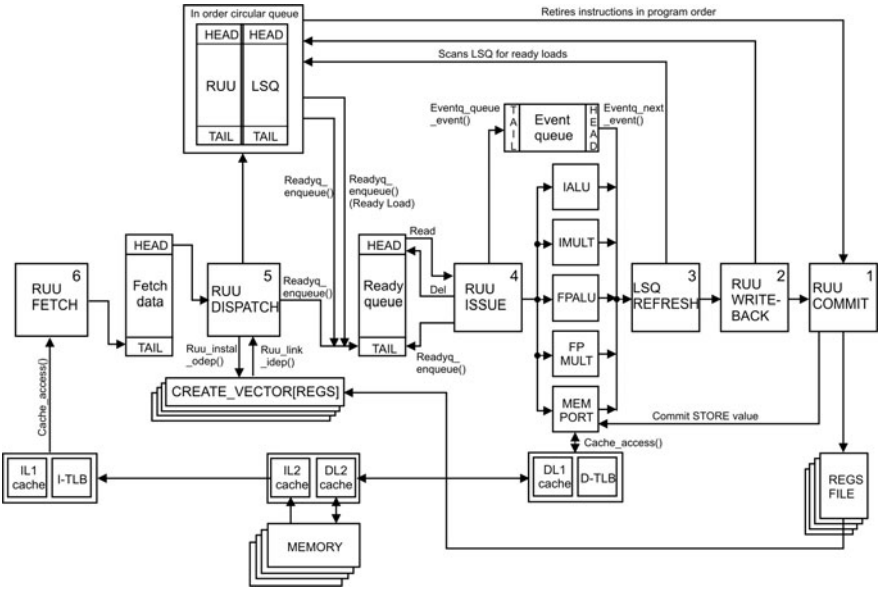


Fig. 4. SS-PMT peripheral processor architecture.

Each active thread is, in this case, a sequence of instructions fetched from peripheral interrupt handlers. Each thread has own identifier (id_thread). This

identifier is attached to each instruction and will accompany the instruction during its processing. All the SimpleScalar simulator blocks were modified in order to be able to identify the thread that the instruction belongs to.

In instruction fetch stage the Register Update Fetch Unit (RUU_Fetch) module fetches the instruction pointed by PC[id_thread] in interrupt handler and put it in the prefetch buffer. In the decoding stage the instruction is read from the prefetch buffer and, after decoding, depending on the class that the instruction belong to, is loaded in the proper instruction queue:

- in RUU (Register Update Unit Queue) if the instruction will not perform any access in data memory;
- in LSQ (Load/Store Queue) if the instruction is Load or Store.

When the instruction is ready to be executed (the operands are available), it will be loaded in the Ready Queue. In the execution stage the instructions extracted from Ready Queue are passed to the functional units. For an accurate Writeback stage handling, in the Execution stage the Event Queue will be set also, according tot the functional units latencies. In the Writeback stage the instruction results are written in RUU/LSQ queues by Event Queue scanning. In the same time, the instructions ready to execute detected in the RUU/LSQ queues (all instructions with data dependencies solved) are transferred to the Ready Queue. The LSQ_Refresh module scans the LSQ queue in order to find ready Loads (the operand is available). If such Loads are found, they will be introduced in Ready Queue. In Commit stage, the instruction result will be read from RUU queue and will be written in the destination register pointed by instruction. The destination register will be selected inside the context of the thread that the instruction belongs to. In this phase the writes pointed by the Store instructions are performed in data cache, also.

5 Simulation Results

For SS-PMT architecture simulation we were using SPEC95 benchmarks [7, 11]. We selected 8 representative benchmarks from SPEC95, which represented the threads concurrently processed in our architectures (each thread represents an interrupt service routine). In this evaluation we have assume that the interrupt handlers (threads) are already loaded in memory. The results were obtained by running 100 millions of dynamic instructions/thread on standard SimpleScalar architecture and our SS-PMT architecture. For each simulation we have used the groups of benchmarks (thread sets) presented in Section 3, Table 2.

We defined three versions of the SS-PMT architectures: two hardware contexts (2 threads), four hardware contexts (4 threads) end eight hardware contexts (8 threads). To avoid the high decreasing of the hit rate together with contexts number increasing, we extended the number of sets for D-cache and I-cache (i.e. the

doubling of the threads number leads to sets number doubling, keeping constant the associatively factor). We evaluated the performance (Instruction per Cycle - IPC) of standard SimpleScalar architecture (scalar configured) using SPEC benchmarks. The scalar processor has separated instruction (I-cache) and data (D-cache) caches, both four-way set associative (32, 64, 128 and 256 KBytes for I-cache and D-cache size). Also we evaluated the SS-PMT processor using the same methods like those used for the scalar processor. Figures 5, 6 and 7 depict the performance of SS-PMT processor with 2, 4 and 8 contexts.

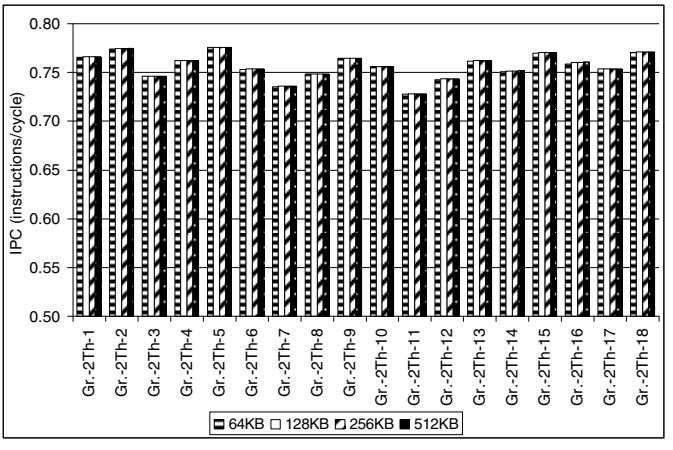


Fig. 5. 2-contexts SS-PMT performance.

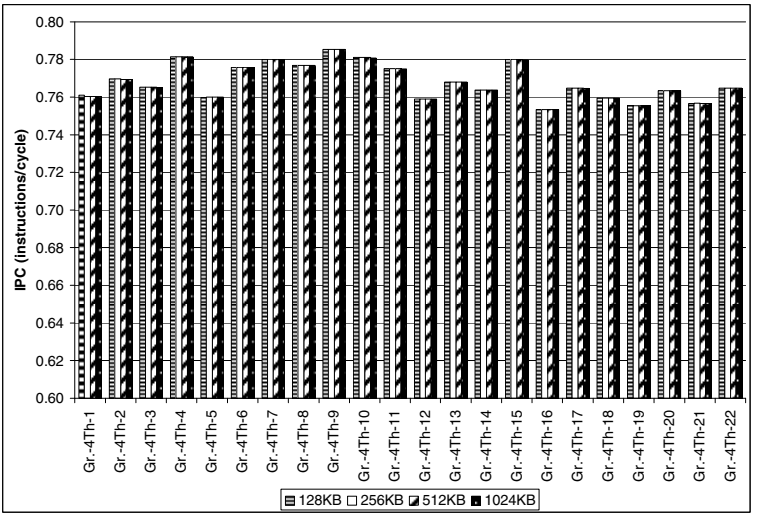


Fig. 6. 4-contexts SS-PMT performance.

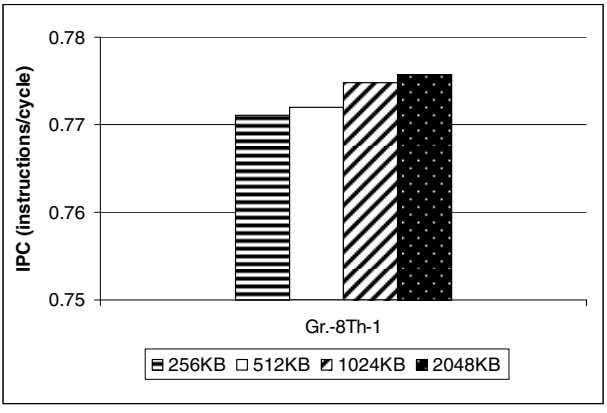


Fig. 7. 8-contexts SS-PMT performance.

6 Conclusions

Figures 8 and 9 depict the overall performance of the SS-PMT embedded architecture versus to the performance of 1-context scalar architecture, using selected benchmarks as threads (interrupt handlers) to feed the multithreading mechanism.

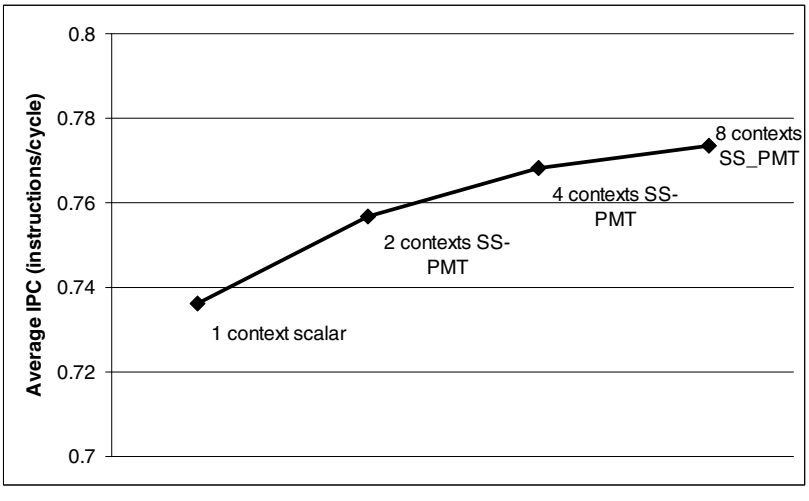


Fig. 8. Average IPC of SS-PMT architecture.

Based on Fig. 8, we may conclude that the performance (IPC) grows with approximately 2.7% by migrating from scalar-1 context processor to 2 contexts SS-PMT, with approximately 1.5% by migrating from 2 contexts SS-PMT to 4

contexts SS-PMT and with 0.7% by migrating from 4 contexts SS-PMT to 8 contexts SS-PMT. The evolution of IPC parameter shows us that the increasing of the contexts number over 8 isn't a good solution for this type of architecture, because it leads to performance degradation.

In figure 9 the D-cache hit rate decreases together with processor architecture evolution. This decrease is due to the threads number increasing, threads which concurrently coexist in D-cache. Basically these architectures are effective when the number of contexts (threads) is equal or lower than the number of pipeline stages. All the more it must appreciate the positive evolution of the IPC parameter, which keeps a positive trend even if it is negatively affected by D-cache hit rate decreasing.

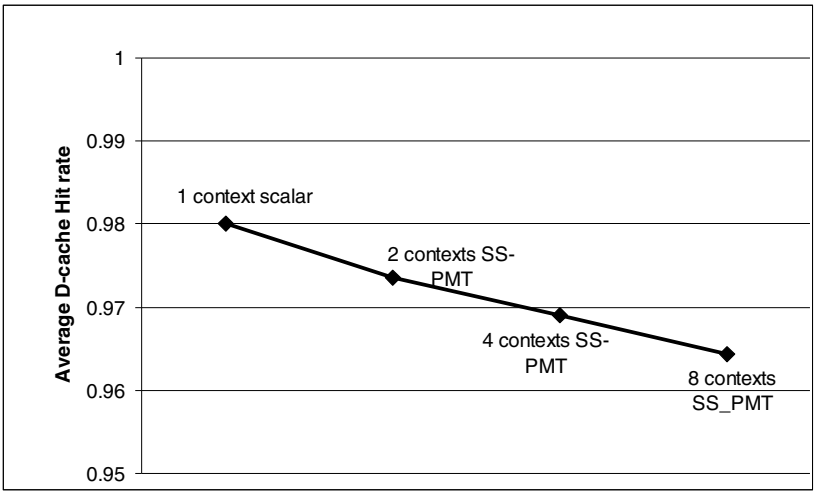


Fig. 9. Average D-cache Hit rate of SS-PMT architecture.

SS-PMT embedded architectures proposed in this chapter may be better than a classical scalar processor, except that this performance can be affected by increasing the number of contexts over the limit that can be determined through experiments. Also, to increase the processing performance, is required that cache size to be correlated with number of contexts implemented in this processor [5]. This fine-tuning of the two parameters can lead to an optimal instance of a multithreaded processor that can be embedded in dedicated systems [8]. Another important aspect, which we have not achieved in this work, is related to increasing the power consumption of the multithreaded processor [13], which has to be taken into account in evaluating its performance.

Acknowledgments. The work was supported by the Romanian Ministry of Education, CNCSIS grant 680/19.01.2009.

References

[1] Burger, D., Austin, T.M.: The SimpleScalar Tool set version 2.0. ACM SIGARCH Comput. Archit. News 25, 13–25 (1997); doi:10.1145/268806.268810

[2] Caprita, H.V., Popa, M.: Design methods of multithreaded architectures for multicore microcontrollers. In: Proc. 6th IEEE Int. Symp. Appl. Comput. Intell. Inform., Timisoara, Romania, pp. 427–432 (2011); doi: 10.1109/SACI.2011.5873041

[3] Chappell, R.S., Stark, J., Kim, S.P., Reinhardt, S.K., Patt, Y.N.: Simultaneous Subordinate Microthreading (SSMT). In: Proc. 26th Annu. Int. Symp. Comput. Archit., Atlanta, GA, USA, pp. 186–195 (1999); doi: 10.1145/300979.300995

[4] Giorgi, R., Popovic, Z., Puzovic, N.: Implementing fine/Medium grained TLP support in a many-core architecture. In: Proc. 9th Int. Workshop Embed Comput. Syst., Heidelberg, Germany, pp. 78–87 (2009); doi: 10.1007/978-3-642-03138-0_9

[5] Ravotto, D., Sanchez, E., Sonza Reorda, M., Squillero, G.: Design validation of multithreaded architectures using concurrent threads evolution. In: Proc. 22nd Annu. Symp. Integr. Circuits Syst. Des., Natal, Brazil (2009); doi: 10.1145/1601896.1601964

[6] Redstone, J., Eggers, S., Levy, H.: Mini-threads: Increasing TLP on small-scale SMT processors. In: Proc 9th Int. Symp. High Perform. Comput. Archit., Anaheim, CA, USA, pp. 19–30 (2003)

[7] Sibai, F.N.: Evaluating the performance of single and multiple core processors with PCMARK®05 and benchmark analysis. ACM SIGMETRICS Perform. Eval. Rev. 35, 62–71 (2008)

[8] Spracklen, L., Abraham, S.G.: Chip multithreading: Opportunities and challenges. In: Proc. 11th Int. Symp. High Perform. Comput. Archit., San Francisco, CA, USA, pp. 248–252 (2005); doi: 10.1145/1364644.1364647

[9] Tullsen, D.M., Eggers, S.J., Emer, J.S., Levy, H.M., Lo, J.L., Stamm, R.L.: Exploiting choice: instruction fetch and issue on an implementable simultaneous multithreading processor. In: Proc. 23rd Annu. Int. Symp. Comput. Archit., Philadelphia, PA, USA, pp. 191–202 (1996); doi: 10.1145/232973.232993

[10] Tullsen, D.M., Brown, J.A.: Handling long-latency loads in a simultaneous multithreading processor. In: Proc. 34th Annu. ACM/IEEE Int. Symp. Micro., Austin, TX, USA, pp. 318–327 (2001)

[11] Uhrig, S.: Evaluation of different multithreaded and multicore processor configurations for SoPC. In: Proc. 9th Int. Workshop Embed Comput. Syst., Heidelberg, Germany, pp. 68–77 (2009); doi: 10.1007/978-3-642-03138-0_8

[12] Ungerer, T., Robic, B., Silc, J.: A survey of processors with explicit multithreading. ACM Comput. Surv. 35, 29–63 (2003); doi:10.1145/641865.641867

[13] Wei, H., Stan, M.R., Gurumurthi, S., Ribando, R.J., Skadron, K.: Interaction of scaling trends in processor architecture and cooling. In: Semicond. Therm. Meas. Manag. Symp., Santa Clara, CA, USA, pp. 198–204 (2010); doi:10.1109/STHERM.2010.5444290

Using Cycle-Approximate Simulation for Bus Based Multi-Processor System-On Chip Analysis

Alexandru Amaricai[1], Alin Dobre[2], Oana Boncalo[1], Andrei Tanase[1], and Camelia Valuch[1]

[1] "Politehnica" University of Timisoara, Department of Computer Engineering, Bd. V. Parvan 2, 300223 Timisoara, Romania
{alexandru.amaricai,oana.boncalo}@cs.upt.ro,
{andrei.tanase,camelia.valuch}@movidius.com
[2] Movidius Ltd, Str. Paris 2A, 300010 Timisoara, Romania
alin.dobre@movidius.com

Abstract. In this chapter, a cycle approximate simulator for multi-processor system-on chip is presented. The aim of this simulation tool is to enable an enhanced software/hardware analysis capability for bus based systems. The most important contributions are represented by its high flexibility (easy configuration of a SoC using dedicated libraries for generic and specific components and easy integration of other simulators and models), accurate modeling of features specific to multi-processor systems (busses, inter processor communication mechanisms, etc), accurate implementation of a wide range of performance metrics and power consumption estimates (for processors that support this) and high simulation speed. This way, the proposed simulator can be used for both hardware architecture design exploration and software development.

1 Introduction

Today's semiconductor technology is characterized by high integration capability. As the clock frequency cannot increase much further, due to high energy consumption, other ways to improve system's performance are sought. On one hand multiple processor cores are placed on the same chip, in order to increase parallelism. On the other hand, dedicated hardware accelerators are used for computational intensive algorithms [1,2]. Therefore, today's systems are characterized by the integration on the same chip of a wide range of components, such as processor or DSP cores (ARM, x86, SPARC, PowerPC etc.), memory modules (cache, flash or DRAM), interfaces for different communication protocols (UART, I2C, USB, EBI, etc), dedicated hardware accelerators, buses and bridges for intra-chip

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 213–224.
springerlink.com

communication, etc. Several challenges have to be addressed when developing such complex systems [3]:

a. Decreased time to market: this goal must be attained for both hardware system and the software application,
b. Optimal hardware configuration: adding hardware components leads to an increase in chip's cost and power consumption; therefore, the added components must significantly increase the overall performance of the designed chip.

Simulation tools can help achieving these goals in two ways:

a. Architectural design exploration – from the hardware point of view, applications can be simulated for different architectures; this way, important architectural decisions can be made, such as cache parameters, bus configurations, integration of specific hardware accelerators, etc
b. Software application development – from the software point of view, simulation offers several important advantages:
    a. Decrease development time – usually, a simulator comes before the hardware prototype; therefore, application development can start before the chip is available.
    b. Increased software quality – the simulator provides a detailed view of the hardware components states at different moments; therefore, debugging activities are simplified.
    c. Software analysis – a simulator may provide performance and power consumption metrics; based on the obtained metrics, optimization techniques may be applied.

Hardware simulation can be performed at different abstraction layers. The most accurate simulation is obtained using RTL models. However, it has a high computational complexity; simulating large software applications is not feasible [4]. Higher abstraction simulation models provide better speed, while obtaining less accurate results [1,2]. Therefore, a trade-off between performance and accuracy must be obtained.

In this chapter, a C++ cycle approximate simulator – moviSim – is presented. Its main goal is to model a wide range of heterogeneous bus based MPSoCs. The main desired features are increased flexibility (in order to easy reconfigure hardware architectures), increased accuracy (in order to facilitate detailed analysis) and high simulation performance. The flexibility is provided by two means: development of a dedicated library of both generic and specific components and providing a standardized interface which allows easy integration of hardware modules.

2 Multi-processor System on Chip Simulation

Several cycle accurate MPSoC simulators have been developed, such as the ones presented in [5,6,7,8,9].The ***M5*** simulator [5] has been specific designed to model computing systems in TCP/IP networks. It simulates bus-based systems similar to the ones used in conventional desktops. Two processor cores are modeled: a SimpleCPU processor core (non-pipelined, in-order) and the O3CPU core (pipelined, superscalar, out-of-order and simultaneous multithreaded). Different instructions set architectures (ISA) are mapped on the O3CPU core, such as Alpha, Sparc, MIPS and ARM. Regarding support for interfaces and accelerators, M5 offers an accurate model for a Network Interface Controller, but also models for serial ports and disk controllers. According to [6], the maximum number of processor cores supported in a computing system is 4. Its main advantage is represented by the fact that it can incorporate in an easy manner a wide range of ISA, however, the processor cores are not modeled accurately (as the instruction set architecture is mapped on the O3CPU). Also, this simulator allows rapid integration of interfaces (such as I/O devices).

The ***Simics GEMS*** simulator [8] has been developed around the Ruby memory system simulator, the OPAL processor model (which uses a partial SPARC v9 ISA) and the Simics functional simulator. Ruby models a flexible multi-processor memory hierarchy, while the entire interconnect network is treated a single monolithic communication channel. The flexibility of this simulator relies on the usage of Simics functional simulator.

The ***SESC*** simulator [9] is considered MPSoC simulator with the highest performance, according to [6,7]. An out-of-order MIPS core is modeled as processing element; however the effects of wrong-path instructions are not modeled. The interconnect model is based on a routing network composed out of switches which routes data (organized as packages) from source to destination. As a drawback, it does not provide support for to integrate or add interfaces and hardware accelerators, which are ubiquitous in every modern SoC.

The ***MC-Sim*** [6] simulator is suitable for network-on chip modeling. It uses SESC processor core module as processing element. It models L1 and shared L2 caches; however it does not provide models for DRAM memories. Accelerators can be integrated using the coprocessor models, either as tight-coupled coprocessor (being part of a network node coupled with a processing core) either as loosely-coupled coprocessor (as a stand-alone node within the network).

The ***MVTSim*** simulator [7] uses a MIPS32 as processor cores. Communication between cores is performed using lightweight threads between the processing elements, instead of data read and write operations. The cache is coupled to the processor, while each core seas only a fraction of the memory space, in order to avoid coherency problems. It does not allow integration of interfaces and hardware accelerators.

All described simulators have the following drawbacks:

1. They do not model in an accurate way bus and bus transactions; thus they cannot be used in day to day software development process (they cannot help a developer to detect for example the clashes on different resources, nor the miss-usage of the resources of the system).
2. No models for inter processor communication and synchronization mechanisms are provided.
3. These simulators lack in their analysis capability, not providing important metrics for performance (stall clock cycles, bus traffic, cache hit/miss count, memory clashes etc) or power consumption estimation which are important during both hardware and software development and optimization processes.

Due to these major drawbacks, the above simulators could not be used for hardware architecture exploration and software application development.

3 moviSim Architecture

moviSim is used to model bus based MPSoC, which are composed of different processor and DSP cores, caches, memory modules, interfaces, accelerators and other modules connected through buses. An example of such system is depicted in Fig. 1. The moviSim has been implemented in C++.Development of the moviSim has been driven by the following goals:

1. *Accurate bus and bus transactions modeling* – generic bus models similar to the ones used in AMBA specifications have been developed [12]; several important bus characteristics can be configured, such as bus width, number of masters or slaves, the ability to support different types of transactions, arbitration policy, etc; every component which connects to a specific bus must contain at least one slave interface (characterized by slave address, address space and slave ID) and possible one or several master interfaces (defined by a master ID); furthermore, a generic bus transaction model is used (Fig 2.a); important features of the transaction model are implemented, such as the address, data, master ID, timestamp (moment of the transaction), burst and burst type; log files containing all the transactions on a specific bus may be generated, and thus, bus traffic can be monitored.
2. Easy configuration of a hardware architecture – in order to achieve this goal, all simulation models for hardware components inherit the ModelInterface class (Fig. 2.b); some of the methods defined by this class are:
    a. *Execute* – it describes the behavior of a bus or hardware component within one clock cycle; this method is addressed at each clock cycle;
    b. *Reset* – it describes the behavior of a hardware component on a reset; is called at each activation of the reset signal;

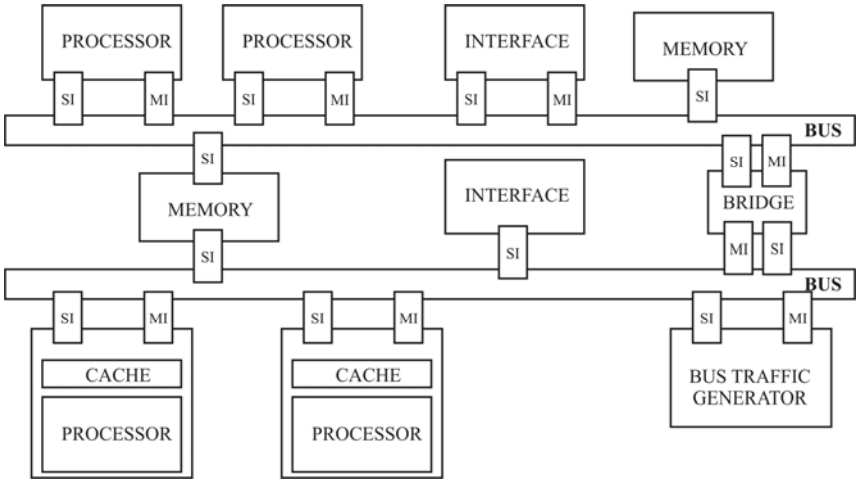


Fig. 1. General architecture modeled by MoviSim [3].

c. *runSimulation* – it is used for processor cores to set the start address of their corresponding code section;
d. *GetState* – it is used in order to determine if a processor core is executing code;
e. *GetProfilingInfo* – this method is used to generate profiling information for the component, in order to increase the analysis capability;
f. *GetMasterInterface* and *GetSlaveInterface* – this method is used to determine the master and slave interfaces of the module.

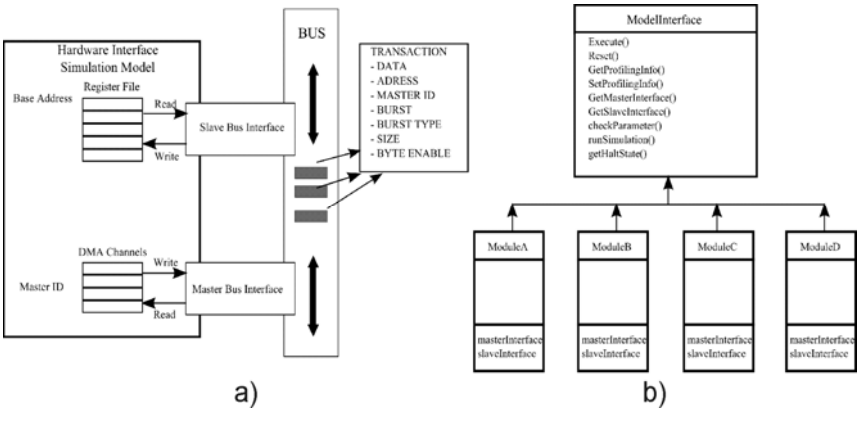


Fig. 2. a) Bus interfaces for a simulation model [3], b) the ModelInterface abstract class.

3. *Integration capability of other simulators* – integration of other simulators is possible, due to the usage of standardized model interface; in order to include other simulators, several steps have to be followed: declaring a wrapper class consistent with the standardized interface; adding an external simulator object member into the wrapper class; calling the cycle execution method and the reset method of the external simulator object within the wrapper class execute and reset methods; defining bus slave and master interfaces and implementing methods for read and write operations; this way, the SystemC based PowerPC Microlib [10] and the SWARM ARM7 [11] simulators have been integrated (Fig. 3).

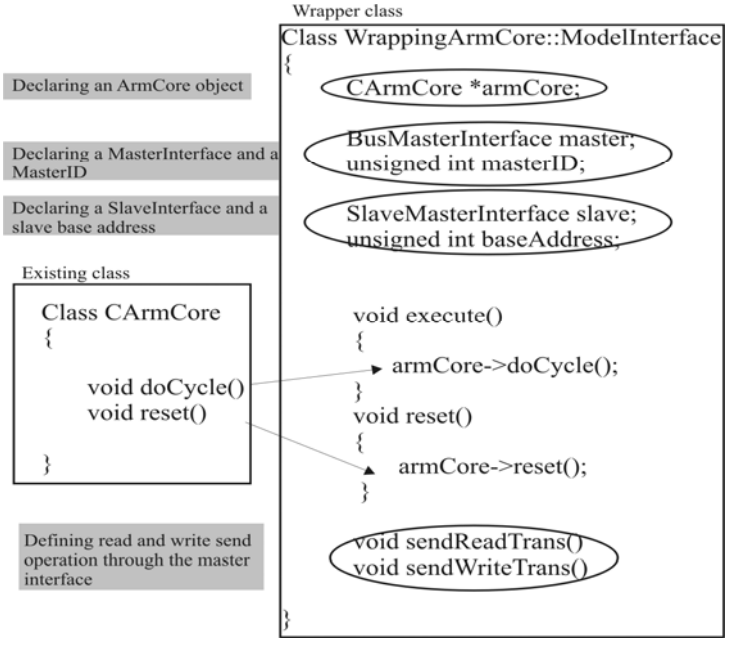


Fig. 3. Integrating the SWARM simulator into MoviSim.

4. *Increased analysis capability* – several metrics have been implemented in the moviSim simulator, such as:
    a. Execution time – the number of clock cycles required for each program to execute is provided by the simulator;
    b. Stalled processor cycles – stalls which appear during the execution of a program on one of the systems processors are provided; moreover, the stall source is provided allowing a stall profile of the simulation;
    c. Cache metrics – these include hit/miss ratio, cache accesses, port clashes, etc.;

d. Instruction level parallelism - this allows software optimizations in order to make use of the hardware more efficiently;
e. Estimated power consumption profile – this feature is implemented for the Movidius SVE cores; extremely useful when analyzing hardware architectures and software application for mobile devices;
f. Profiling information – these include assembly line profiling, code and data labels profiling, etc.;

Furthermore, more metrics can be added, depending on the desired analyses which will be performed.

5. *Inter-processor mechanism modeling* - two types of inter processor communication and synchronization mechanisms have been implemented: a FIFO module and a hardware MUTEX module. These components can be connected to a processor core. Regarding a FIFO, other processor cores may write to it, while reading can be performed only by the core which owns the FIFO. Regarding MUTEX modules, these are used for dedicated hardware mutual exclusion mechanisms, which are frequently used when dealing with multi-core and multi-threaded systems.

Configuring a system employs a standardized XML file, which contains the important characteristics of the hardware component inserted into the SoC. This way, a simulation model for a target SoC is performed into an easy manner.

4 Modeling Hardware Modules

Two types of components have been used: generic components and specific components. The generic components have been implemented in order to increase the flexibility of the simulator. The following modules have been modeled: buses, memories, caches and bus bridges.

In order to have accurate models for memories and caches, a wide range of parameters can be configured:

- for memories: size, read/write ports, latency (in clock cycles), error detection and correction,
- for caches: size, set-associativity, write policy, replacement policy, coherency protocols.

The most important class of specific components is represented by the processor cores. Two types of processor cores have been implemented:

1. Leon Sparc V8 processor – a 32 bit RISC processor with floating point unit.
2. Movidius DSP core (SVEs) – a 128 bit VLIW low power DSP with 8 parallel execution units (including an integer, scalar and vector SIMD floating point units – for both 16 and 32 bits floating point arithmetic).

Although the performance has been affected, an accurate implementation of these two processors has been chosen. This is due to the desire to obtain increased analysis capability of different parameters, such stall clock cycles, instruction level parallelism, and power consumption.

Furthermore, a wide range of hardware interface and modules typical to an SoC have been implemented. These include [3]:

1. *Camera interface module* – the camera interface module is used for simulating the data sent by a digital camera; the input is a data file that contains a stream containing frames that emulate the activity of a camera; the supported image formats are planar YCbCr4:2:2, RGB888 and raw Bayer format (12 bits), while output formats are YCbCr 4:2:2, YCbCr 4:2:0 and YCbCr 4:4: both planar and linear. It has four DMA channels, one for linear RGB with review, and three for YCbCr formats in both planar and linear format; the processing modules implemented are windowing, color correction, color space conversion (from RGB to YCbCr), color mapping and sub-sampling module and Bayer demosaicing module. Another feature implemented is the preview DMA channel that transfers a frame in RGB888 image format. Furthermore, each DMA channel can be enabled independently. The camera interface and its DMAs are configured using the register file, in the same way as the hardware. The model is capable of generating interrupts such as the ones for HSYNC, VSYNC (end of frame) or DMA transfer done.
2. *LCD interface module* – it is used to simulate the data transfers utilized for the display of frames on a LCD screen; its inputs are represented by a stream of frames from the system's memory, while the outputs are represented by a file that emulates the activity of the LCD; it implements operations such as load source address, layers initialization, convert to RGB function, convert to YUV function, execute transfer, write to output file and write each component; it has three DMA channels for the three video layers. LCD supports blending and transparency effects; the second category of features is related to the management functions on the bus; special attention is given to modeling the DMA functionality in order to accurately model the bus traffic; read requests are issued to fill the DMA FIFOs for as long as there is space in the DMA FIFOs; the input formats supported are: RGB888 linear, YCbCr4:2:2 linear, YCbCr4:2:0 planar, while the supported output formats are: YCbCr4:2:2 linear, YCbCR4:4:4 linear and RGB888;
3. *DMA controller* – it is used to transfer data between two memory locations with multi-task option; it implements functions for registry setting, calculating source and destination address, data transfer; transfer supports FIFO mode and stride mode (transfer length byte and pass over stride bytes).
4. *NAL encoder and decoder* – these two modules can be used to encode/decode NAL units of type 1,5 and 20 for H.264 encoding/decoding

[13]; the implemented decoder takes as input a NAL unit from the memory and via the receiver DMA channel, and outputs the slice header (254 words per slice), macroblock fixed part (60 words per macroblock) and the macroblock variable component (the macroblock coefficients), and it transfers them on the transmission DMA channel, while the encoder performs the opposite operations; the encoder/decoder configuration registers contain the sequence parameter set and picture parameter set used to encode/decode the NAL unit; it performs both CABAC and CAVLC encoding/decoding.

5. *Bus traffic generator* [3] – this module can substitute other modules, allowing analysis of different bus configurations (width, arbitration, etc.); it comes in 2 configurations: random and pattern; in random mode, it generates random transactions on the bus; in pattern mode, it can be used as a replacement of a processor or hardware component regarding their bus behavior; this module can be used for early development of the simulation system (which could be replaced later by a properly implemented module), or for the modules which aren't relevant from the analysis' point of view, but which may affect the simulation profile; moreover, size and parameters of the injected transactions can be configured.

Future hardware modules can be modeled, as long as their simulation model will inherit the ModelInterface class and will use the standardized bus slave and master interfaces.

5 Simulation Results

The simulated MPSoC is similar to the one depicted in Fig. 4. It consists of:

- 8 Movidius SVEs, each with 1 FIFO,
- 1 Leon RISC core,
- 8x128 kB CMX RAM memory directly link to each SVEs,
- 8 hardware MUTEX for the CMX RAM memory,
- 128 kB L2 Cache with LRU replacement policy,
- 1x128-bit bus, 2x64-bit bus and 1x32-bit bus,
- 6 bridges,
- 16 MB DRAM memory,
- 32 kB SRAM memory,
- UART interface, JTAG port, camera interface, LCD interface, 1 timer module with 8 timers, interrupt controller,
- NAL encoder and decoder.

The simulations have been performed on an Intel I5 processor, 4GB DDRAM memory desktop computer with Windows 7 operating system. We have simulated a full software H.264 decoding and a full software H.264 encoding on the

described MPSoC. The H.264 decoding required the use of 3 SVEs, while the H.264 encoder has been implemented on 5 SVEs. Results are presented in Table 1.

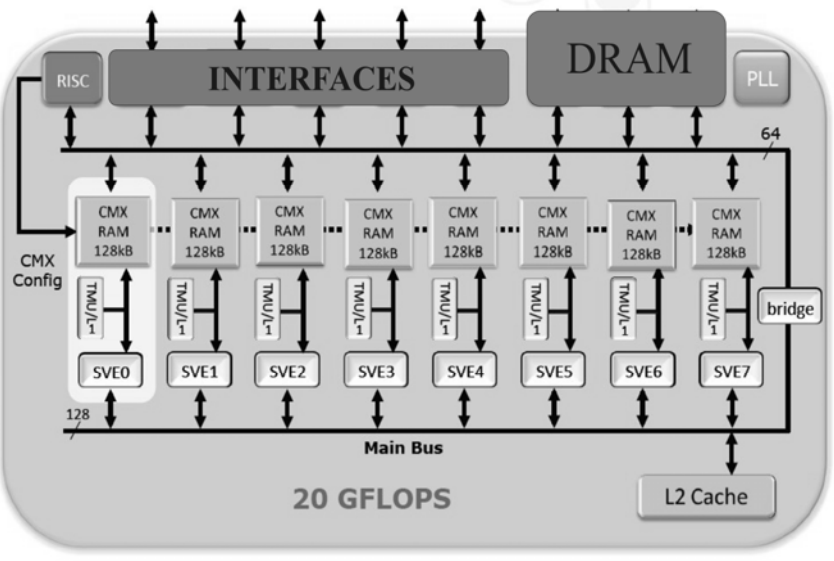


Fig. 4. An overview of the modeled system used in simulations.

Table 1. Simulation results

Algorithm	Image/Video Stream Properties	Simulation Time	Number of Executed Instructions	Instruction Level Parallelism	Clock Cycles	Stall Clock Cycles
H.264 Encoding	8 frames 128*144 pixel YCbCr 4:2:2	1 min 40 s	SVE0:974,255 SVE1:916,517 SVE2:1,044,114 SVE3:700,888 SVE4:761,557	SVE0:2.48 SVE1:2.47 SVE2:2.50 SVE3:2.49 SVE4:2.89	SVE0:1,304,358 SVE1:1,304,367 SVE2:1,304,369 SVE3:1,304,370 SVE4:1,304,303	SVE0: 330,104 SVE1:387,851 SVE2:260,256 SVE3: 603,482 SVE4: 542,747
H.264 Decoding	1 frame 720*432 pixel YCbCr 4:2:0	1 min 13 s	SVE0: 293,405 SVE1: 3,637,891 SVE2: 3.173.583	SVE0: 2.08 SVE1: 2.12 SVE2: 2.32	SVE0: 4,322,465 SVE1: 4,534,537 SVE2: 4,478,454	SVE0: 4,029,061 SVE1: 896,647 SVE2: 1,304,872

The simulation results show that the proposed simulator has an execution speed of more 40 kIPS when the software ran on 5 DSP cores and more than 90 kIPS when the software used 3 SVEs. Some factors which limit the performance of the MoviSim are:

- complex simulated SoC system
- complex VLIW architecture of the Movidius DSP core;
- accurate modeling of bus and bus transactions modeling of interrupts and the inter-processing mechanisms, which in the described applications are frequently used

The proposed simulator presents a simulation performance similar to MC-Sim and higher with respect to Simics-GEMS or M5.

6 Conclusions

This chapter presents a cycle accurate simulator, implemented in C++ used for modeling bus based MPSoC. The most important advantages of the moviSim simulator are:

1. Enhanced analysis capability for bus traffic by accurate modeling of bus and bus transactions and by implementing a bus traffic generator.
2. Enhanced analysis capability of systems performance by implementing stall clock cycles count, instruction level parallelism computation and cache hit/miss counters.
3. Easy configuration of a SoC system by providing standardized interface for hardware components and a library of generic modules.
4. Easy integration of other cycle accurate simulators – the SWARM ARM7 and the MicroLib PowerPC750 simulators have been integrated into moviSim.
5. Modeling of interrupts, FIFOs and hardware MUTEXes, which are used as means of communication and synchronization between processors or between processors.

The moviSim simulator has been used for architectural design exploration in the development of the Movidius Myriad3D platform, which uses 20 GFLOPS Movidius MA1133 graphic coprocessor, and for the Movidius 3D applications development.

Acknowledgments. This work has been funded by the EU Falx Daciae - SUIM 499/11844, POS CCE O2.1.1 research program.

References

[1] Cay, L., Gajsky, D.: Transaction level modeling: An overview. In: Proceedings of 1st IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, Newport Beach, USA, pp. 19–24 (2003)
[2] Weaver, V.M., McKee, S.A.: Are cycle accurate simulations a waste of time? In: Proceedings of 7th Annual Workshop on Duplicating, Deconstructing, and Debunking (WDDD), Beijing, China, pp. 2–13 (2008)
[3] Amaricai, A., Dobre, A., Boncalo, O., Tanase, A., Valuch, C.: Models and implementations of hardware interface modules in a multi-processor system-on-chip Simulator. In: Proceedings of the 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, pp. 433–437 (2011)
[4] Lee, J., Kim, J., Jang, C., Kim, S., Egger, B., Kim, K., Han, S.: FaC Sim: a Fast and Cycle-Accurate architecture simulator for embedded systems. In: Proceeding of 2008 ACM SIGPLAN-SIGBED Conference on Languages, Compilers and Tools for Embedded Systems, Tucson, USA, pp. 89–100 (2008)
[5] Binkert, N., Dreslinski, R., Hsu, L., Lim, K., Saidi, A., Reinhardt, S.: The M5 simulator – Modeling networked systems. IEEE Micro 26, 52–60 (2006)
[6] Cong, J., Gururaj, K., Han, G., Kaplan, A., Naik, M., Reinman, G.: MC-Sim: An efficient simulation tool for MPSoC design. In: Proceeding of 2008 IEEE/SCM Conference on Computer-Aided Design (ICCAD), San Jose, USA, pp. 364–371 (2008)
[7] Makela, J.M., Paakkuleinen, J., Leppanen, V.: MVTSim – Software simulator for multicore on chip parallel architectures. In: Proceedings International Conference on Computer Systems and Technologies – Comp. Sys. Tech., Ruse, Bulgaria, Article No. 15, p. 8 (2009)
[8] Martin, M., Sorin, D., Beckmann, B., Marty, M., Xiu, M., Alameldeen, A., Moore, K., Hill, M., Wood, D.: Multifacet's General Execution Multiprocessor Simulator (GEMS) toolset. ACM SIGARCH Computer Architecture News 33, 92–99 (2005)
[9] Renau, J., Fraguela, B., Tuck, J., Liu, W., Prvulovic, M., Ceze, L., Sarangi, S., Sack, P., Strauss, K., Montesinos, P.: SESC Simulator (2005), `http://sesc.sourceforge.net`
[10] Mouchard, G.: MicroLib PowerPC750 Simulator (2002), `http://www.lri.fr/~mouchard/PowerPC750/`
[11] Dales, M.: SWARM Simulator (2003), `http://www.cl.cam.ac.uk/~mwd24/phd/swarm.html`
[12] AMBA Bus Specification, `http://infocenter.arm.com/help/index.jsp`
[13] International Telecommunication Union, Advanced Video Coding for Generic Audiovisual Services – H264 Standard (2005)

A Case Study on Hardware/Software Codesign in Embedded Artificial Neural Networks

Jonathan Parri, John-Marc Desmarais, Daniel Shapiro,
Miodrag Bolic, and Voicu Groza

Computer Architecture Research Group, School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward Avenue, Ottawa, Ontario, Canada
{jparri,jdesm068,dshapiro,mbolic,vgroza}@uottawa.ca

Abstract. Software/hardware codesign is a complex research problem that has been slowly making headway into industry-ready system design products. Recent advances have shown viability to this direction within the design space exploration scope, especially with regards to rapid development cycles. Here, we exploit the hardware/software codesign landscape in the artificial neural network problem space. Automated tools requiring minimal technical expertise from Altera and Tensilica are examined along with newer advances solely within hardware/software codesign research domain. The design space exploration options discussed here look to achieve better software/hardware partitions using instruction-set extensions and coprocessors. As neural networks continue to find usage in embedded systems, it has become imperative to efficiently optimize their implementation within a short development cycle. Modest speedups can be easily achieved with these automated hardware/software codesign tools on the benchmarks examined.

1 Introduction

Application-Specific Instruction-set Processors (ASIP) and coprocessors have become common within the embedded marketplace, targeting a variety of industrially relevant real-world problems. Automated and manual methods for identifying efficient partitions between hardware and software are available. In this chapter, we examine the advantages and disadvantages of each approach using published artificial neural network examples [1, 2] and industry standard tools [3, 4]. These example programs were selected as neural network implementations with a status of legacy code that is still in use today.

In an ASIP, the instruction-set is customized to accelerate a particular application or application type. The design of an ASIP brings both hardware and software together; exemplifying the hardware/software codesign methodology.

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 225–237.
springerlink.com

Neural networks are an application of interest to the hardware acceleration community as a neural network ASIP implementation can lead to lower power consumption, faster task execution and higher task throughput compared to traditional processor-centric software implementations. Here, we examine how an embedded system can be developed and tailored to increase the performance of neural network applications.

When custom instructions are added to an existing instruction-set, they are called instruction-set extensions [5]. The automated identification of instruction-set extensions is a continually advancing research direction. In contrast, another solution involves adding a coprocessor or hardware accelerator to the bus that is shared with a processor. This approach does not require any change in processor architecture. Hardware/software codesign can be achieved in many ways, some more automated than others, and spanning a spectrum of design space exploration options. Human guided software to hardware conversion involves a skilled engineer selecting hotspots in an application based upon profiling and inspection of the code. For example, code surrounded by several nested *for* loops would, at a glance, seem to be an interesting candidate for hardware acceleration as unrolled loops. Conversely, automated implementation of custom instructions relies on modeling of the execution time behavior of the source code. Both approaches should result in a speedup.

Both the bidirectional associative memory (BAM) and Hopfield auto-associative memory networks (HAM) code samples [1, 2] were subjected to automated instruction-set extension identification and coprocessor implementation algorithms using commercial tools. We compare the manually specified coprocessors with the automated custom instruction acceleration to reveal the speedup possibilities exposed by each technique.

1.1 Our Contribution

In this chapter, it is shown that hardware/software codesign has become a viable solution for artificial neural network (ANN) applications within a rapid development life cycle. Trained experts are still needed for optimal hardware/software partitioning but today's results show promise with automated tools offering quick and effective solutions within the neural network problem space. Legacy code has become a large issue in the embedded system arena. Code rewrites and manual system augmentations can be easily minimized with hardware/software codesign tools while maintaining strict adherence to performance and power requirements. Automation is ready for primetime and is shown to cover two different neural network implementations effectively. Previous work [6], showcased one automated method of hardware software partitioning for neural network applications. Here, we expand on this concept and examine other commercial options.

2 Background and Related Work

Research into artificial neural networks began in the 1940s as a way of modeling the biological pattern classification techniques present in the brain. Modeling neurons and their interconnections lead to a powerful machine capable of effectively and efficiently solving some types of automatic classification problems [7]. In the 1980s neural network research moved into the area of memory modeling.

We analyzed two memory modeling algorithms, the auto-associative Hopfield model (HAM) and the hetero-associative Bidirectional Associative Memory (BAM) [8, 9].

2.1 Implementation and Optimization Approaches

Once working algorithms have been developed, techniques are developed to optimize their performance. These optimization techniques can be categorized as software, hardware or a hybrid hardware/software solution.

In software, several supercomputer implementations of ANNs have been created. In [10], the development of a menu driven neural network system is explored to run on a Cray supercomputer. The IBM Almaden Research Lab and the University of Nevada running on a Dawn Blue Gene/P supercomputer simulated 1 billion neurons with 10 trillion synapses [11].

ANN software implementations are processor intensive hence, the CPU is the bottleneck in algorithm processing speed. As such, in order to speed up the processing of an ANN implementation, the CPU clock rate needs to be increased, or specialized hardware needs to be developed [12, 13].

The ANN algorithms have also been developed in hardware. These have been done on a single FPGA, a network of FPGAs and in silicon. These hardware implementations require a datapath and control logic for each neuron as well as a physical synapse connection between them [14, 15]. Several custom hardware implementations of ANN algorithms exist. Examples include BrainScaleS [16], MoNeta [17], and IFAT 4G [18, 19]. Application specific ANN implementations also exist for solving fuzzy logic [20] and trigonometric [21] problems.

Constraints on these hardware designs are typically the number of connections possible between neurons and the amount of chip area per neuron [22]. Using common network techniques, it is possible to set up routing protocols which allow for dynamic route creation between neurons [23, 24]. However, the savings in number of connections is offset by the extra chip space required for the routing logic. Multichip solutions are also viable, increasing hardware requirements at the cost of bandwidth.

Both the hardware and software approaches have their advantages and drawbacks. A hybrid hardware/software codesign approach has been explored in

[25]. Cray XD1 systems have been paired with FPGAs [14, 15] and [26] uses graphics processors to simulate neural networks. In the following pages, different options for this hardware/software codesign approach are explored.

3 Hardware/Software Codesign

Typically, developers tackle a system design problem in two separate steps, the hardware design and the software design. Hardware/software codesign has emerged as a way to envision the entire system design as both software and hardware components simultaneously, streamlining the design process. This streamlining reduces design costs, speeds up development time, limits rework and eases integration for SoC based systems.

As most designs work from existing microprocessor intellectual property, common codesign efforts involve hardware acceleration based on existing intellectual property cores. Typical hardware acceleration techniques used within the hardware/software codesign methodology are coprocessors and instruction-set extensions. Both directions are explored next with two neural network legacy application examples.

3.1 Automated Coprocessor Acceleration

Coprocessors are hardware accelerators that typically interact with a processor through the bus. Coprocessors can be specialized, offloading complex tasks away from the primary processor. Examples of common coprocessor functions include cryptographic functions, signal processing or graphics.

Altera offers an automated software to hardware generation tool [4] which takes compatible C functions and converts them into HDL which becomes integrated as coprocessors. The coprocessors are connected to the Avalon fabric which connects the NIOS processors to other SoC components such as memory controllers or parallel input/output (PIO) interfaces. Certain restrictions apply to hardware conversion such as operator limitations, data types or runtime library use within functions. To utilize the tool, a user simply selects a function within the C program which they wish to accelerate. Minimal code changes are required for quick design space exploration; however, the best mappings require code tailoring to optimally take advantage of the coprocessor architecture. C2H guidelines are available describing the best way to code the initial C program for optimal hardware generation. Fig. 1 shows the automated coprocessor generation process as a sequence of steps.

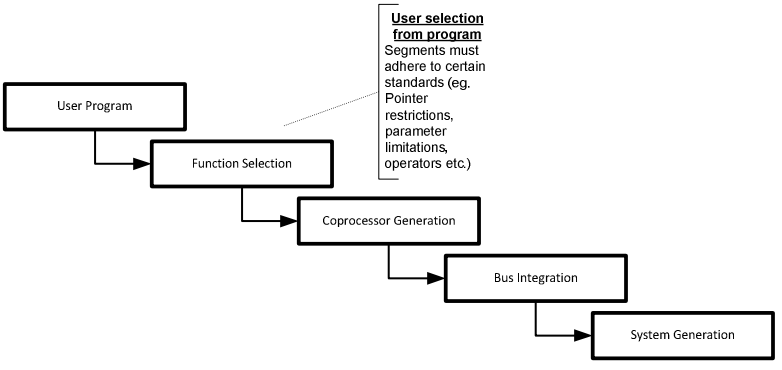


Fig. 1. Automated Coprocessor Generation.

3.2 Instruction Set Extension Identification

ASIPs are processors which have had their instruction sets augmented and tailored towards a specific application. Hardware accelerators are created as custom instructions and added into a general-purpose processor Instruction-Set Architecture (ISA). Selected custom instructions and the matching hardware description are called Instruction-Set Extensions (ISEs), and the custom instruction discovery process is called ISE identification. ISEs are typically more closely coupled and fine-grained compared to their coprocessor counterparts as they integrate directly into the datapath of the processor and become visible through the ISA, not a memory interface.

One of the most popular tools for ISEs is provided by Tensilica [3]. Tensilica is a leader in processor optimizations and instruction-set augmentations for custom hardware, all of which is dependent on their base Xtensa processor. For the purposes of this work, we examine the automated Tensilica compiler XPRES as it attempts to discover and generate instruction-set extensions. Fig. 2 illustrates the standard automated ISE design flow used in this problem domain. Automated methods examine the low level representations of programs and attempt to find patterns which would benefit from hardware acceleration within the datapath. The level of analysis usually takes place on that of the dataflow graphs (DFG) on basic blocks. Basic blocks are code segments with one entry point and one exit point. Selection is made based on constraint based optimization or a library of known template patterns. Once a selection is made, hardware is generated and the graph nodes are compressed into the new ISA instruction. The algorithm described in [5] is also used to automatically discover ISEs as shown in [6].

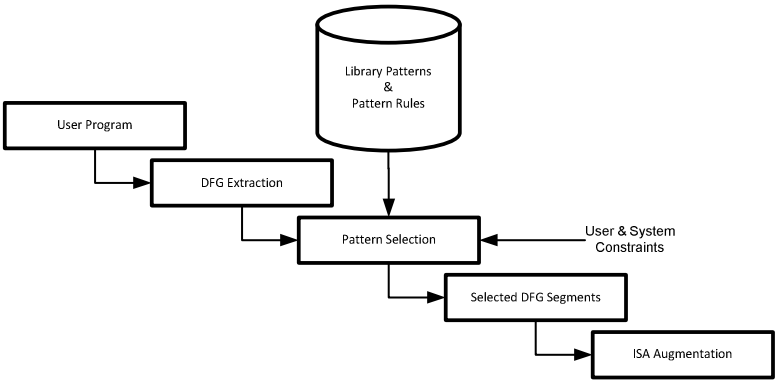


Fig. 2. Instruction-Set Extension Identification Design Flow.

4 Experimentation

Two experiments were conducted using both BAM and HAM neural network simulations as test benchmarks [1, 2]. The objective of the experiment series is to examine the various automated hardware/software codesign techniques for neural network applications. These techniques can be used to quickly and efficiently facilitate rapid development. We compare industry results to research intensive automated tools.

1. The first test utilizes the Tensilica XPRES compiler to automatically generate instruction-set extensions. A clear trade-off between performance and hardware size is available. No code augmentations were made to the benchmarks.
2. The second test uses the Altera C2H tool to convert selected C program functions into coprocessors. Minor code changes were made removing file handles to allow for conversion.

5 Results

Results of the two experiments are shown in this section. The subsequent section analyses the outcome of the tests compared to initial work presented in [6].

5.1 Experiment 1

In this experiment, the Xtensa XPRES Compiler Version 4.0.2 was used. Its function was to analyze the C implementations of the BAM and the HAM algorithms in order to generate and simulate the creation of additional hardware instructions, providing information on the additional logic that would be required to maximize speedup.

The XPRES compiler has four main acceleration techniques, FLIX, SIMD, fusion and specialization. FLIX allows for the creation of VLIW type custom instructions. SIMD allows for a single instruction to work on several data concurrently. Fusion fuses consecutive arithmetic or logic instructions to operate in a single clock cycle. Finally, specialization resizes existing instruction calls so that they are more likely to fit in a VLIW instruction. ISEs were automatically generated as TIE (Tensilica Instruction Extension), Tensilica's proprietary HDL language.

During software profiling, the XPRES compiler determined that SIMD instructions would not help to increase the performance of our algorithms so FLIX and fusion were enabled. Tensilica results are shown in Table 1.

Table 1. BAM/HAM Tensilica Results

Xtensa with XPRES	Baseline Clock Cycles	Accelerated Clock Cycles	Speedup
BAM	544,814	393,190	1.387x
HAM	9,531,886	6,341,179	1.503x

5.1.1 BAM

During the automated design space exploration, considering ISE options, tradeoffs between hardware and performance become available. Fig. 3 shows how the considered ISEs affect hardware size and performance. The maximum performance gain is achieved with the largest additional hardware of 35,003 gates. The baseline processor execution takes 544,814 clock cycles for the application. The maximum feasible performance gain comes in at 393,190 clock cycles which is relatively consistent with the plateau seen after 19,500 additional gates for the ISEs. The optimal tradeoff point between performance and additional hardware can be found in this area.

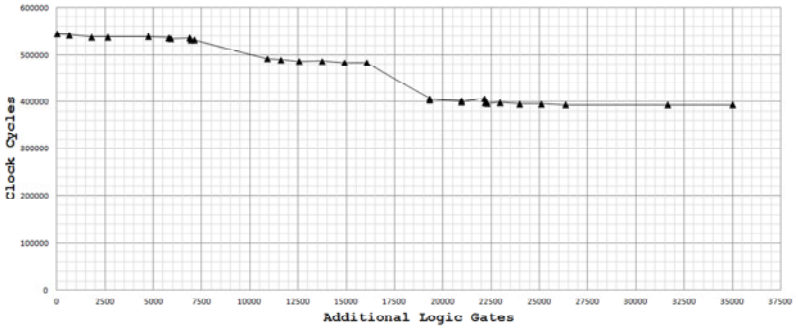


Fig. 3. Xtensa Analysis of BAM Algorithm - Clock Cycles vs. Additional Hardware Requirement.

5.1.2 HAM

There are three clear performance plateaus for the HAM benchmark after being run through the XPRES compiler as shown in Fig. 4. The baseline processor required 9,531,886 clock cycles to execute the program. Utilizing the maximum number of ISEs and around 27,000 additional gates, we can get this execution down to 6,341,179 cycles. Note that the speedup actually decreases with the final hardware additions. A maximum speedup of 1.5x was achieved with an additional 26,829 gates which can be interpolated from the data presented below.

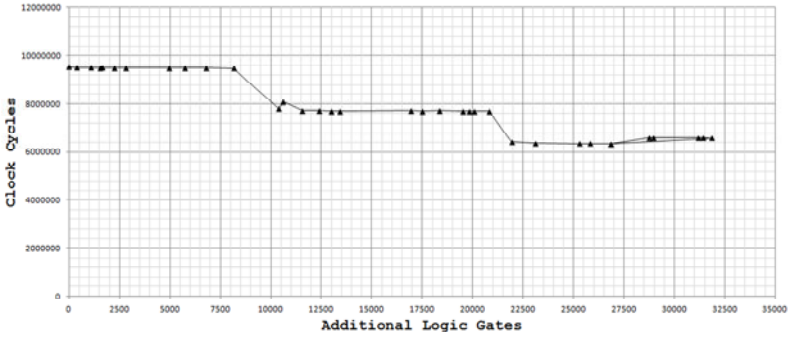


Fig. 4. Xtensa Analysis of HAM Algorithm - Clock Cycles vs. Additional Hardware.

5.2 Experiment 2

For the Altera C2H tool examination, the net initialization, weight calculation and layer propagation functions were converted to hardware since they were on the critical path. File output functions for results were then changed to terminal output functions in an effort to remove file handles. The three selected functions were

then automatically converted to hardware coprocessors via the C2H tool and minor pointer reassignments were done. The HDL was automatically generated and integrated into the complete embedded system. Results are available in Table 2. Each test was physically run on an Altera DE2 development board using a Cyclone II FPGA.

Table 2. BAM/HAM Altera C2H.

Altera C2H	Baseline Clock Cycles	Accelerated Clock Cycles	Speedup
BAM	17,445,554	12,918,407	1.350x
HAM	305,475,945	260,245,741	1.174x

5.3 Summary

The results of both experiments are available in Table 3. The summary shows the physical and simulated embedded system architectures along with the base intellectual property processors. Different hardware/software codesign methods described in the previous experiments are listed for each value.

Table 3. Results of Design Space Exploration.

System Name	Best Execution Time (s)		CPU	Memory	Clock Rate (MHz)
	BAM	HAM			
Baseline NIOS II	0.17446	3.05476	NIOS II/f	4KB I-Cache, 2KB D-Cache, 8MB RAM	100
NIOS II with BAM coprocessors	0.12980	-	NIOS II/f	4KB I-Cache, 2KB D-Cache, 8MB RAM	100
NIOS II with HAM coprocessors	-	2.60244	NIOS II/f	4KB I-Cache, 2KB D-Cache, 8MB RAM	100
Baseline Xtensa	0.00545	0.09532	Xtensa LX 4.0	1KB I-Cache, 1KB D-Cache, RAM not configured in simulation	100
Customized Xtensa	0.00393	0.06341	Xtensa LX 4.0	1KB I-Cache, 1KB D-Cache, RAM not configured in simulation	100

5.4 Comparison and Discussion

Various automated codesign methods were explored in this work including coprocessor and ISE solutions. Speedups were achieved with little development effort on the part of the engineer or developer. It should be noted that the baseline Tensilica compilation with no additional hardware had the fastest baseline execution time. It is believed that the underlying VLIW architecture and O3 compiler flags caused such a high-performance baseline for the Xtensa processor. There is a clear initial decision that must take place where unaugmented targets are compared. It may in fact be possible that a proper target selection may alleviate the need for embedded system augmentation through coprocessors or ISEs. A tailored ISA or the use of coprocessors allows for the possibility of lower power consumption while maximizing performance.

During the automated analyses for the ISEs, it was noted that initialization and the propagation stages posed the most computationally intensive. The manually selected functions in C2H closely matched where the low-level ISE identifications occurred and again with the generated TIE segments from Tensilica. This is similar to the results obtained in [6] with a major selected graph segment shown below in Fig. 5.

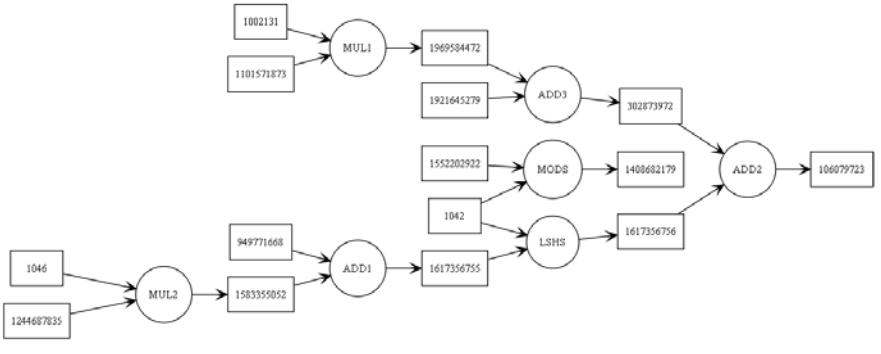


Fig. 5. An identified ISE from the BAM program which initializes the network with data.

This particular identified segment can accelerate an operation found within three levels of for-loops for initialization. The instruction contains two multipliers, three adders, a modular arithmetic unit, and a signed left shift. The circles represent computation, and the operation type is noted. The square boxes represent registers, and the numbers in these boxes are the labels of the associated variables extracted from the low-level intermediate representation (LIR) of the program within the ISE identification pass of the COINS compiler as described in [5]. The arrows represent signal paths in the ISE datapath.

In our previous work [6], based off bleeding-edge custom hardware identification algorithms, a speedup of between 1.18x and 2.1x was found for the example applications. These results are close to the commercial tools available on

the market today as concluded from the results shown in the previous section. A direct comparison cannot be easily made as each target architecture is different and different compilers were used.

The use of both ISEs and coprocessors simultaneously has been explored in works such as [27], increasing the design space exploration breadth while using similar techniques. Here, we were able to successfully utilize automated codesign tools with neural network applications to achieve acceptable performance results with minimal effort. It is believed that more in-depth engineer involvement in the codesign process will yield increased performance; however, the goal of our work is to show that automated tools are in fact ready for primetime with regards to legacy neural network application code.

6 Conclusions

In this work, we have examined the use of ever advancing hardware/software codesign tools for legacy neural network applications. Modest performance speedups were easily achieved with little developer involvement. There is a clear benefit to using existing hardware/software codesign tools for legacy applications. They reduce both time to market and staff knowledge requirements while offering performance benefits that are comparable to hand rolled solutions.

References

[1] Kutza, K.: Neural Networks at your Fingertips – The Hopfield Model , http://www.neural-networks-at-your-fingertips.com/hopfield.html (accessed September 22, 1996)

[2] Kutza, K. (1996) Neural Network at your Fingertips – Bidirectional Associative Memory,http://www.neural-networks-at-your-fingertips.com/bam.html (accessed September 22, 2011)

[3] Tensilica: XPRES Product Brief (2005), http://www.tensilica.com/uploads/pdf/XPRES%201205.pdf (accessed September 22, 2011)

[4] Altera: NIOS II C2H Compiler User Guide (2009), http://www.altera.com/literature/ug/ug_nios2_c2h_compiler.pdf (accessed September 22, 2011)

[5] Shapiro, D.: Design and implementation of instruction set extension identification for a multiprocessor system-on-chip hardware/software co-design toolchain. Dissertation. University of Ottawa, Ottawa, Canada (2009)

[6] Shapiro, D., Parri, J., Desmarais, J.-M., Groza, V., Bolic, M.: ASIPs for artificial neural networks. In: Proceedings of 6th IEEE Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisoara, Romania, pp. 529–533 (2011)

[7] Dreyfus, G.: Neural Networks: Methodology and Applications. Springer, Heidelberg (2005)

[8] Kosko, B.: Bidirectional associative memories. IEEE Trans. on Syst. Man and Cybern. 18, 49–60 (1988)

[9] Hopfield, J.: Neural networks and physical systems with emergent collective computational Abilities. Proc. of the Natl. Acad. of Sci. 79, 2554–2558 (1982)
[10] Whitson, G., Wu, C., Taylor, J., Ermongkonchai, A.: CANS: an interactive neural network system for Cray supercomputers. In: Proceedings of 1992 ACM/SIGAPP Symp. on Appl. Comput (SAC 1992), Las Vegas, USA, pp. 665–668 (1992)
[11] Ananthanarayanan, R., Esser, S.K., Simon, H.D., Modha, D.S.: The cat is out of the bag: cortical simulations with 109 neurons, 1013 synapses. In: Proceeding of Conf on High Perform Comput Netw Storage and Analysis (SC 2009), pp. 63:1–63:12 (2009)
[12] Shapiro, D., Digeser, P., Tubolino, M., Klemm, M., Sikora, A., Bolic, M.: Instruction set extensions for computation on complex floating point numbers. In: Proceedings of IEEE 26th Convention of Electr. and Electron Eng. in Israel (IEEEI 2010), Eilat, Israel, pp. 938–941 (2010)
[13] David, G., Darin, P.: Automatic generation of application specific processors. In: Proceedings of 2003 Int. Conf. on Compil. Archit. and Synth. for Embed. Syst. (CASES 2003), New York, United States, pp. 137–147 (2003)
[14] Vutsinas, C., Taha, T., Rice, K.: A context switching streaming memory architecture to accelerate a neocortex model. Microprocess Microsyst. 33, 117–128 (2009); doi:10.1016/j.micpro.2008.08.011
[15] Rice, K., Taha, T., Vutsinas, C.: Scaling analysis of a neocortex inspired cognitive model on the Cray XD1. J. Supercomput. 47, 21–43 (2009); doi:10.1007/s11227-008-0195-z
[16] BrainScaleS: The BrainScaleS project (2011), http://brainscales.kip.uni-heidelberg.de/index.html (accessed September 22, 2011)
[17] Versace, M., Chandler, B.: The brain of a new machine. IEEE Spectrum 47(12), 30–33 (2010)
[18] Russell, A., Orchard, G., Dong, Y., Mihalas, S., Niebur, E., Tapson, J., Etienne-Cummings, R.: Optimization methods for spiking neurons and networks. IEEE Trans. on Neural Netw. 21, 1950–1962 (2010)
[19] Johns Hopkins University Computational Sensory-Motor Systems Lab Integrate and Fire Array Transceiver (2011), http://etienne.ece.jhu.edu/projects/ifat/index.html (accessed September 22, 2011)
[20] Parri, J., Ratti, S.: Trigonometric function approximation neural network based coprocessor. In: Proceedings of 2nd Microsyst. and Nanoelectron Res. Conf. (MNRC 2009), Ottawa, Canada, pp. 148–151 (2009)
[21] Thareja, V., Bolic, M., Groza, V.: Design of a fuzzy logic coprocessor using Handel-C. In: Proceedings of 2nd Int. Conf. on Soft. Comput. Appl (SOFA 2007), Gyula, Hungary, pp. 83–88 (2007)
[22] Salapura, V., Gschwind, M., Maischberger, O.: A fast FPGA implementation of a general purpose neuron. In: Proceedings of 4th Int. Workshop on Field-Program Log and Appl. (FPL 1994), pp. 175–182 (1994)
[23] Tsui, K., Yanco, H.: Simplifying wheelchair mounted robotic arm control with a visual interface. In: Proceedings of AAAI Spring Symp. on Multidiscip. Collab. for Soc. Assist. Robot. (AAAI SS07-2007), Palo Alto, USA, pp. 247–251 (2007)
[24] Wang, Q., Li, A., Li, Z., Wan, Y.: A Design and Implementation of Reconfigurable Architecture for Neural Networks Based on Systolic Arrays. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3973, pp. 1328–1333. Springer, Heidelberg (2006)

[25] Esmaeilzadeh, H., Farzan, F., Shahidi, N., Fakhraie, S., Lucas, C., Tehranipoor, M.: NnSP: embedded neural networks stream processor. In: Proceedings of 48th Midwest Symp. on in Circuits and Syst. (MWSCAS 2005), Cincinnati, USA, vol. 1, pp. 223–226 (2005)

[26] Nageswaran, J.M., Dutt, N., Krichmar, J.L., Nicolau, A., Veidenbaum, A.V.: 2009 special issue: A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors. Neural Netw. 22, 791–800 (2009)

[27] Parri, J., Bolic, M., Groza, V.: Custom instruction hardware integration within a SoC hybrid environment. In: Proceedings of 6th IEEE Symposium on Applied Computational Intelligence and Informatics (SACI 2011), pp. 517–522 (2011)

Pragmatic Method to Obtain Optimal Control Laws for Small Windgenerators

Radu Boraci, Octavian Prostean, Nicolae Budisan, and Cosmin Koch-Ciobotaru

"Politehnica" University of Timisoara, Department of Automation and Applied Informatics, Bd. V. Parvan 2, 300223 Timisoara, Romania
{radu.boraci,octavian.prostean,nicolae.budisanu, cosmin.koch}@aut.upt.ro

Abstract. The chapter presents a pragmatic method to obtain optimal implementable control methods of variable speed fixed blades small windgenerators, depending of available information values about the control object (wind speed, rotation speed, air density, blade's position in the air flow). The elaborated method presumes the existence off a laboratory model having: an electromechanical analog model of considered turbine in accordance with a pre-established turbine theoretical mathematical model and the full generator-grid system, identical with the implemented on the site one. The pre-established turbine theoretical model allows to calculate for different wind speed values, in the established working domain, the optimum turbine values $n_{T\,opt\,k}$, $M_{T\,opt\,k}$, $P_{T\,opt\,k}$, k=1, 2, ... Having the obtained optimum turbine operation values, these optimum regimes may be experimentally obtained on the laboratory model, and all operation values of different elements of the conversion line "generator - power electronic interface – grid" may be measured. Different obtained parameter pairs may be use to determine regression functions that may be used as optimum wind control laws. Considering the hardware structure of studied windgenerator were chosen some optimum control laws. The obtained control laws was implemented and verified on Matlab / Simulink model considering different wind speed variations. The simulation results confirm the opportunity and the quality of the adopted optimal control law.

1 Introduction

In literature, a large number of papers deal with the operation regimes of windgenerators, small ones inclusively, and their control strategies implemented in dedicated software [1 - 4].

The state of the art of small windgenerators shows the following main trends:

- direct coupling of the wind turbine and the electrical generator,
- permanent synchronous generator,

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 239–248.
springerlink.com

- unregulated turbine blades,
- lack of wind speed transducer,
- lack of rotation speed transducer,
- wind turbine power optimum control.

The considered in this chapter windgenerator has the mentioned features.

At "Politehnica" University from Timisoara, Romania UPT, was studied the control system of 5 kW horizontal windgenerator, under one European Economic Aria Grant [2], following the next main targets:

1) The start and grid/consumers connection of the wind generator, at a wind speed at which the wind turbine power is larger than that of the power losses in the conversion system and, consequently, the electrical energy delivery begins [1, 2]

2) To assure the wind generator operation at optimum power of the wind turbine for wind speed range between the start and rated values [1, 2, 5]

3) To assure the windgenerator operation at $I_{HDC\ rated} = const$, for wind speed range upper the rated one.

4) Wind generator stop in normal operation conditions at the human operator request

5) Wind generator automatic safety stop in fault operation conditions (extreme wind speed, short-circuit, vibrations, blade damage, mechanical breakdown) [6, 7].

The structure of the considered windgenerator is given in Fig. 1, where:

- WT – wind turbine rotor
- BRK – safety system brake
- ELBRK – electrical brake resistor
- PMSG – permanent magnet synchronous generator
- DB – diode bridge
- HDC – hybrid dc-dc converter
- BVI – boost voltage inverter.

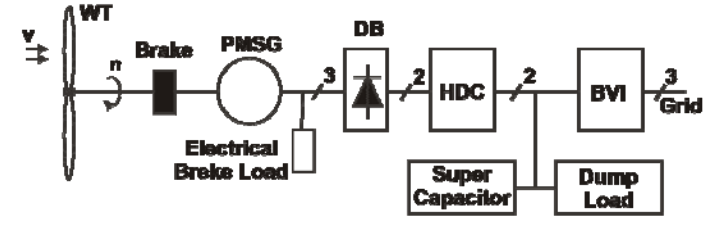


Fig. 1. Wind generator structure.

In this chapter, the second mentioned operation regime of the UPT-EEA 5 kW wind generator there is considered [8].

2 Optimal Operation Regime and Control Strategy

At present day are known two approaches of the optimal control:

- Maximum power point tracking MPPT;
- Control based on the turbine mathematical model knowledge.

In the considered application there was considered the second mentioned approach, with the following mathematical model

$$M_T = C_M \rho S R v^2 / 2 \tag{1}$$

from witch results

$$P_T = M_T \omega \tag{2}$$

where: S = 19.6 m^2, R = 2.50 m, v - wind speed, ω - turbine rotor angular speed, C_M - torque coefficient given by the relation:

$$C_M = C_{M0} + a \cdot TSR - b \cdot TSR^{2.5} \tag{3}$$

with: tip speed ratio TSR_0 =3, CM_0=0.0222, a =0.0986, b = 0.0113.

It is a known fact the importance of the preliminary experimental test of technical systems as a whole or of parts of such systems, from different reasons:

a) To verify the correctness of theoretical approaches/estimation of their characteristics used in their design for fabrication or for preliminary considerations of their operation characteristics;

b) To verify the fabrication and assembling technology.

The tunnel tests are specific for air-dynamic devices or systems.

In case when tunnel tests are not accessibly, a windgenerator laboratory model may be used, with full scale or reduced power real and / or analogical elements.

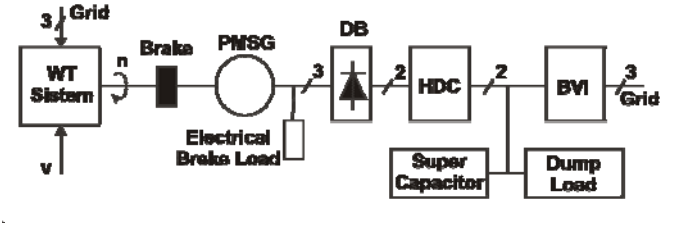


Fig. 2. The mixed physical-analog laboratory model of UPT-EEA 5 kW windgenerator.

In Fig. 2 there is presented the structural scheme of the considered windgenerator laboratory model.

The laboratory model contains the full scale elements of the considered windgenerator except the wind turbine, instead of it an analog electromechanical model was adopted. The turbine model consists from an induction motor supplied by a frequency converter having as preset values the turbine torques corresponding to every wind speed and every rotation speed of the real turbine (1).

It must be mentioned that it exists many methods of controlling the optimal power of the wind turbine, considering one of a multitude pairs of working values in optimal operating regimes:

- $n_{opt}, P_{T\,opt}, \left(P_G\right)_{P_{T\,opt}}, V_{G\,opt}, I_{G\,opt}, I_{DB}, V_{DB}, P_{HDC\,opt}, V_{HDC\,opt}, I_{HDC\,opt} = F\left(v\right)$;
- $P_{T\,opt}, \left(P_G\right)_{P_{T\,opt}}, V_{G\,opt}, I_{G\,opt}, I_{DB}, V_{DB}, P_{HDC\,opt}, V_{HDC\,opt}, I_{HDC\,opt} = F\left(n\right)$;
- $V_{G\,opt}, I_{G\,opt}, I_{DB}, V_{DB}, P_{HDC\,opt}, V_{HDC\,opt}, I_{HDC\,opt} = F(\left(P_G\right)_{P_{T\,opt}})$;
- $I_{HDC} = V_{G\,opt}$;
- $I_{HDC\,opt} = F\left(V_{HDC}\right)$;
- et al.

For the UPT-EEA 5 kW windgenerator the optimal control law $I_{HDC\,opt} = F(V_{HDC})$ was considered, because of accessibility of the parameters $I_{HDC\,opt}$ and V_{HDC}.

The mentioned values may be theoretically determined using the mathematical models of the windgenerator components or may be obtained from the tunnel or from laboratory model experimental results as in the considered study case, results given in Table 1.

In Table 1 the values of n_{opt} and M_T corresponding to $P_{T\,opt}$ were calculated in advance using the wind turbine power model obtained from (1) and (2) are given.

The output values, $V_{HDC\,opt}, I_{HDC\,opt}$ where determined experimentally, for each pair $n_{opt}, \left(M_T\right)_{P_{T\,opt}}$.

Considering the experimentally obtained pairs $(V_{HDC\,opt\,k}, I_{HDC\,opt\,k})$, $k = 1,2$ ..., of the two variables of adopted control function $I_{HDC\,opt} = F(V_{HDC})$, may be obtained the needed control function using the method of regression functions.

Doing as mentioned above, was obtained the control function (4) represented in Fig. 3.

$$I^{*}_{HDC\,opt} = -89.590998 + 7.8627425\,V_{HDC} - \\ -0.18449363\,V^{2}_{HDC} + 0.0014904265\,V^{3}_{HDC} \tag{4}$$

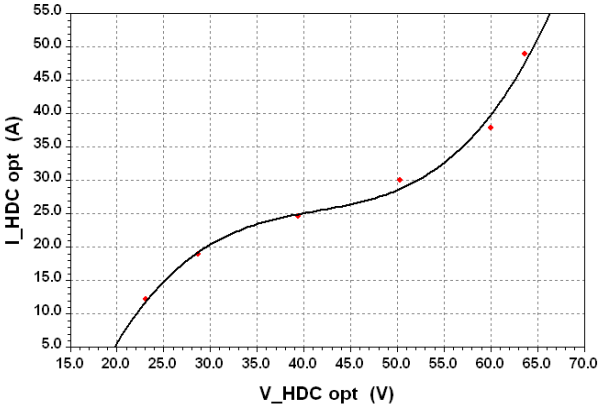


Fig. 3. The UPT-EEA 5 kW windgenerator optimum control function.

By means of the obtained control function, can be preset, in dependence of the V_{HDC} , the optimum value of $I_{HDC\,opt}$ corresponding to the optimum value of the turbine power $P_{T\,opt}$, as it is shown in Fig. 3.

Table 1. Experimental determined values $I_{HDC\,opt}$ AND V_{HDCopt} for calculated in advance optimum $(n_{OPT}, (M_T)_{PTOPT})$ windgenerator regime

n_{opt} [rpm]	$(M_T)_{PT\,opt}$ [Nm]	$V_{HDC\,opt}$ [V]	$I_{HDC\,opt}$ [A]	$P_{T\,opt}$ [W]
46	66,90	23,1	12,19	280,58
57	105,60	28,7	19,01	545,67
69	150,80	39,4	24,68	970,08
80	206,40	50,3	30,01	1506,79
92	268,00	60,0	37,97	2280,02
103	340,80	63,7	49,01	3184,58

Using other experimental values, measured as those shown in Table 1, may be obtained, too, if this is practically more convenient, other control functions/laws

mentioned in Section 1. There were calculated two control laws $I_{HDC} = F(n)$ and $P_{HDC\ input} = F(n)$. After some simplifications of the obtained corresponding regression functions, were retained and experimented successfully the following control laws.

$$I_{HDC} = 0.0008\, n^3 \tag{5}$$

and

$$P_{HDC\,input} = 0.0028\, n^3 . \tag{6}$$

3 Simulation Model

The simulation model regarding the control of transient optimal regimes of the considered above wind generator is developed in Matlab / Simulink and presented in Fig. 4 [2, 8], where the operational characteristics of the system elements are known from manufacturer tests as well from experimental determination in the UPT laboratories.

The controller from Fig. 5 is presented in Fig. 6.

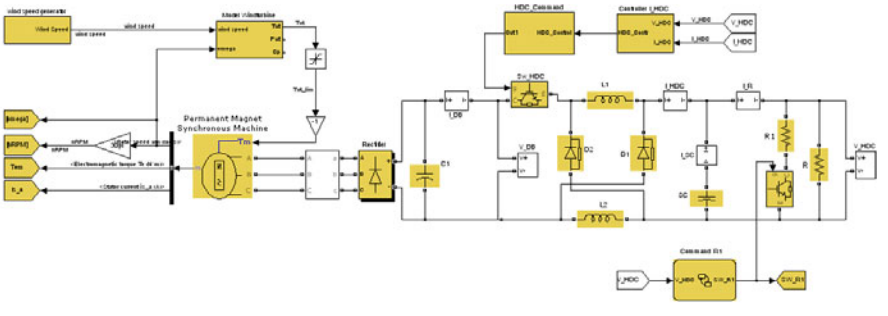


Fig. 4. Simulation model for the wind generator control of transient optimal regimes.

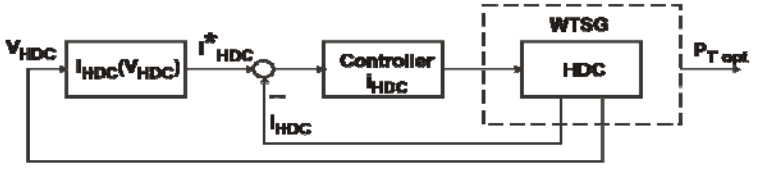


Fig. 5. The structure of the turbine power optimization system.

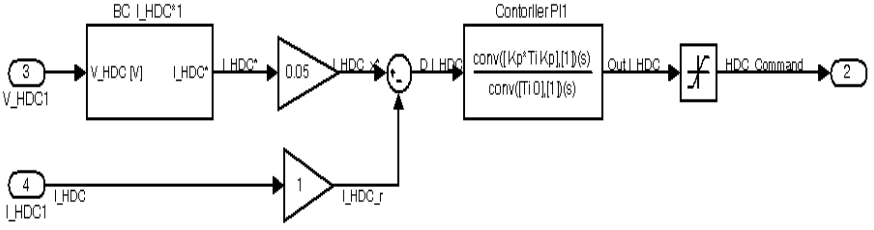


Fig. 6. Controller simulation model.

4 UPT-EEA 5kW Windgenerator Optimum Control Simulation

The simulation model regarding the control of transient optimal regimes of the considered above wind generator is developed in Matlab / Simulink and presented in Fig. 5 [2, 4], where the operational characteristics of the system elements are known from manufacturer tests as well from experimental determination in the UPT laboratories.

Using the obtained simulation model from Fig. 5 were studied some transient regimes of the windgenerator, considering, from many possible:

a) specific increasing wind speed of 15% ... 100%, expressed by the relation:
- step function;
- increasing linear function;

b) admitted by the considered windgenerator small wind gusts [6], expressed by the relation (represented in Fig. 7):

$$V(z,t)=\begin{cases} V(z);\ t<0 \ \text{ and } \ t>T \\ V(z)-0.37V_{gustN}\sin(3\pi t/T)\left(1-\cos(2\pi t/T)\right);\ 0\le t\le T \end{cases} \tag{7}$$

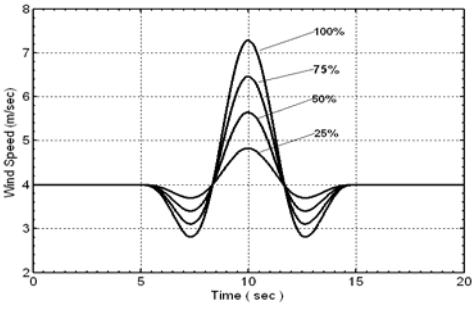


Fig. 7. Small wind gusts example.

Simulation results of above mentioned regimes are presented in the following:

- in Fig. 8, for 50% wind speed step variation;
- in Fig. 9, for 50% wind speed increasing linear function;
- in Fig. 10, for 50% small gusts wind speed.

In all cases, the transient values of specific parameters do not overcome the admissible ones.

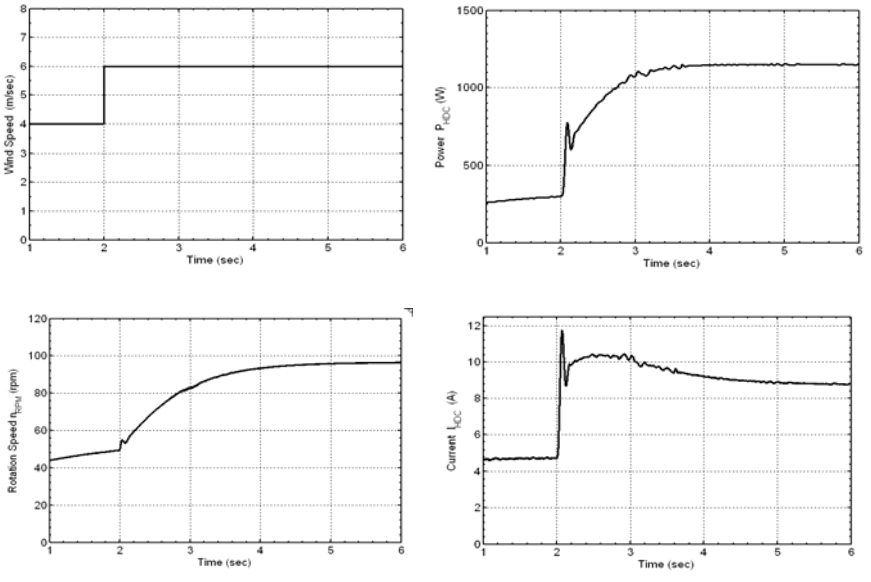


Fig. 8. Simulation results for wind speed step variation.

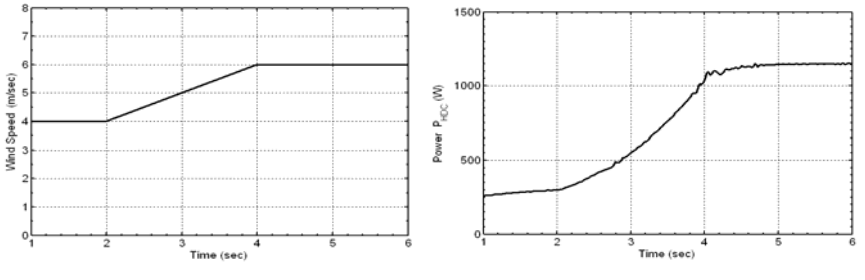


Fig. 9. Simulation results for wind speed linear increasing function.

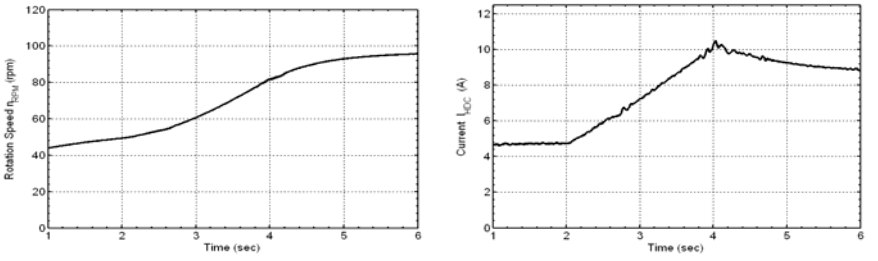


Fig. 9. *(continued)*

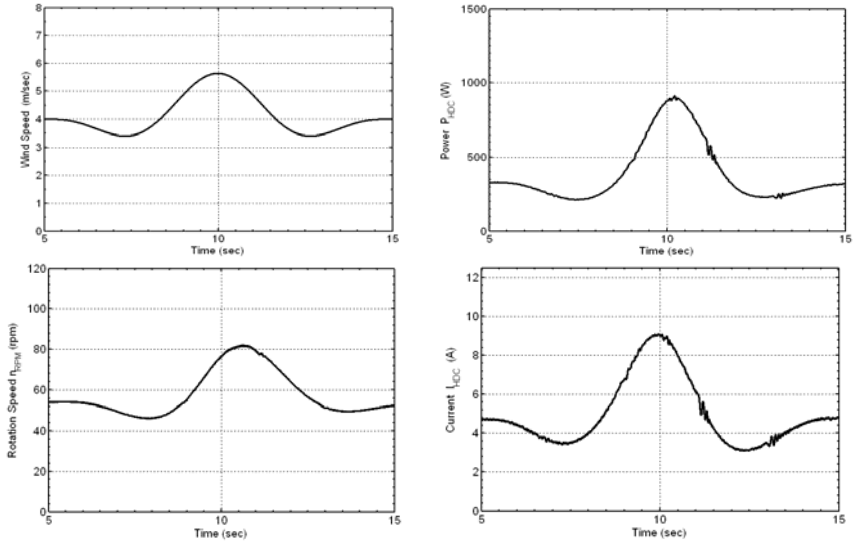


Fig. 10. Simulation results for small wind gusts.

5 Conclusions

Two approaches to control the optimal power of wind turbines are known: the maximum power point tracking (MPPT) and the control based on the turbine mathematical model knowledge. The second mentioned approach was considered in the application presented in this chapter.

From the many existing solutions to control the optimal power of wind turbines, in this chapter, for the UPT-EEA 5 kW windgenerator there was accepted the control law $I_{HDC\ opt} = F(V_{HDC})$ for the considerations mentioned in Section 2.

The simulation results confirm the opportunity and the quality of the adopted optimal control law.

Using the experimental results from Table 1 other control functions / laws mentioned in Section 2 may be obtained, too, if this is more convenient.

Acknowledgments. This work was supported by the European Economic Area (EEA) project "RO 018 Improvement of the Structures and Efficiency of Small Horizontal Axis Wind Generators with Non-Regulated Blades". This work was partially supported by the strategic grant POSDRU 2009 project ID 50783 of the Ministry of Labor, Family and Social Protection, Romania, co-financed by the European Social Fund – Investing in People.

References

[1] Chen, Z., Guerrero, J.M., Blaabjerg, F.: A review of the state of the art of power electronics for wind turbines. IEEE Trans. Power Electron (2009);
[2] doi:10.1109/TPEL.2009.2017082
[3] Budisan, N., Muntean, N., Boraci, R., et al.: Electronic conversion system and speed-control strategy for small wind generators. In: Proc. Int. Conf. Comput. Cybern and Technol. Inform, ICCC-CONTI (2010); doi:10.1109/ICCCYB.2010.5491231
[4] Hong, Y.Y., Lu, S.D., Chiou, C.S.: MPPT for PM wind generator using gradient approximation. Energy Convers. Manag. (2009); doi:10.1016/j.enconman.2008.08.035
[5] Boraci, R., Koch-Ciobotaru, C., Prostean, O., Budisan, N.: Experimental determination of an optimal control law of a small windgenerator. In: Proc. IEEE Int. Symp. Appl. Comput. Intel. and Inform. (2011); doi:10.1109/SACI.2011.5873063
[6] Lazarov, V., Roye, D., Spirov, D., Zarkov, Z.: Study of control strategies for variable speed wind turbine under limited power conditions. In: Proc. Power Electron and Motion Control Conf. (2010); doi:10.1109/EPEPEMC.2010.5606574
[7] Koch-Ciobotaru, C., Boraci, R., Filip, I., Vasar, C.: Study of brake transient regimes for a small wind generator. In: Proc. Int. Symp. on Exploit Renew. Energy Sources (2011); doi:10.1109/EXPRES.2011.5741797
[8] CEI/IEC 61400-2, Part 2: Design requirements for small wind turbines International Standard, `http://webstore.iec.ch/preview/info_iec61400-2%7Bed2.0%7Den_d.pdf` (2006)
[9] Koch-Ciobotaru, C., Boraci, R., Filip, I., et al.: Control Strategy for a Variable-Speed Wind Turbine Using DC Bus Measurements. In: Proc. IEEE Int. Symp. on Intel Syst. and Inform. (2010); doi:10.1109/SISY.2010.5647425

Applicability of Asymptotic Tracking in Case of Type 1 Diabetes

Péter Szalay and Levente Kovács

Department of Control Engineering and Information Technology, Budapest University of Technology and Economics, Magyar tudósok krt. 2, H-1117 Budapest, Hungary
{szalaip,lkovacs}@iit.bme.hu

Abstract. The alarming increasing tendency of diabetes population attracts technological interest too. From an engineering point of view, the treatment of diabetes mellitus can be represented by an outer control loop, to replace the partially or totally deficient blood glucose control system of the human body. To acquire this "artificial pancreas" a reliable glucose sensor and an insulin pump is needed as hardware, and a control algorithm to ensure the proper blood glucose regulation is needed as software. The latter is a key point of the diabetes "closing the loop" problem and its primary prerequisite is a valid model able to describe the blood glucose system. In the current chapter one of the most widely used and complex nonlinear model will be investigated with a dual purpose. Specific control aspects are discussed in the literature only on linearized versions; however, differential geometric approaches give more general formalization. As a result our first aim is to hide the nonlinearity of the physiological model by transforming the control input provided by a linear controller so that the response of the model would mimic the behavior of a linear system. Hence, the validity of linear controllers can be extended from the neighborhood of a working point to a larger subset of the state-space bounded by specific constraints. On the other hand, applicability of the nonlinear methodology is tested on a simple PID control based algorithm compared with LQG optimal method. Simulations are done under MATLAB on realistic input scenarios. Since the values of the state variables are needed Kalman filtering is used for state estimation.

1 Introduction

According to the data provided by the World Health Organization (WHO), diabetes mellitus is predicted to be the "disease of the future" especially in the developing countries. The diabetic population (4% of World's population in 2000) is predicted to be doubled by 2030 (being estimated to be 5,4% of the World's total

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 249–260.
springerlink.com

population) [1]. Recent statistics show the same prevalence within the 2010-2030 period, but the overall diabetes population increased significantly (6.4% in 2010 and estimated to be 7.7% in 2030) [2].

The normal blood glucose concentration level in the human body varies in a narrow range (70 - 120 mg/dL). If for some reasons the human body is unable to control the normal glucose-insulin interaction (e.g. glucose concentration level is constantly out of the above mentioned range), diabetes is diagnosed. The consequences of diabetes are mostly long-term; among others, diabetes increases the risk of cardiovascular diseases, neuropathy, retinopathy [3]. Hence, diabetes mellitus is a serious metabolic disease, which should be artificially controlled.

From an engineering point of view, the treatment of diabetes mellitus can be represented by an outer control loop, to replace the partially or totally deficient blood glucose control system of the human body. The quest for artificial pancreas can be structured in three different tasks [4, 5]:

- continuous glucose sensor for measurements;
- insulin pump for infusion;
- control algorithm.

To design an appropriate control, an adequate model is necessary. In the last few decade different mathematical models of the human blood glucose system appeared. A brief overview can be found in [6]. The minimal model [7] proved to be the simplest one and was developed for type 1 diabetes patients under intensive care.

However, its simplicity proved to be its disadvantage, since in its formulation a lot of components of the glucose-insulin interaction were neglected.

To address this problem other more general, but more complicated models appeared [8-12]. The nonlinearity in each of them makes artificial pancreas a challenging control problem and the applied strategies are usually developed for the linearized version of the models.

However, generalization of this concept can be realized using control methods developed directly for nonlinear systems [13-16]. The current chapter investigates this aspect in terms of differential geometric approach. Our aim is to hide the nonlinearity of the physiological model by transforming the control input provided by a linear controller so that the response of the model would mimic the behavior of a linear system. Hence, the validity of linear controllers can be extended from the neighborhood of a working point to a larger subset of the state-space bounded by specific constraints. Although, this approach might not increase the performance of the controllers to a great extent, but the reliability of tight glucose control can directly affect the quality of life of a type-1 diabetes patient. Our goal is a control algorithm whose stability and performance can be guaranteed and mathematically proven according to adequate medical specifications.

The chapter is structured as follows. First, the widely used and complex nonlinear glucose-insulin model of Magni et al [9] is presented. This is followed by a

brief theoretical summary of the applied nonlinear control theory aspects (exact linearization via state feedback, asymptotic output tracking) and aspects of Kalman filtering. Section 4 presents and explains the obtained results and simulations. Finally, Section 5 concludes the chapter and formulates further directions.

2 The Investigated Model

The model of Magni et al. [9] dates back to the complex glucose-insulin model of [8]. The in-silico glucose metabolism model of [9] is presented with compartments to describe subcutaneous insulin delivery and subcutaneous continuous glucose monitoring, as well as an intestinal glucose absorption model.

Although the model has a modular build, here it will be briefly summarized as a single system. Details can be found in [8, 9].

The system has 10 state variables:

- G_M – subcutaneous glucose concentration (mg/dL);
- G_p – glucose in plasma and rapidly equilibrating tissues (mg/kg);
- G_t – glucose in slowly equilibrating tissues (mg/kg);
- X – insulin in interstitial fluid (pmol/L);
- I_d, I_1 – state variables for delayed insulin signal (pmol/L);
- I_p – insulin mass in plasma (pmol/kg);
- I_l – insulin mass in liver (pmol/kg);
- S_2 – monomeric insulin in the subcutaneous tissue (pmol/kg);
- S_1 – polymeric insulin in the subcutaneous tissue (pmol/kg).

The inputs of the system are the u injected insulin flow (pmol/min) and the Ra glucose rate of appearance in plasma (mg/min). The parameters of the model are as follows: V_G the distribution volume of glucose (dL/kg); U_{ii} the insulin-independent glucose utilization (mg/kg/min); k_1, k_2 rate parameters of the glucose subsystem (min^{-1}); k_{e1} the renal glomerular filtration rate (min^{-1}); k_{e2} renal threshold (mg/kg); V_i insulin distribution volume (L/kg); m_1, m_2, m_3 and m_4 rate parameters of the insulin subsystem (min^{-1}); BW body weight (kg); k_{p1} the extrapolated endogenous glucose production at zero glucose and insulin (mg/kg/min); k_{p2} the liver glucose effectiveness (min^{-1}); k_{p3} the indicator of effect of a delayed insulin signal (mg·L/kg/min/pmol); k_i the parameter of delayed insulin signal (min^{-1}); I_b the basal level of plasma insulin concentration (pmol/L); p_{2U} the rate constant of insulin action (min^{-1}); K_{m0}, K_{mx}, V_{m0} and V_{mx} model parameters for insulin-dependent glucose utilization (-); k_d degradation constant (-); k_{a1}, k_{a2} absorption constants (-); $k_{s.c.}$ rate constant for the subcutaneous glucose compartment (-):

$$
\begin{aligned}
\dot{G}_M(t) &= -k_{s.c.} G_M(t) + \frac{k_{s.c.}}{V_G} G_p(t) \\
\dot{G}_p(t) &= -k_1 G_p(t) + k_2 G_t(t) - k_{e1}\, max\{0, G_p(t) - k_{e2}\} + \\
&\quad + \frac{Ra(t)}{BW} - U_{ii} + max\{0, k_{p1} - k_{p2} G_p(t) - k_{p3} I_d(t)\} \\
\dot{G}_t(t) &= k_1 G_p(t) - k_2 G_t(t) - \frac{V_{mx} G_t(t) X(t)}{K_m + G_t(t)} - \frac{V_{m0} G_t(t)}{K_m + G_t(t)} \\
\dot{X}(t) &= -p_{2U} X(t) + \frac{p_{2U}}{V_i} I_p(t) - p_{2U} I_b \\
\dot{I}_d(t) &= -k_i I_d(t) + k_i I_1(t) \\
\dot{I}_1(t) &= -k_i I_1(t) + \frac{k_i}{V_i} I_p(t) \\
\dot{I}_p(t) &= -(m_2 + m_4) I_p(t) + m_1 I_1(t) + k_{a2} S_2(t) + k_{a1} S_1(t) \\
\dot{I}_1(t) &= -m_2 I_p(t) + (m_1 + m_3) I_1(t) \\
\dot{S}_2(t) &= -k_{a2} S_2(t) + k_d S_1(t) \\
\dot{S}_1(t) &= -(k_{a1} + k_d) S_1(t) + \frac{u(t)}{BW}
\end{aligned}
\tag{1}
$$

A big advantage of the model is the integrated, three-compartment intestinal glucose absorption model [8]. This describes the glucose transit through the stomach and intestine to the plasma in case of enteral feeding.

There are some differences in the notations compared to [9]. Here endogenous glucose production (*EGP*) is defined as follows:

$$EGP = max\{0, k_{p1} - k_{p2} G_p(t) - k_{p3} I_d(t)\} \tag{2}$$

where k_{p1} integrates three constans of [9]: the basal EGP_b value, the I_b endogenous insulin production basal value and the G_{pb} basal glucose level in plasma. In other words, $k_{p1} = EGP_b + k_{p2} G_{pb} + k_{p3} I_b$.

Another difference is that although the three-compartment model representing intestinal glucose absorption is used in the simulations to acquire glucose absorption profile, it is not regarded as part of the model. Detailed description of the absorption model can be found in [8]. The dimension of the *Ra* and *u* is (mg/min) and (pmol/min) instead of (mg/kg/min) and (pmol/kg/min) respectively; therefore division with body weight (*BW*) is added.

3 Methods: Nonlinear Control and Kalman Filtering

The concept of exact linearization of a nonlinear system via nonlinear state feedback control was introduced in [16]. Consider a SISO nonlinear system:

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + g_i(x(t))u(t) \\ y(t) &= h(x(t)) \end{aligned} \tag{3}$$

where f and g are smooth R^n-valued mappings and h is a smooth real-valued mapping defined on an open set $U \subset R^n$.

For system (3) the concepts of relative degree analysis, exact linearization and asymptotic output tracking can be considered according to [16] as it was presented in [15] in details. Let us choose a prescribed reference linear system of the form:

$$\begin{aligned} \dot{\varsigma} &= A \cdot \varsigma + B \cdot w \\ y_R &= C \cdot \varsigma \end{aligned} \tag{4}$$

Then, the control law (5) can be used for asymptotic tracking of the output of the reference system (4) with the original (3) nonlinear system.

$$\begin{aligned} u(t) = \frac{1}{L_g L_f^{r-1} h(z(t))} \Bigg(& - L_f^r h(z(t)) - \sum_{i=0}^{r-1} \left(a_i \cdot L_f^i h(z(t))\right) + \\ & + b_r \cdot CA^{r-1} Bw(t) + C \sum_{j=0}^{r} \left(b_r \cdot A^i\right) \varsigma(t) \Bigg) \end{aligned} \tag{5}$$

$L_f^i g$ represents the i-th Lie derivative of the scalar valued function g with respect to the vector field f, a_i (i=1…r) and b_j (j=0…r) are the parameters of the tracking dynamics.

Both exact linearization and asymptotic output tracking need the values of the state variables of the nonlinear system. However, in practice the only measured quantity is the intravenous or subcutaneous glucose concentration. The sensors used in measurements have relatively high noise and a sampling time of 3-5 minutes. Assuming the model represents the dynamics of the system well and we know the precise values of the model parameters an extended Kalman filter (EKF) can provide adequate state-estimation. For this case we used the algorithm which is detailed in [17].

4 Results

The model presented in (1) is not easy to be handled with methods based on differential geometry, because it has a relatively high number of state variables and it has a relative degree that is almost the half of that value. Moreover, some modifications need to be done in order to have smooth mappings in the system. After examining the connections between each compartment, we can notice that the system can be divided into three subsystems.

The first subsystem (6) has 4 states, with injected insulin flow (*u(t)*)as input, and I_p (x_1 in this case) as output. It is basically linear with a relative degree of 2:

$$\begin{aligned}
\dot{x}_1(t) &= -(m_2 + m_4)x_1(t) + m_1 x_2(t) + k_{a2} x_3(t) + k_{a1} x_4(t) \\
\dot{x}_2(t) &= m_2 x_1(t) - (m_1 + m_3)x_2(t) \\
\dot{x}_3(t) &= -k_{a2} x_3(t) + k_d x_4(t) \\
\dot{x}_4(t) &= -(k_{a1} + k_d)x_4(t) + \frac{u(t)}{BW}
\end{aligned} \tag{6}$$

The second subsystem (7) has 5 states, with I_p as input and G_p as output (noted as x_1 in this case). Its relative degree is 3:

$$\begin{aligned}
\dot{x}_1(t) &= -k_1 x_1(t) + k_2 x_2(t) - k_{e1} \frac{(x_1(t) - k_{e2})}{1 + exp(M_1(k_{e2} - x_1(t)))} - \\
&- U_{ii} + \frac{Ra(t)}{BW} + \frac{\left(k_{p1} - k_{p2} x_1(t) - k_{p3} x_4(t)\right)}{1 + exp\left(M_2\left(k_{p2} x_1(t) + k_{p3} x_4(t) - k_{p1}\right)\right)} \\
\dot{x}_2(t) &= k_1 x_1(t) - k_2 x_2(t) - \frac{V_{mx} x_2(t) x_3(t)}{K_m + x_2(t)} - \frac{V_{m0} x_2(t)}{K_m + x_2(t)} \\
\dot{x}_3(t) &= -p_{2U} x_3 - p_{2U} I_b + \frac{p_{2U}}{V_i} I_p(t) \\
\dot{x}_4(t) &= -k_i x_4(t) + k_i x_5(t) \\
\dot{x}_5(t) &= -k_i x_5(t) + \frac{k_i}{V_i} I_p(t)
\end{aligned} \tag{7}$$

The last subsystem is a first order linear system with G_p as input and G_M as output.

$$\dot{G}_M(t) = -k_{s.c.} G_M(t) + \frac{k_{s.c.}}{V_G} G_p(t) \tag{8}$$

We can perform exact linearization on the first subsystem using the following local coordinate transformation:

$$\Phi(x)=\begin{pmatrix} x_1 \\ -(m_2+m_4)x_1+m_1x_2+k_{a2}x_3+k_{a1}x_4 \\ -k_{a2}x_3+k_dx_4 \\ -(k_{a1}+k_d)x_4 \end{pmatrix} \tag{9}$$

$\Phi(x)$ is a local diffeomorphism in all points of the whole state-space, since:

$$\left|\frac{\partial \Phi}{\partial x}\right| = m_1\left(k_{a2}\left(k_{a1}+k_d\right)\right) \tag{10}$$

The zero dynamics of the system are uniformly asymptotically stable in Lyapunov-sense:

$$\begin{aligned} \dot{z}_3 &= -k_{a2}z_3+k_dz_4+\frac{k_d}{BW}u \\ \dot{z}_4 &= -(k_{a1}+k_d)z_4-\frac{(k_{a1}+k_d)}{BW}u \end{aligned} \tag{11}$$

After applying the proper control law based on [16], it is essential to transform the resulting series of integrators into an asymptotically stable system with poles p_1 and p_2 guaranteeing that the zero dynamics of the next subsystem will be asymptotically stable as well:

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= -p_1p_2z_1-(p_1+p_2)z_2+u \end{aligned} \tag{12}$$

Consequently, the structure of the current control problem can be delimited in two loops (Fig. 1): the first responsible for the exact linearization via feedback and Kalman-filtering, while the second loop for the asymptotic output tracking.

In the second loop, the asymptotic output tracking is realized on the series of system (12) and the second subsystem (7). The relative degree of this system is 5; therefore, two 5th order linear systems are needed: $W_1(s)$ the linear system to be tracked (4), and $W_2(s)$ the tracking dynamics with parameters a_i and b_j (i=1…r, j=0…r). Due to the subsystems' nonlinearities, the coordinate transformation and the control law have limited applicability with several inequality-constrains to be taken into consideration, and singular points to be avoided. Since the steady-state linearization of the series of the first two subsystems results in a 9th order linear system, only one extra pole should be added to complete W_1(s) and W_2(s). In Fig. 1, $C_1(x)$ represents the state-feedback for exact linearization of the first subsystem, while $C_2(z)$ represents the control law (5).

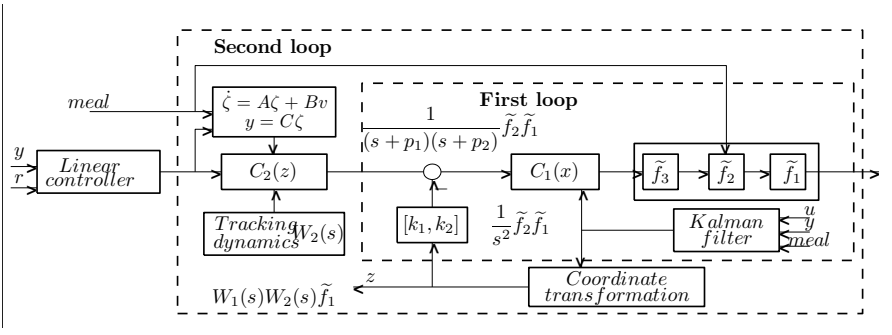


Fig. 1. Structure of the tracking controller.

The difference between the steady-state linear system, the model, and the model combined with the control law introduced above is shown in Fig. 2. Although the asymptotic output tracking response converges to the linearized model, slight deviations occur when there is a sudden change in the meal absorption input. This is caused by the fact that the relative degree of the model for the absorbed glucose input (*Ra*) is less than the relative degree for the injected insulin input. We should also mention that for the original system the deviation of the output from its basal value (250 mg/dL) is displayed.

For the estimation of state variables an EKF was used on the simulated output, to which a zero-mean white Gaussian noise was added with 51 mg^2/dL^2 variance, similarly to the simulations presented in [18]. The sampling time was 1 minute. The performance of the Kalman-filter is displayed on Fig. 3. It can be seen that the output is well filtered even with a relatively big measurement noise.

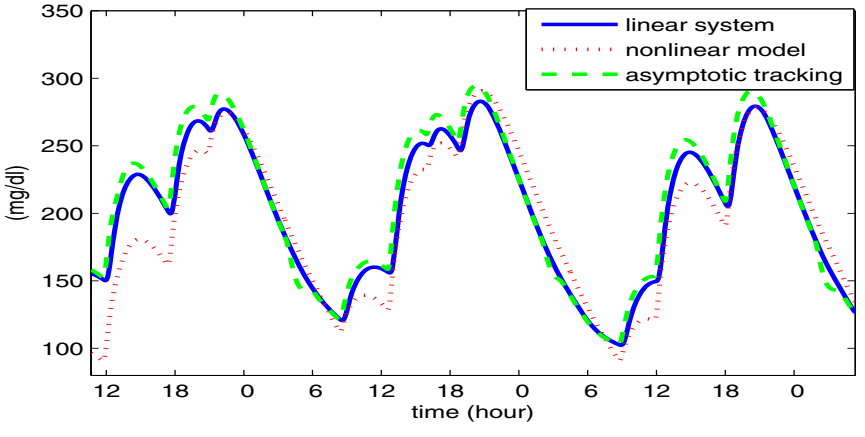


Fig. 2. Comparison of the steady-state linearization of model (1) with asymptotic output tracking and with the deviation of the output for the original model from its basal value (250 mg/dL).

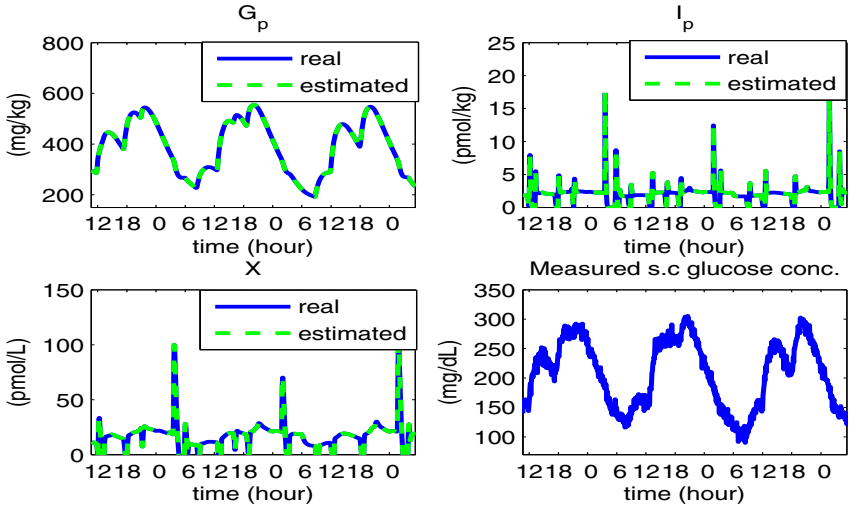


Fig. 3. The real and estimated values of three state variables (G_p, I_p and X) acquired from the simulated output samples with added Gaussian measurement noise (bottom right) using EKF.

To show the effectiveness of the method, the performance of two widely used controllers [17] have been compared: a classical PID controller and an LQG controller (LQ optimal control with Kalman filtering), both fitted for the steady-state linearization of model (1). The structure of the PID controller was first determined in continuous time domain, and then transformed into a discrete-time with the same 1 minute sampling frequency that was used in the sensor model (Fig. 4):

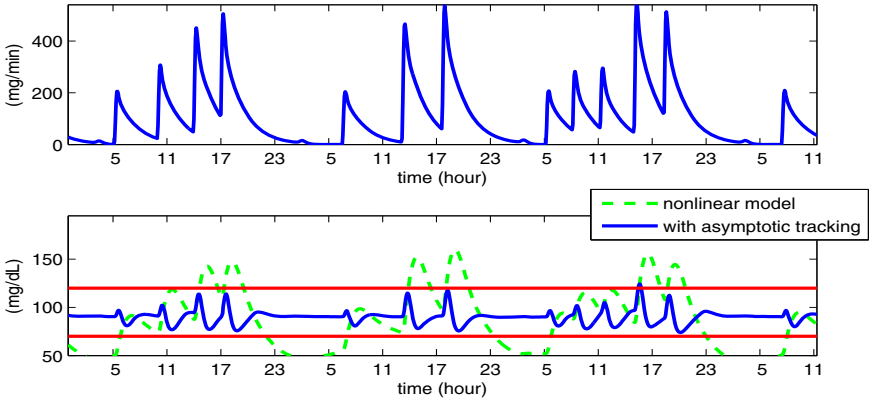


Fig. 4. Glucose absorption profile used in the simulations (top) and the performance of a PID controller with and without asymptotic output tracking (bottom).

$$W_{PID}(s) = A_p \frac{(T_I s + 1)}{T_I s} \frac{\left(T_D^2 s^2 + 2\xi T_D s + 1\right)}{\left(\frac{T_D^2}{n^2} s^2 + 2\frac{T_D}{n} s + 1\right)} \tag{13}$$

$$D_{PID}(z) = \left(1 - z^{-1}\right) Z\left\{L^{-1}\{W_{PID}(s)\}\right\} \tag{14}$$

In case of the LQG controller, an integrator has been added to ensure servo properties. The structure of the controller is presented in (Fig. 5), while the model output with and without asymptotic tracking is displayed on (Fig. 6). It can be seen that the nonlinear designed approach is able to keep the glucose level inside the defined 70-120 mg/dL interval in both cases, while during simulation of the designed controllers applied directly on the original nonlinear system both hypo- and hyperglycemia occur. For all simulations a real dataset of a 17 year old boy's recorded feeding profile has been used (Fig. 4 and 6).

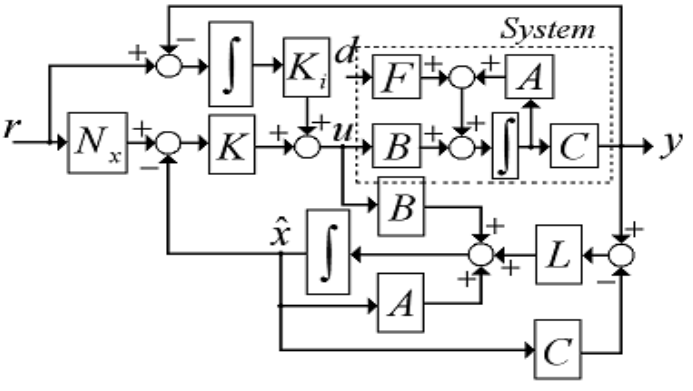


Fig. 5. LQR controller structure.

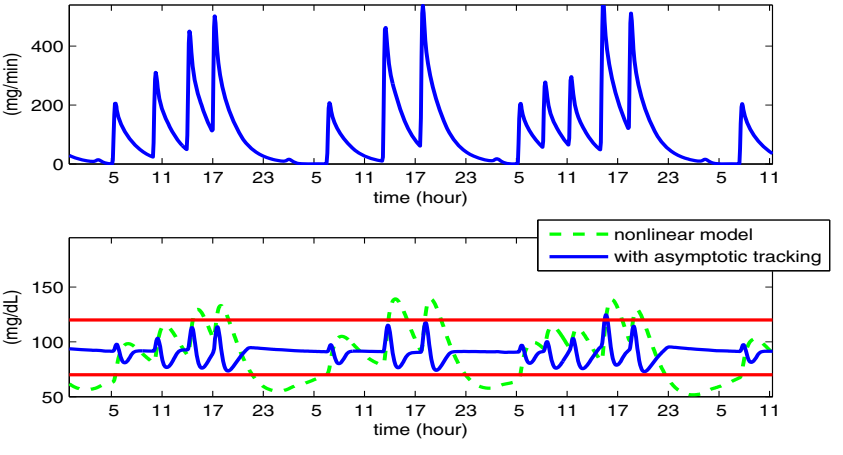


Fig. 6. Glucose absorption profile used in the simulations (top) and the performance of the LQR controller with and without asymptotic output tracking (bottom).

5 Conclusions

The aim of the current chapter was to apply asymptotic tracking for the highly nonlinear model [9]. We managed to hide the nonlinearity of the physiological model by transforming the control input provided by a linear controller so that the response of the model would mimic the behavior of a linear system. In addition, an EKF was designed to estimate the values of the state variables.

It can be concluded from the simulation results that the methods presented have the potential to extend the validity of linear controllers.

However, due to its limitations, several practical issues should be considered in the future:

1. the commercially available sensors work with higher sampling time, therefore model reduction might be needed to avoid the violation of the Shannon sampling theorem.
2. the EKF is sensitive to structural and parameter inaccuracies hence more robust filtering techniques should be used.
3. the meal absorption can only be estimated, which will inject further errors and disturbances into the controller.

However, by choosing different reference system and tracking dynamics for the asymptotic tracking it is possible to reduce the sensitivity and provide a better linear system for the linear controller loop to work with.

Acknowledgments. This work was supported in part by Hungarian National Scientific Research Foundation, Grants No. OTKA 82066, CK80316. It is connected to the scientific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project, supported by the New Hungary Development Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002). The authors say special thanks to Dr. Enikő Felszeghy from University of Debrecen Medical- and Health Science Center, Pediatrics Department, Debrecen, Hungary for the provided real patients' data.

Abbreviations

EGP	Endogenous Glucose Production
EKF	Extended Kalman Filter
LQG	Linear Quadratic Gaussian control
PID	Proportional Integrator Derivative
SISO	Single Input Single Output
WHO	World Health Organization

References

[1] Wild, S., Roglic, G., Green, A., Sicree, R., King, H.: Global prevalence of diabetes - Estimates for the year 2000 and projections for 2030. Diab Care 27(5), 1047–1053 (2004)

[2] Shaw, J.E., Sicree, R.A., Zimmet, P.Z.: Global estimates of the prevalence of diabetes for 2010 and 2030. Diab. Res. Clin. Pract. 87, 4–14 (2010)

[3] Fonyó, A., Ligeti, E.: Physiology. Medicina, Budapest (2008) (in Hungarian)

[4] Cobelli, C., Dalla Man, C., Sparacino, G., Magni, L., Nicolao, G., Kovatchev, B.: Diabetes: Models, Signals, and Control (Methodological Review). IEEE Rev. Biomed. Eng. 2, 54–96 (2009)

[5] Harvey, R., Wang, Y., Grossman, B., Percival, M., Bevier, W., Finan, D., Zisser, H., Seborg, D., Jovanovic, L., Doyle, J.F., Dassau, E.: Quest for the artificial pancreas. IEEE Eng. Med. Biol. Mag. 29(2), 53–62 (2010)

[6] Chee, F., Tyrone, F.: Closed-loop control of blood glucose. LNCS, vol. 368. Springer, Heidelberg (2007)

[7] Bergman, B.N., Ider, Y.Z., Bowden, C.R., Cobelli, C.: Quantitative estimation of insulin sensitivity. Am. J. Physiol. 236, 667–677 (1979)

[8] Dalla Man, C., Rizza, R., Cobelli, C.: Meal simulation model of the glucose-insulin system. IEEE Trans. Biomed. Eng. 54(10), 1740–1749 (2007)

[9] Magni, L., Raimondo, D.M., Dalla Man, C., Nicolao, G., Kovatchev, B., Cobelli, C.: Model predictive control of glucose concentration in type I diabetic patients: An in silico trial. Biomed. Signal Process Control 4(4), 338–346 (2009)

[10] Hovorka, R., Canonico, V., Chassin, L.J., Haueter, U., Massi-Benedetti, M., Federici, M.O., Pieber, T.R., Schaller, H.C., Schaupp, L., Vering, T., Wilinska, M.E.: Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes. Physiol. Meas. 25, 905–920 (2004)

[11] Parker, R.S., Doyle III, F.J., Ward, J.H., Peppas, N.A.: Robust H ∞ glucose control in diabetes using a physiological model. AIChE J. 46(12), 2537–2549 (2000)

[12] Sorensen, J.T.: A physiologic model of glucose metabolism in man and its use to design and assess improved insulin therapies for diabetes. PhD Thesis, Dept. of Chemical Engineering Massachusetts Institute of Technology, USA (1985)

[13] Palumbo, P., Pepe, P., Panunzi, S., Gaetano, A.: Glucose control by subcutaneous insulin administration: a DDE modeling approach. In: Proc. 18th IFAC World Congress, Milano, Italy, pp. 1471–1476 (2011)

[14] Kovács, L., Szalay, P., Benyó, B., Chase, G.J.: Asymptotic output tracking in blood glucose control. A case study. In: 50th IEEE CDC & ECC Conf., Orlando, USA (2011) (in press)

[15] Szalay, P., Kovács, L.: Applicability of asymptotic tracking in case of Type 1 Diabetes. In: Proc. 6th. IEEE Int. Symp. Appl. Comput. Intell. and Inform., Timisoara, Romania, pp. 623–628 (2011)

[16] Isidori, A.: Nonlinear control systems, 3rd edn. Springer, Berlin (1995)

[17] Lantos, B.: Theory and design of control systems II. Akademia Press, Budapest (2003) (in Hungarian)

[18] Facchinetti, A., Sparacino, G., Cobelli, C.: An online self-tunable method to denoise CGM sensor data. IEEE Trans. Biomed. Eng. 57(3), 634–641 (2010)

Optimal Energetic Conditions for Cell Seeding of Scaffolds

Andreea Robu[1], Lacramioara Stoicu-Tivadar[1], and Adrian Neagu[2]

[1] Department of Automation and Applied Informatics, "Politehnica" University of Timisoara, Bd. V. Parvan 2, 300223 Timisoara, Romania
{andreea.robu,lacramioara.stoicu-tivadar}@aut.upt.ro
[2] Department of Biophysics and Medical Informatics, "Victor Babes" University of Medicine and Pharmacy Timisoara, P-ta Eftimie Murgu 2, 300041 Timisoara, Romania
neagu@umft.ro

Abstract. Tissue Engineering is a novel area of biomedical research that combines the principles and methods of engineering and biology, in order to develop living tissues *in vitro*. The creation of functional tissue constructs presumes that living cells are seeded on different types and structures of biomaterials. Since the laboratory experiments are expensive and hard to perform, the computational approaches to tissue engineering are a cost-efficient alternative for predicting the growth of tissue constructs *in vitro*. This study developed computational models of biological systems formed by a cellular aggregate located on the plane surface of a biomaterial, respectively by a cellular aggregate located on a porous scaffold. Based on the Metropolis Monte Carlo method, we simulate the evolution of a cellular aggregate on the biomaterial's surface and we identify the energetic conditions that lead to a uniform and rapid cell spreading. We have monitored the evolution of the centre of mass of the cells in the system and the number of cells attached to the substrate after running a certain number of Monte Carlo steps and we found that cell-cell interactions disfavor cell spreading, while cell-biomaterial interactions favor cell spreading. We have also simulated the distribution of cells in a porous scaffold, analyzing the energetic conditions that lead to a successful cell seeding.

1 Introduction

The loss or damage of organs or tissues is one of the major problems in health care systems. Tissue engineering (TE) is a relatively new, multidisciplinary field that applies the principles of biology, engineering and material science toward the development of functional tissue constructs. These constructs are obtained by seeding living cells on different types and structures of porous and absorbable

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 261–272.
springerlink.com

scaffolds. Upon implantation into the host organism, the scaffold gradually disappears, while the cells continue to develop until they reach the form and the functionality of the desired tissue. The engineered constructs should be capable to restore, replace or improve tissue portions or the whole tissue itself (such as bone, cartilage, blood vessels or bladder) and to assure certain mechanical and structural properties, similarly to the native tissue [1, 2]. Understanding how cells organize into structured tissues is of major interest in all these fields.

One of the most important principles that underlies many discreet cellular models, and which is taken into account also in our study, is the differential adhesion hypothesis (DAH) proposed by Steinberg [3]. DAH states that cells tend to establish firm bonds with their neighbors. More precisely, DAH states that cells move in their environment until they reach the minimum energy configuration [3].

Forces acting between cells and between cells and adjacent media, such as the extracellular matrix (ECM), are transmitted by receptors: proteins embedded in the cell membrane and anchored to the cytoskeleton. One class of such receptors is the family of integrins, which mainly bind to components of the ECM. Integrins mediate cell-biomaterial adhesion indirectly: ECM proteins from the culture medium adsorb to the biomaterial and cells attach to these ECM proteins [4]. Another class of receptors is the family of cell adhesion molecules, such as cadherins, calcium-dependent transmembrane proteins that interact with their counterparts on neighbouring cells; thereby, cadherins mediate cell-cell interaction (cohesion) [4].

In the first part of our work, we develop a computational model of a biological system formed by a multicellular aggregate, bathed in cell culture medium, located on the plane surface of a biomaterial. Based on this model, we perform computer simulations of the experiments of ref. [5], identifying the optimal conditions that lead to rapid spreading of the cells on the surface of the biomaterial. By our *in silico* study we aim to validate the Monte Carlo approach proposed by our group for simulating cell seeding of tissue engineering scaffolds [6].

In the second part of our work, we create a computational model of a cellular aggregate located on a porous scaffold, bathed in culture medium. Using the Metropolis Monte Carlo method, we simulate the migration of the cells into the scaffold. We identify the optimal interactions between cells and between cells and biomaterial for a successful cell seeding [11].

Taking into account that the quality of the tissue constructs obtained and the time in which they are achieved *in vitro*, are two essential parameters in the tissue engineering area, our study provides information regarding the conditions that lead to an uniform and rapid distribution of the cells in a porous scaffold. It has been demonstrated that if cells are distributed uniformly in the scaffold, there is a higher cellular viability and the obtained construct has mechanical properties closer to the ones of native tissues [12].

2 Methods

The computational model of a cellular aggregate placed on the plane surface of the biomaterial is built on a cubic lattice composed of 50 × 50 × 63 nodes, the Oz axis representing the system's axis of symmetry. The distance between two adjacent lattice sites is equal to one cell diameter. In the region where $z>z_0$ (with z_0=3 in this study) the cell aggregate is represented by a sphere of 20 units in diameter, each site of the sphere being occupied by a cell. The lattice sites located around the cell aggregate at $z>z_0$ are occupied by medium particles, thus achieving a model for the cell aggregate bathed in culture medium. In the region where $z<=z_0$, each node of the network is occupied by an immobile particle, this area representing the biomaterial; we study the spreading of cells on the surface of this biomaterial [6].

To visualize the model system, we use the Visual Molecular Dynamics (VMD) software [9]. Fig. 1 depicts the initial configuration.

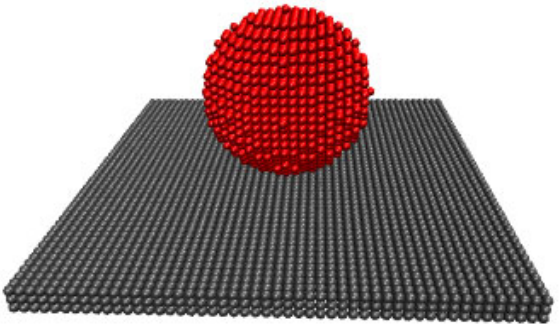


Fig. 1. The model associated to the studied biological system: a cellular aggregate located on the surface of a parallelepipedic piece of biomaterial.

The principle that guides the simulations of cellular rearrangements in the vicinity of the biomaterial is the evolution of the system towards the state of minimum energy of adhesion. This means that the cells tend to form the largest possible number of strong connections with their neighbors (other cells or biomaterial particles) [3, 10].

The total adhesion energy of a system composed of t types of cells in the vicinity of a substrate can be brought to the form [8, 10]:

$$E = \sum_{\substack{i,j=0 \\ i<j}}^{t-1} \gamma_{ij} \cdot B_{ij} + \sum_{i=0}^{t-1} \gamma_{is} \cdot B_{is}\,, \tag{1}$$

where Bij is the number of links between two particles (of type i and j), B_{is} the number of links between the cells of type i and the substrate, $\gamma_{ij} = (\varepsilon_{ii} + \varepsilon_{jj})/2 - \varepsilon_{ij}$

is the cell-cell interfacial tension, whereas $\gamma_{is} = \varepsilon_{ii}/2 - \varepsilon_{is}$ is the cell-substrate interfacial tension, ε_{ii} is a measure of interaction between cells, ε_{is} is a measure of interactions between cells and substrate [10].

To simulate the evolution of cell spreading on the plane surface of the biomaterial (Fig. 1), we used the Metropolis Monte Carlo algorithm. An elementary movement in the system is the swap of a cell – randomly chosen, with a volume element of cell culture medium from its vicinity – randomly chosen.

A Monte Carlo step (MCS) is the sequence of operations in which each cell is given the chance to make one move. A move is accepted with a probability

$$P = \min(1, \exp(-\Delta E / E_T)), \tag{2}$$

where ΔE is the difference between the adhesive energy of the system before and after the movement, whereas E_T is the biological equivalent of the energy of thermal motion [6,7], a measure of cell motility [10].

The *in silico* study of cell spreading on a planar substrate presented in this chapter is based on Monte Carlo simulations performed for different values of the following model parameters: (i) the cohesion energy between cells, (ii) the adhesion energy between cells and scaffold. As output parameters we monitored (i) the centre of mass of the cells, (ii) the number of cells attached to the substrate. The centre of mass of the cells is an indicator of the global motion of the cellular mass towards the substrate. The number of cells that attach to the substrate, on the other hand, describes the rate of cell spreading on the substrate's surface – a quantity measured in the experiments of Ryan et al. [5].

We also study the evolution of a spherical multicellular aggregate located on the surface of a porous scaffold, for different values of the cell-cell interaction energy and the cell-biomaterial interaction energy [11].

The computational model of the scaffold consists of a cubic network of 50×50×80 dimension along Ox, *Oy, Oz*, a node from this network being occupied by a biomaterial particle, or by a medium particle. The porosity of the scaffold is achieved in the model by taking into account the radius of the pores as well as the radius of the circular orifices that connect the pores. The cellular aggregate surrounded by cell culture medium is represented in the area $z>z0$ ($z0=80$) by a sphere with the radius $R=10$, each element of the sphere being occupied by a cell [11].

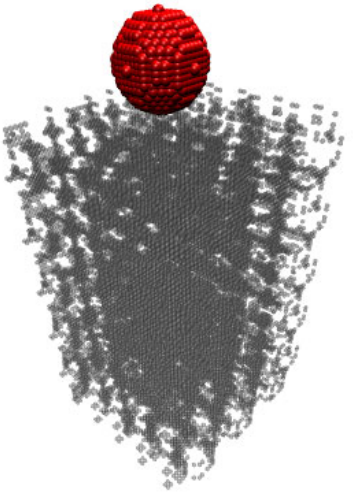


Fig. 2. The model associated to the studied biological system: a cellular aggregate located adjacent to porous scaffold.

3 Results and Discussions

In order to study the influence of cell-cell cohesion energy and cell-substrate adhesion energy on the spreading of cells on a planar substrate, a series of Monte Carlo simulations have been conducted.

Table 1 lists the values of the input parameters used in the simulations, the number of performed MCS, and the values of the obtained output parameters.

Table 1. Values of input and output parameters in representative simulations

Cell-cell interaction energy ε_{cc} / E_T	Cell-scaffold interaction energy ε_{cs} / E_T	MCS	Value of Z_{CM} for all cells	Number of cells attached to the substrate	Set of simulations, Figure
1	0.6	80 000	7	500	I, Fig. 3 – A, Fig. 4, Fig. 5
1	0.8	80 000	5.5	900	II, Fig. 3. – B, Fig. 4, Fig. 5
1	1	80 000	4.5	1400	III, Fig. 3. – C, Fig. 4, Fig. 5
1	1.2	80 000	4	2100	IV, Fig. 3. – D, Fig. 4, Fig. 5

In Fig. 3, regarding the four representative simulations presented in Table 1, we can observe that the spreading of cells on the substrate's surface is uniform and

the rate of cell attachment to the substrate is governed by the value of the cell-substrate interfacial tension [10].

After running 80 000 Monte Carlo steps the attachment of cells on the plane surface of the biomaterial is faster as the cell-substrate interfacial tension is lower (the cell-cell interaction energy remains constant, and the cell-substrate adhesion energy becomes higher).

Fig. 4 presents the evolution of the center of mass of the cells from the system for different values of the input parameters given in Table 1.

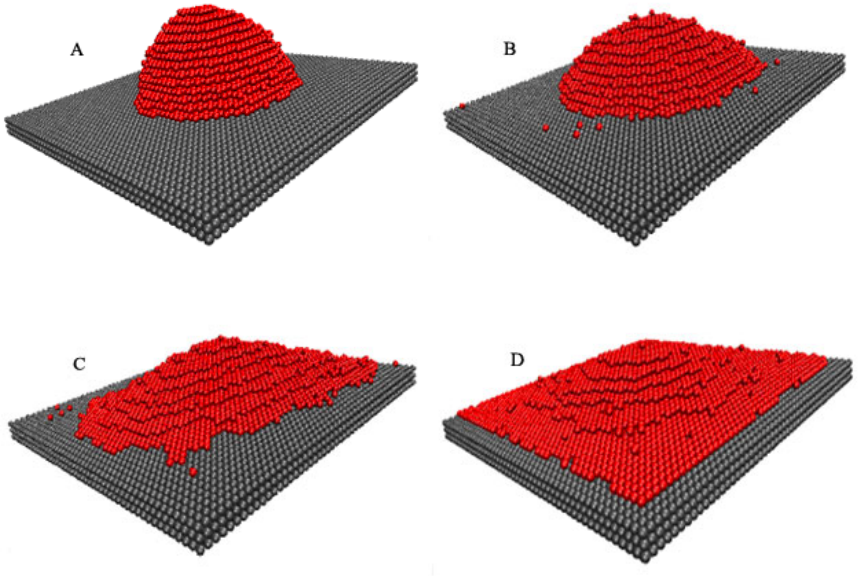


Fig. 3. Cell spreading on the plane surface of a biomaterial after running 80 000 MCS (A – Table 1, row 1; B – Table 1, row 2; C – Table 1, row 3; D – Table 1, row 4).

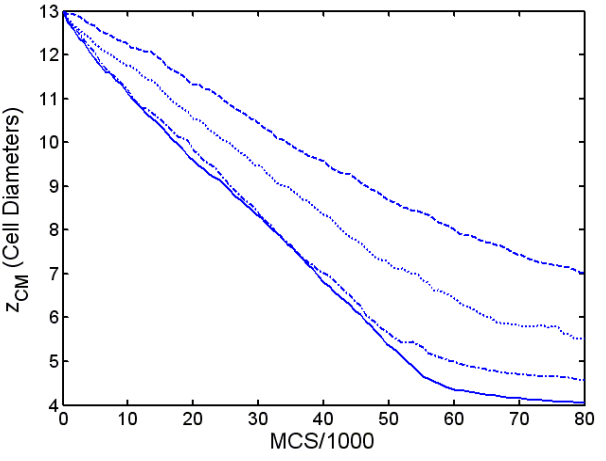


Fig. 4. The centre of mass of all cells during the simulations with parameters given in Table 1: Simulation I, row 1, dashed line; Simulation II, row 2, dotted line, Simulation III, row 3, dash-dot line; and Simulation IV, row 4, solid line.

In the initial state, the center of mass of the cells has the value of 13 lattice spacings (the cellular aggregate with a radius of 10 units is placed on a slab of biomaterial with a width of 3 units). As shown in Fig. 4, as the value of the interfacial tension becomes smaller, the centre of mass of the cells decreases, getting closer to the value 4, which corresponds to a cell monolayer attached to the surface of the substrate [10].

Fig. 5 plots the number of cells that managed to attach to the surface of the substrate versus elapsed MCS. As the cell-substrate interfacial tension becomes smaller, the rate of cell spreading on the substrate becomes also larger. In Simulation IV (solid line on Fig.5) an equilibrium state is reached in about 6×10^4 MCS because the cells cover the entire surface of the biomaterial [10].

By systematically varying cell-cell and cell-substrate interactions, Ryan et al. [5] generalized DAH by including also cell-biomaterial adhesion. Their results show that DAH allows for a rational control of cell spreading on the surface of biocompatible materials. Cell-cell cohesion was found to disfavor cell spreading, whereas cell-biomaterial adhesion was found to favor cell spreading [5].

In order to validate the Metropolis Monte Carlo method, we simulated cell spreading for different values of the cell-cell cohesion energy and cell-substrate adhesion energy and found that, indeed, the competition of these interactions explains the experimental data.

Table 2 lists the values of the cell–cell cohesivity and cell-substrate adhesivity parameters used in the simulations and the number of performed MCS.

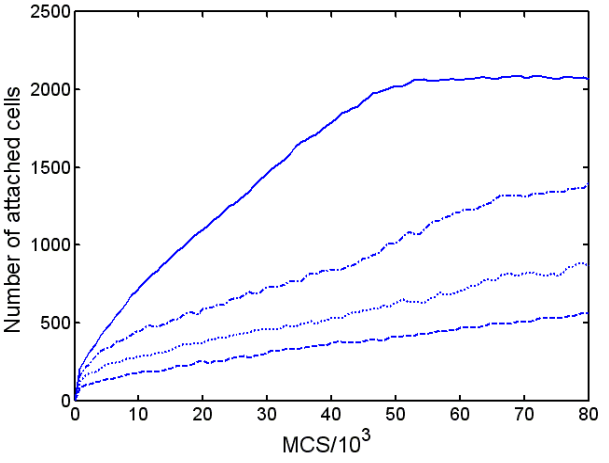


Fig. 5. The number of cells attached to the surface of the substrate. Parameters are given in Table 1: Simulation I, row 1, dashed line; Simulation II, row 2, dotted line; Simulation III, row 3, dash-dot line; and Simulation IV, row 4, solid line.

Analyzing Fig. 6, we can observe that at a constant level of cell-cell adhesion, cell spreading on the biomaterial depends on the cell-substrate adhesion energy. Increasing cell-cell cohesivity decreases aggregate spreading rate, while increasing cell-substrate adhesivity increasing aggregate spreading rate. Our results are in qualitative agreement with the experimental findings of ref. [5]. Thus, our simulation program correctly captures the essential features of the DAH.

In order to study the influence of cell-cell cohesion energy and cell-substrate adhesion energy in the cell seeding process of a porous scaffold, a series of Monte Carlo simulations have been performed [11].

Table 2. Values of input parameters in representative simulations

Cell-cell interaction energy ε_{cc} / E_T	Cell-scaffold interaction energy ε_{cs} / E_T	MCS	Set of simulations, Figure
0.8	1.8; 1.2, 0.6	80 000	I, Fig. 6 - A, B, C
1.4	1.8; 1.2, 0.6	80 000	II, Fig. 6 - D, E, F
2	1.8; 1.2, 0.6	80 000	III, Fig. 6 - G, H, I

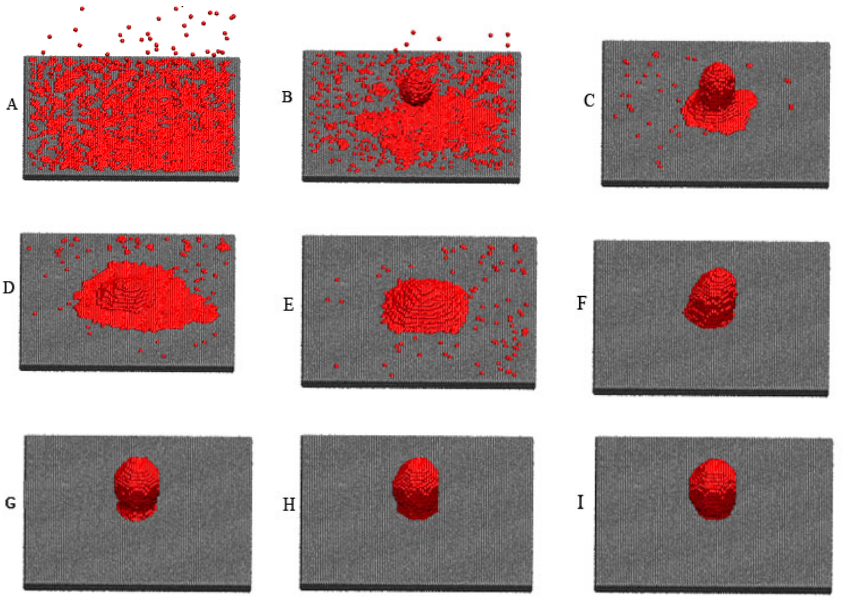


Fig. 6. Matrix of simulation results showing the influence of cell-cell cohesivity and cell-substratum adhesivity. (A, B, C –Table 2, row 1;D,E,F – Table 2, row 2;G,H,I – Table 2, row 3)

Table 3. Values of input parameters in representative simulations

Cell-cell interaction energy ε_{cc} / E_T	Cell-scaffold interaction energy ε_{cs} / E_T	MCS	Set of simulations, Figure
1	0.8	200 000	I, Fig. 7 – A, B
1	1	200 000	II, Fig. 8 – A, B
1	1.2	200 000	III, Fig. 9 – A, B
1	1.4	200 000	IV, Fig. 10 – A, B

Figs. 7-10, regarding the four set of simulations presented in Table 3, show the manner in which the aggregate gradually penetrates the scaffold and spreads more and more in its volume [11].

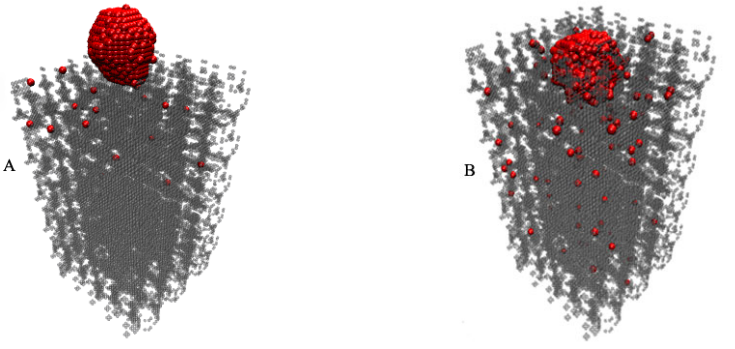


Fig. 7. Cell seeding of a porous scaffold. Simulation I (A – Table 3, row 1, Configuration at 50 000 MCS; B – Table 3, row 1, configuration at 200 000 MCS).

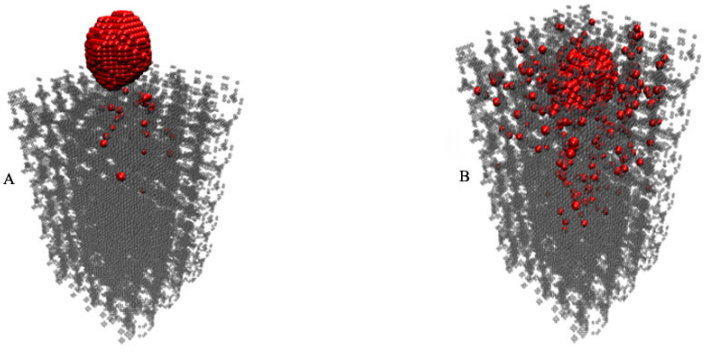


Fig. 8. Cell seeding of a porous scaffold. Simulation II (A – Table 3, row 2, Configuration at 50 000 MCS; B – Table 3, row 2, Configuration at 200 000 MCS).

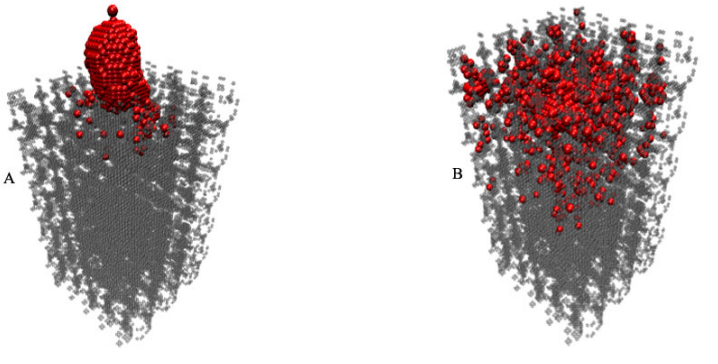


Fig. 9. Cell seeding of a porous scaffold. Simulation III (A – Table 3, row 3, Configuration at 50 000 MCS; B – Table 3, row 3, configuration at 200 000 MCS).

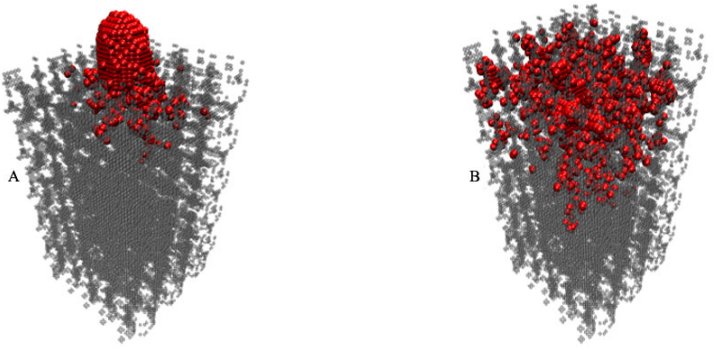


Fig. 10. Cell seeding of a porous scaffold. Simulation IV (A – Table 3, row 4, Configuration at 50 000 MCS; B – Table 3, row 4, configuration at 200 000 MCS).

Analyzing these simulation results, we can observe that the cell seeding of a scaffold is governed by the values of cell-substrate and cell-cell interaction energy. The higher is the cell-substrate interaction energy (the smaller is the cell-substrate interfacial tension), the faster is the seeding of cells in the volume of the scaffold. We found that the optimum cell-substrate interaction energy in our simulations is about 1.4, conducting to a rapid and uniform distribution of the cells in the scaffold [11].

4 Conclusions

We developed computational models for tissue engineering constructs in order to study the evolution of cellular aggregates in the proximity of biomaterials. Using the Metropolis Monte Carlo method, we simulated the cell seeding process and we identified the optimal energetic conditions that lead to an uniform and rapid distribution of the cells on the substrate or in the scaffold.

Since cell-biomaterial adhesion favors spreading and cell-cell cohesion disfavors spreading it has been conjectured that the relevant parameter is the adhesion/cohesion ratio [4]. Our model sheds new light on the competition of adhesion and cohesion during cell spreading, suggesting that the relevant parameter is the cell-substrate interfacial tension, equal to half of the cohesion energy minus the adhesion energy.

This study enables the optimization of the cell seeding process, a very important factor for the quick development of tissue constructs in the laboratory and for insuring mechanical properties similar to native tissues.

The creation of functional tissue constructs lays at the basis of the development of new therapies for patients that have diseased or totally damaged tissues. Since laboratory experiments are very expensive and time consuming, this study plays a very important role, supplying the researchers with a set of data procured from the simulations of the process of attaching the cells to the scaffold and thus reducing both the time and costs of fabricating tissue constructs.

References

[1] Langer, R., Vacanti, J.P.: Tissue Engineering. Science 260, 920–926 (1993)
[2] Ingber, D.E., Levin, M.: What lies at the interface of regenerative medicine and developmental biology? Development 134, 2541–2547 (2007)
[3] Steinberg, M.S.: Differential adhesion in morphogenesis: a modern view. Curr. Opin. Genet. Dev. 17, 281–286 (2007)
[4] Lauffenburger, D.A., Griffith, L.G.: Who's got pull around here? Cell organization in development and tissue engineering. Proc. Natl. Acad. Sci. USA 98, 4282–4284 (2001)
[5] Ryan, P.L., Foty, R.A., Kohn, J., Steinberg, M.S.: Tissue spreading on implantable substrates is a competitive outcome of cell– cell vs. cell–substratum adhesivity. Proc Natl. Acad. Sci. USA 98, 4323–4327 (2001)
[6] Neagu, A., Kosztin, I., Jakab, K., Barz, B., Neagu, M., Jamison, R., Forgacs, G.: Computational modeling of tissue self-assembly. Mod. Phys. Lett. B 20, 1217–1231 (2006)
[7] Robu, A.P., Neagu, A., Stoicu-Tivadar, L.: A computer simulation study of cell seeding of a porous biomaterial. In: Proceedings of International Joint Conference on Computational Cybernetics and Technical Informatics (ICCC-CONTI 2010), Timisoara, Romania, pp. 225–229 (2010)
[8] Doaga, O., Savopol, T., Neagu, M., Neagu, A., Kovács, E.: The kinetics of cell adhesion to solid scaffolds: An experimental and theoretical approach. J. Biol. Phys. 34, 495–509 (2008)
[9] Humphrey, W., Dalke, A., Schulten, K.: VMD – Visual Molecular Dynamics. J. Mol. Graph 14, 33–38 (1996)
[10] Robu, A., Stoicu-Tivadar, L., Neagu, A.: Cell spreading on biocompatible materials studied by computer simulations. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisoara, Romania, pp. 641–644 (2011)
[11] Robu, A., Stoicu-Tivadar, L., Neagu, A.: Simulation of cellular aggregate seeding of porous scaffolds. In: Proceedings of International Conference on Computers and Computing (ICCC 2011), Lanzarote, Canary Islands, Spain, pp. 100–105 (2011)
[12] Vunjak-Novakovic, G., Obradovic, B., Martin, I., Bursac, P.M., Langer, R., Freed, L.E.: Dynamic cell seeding of polymer scaffolds for cartilage tissue engineering. Biotechnol. Prog. 14, 193–202 (1998)

Ideas on a Pattern of Human Knowledge

Claudiu Pozna[1] and Radu-Emil Precup[2]

[1] Transilvania University of Brasov, Department of Product Design and Robotics,
Bd. Eroilor 28, RO-500036 Brasov, Romania
cp@unitbv.ro
[2] "Politehnica" University of Timisoara, Department of Automation and Applied Informatics,
Bd. V. Parvan 2, RO-300223 Timisoara, Romania
radu.precup@aut.upt.ro

Abstract. This chapter presents some ideas that concern a pattern of human knowledge. This pattern is based on the experimentation of causal relations. The cultural origin of the patterns is analyzed in terms of philosophical, psychological and linguistic points of view. An application scenario related to a robot integrated in a cognitive system is described. The definitions of signatures and of signature classes are given as useful steps in an alternative modeling approach to the observation process.

1 Introduction

The pattern of human knowledge analyzed here represents a stage of the "Research on new cognition system based on the experimentation of the causal relations" project. The big challenge regarding the studies on human cognition is to answer the question on how can human mind obtain synthesis from experimental data. This deals in fact with how the human mind can build veridical models of reality only from disturbed and unsubstantial data [1], the only information we can get through our senses.

The domain under investigation requires the solving of this problem as an interdisciplinary one. Two aspects are investigated in this context, the psychological and the technical one. The psychological approach, known as Cognitive Psychology (CP), has in view, as an objective, the understanding of the human knowledge phenomenon, the obtaining of models of the processes which occur during this phenomenon. The technical approach, referred to as Machine Learning, Robotics and Artificial Intelligence, aims to build a mathematical model and then to design and achieve a product.

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 273–286.
springerlink.com

The new results presented in this chapter are supported by the proposal of new pattern of human knowledge which is based on the experimentation of the causal relations according to [2]. This chapter includes new definitions of signatures and of classes of signatures that emerge from the fuzzy signatures as convenient hierarchical symbolic representations of data [3]–[8].

The implementation of the theoretical approaches leads to a product which is a robot in our case. This robot has several attributes including intelligence and capacity of knowledge.

This chapter is structured as follows. The analysis of the state-of-the-art in the field is conducted in the next section. Section 3 presents the project mentioning its objective and steps. Section 4 highlights the results expressed as a new cognition model. The definitions of signatures and of classes of signatures are presented in Section 5. The conclusions and directions of future research are pointed out in Section 6.

2 Literature Review

The research directions in CP from the point of view of patterns of human knowledge deal with attention, perception memory and reasoning [9]. Neural networks are employed in this context [10].

The research concerning the attention aims the explanation of the following phenomenon: human being has to deal with a great amount of information because of numerous stimuli from the environment, but only a small part of them are used. The explanation has been given through the following models: the model of the selective attention (Broadbent), which explains the attention by the existence of a sensorial filter [11] (the model is limited regarding the explanation of the "cocktail party" phenomenon: a person who has focused on a conversation is still able to seize relevant information from another source as well); the model of attenuation [12] replaces the "all or nothing" principle of the previous model with the principle of selecting the main channel and the attenuation of the other channels (the main limit of the model consists in the fact that it does not specify clearly what is understood by the informational attenuation of the signal); the model of the selective attention which relies on resources [13], according to this model the selection takes place at different levels of processing, and the closer to the final point the processing takes place the more resources are used.

The research in perception deals with modeling the process in which the information supplied by environment is interpreted in order to find its significance and meaning. The perception consists in the interpretation of the information collected by using the sensations. Several approaches are suggested in the literature: the model of stamps [14] in which the information is compared to a set of models stored in the memory, and the limits of the model are given by the fact that the stamp must match perfectly, an infinite number of models would be needed, and the procedure is time-consuming; the prototype model where the natural objects

are approximations of idealized prototypes [11]; the model based on the analysis of the features [15] which suggests a hierarchy of remarkable features; the model of the scenic analysis [16] where the patterns can be easily recognized because we expect to find certain forms in certain places; the cyclic model of perception [17] described as a combination of synthesis-analysis-synthesis expressed as the synthesis elaborates the perceptive model and the analysis extracts the information from the environment in view of its further correction.

The research in memory is focused on the processes by which the storing and the updating of information are made. The processes investigated in this context are those which allow the storage of information only for the time needed to use it i.e. the short-standing memory, the storage of the information for a longer period of time representing the long-standing memory, and the way in which the information is organized so as to be stored. The current approaches include: the constructivist model [14] which stresses the importance of memorizing the meaningful information and of the avoiding of meaningless association; the modal model of memory [18] which explains the way in which the information is acquired, stored and updated; the model of the work memory [19] which states the existence of several parts: an articulator loop which stores the information in a verbal form, a first acoustic storage with a limited capacity, a sensorial filter which permits the taking of the visual information, a central instance at which the input is connected.

The reasoning is defined as representing the processes implied by symbolizing and manipulating the information [14]. It is important to outline here: the Piage model, in which the reasoning is seen as an equilibration process necessary to replace the lack of balance provoked by the new (the equilibration can be done by the assimilation of the way in which the new is understood, based on the already existent concepts or by the accommodation which implies the modifying of the concepts in order to understand the new); models that belong to the cognitive perspective (Miller, Newell and Simon) stating that a complexity of a problem can be reduced by solving a series of sub-problems; the models of the forming of concepts explain the way in which a person extracts the essential features from the stimulus and places the result in a category; the Bruner models [14] which use four types of strategies to identify the concepts: the conservative focusing; the game of chance-type focusing; the successive scanning; the simultaneous scanning; the Levine model [20] of successive hypotheses which merge into a work hypothesis; the behaviorist model [21] according to which the reasoning is nothing else but a voiceless speaking which explains the fact that when a man is trying to solve a problem, he also solicits language at some extent; the rationalist model [22] explains the language acquisitions by genetic endowment, stating that there are linguistic universal patterns in every language and there are certain people who are genetically endowed such that to recognize them.

From the technical point of view, according to EU's IST 2002 23.24 a cognitive system is a system that understands, learns and develops itself by social and individual interactions [9]. The desired objectives are based on the achievements referring to the three components of a cognitive system: the action which is external

(the robot control) [14] and internal [23] (the approaching of a behavior through which the system focuses on a certain problem); the perception, which suppose the interpreting of the signals obtained from the sensorial system and the using of this interpretation in describing a situation that triggers the reasoning; the reasoning which is the one that co-ordinates the action and the perception. According to [14], the reasoning is made of two functions, symbolizing and manipulating the symbols.

The following conclusions can be pointed out on the basis of this literature review. They can be viewed as useful research hints.

A cognitive system is a system which understands and learns by social and individual interactions. More precisely, the birth of cognitive systems has lead to a new generation of robots.

The main difference between an intelligent robot and a robot integrated in a cognitive system is that the latter can (in time) execute more tasks that he was programmed to do. More precisely, in case of robots which have a tactical level of control there are many solutions (known a priori) from which the robot will choose the one that fits the problem he has to solve: the knowledge (the set mentioned and the rules of their arbitration) are introduced by the designer; the development of knowledge is made by a procedure established a priori.

In case of robots which have a strategic level of control there is a set of behaviors which permit the planning of tasks. The knowledge (the behaviors and they rules of composition) are established a priori. The development of knowledge is made in accordance to a procedure (established a priori).

While in the previous cases the knowledge was managed a priori by the designer (these rules do not change, no matter of the knowledge of the robot) in the case of the cognitive system, the task of organizing the knowledge process becomes the task of the system itself. This task is a dynamic processing where its organization depends on the acquired knowledge, viz. it modifies the rules according to the new knowledge.

The achievement of a cognitive system is an interdisciplinary problem dealing with CP, Robotics and Artificial Intelligence (R+AI). The CP offers models of human reasoning, and the R+AI transforms these models and combines them with the own knowledge.

The complexity of the cognitive system revealed the fact that the achievement of such a system must benefit from the Machine Learning and AI algorithms.

The following requirements are pointed out regarding the limitations of the domain and the less approached directions (because of the novelty of the domain):

- Up to now the research focused on the organizing of the cognitive system. We consider in [24] that the auto-organizing of the knowledge process can be obtained by knowledge process itself. The mentioned process is controlled by a certain structure of knowledge which is modified because of the new knowledge. We consider that the mentioned dynamics can be known by using the basic level of knowledge in terms of [24]: the use of the experimentation of the causal relations as a source of knowledge.

The necessity to define the ratio between the initial knowledge of the system (a priori knowledge) and the knowledge the system can acquire (a posteriori

knowledge). The necessity of determining the potential of the a priori knowledge and of the organization of this knowledge focuses the further performance of the cognitive system.

3 Control Objectives and Steps in Cognition Systems

The analysis of the actual stage of knowledge revealed the fact that a cognitive system is in fact a new generation of robots. The main characteristic of this generation consists in its knowledge capacity. The analysis indicates one research direction, the study of the connections between the auto-organization of knowledge and the knowledge process itself.

The main source of human knowledge is the experience. The basic level of this way of knowledge is represented by the experimentation of the causal relations.

The objective of the project appeared from the corroboration of the two previous ideas: the creation of a cognitive system able to auto-organize its knowledge process by experimenting. More precisely, we speak about the building of a robot able to develop its knowledge by auto-organizing the experiments that involve causal relations. The project outcome is a robot integrated in a cognitive system. In the following we will refer to this system like robot.

The following scenario is aimed (Fig. 1):

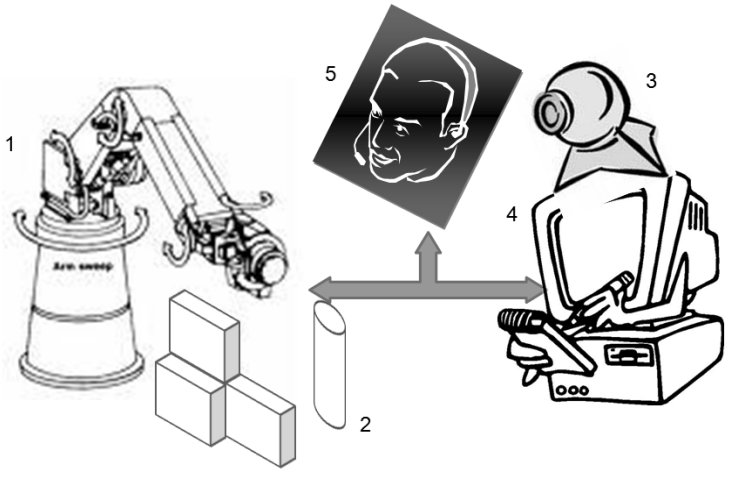


Fig. 1. The scenario scene: 1 - the robot arm, 2 - the object in the artificial environment, 3 - the video camera, 4 - the computer, and 4 - the programmer.

- The robot self-organizes its knowledge.
- This knowledge is of an experimental type has two sources: the first refers to the causal relations between phenomena that appear in the artificial environment in which the robot is immersed and the second by dialog with its programmer.

- From the first source, in order to acquire experience, the robot actions upon the objects and monitors, through a sensorial system, the phenomena appeared. This monitoring information is transformed by the control system of the robot in perception which, in its turn, leads to learning.
- From the second source, in order to acquire experience, the robot constructs questions and asks his programmer about solutions. The answers are next analyzed and integrated.

The aimed benefits are: on a short term, this represents the theoretical base in the building of a robot which auto-organizes its knowledge process: it is a qualitative leap in the field of intelligent robots. The long-term output of the strategy developed here can be implemented in service robots and mobile robots including the autonomous vehicles, etc. Our concept can be used in various applications that use robots [25]–[32].

The main objective of the project is the building of a cognitive system which auto-organizes its knowledge on the basis of the experience from causal relations. The achievement of this system implies the fulfilling of the following objective and intermediary steps.

1. The process of getting a cultural (philosophic and psychological) model of the human knowledge phenomenon. The model which is expressed in a natural language (using the PC concepts is a necessity, because it represents the starting point, the information needed in mathematical modeling (representing the thing that should be done). It cannot be used directly in robotics because the natural language is vague, that is the reason why it has to be translated into mathematical language.

2. The transfer of the obtained psychological model in a mathematical model that can be used in robotics. The model will contain algorithms which describe the acquiring of the knowledge through auto-organizing the experience. The elements of the model are expressed in mathematical language. This is a necessity because it made it possible to make the transfer from the natural language of the psychological model to the mathematical one (representing the thing that can be done). It cannot be used directly in robotics, because it needs to be materialized (i.e., programming, building the sensorial system, etc.).

3. The building of the knowledge system. Expected results are the materializing of the cognitive system obtained by the systemic junction of the robot arm, the sensorial system and the control system (the system is immersed in an artificial environment).

It is accepted that the system has a priori knowledge about understanding experiences. A posteriori, the system enriches its knowledge: it tests connection relations implying identification of causal relations, it acquires new knowledge, by composing knowledge it identifies new causal relations or new connections, it perceives alterations of the environment provoked by the person who uses the system and it identifies new connection relations and new experience subject.

4. The use of the cognitive system. We expect that the using of the cognitive system will permit us to find answers to the following questions: How must the knowledge be organized such that to assure the capacity of self-organizing the experience? How can one stimulate the knowledge?

The last question refers to new a priori knowledge or offering new connection relations in the context of environmental modifications provoked by the user, etc.

The four steps substantiate the cultural model construction. This model will be described in the next section.

4 Pattern-Based Model and Process of Knowledge

We focus on the following thesis: the human knowledge and intelligence – used as sources of inspiration in order to obtain systems of artificial knowledge – are cultural phenomena. Therefore the understanding of these phenomena requires the use of specific investigation methods.

These specific methods are philosophy and psychology. Consequently the first part of the project describes a pattern of human knowledge based on the investigation methods mentioned above. The following phases of the project will take this model, transpose it into a mathematical language and, based on specific procedures, will turn it into an artificial system.

The originality of our approach is its operational nature. More precisely, specialized information [33] was systematized in order to obtain a pattern which in the following phases could be described mathematically, turned into an algorithm and, eventually, transposed on a technical system.

Pattern description includes: expressing the initial principles of the pattern, the actual description of the pattern, describing how it works, underlining the sources of knowledge, the specification of a priori elements required in order for the pattern to function.

The initial principles of this pattern are threefold:

- The knowledge is a cultural process because it belongs to a certain culture and it takes place during a period of time. It can be described in two stages, upward and downward.
- The mode of human knowledge is intelligence driven. Intelligence is a process which has as a goal to increase the space-temporary stability of the individual. Intelligence requires the existence of the two stages previously described and which can be expressed as acquiring experience and understanding it.
- The human knowledge modes are facilitated by the interaction between culture and environment.

The elements that build the pattern are described as follows. First the downward (rationalist) stage is characterized by the following aspects.

The knowledge is the process through which the particular (the experience) is arranged according to a system of categories. The schematic is the basis of the arranging process which allows the general to incorporate the particular. This means that for each of the 4x3 Kantian categories the subject who acquires knowledge has a set of schemes he can use in the arranging process.

The scheme is a basic ensemble which allows the structuring of the information. The information is acquired by the subject due to his senses and structured by the schemes he/she owns.

The scheme allows first the distinction between the relevant and irrelevant information. It offers next the focus on the connections provided by the relevant information.

The comparison mentioned above enables the perception and, in a larger sense, experience. Perception is the agreement to the fact that the structure of relevant information which is obtained by the subject is or not in accordance with the structure of the scheme he/she owns in the aforementioned collection.

An experience represents an ensemble of perceptions (4x3). Although the sensorial data do not permit the accomplishment of the entire ensemble of perception, the subject will add the missing elements. Linguistically, the experience can be considered as answers to the questions asked by the 4x3 categories.

Second, the upward (empiricist) stage has the following features.

The knowledge is the process through which the elements of the categories system are transformed based on experience. Learning stands for the basis of the transformation process which allows the modification of the general for the purpose of incorporating a large variety of particulars.

The experience allows the establishment of the difference between the used schemes and the experimented structure.

The new structures allow the building of new schemes. The new schemes lead next to the modifications of the concepts.

The operation mode of our new pattern is described on the basis of the model presented before. The description includes the next succession of stages (SC) the subject goes through in order to improve himself/herself:

- The object is revealed through its senses to the SC which learns in terms of an ensemble of unfiltered and unstructured information.
- For each of the 12 categories the SC makes use of groups of schemes which enable the categorizing of information as essential and nonessential.
- The focus on the group of essential information (viz., information filtering) allows the use of the scheme in order to identify certain specific structures.
- Identifying a certain structure helps the appearance of the perception phenomenon.
- Experience is an ensemble of perceptions and its dimension equals that of the categories.
- Experience generates information about the process of perception.

- This information may lead to the building of new structures and/or adding new elements to the existing structures.
- The new structures allow the modification of concepts.

The scheme presented in Fig. 2 illustrates the succession pointed out before.

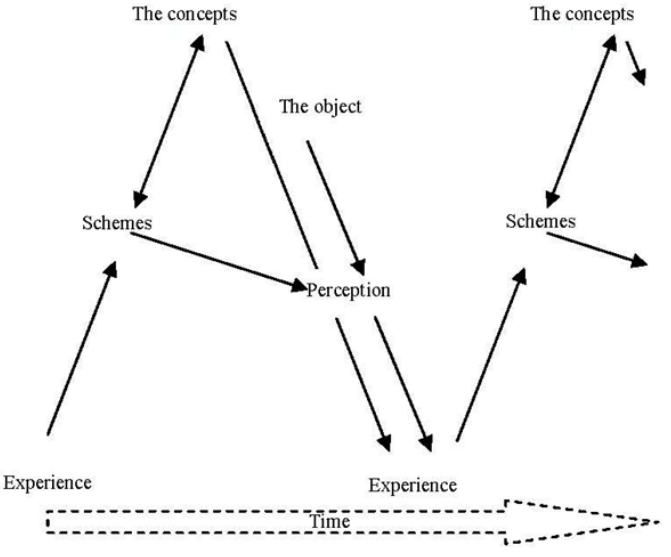


Fig. 2. The information flow in the dynamic process of knowledge.

The sources of knowledge are in accordance with the stages presented before. The mode of human knowledge is facilitated by cultural and environmental interaction. The following aspects are important with this regard.

- The individual learns to use the first schemes during his education.
- The individual is taught how to recognize classes, categories, etc.
- The interactions with other individuals make the individual communicate through concepts.
- The interaction with the environment leads to learning the concept refinement.

A priori elements of the pattern appear from its functioning. That can be expressed also as the basis needed to activate this pattern.

5 Signatures and Classes of Signatures

Let $\mathbf{R}$ be the set of real numbers. The set $S^{(n)}$ is defined recursively as

$$S^{(n)} = \prod_{i=1}^{n} S_i \,, \tag{1}$$

where $S_i = \mathbf{R}$, $i = \overline{1,n}$ (i.e., $i \in \{1,2,...,n\}$), or $S_i = S^{(m)}$, and $\prod$ is the Cartesian product.

Let X be a nonempty set. The collection of signatures is defined as the function $A: X \to S^{(n)}$, and the signature of the element $x \in X$ is

$$A(x) = \begin{bmatrix} ... \\ a_i \\ \begin{bmatrix} a_{i+1,1} \\ a_{i+1,2} \end{bmatrix} \\ \begin{bmatrix} a_{i+2,1} \\ \begin{bmatrix} a_{i+2,2,1} \\ a_{i+2,2,2} \end{bmatrix} \end{bmatrix} \\ ... \end{bmatrix}. \tag{2}$$

The matrix elements values which appear in the signature defined in (2), i.e., $a_1, a_2, ..., a_n, a_{i,1}, a_{i,2}, ..., a_{i,m}, ..., a_{j,k,l}, ...$, are referred to as signature values.

The transposition of the signature $A(x)$ is

$$A^T(x) = [... \quad a_i \quad [a_{i+1,1} \quad a_{i+1,2}] \quad [a_{i+2,1} \quad [a_{i+2,2,1} \quad a_{i+2,2,2}]] \quad ...]. \tag{3}$$

A signature $A(x)$ with the values $a_1, a_2, ..., a_n, a_{i,1}, a_{i,2}, ..., a_{i,m}, ..., a_{j,k,l}, ...$, is expressed as $a^{\cdots}$. Two signatures $a^{\cdots}$ and $b^{\cdots}$ have the same structure if and only if for each value $a_{i_1,i_2,...,i_s}$ of the signature $a^{\cdots}$ there exists the value $b_{i_1,i_2,...,i_s}$ of the signature $b^{\cdots}$.

A class of signatures is a set of signatures with the same structure. The notation $\hat{a}^{\cdots}$ is used to highlight the class of signatures with the same structure as that of the signature $a^{\cdots}$.

6 Conclusions

This chapter has offered new results concerning a pattern of human knowledge proposed in [2]. It has reached a twofold goal:

- It models the cognitive psychology which aims to explain the phenomena which appear during human knowledge: attention, perception, memory and thinking. The analysis of the results obtained in this context permits to identify new, original direction for research.

- It describes the research aimed on the design and implementation of the above mentioned directions.

Our model is advantageous with respect to the previous literature because it captures well the knowledge phenomenon as an ensemble from two phases, the building of experience and the concept modification based on this experience. It is a step forward and an extension with respect to the results given in [34].

Future research will deal with the transformation of the model suggested here into a mathematical one using several tools [35]–[46]. The complexity reduction will be accounted for [47]–[49], and the process of building the model will be expressed in terms of a procedure which will enable the building an artificial system of knowledge in terms of different representations [50]–[53].

Acknowledgments. This work is supported by the Romanian Ministry of Education, Research and Innovation through the PNII Idei project 842/2008 and by the CNCS – UEFISCDI of Romania.

References

[1] Griffiths, T.L., Kemp, C., Tenenbaum, J.B.: Bayesian models of cognition. In: Sun, R. (ed.) The Cambridge Handbook of Computational Psychology, pp. 59–100. Cambridge University Press, Cambridge (2008)

[2] Pozna, C., Precup, R.E.: A new pattern of knowledge based on experimenting the causality relation. In: Proceedings of 14th International Conference on Intelligent Engineering Systems (INES 2010), Las Palmas, Spain, pp. 61–66 (2010)

[3] Kóczy, L.T., Vámos, T., Biró, G.: Fuzzy signatures. In: Proceedings of 4th Meeting of the Euro Working Group on Fuzzy Sets and the 2nd International Conference on Soft and Intelligent Computing (EUROPUSE-SIC 1999), Budapest, Hungary, pp. 210–217 (1999)

[4] Vámos, T., Kóczy, L.T., Biró, G.: Fuzzy signatures in data mining. In: Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Vancouver, BC, Canada, vol. 5, pp. 2842–2846 (2001)

[5] Wong, K.W., Gedeon, T.D., Kóczy, L.T.: Construction of fuzzy signature from data: An example of SARS pre-clinical diagnosis system. In: Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2004), Budapest, Hungary, pp. 1649–1654 (2004)

[6] Wong, K.W., Gedeon, T.D., Kóczy, L.T.: Fuzzy signature and cognitive modelling for complex decision model. In: Castillo, O., Melin, P., Montiel Ross, O., Sepúlveda Cruz, R., Pedrycz, W., Kacprzyk, J. (eds.) Theoretical Advances and Applications of Fuzzy Logic and Soft Computing. AISC, vol. 42, pp. 380–389. Springer, Heidelberg (2007)

[7] Hadad, A.H., Gedeon, T.D., Mendis, B.S.U.: Finding input sub-spaces for polymorphic fuzzy signatures. In: Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2009), Jeju Island, Korea, pp. 1089–1094 (2009)

[8] Ballagi, Á., Kóczy, L.T., Gedeon, T.D.: Robot cooperation without explicit communication by fuzzy signatures and decision trees. In: Proceedings of Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference (IFSA-EUSFLAT 2009), Lisbon, Portugal, pp. 1468–1473 (2009)

[9] http://www.fp6.cordis.lu/fp6/call
[10] McClelland, J.L., Rumelhart, D.E.: The PDP research group: parallel distributed processing: Explorations in the microstructure of cognition, Psychological and biological models, vol. 2. MIT Press, Cambridge (1986)
[11] Broadbent, D.E.: Perception and communication. Pergamon Press, London (1958)
[12] Treisman, M., Gelade, G.: A feature integration theory of attention. Cogn. Psychol. 12, 97–136 (1980)
[13] Johnston, W.A., Wilson, J.: Perceptual processing of nontargets in an attention task. Mem. Cogn. 8, 372–377 (1980)
[14] Malim, T.: Cognitive processes. Editura Tehnica, Bucharest (1999) (in Romanian)
[15] Lindsay, P.H., Norman, D.A.: Human information processing: An introduction to psychology. Academic Press, New York (1972)
[16] Biederman, I.: Recognition by components: A theory of human image understanding. Psychol. Rev. 94, 115–147 (1987)
[17] Neisser, U.: Memory observed: remembering in natural contexts. W. H. Freeman, San Francisco (1982)
[18] Atkinson, R.C., Rauch, M.G.: An application of the mnemonic keyword method to the learning of a Russian vocabulary. J. Exp. Psychol. Hum. Learn. Mem. 1, 126–133 (1975)
[19] Baddeley, D.: The fractionation of human memory. Psychol. Med. 14, 259–264 (1984)
[20] Levine, M.A.: A cognitive theory of learning. Lawrence Erlbaum, Hillsdale (1985)
[21] McNeill, D.: The creation of language. In: Oldfield, R.C., Marshall, J.C. (eds.) Language, pp. 21–31. Penguin Books, London (1968)
[22] Yoshida, E., Kokaji, S., Murata, S., Tomita, K., Kurokawa, H.: Miniaturization of self-reconfigurable robotic system using shape memory alloy actuators. J. Robot Mechatronics 12, 96–102 (2000)
[23] Chater, N., Tenenbaum, J.B., Yuille, A.: Theory-based Bayesian models of inductive learning and reasoning. Trend Cogn. Sci. 10, 309–318 (2006)
[24] Pozna, C.: An introduction on cognition system design. Acta Polytechnica Hungarica 4, 33–47 (2007)
[25] Harmati, I., Skrzypczyk, K.: Robot team coordination for target tracking using fuzzy logic controller in game theoretic framework. Robot Auton. Syst. 57, 75–86 (2009)
[26] Hladek, D., Vaščák, J., Sinčák, P.: Multi-robot control system for pursuit-evasion problem. J. Electr. Eng. 60, 143–148 (2009)
[27] Hermann, G., Kozlowsky, K.R., Tar, J.K.: Design of a planar high Precision Motion Stage. In: Kozlowsky, K.R. (ed.) Robot Motion and Control 2009. LNCIS, vol. 396, pp. 371–379 (2009)
[28] Wang, Y., Lang, H., de Silva, C.W.: A hybrid visual servo controller for robust grasping by wheeled mobile robots. IEEE/ASME Trans. Mechatronics 13, 757–769 (2010)
[29] Skoglund, A., Iliev, B., Palm, R.: Programming-by-demonstration of reaching motions - A next-state-planner approach. Robot Auton. Syst. 58, 607–621 (2010)
[30] Vaščák, J., Madarász, L.: Adaptation of fuzzy cognitive maps - a comparison study. Acta Polytechnica Hungarica 7, 109–122 (2010)
[31] Klančar, G., Matko, D., Blažič, S.: A control strategy for platoons of differential-drive wheeled mobile robot. Robot Auton. Syst. 57, 57–64 (2011)
[32] Linda, O., Manic, M.: Online spatio-temporal risk assessment for intelligent transportation systems. IEEE Trans. Intell. Transp. Syst. 12, 194–200 (2011)

[33] Pozna, C., Precup, R.E., Minculete, N., Antonya, C.: Characteristics of a new abstraction model. In: Proceedings of 4th International Symposium on Computational Intelligence and Intelligent Informatics (ISCIII 2009), Egypt, pp. 129–134 (2009)
[34] Pozna, C., Precup, R.E.: Results concerning a new pattern of human knowledge. In: Proceedings of 2nd International Conference on Cognitive Infocommunications (CogInfoCom 2011), p. 18 (2011)
[35] Horváth, L., Rudas, I.J.: Modelling and solving methods for engineers. Elsevier, Academic Press, Burlington, MA (2004)
[36] Škrjanc, I., Blažič, S., Agamennoni, O.E.: Identification of dynamical systems with a robust interval fuzzy model. Automatica 41, 327–332 (2005)
[37] Johanyák, Z.C., Kovács, S.: Fuzzy rule interpolation based on polar cuts. In: Reusch, B. (ed.) Computational Intelligence, Theory and Applications, pp. 499–511. Springer, Heidelberg (2006)
[38] Kovács, L., Paláncz, B.: Glucose-Insulin Control of Type1 Diabetic Patients in H_2/H_∞ Space via Computer Algebra. In: Anai, H., Horimoto, K., Kutsia, T. (eds.) AB 2007. LNCS, vol. 4545, pp. 95–109. Springer, Heidelberg (2007)
[39] Hulko, G., Belavy, C., Bucek, P., Ondrejkovic, K., Zajicek, P.: Engineering methods and software support for control of distributed parameter systems. In: Proceedings of 7th Asian Control Conference (ASCC 2009), Hong Kong, pp. 1432–1438 (2009)
[40] Wu, M., Yan, J., She, J.H., Cao, W.H.: Intelligent decoupling control of gas collection process of multiple asymmetric coke ovens. IEEE Trans. Ind. Electron 56, 2782–2792 (2009)
[41] Bobal, V., Kubalcik, M., Chalupa, P., Dostal, P.: Self-tuning control of nonlinear servo system: Comparison of LQ and predictive approach. In: Proceedings of 17th Mediterranean Conference on Control and Automation (MED 2009), Thessaloniki, Greece, pp. 240–245 (2009)
[42] Ahn, K.K., Anh, H.P.H.: Inverse double NARX fuzzy modeling for system identification. IEEE/ASME Trans. Mechatronics 15, 136–148 (2010)
[43] Sanchez Boza, A., Haber Guerra, R.: A first approach to artificial cognitive control system implementation based on the shared circuits model of sociocognitive capacities. ICIC Express Lett. 4, 1741–1746 (2010)
[44] Abiyev, R.H., Kaynak, O.: Type 2 fuzzy neural structure for identification and control of time-varying plants. IEEE Trans. Ind. Electron 57, 4147–4159 (2010)
[45] Johanyák, Z.C.: Student evaluation based on fuzzy rule interpolation. Int. J. Artif. Intell. 5, 37–55 (2010)
[46] Garcia, A., Luviano-Juarez, A., Chairez, I., Poznyak, A., Poznyak, T.: Projectional dynamic neural network identifier for chaotic systems: Application to Chua's circuit. Int. J. Artif. Intell. 6, 1–18 (2011)
[47] Baranyi, P., Lei, K.F., Yam, Y.: Complexity reduction of singleton based neuro-fuzzy algorithm. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC 2000), Nashville, TN, USA, vol. 4, pp. 2503–2508 (2000)
[48] Baranyi, P., Tikk, D., Yam, Y., Patton, R.J.: From differential equations to PDC controller design via numerical transformation. Comp. Ind. 51, 281–297 (2003)
[49] Baranyi, P., Yam, Y., Várkonyi-Kóczy, A., Patton, R.J.: SVD based reduction to MISO TS fuzzy models. IEEE Trans. Ind. Electron 50, 232–242 (2003)
[50] Angelov, P., Buswell, R.: Identification of evolving rule-based models. IEEE Trans. Fuzzy Syst. 10, 667–677 (2002)

[51] Pedrycz, W.: Evolvable fuzzy systems: some insights and challenges. Evolving Syst. 1, 73–82 (2010)
[52] Tamani, K., Boukezzoula, R., Habchi, G.: Application of a continuous supervisory fuzzy control on a discrete scheduling of manufacturing systems. Eng. Appl. Artif. Intell. 24, 1162–1173 (2011)
[53] Kasabov, N., Hamed, H.N.A.: Quantum-inspired particle swarm optimisation for integrated feature and parameter optimisation of evolving spiking neural networks. Int. J. Artif. Intell. 7, 114–124 (2011)

Structural Improvements of the OpenRTM-aist Robot Middleware

Zoltán Krizsán and Szilveszter Kovács

Department of Information Technology, University of Miskolc, Miskolc-Egyetemváros, H-3515 Miskolc, Hungary
{krizsan,szkovacs}@iit.uni-miskolc.hu

Abstract. The robot middleware is a key concept in developing complex robot systems even in geographically distributed environment. Giving a handy API, reusable standardized components and communication channels together with some automatism, the robot middleware helps the user to build easily reconfigurable systems. The behavior of the components and the manner of interaction among them are standardized by the Robotic Technology Component (RTC) specification. One implementation of the RTC specification is the OpenRTM-aist. It is a well written and convenient modular system built upon the Common Object Request Broker Architecture (CORBA). The first version of CORBA is released in 1991. Nowadays the CORBA is getting more and more out of date. Our ultimate goal is developing a new robot middleware in which we apply a modern distributed framework instead of CORBA. As a first step of the substitution we suggest some practical extensions of the OpenRTM-aist by the adaptation of the Internet Communication Engine (ICE) for component communication and the introduction of the web application concept for system editing and control. Having these structural modifications the resulting system became more powerful than the original system has been. The chapter introduces some structural and implementation details of the OpenRTM-aist together with the results of the experiments done for the performance comparison of the original and extended system.

1 Introduction

Within the robotics area the task of robot systems can change quickly. If the job and circumstances are changing frequently, reusable and reconfigurable components are needed as well as a framework which can handle these changes. The effort needed to develop such components, depends on the used programming language, developing environment and other framework. Many programming languages can be applied for this process, but developing a brand new framework is a difficult task. Applying an existing framework as a base system, e.g. the OpenRTM-aist in our case could dramatically simplify the job.

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 287–300.
springerlink.com

In a small program development team where the members are not programmers, starting from an existing framework is the best way for further work. Frameworks and middlewares are gaining popularity through their rich set of features helping the development of complex systems. Joining together any robot framework and robot drivers can form a complex and efficient system with relatively minimal efforts. In many cases the task of the system designer is minimized to configuration of an already existing framework. On the other hand when an existing framework needs only some new features it can be simply extended with them. Improving an existing framework is also an easier job than developing a new one because the design and implementation of such system require special knowledge and skills (design and implementation patterns). In most cases the missing functionalities can be simply embedded to an existing framework, but there are some cases when it is hard to do so. An existing framework can be improved simply only in the case if it is well designed and implemented. In spite of this, building a brand new system from the basis is a much long process with a lot of efforts. In the area of robotics there are a lot of robot parts sharing similar features, so the concept of the robot middleware as a common framework for complex robot systems is obvious. Probably there is no framework which can fulfill all the above requirements entirely.

The primary goal is to find a robot framework in this environment which is easy to use, reliable and also easy to extend. If it is impossible to find a perfect one the secondary goal could be the improvement of an existing one. In final case when no acceptable framework exists we have to implement a brand new one from the basis.

In this chapter first we would like to compare some robot frameworks for giving starting impression, then we suggest some improvement idea for an existing framework.

From the existing robot middleware frameworks for further development, we have chosen the promising open source OpenRTM-aist system (it can be downloaded from [5]).

2 Robot Middlewares

In this section we try to introduce and compare some of the existing robot middleware approaches. The key consideration in robotics middleware is to give a handy API and as much as possible and automatism. Moreover the middleware also have to support as many operating systems and programming languages as possible. If any of the frameworks is applied for a specific job then some efforts are also needed for understand the concept and the application details.

2.1 Yet Another Robot Platform (YARP)

Communication in YARP (with more details in [1]) generally follows the *Observer* design pattern. Every YARP connection has a specific type of carrier associated with it (e.g., TCP, UDP, MCAST (multi-cast), shared memory, in process). Ports can be protected by SHA256 based authentication. Each Port is assigned a unique name. Every Port is registered by name with a name server. The YARP name server, which is a special YARP port, maintains a set of records, whose key is a text string (the name of a Port). The rest of the record contains whatever information is needed to make an initial connection to a given Port.

2.2 OpenRDK

The user's robot system can be developed from Agents, which is a simple process. A module is a single thread inside the agent process. Every module has a repository in which they publish some of their internal properties. Inter-agent (i.e., inter-process) communication is accomplished by two methods: through property sharing and message sending. RConsole is a graphical tool for remote inspection and management of modules. It can be used as both the main control interface of the robot and for debugging while we develop the software. It is just an agent that happens to have some module that displays a GUI. The framework can be downloaded from [2].

Table 1. Main features of YARP, OpenRDK and OpenRTM-aist

Name of the middleware	YARP	OpenRDK	OpenRTM-aist
Logical basic unit	-	Agent	Component
Execution basic unit	-	Module	Module
Information/functionality sharing	Encrypted data port	Property sharing, messages	Data port, service port
Registered item	Ports	Properties of Agent	Components
Communication possibility	TCP, UDP, MCAST (multi-cast), shared memory, in process	TCP	TCP
Applied RMI middleware	CORBA	CORBA	CORBA
Concept	Communication centric	Data centric	Functionality (data) centric

The YARP simplifies the problem of communication so it stores only the port and gives a flexible and extendable communication mechanism. The OpenRDK uses Agent concept and creates a repository for common data (property or message). The OpenRTM-aist manages the components not just stores information about it. It supports states of components and introduced the Execution concept

which gives some patterns for revealing robot jobs (periodic sampled data processing, stimulus response).

All of these middlewares supports the online components only so the unavailable components cannot be used, which is a strong coercion. Every time when the user would like to edit only the system all process of robot part have to run at the same time. If the robot system of user needs such filter or processing part, which is not available no chance to create it that time.

3 The OpenRTM-aist

The OpenRTM-aist middleware is a well written and convenient modular system built upon a CORBA basis. It is developed by the National Institute of Advanced Industrial Science and Technology - Intelligent Systems Research Institute - Task Intelligence Research Group. The OpenRTM-aist components are coequal, which means all participants have the same rights (editor, components). It has no central logic. The end-user can search for the online OpenRTM-aist components only at runtime using a graphical system editor. The online component search is supported by the CORBA naming service, which is a simple process act as a servant object used remotely by components and a graphical system editor as well.

The specification of RT-Middleware is defined in the "Robotic Technology Component 1.0" and "Super Distributed Object 1.1" standard (with the full specification in [3]). These standards describe the concept and the structure of a modularized robot system and its behavior. In the RT-Middleware (originally introduced by Noriaki Ando et al. in [4]) the software is modularized into components of RT functional elements (called RT-Component). Each RT-Component has an interface called Port for communication with other components. The RT system is constructed by connecting the ports of multiple components to each other for aggregating the RT-Component functions.

The advantage of OpenRTM-aist is an easy and user friendly way to create and use robot parts. Two actors of system exist:

Component developers, who make the components. For components development many programming languages can be used such as C++, Java and Python. Some tools help the process-development by automatically generating the skeleton of the components. The developer has to fill the generated skeleton source, by concentrating on the business logic only.

The end-user is a person, who has no knowledge about the programming language, as he has no intention to develop components. These people want to use the components only for their work. In their case only the address of a name server must be known to be able to use the system. After the editor connects to the name server it sends the list of the available components, from which the end-user can built his robot system.

For component creation the middleware also contains a graphical editor, named RTC Builder. After the definition of component, its source files are automatically

generated and stored in an xml file. The developer's job is just filling the body of the generated source code. The number of applications proves that it is an easy, and user friendly application.

4 Extending the OpenRTM-aist

For extending the features of the OpenRTM-aist, our first goal was to improve the support of the offline components. The complete robot system, which parts can run in different places, can be explored, and edited by a graphical application called RTC system editor. The user can create, and activate a new system by allocating, and connecting existing online RTC components. The structure of a robotic system and its communication details are shown in Fig. 1. Behind the scenes, the communication among allocated components can be established via CORBA middleware as all components are implemented as CORBA servant objects. These online components can be used controlling by the RTC graphical editor based on the naming service of CORBA. If the connection between editor and any of the components is broken, from the editor's point of view the component has become offline, and hence robot part is automatically removed from the user's system (the component together with its already defined connections if there are any).

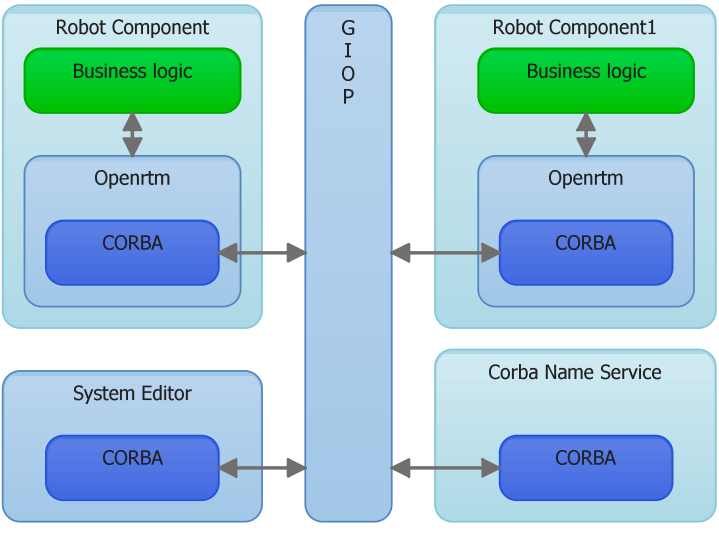


Fig. 1. Structure and communication among parts of OpenRTM-aist.

The second improvement we need to make is related to the client authentication together with the applied access policy. In the current system any of the available components can be used freely for every user without any security restrictions. If

the IP of the name service or the parameters of components (IP and port) are known then any person can use the resources of the host.

The third improvement needs to be made is the development of a lightweight graphical system editor. The current system editor's size is about 100 MByte, which results in a long download time. The updates of the editor must be downloaded and installed by the end-user explicitly.

Building blocks of the system are RT components (e.g. robots) with well-defined architecture and the ability of running independently from each other. Their connections (communications) are possible only via ports. The components can send/receive data to/from each other via a dataport. The components can live, work and die. They can even have more states controlled by the system and the user.

5 Details of the Extension

To be the first step of the OpenRTM-aist extension we had two main goals:

1. To provide a new communication channels for components based on ICE, in which the components can share their behaviors.
2. Development of a new platform-independent lightweight graphical system editor.

5.1 The ICE Service Port Extension

One of the key underlying ideas of the improved system is the ICE technology (with more details in [6]) integration. The OpenRTM-aist system is already contains a service port, which is the sole way to publish the behavior of components among other components, but this port is based on CORBA. Like CORBA, the ICE Framework is also an object oriented distributed platform that can run on several different operating systems and many programming languages e.g. C++, .NET, Java, Python Objective-C, Ruby, PHP, and ActionScript. Compared to CORBA, ICE also supports .NET and more programming languages (about 60). It could be a new platform for replacing CORBA in the robot middleware keeping its strengths, while avoiding its weaknesses. Moreover ICE can also simplify difficult programming model of CORBA.

The integration of ICE system leads to a number of benefits, which will be available for the robot middleware components too.

The first real advantage of IcePort is the *simple, easy to understand* and *use*, programming model (i.e. it is a simplified programming API). Developers do not need to learn lashing of documentation, which makes the development process effortless and quick. The *dynamic invocation* and *dispatch interfaces* allow the

developer to write generic clients and servers that need not have compile-time knowledge about the types used by an application. This opens the possibility of creating applications such as object browsers, protocol analyzers, or protocol bridges.

During the specification of the IcePort extension we have recognized a novel feature of the extended system which was inexistent in the original CORBA based version. In some situations it is necessary to make *connection among one consumer and multiple service provider components*. This type of communication is not supported in the original system, nevertheless it is often necessary for robot configuration, (e.g., one robot item can be controlled by multiple input devices). In case of ICE technology this type of communication can be easily implemented. The communication encryption is a complicated task in the free implementation of CORBA, therefore an easy to use and easy to configure encryption layer is required. Ice also provides an *encrypted data stream* between multiple ports, which can easily introduced at the end-user side solving the problem of secure communication with minimal effort. The communication stream is usually TCP based, but in some situations the UDP datagram could have some benefits. The *UDP protocol* can also be used for one-way operation, which is a faster than TCP e.g, broadcasting information in a publish-subscribe kind of application or logging activity. The overhead in negotiating for a TCP socket and handshaking the TCP packets is huge, hence for one way communication UDP is at least two times faster than TCP. In ICE the UDP is also supported by API. The components can be organized into farms, so the system can be structured to be more reliable.

5.2 Web Based Editor System Introduction

The main components of a robot system usually have a rich graphical interface. E.g. the VirCA system (will be discussed more detailed in the next session) the main component has an interface in three-dimensional virtual space. It is a time consuming process to switch between the separate editor and main graphical interface. It would be much better if the editor surface could be displayed in the main graphical interface.

The new editor is a lightweight System Editor that can work within a browser. It can be embedded in a web application. The actual robot system can be saved for further editing in the client side in xml structure, or saved on the server in a database. Our editor can also use favorite components which are stored in a searchable database.

As it is discussed in the introductory section, in OpenRTM-aist if the network connection is broken, then the box of component is disappeared and all the related connections are deleted too. In our extension, the editor system just illustrates the "not available" state of component, but the user can still use it further. This is one of the key improvements of our extension. During the design time all components and connections of the robot system are grayed, showing that the component

status is unknown. Compared to the real situation one or more components of the grayed ones can be unavailable meantime, because the system displays the last known parameters of missing parts. When the robot system is activated by the end-user, the extension logic tries to activate the appropriate components and to re-establish the connections.

The new editor provides the virtual *master component* concept in the form of special components. The virtual item allows the division of the system into several smaller parts (Divide et impera). Having this extension the edition of the parts can be simpler as the unnecessary and distracting details can be hided from the user. The virtual item can include other virtual or regular components too. The activation command this case means the activation of all the included real items. Any ports can be published as new port of the virtual item.

The original editor is a plugin of Eclipse, so running the original system editor the user must have a big Eclipse environment (about 100 Mbyte), and a high Eclipse dependency with limited graphical opportunities.

In our lightweight system editor the end-users need only a browser to use the system, as the editing system now is a web application. At the beginning of each service usage the browser always checks the URL, so the last version will be accessed automatically, hence the version control is also guaranteed.

In the new system, thanks to the appearance of the Web layer in the structure, the further extensions become simpler, only minimum expenditure investment is needed (such as advanced user management, integration into other systems, support legacy system, searching the components, advertisement).

In our case between the host and the end-user there is an XML-based communication over the HTTP protocol. The applied Web application supports all operating systems with any browser, so building heterogeneous systems are also supported.

For an end-user, who does not want to run own robot components, there is no need to use the CORBA protocol or install any CORBA endpoint, because the lower level of OpenRTM-aist is isolated by the web application tier. The extended robot system has more parts as it is shown in Fig. 2.

In the new OpenRTM-aist architecture, some new items are appearing:

The Client tier: At the end-user side a flash application is running in a browser. It has only one reason, to present an interface to the user. It sends requests to the presentation layer and gets responses. The basic editor function is the handler here (e.g., system editor area, shape for component, component insertion). Supporting the offline components and minimizing the communication efforts, the required connections will be created just in time when the robot system is starting. After the presentation tier gets the commands and parameters from the client tier, the given values are checked here, and only valid commands are forwarded. For this task it can use a database tier and a common logic layer too. This layer is also independent from OpenRTM-aist. Two varieties of this layer exist: console and web typed. For automatic running (and integration test) our editor system has a command line interface called OpenRTMExt_Tools (Java application) and a graphical

interface (Java web application) as well. The server logic is implemented as jsp pages and servlets to serve the request. In case of console application it is a simple text (generated by java application), but in case of graphical application the form of communication is XML based.

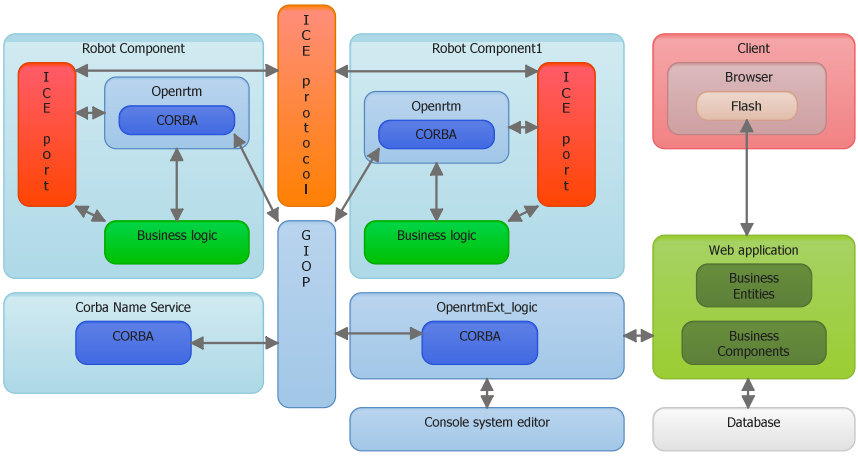


Fig. 2. The architecture of extended OpenRTM-aist.

The Web application and the console system editor tier are belongs to the presentation layer running on server. They translate the request by the server logic. The results are simple texts or XML documents sent to the client tier.

The OpenrtmExt_logic layer is used by Presentation layer. This layer coordinates the application collections of Java Bean which are worked together in this layer. The logic is run in threads for better performance. The Master component concept is also introduced here. It gives virtual elements which simplify the topology of the user's robot system. It is a platform independent programming library (lib). This tier depends on low level communication such as CORBA and ICE. This layer hides the details of OpenRTM-aist, introduces some new features and establishes some new complex instruction which is not available in the original system. The request, which is get from client is translated to CORBA method invocation and forwarded to any participant. After the appropriate remote object(s) has done its work (any value can be sent back), the result(s) goes back to the presentation tier. We have implemented this layer to guarantee the same behavior in case of the console and the graphical interface as well.

The Database tier stores the profile of users and the details of the robot systems. Applying a stored procedure some functions can be moved to the database tier. It ensures a quick and searchable data background to the system.

6 An Application Example, the VirCA System

The Virtual Collaborating Arena (VirCA) is a virtual laboratory, which gives an opportunity to a team of researchers or industrial engineers to collaborate with each other. It is possible in an augmented environment to control virtual and physical devices remotely in a simple way. The VirCA system was developed in the Cognitive Informatics Group of MTA SZTAKI research institute, more information about it can be found under [7] and [8]. The main idea of VirCA is placing the physical device in a generated virtual space, as it is presented in Fig. 3.

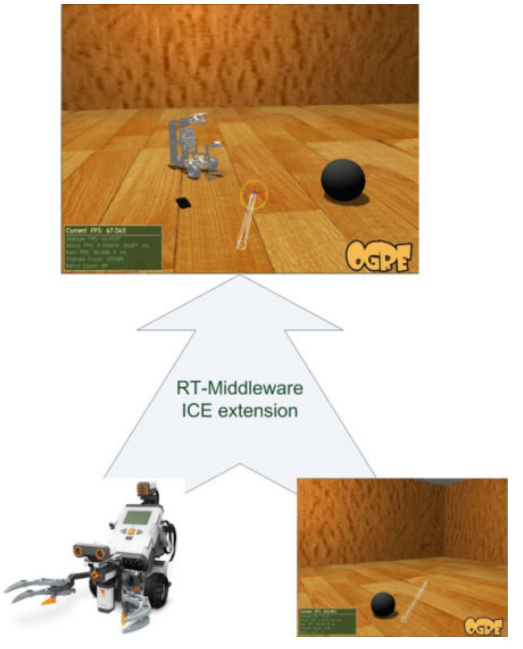


Fig. 3. The concept of VirCA system.

Behind the scene this system consists of more extended OpenRTM-aist components. Special components run in all places connecting to a simple CORBA name server to register their device lists. When the components are up and registered on the network, the operators can choose their input and output devices (cognitive informatics equipment and robots) and use them to solve the given task.

The main task of the VirCA is the visualization of physical devices, connecting and controlling them and establishing interaction among real and virtual objects. As a secondary goal it is also able to connect more virtual spaces form an extended virtual reality environment. The VirCA had been established as an extended OpenRTM-aist system. Some cognitive components had been also developed for the VirCA. The members of the development team are engineers, not programmers and they do not need to use CORBA code.

In the VirCA system the following mandatory components are needed:

3D Space: A 3D visualization component which continuously renders the virtual word. After the user connects (using a graphical interface) any device to this main component, the 3D space shows the model of the physical device in the appropriate position of the virtual space. The virtual object can also "sense" the outer object's presence (e.g. virtual and physical objects can react each other). The user can send a command to a physical device via graphical (3D) interface. The 3D Space component has two interfaces: Register interface and Commander interface. The Register interface is an ICE service provider. The physical (cyber) devices can register/unregister themselves into the virtual world, or send their new positions. The Commander interface is an ICE service consumer. It sends commands to the physical entity, e.g. move to the specified position or take something or do a job, etc. The 3D space component has two kinds of implementation. One can run on normal personal computer as usual graphical desktop application. The other is running in the CAVE immersive 3D environment, which is more realistic because the user can stand inside a cube of projected 3D screens having the feeling to be a part of the surrounding virtual space.

Cyber devices: Several different physical robot devices can exist at the same time in different places. Each type of robot parts can have its own component, which can exist in several instances in the same time. We support more devices: e.g. Lego NXT, KUKA industrial and NAO humanoid robot. The cyber device component must have two mandatory interfaces: the *Commander interface* which is an ICE service provider and the *Register interface* which is an ICE service consumer. The *Commander interface* gets commands from 3D main components or from any other components, which want to control them. The *Register interface* registers into, or unregisters from the virtual space.

There are some optional components too. The *Camera component* is providing pictures or movie about a physical space. The *Observer component* gets the picture, which is provided by a camera component, identifies the objects and sends events to 3D engine, which notifies the others. A variety of *Controller components* forms can exist. The user sends commands to the cyber device via this component by voice or hand-gestures. The input of these components can be provided by camera or microphone or any other cognitive sources. The system needs to be improved in such case if it uses more cyber devices then the Commander interface or the 3D system consumes more Commander interfaces of more cyber devices.

7 Experiments

Having a new and improved system with some novel functionalities, the key question is the system performance. The main goal of the test is the comparison of the performance of CORBA and ICE service ports. We have implemented two test components, the test provider and the test consumer. The provider has an IcePort and a CorbaPort. These two ports have the same interface (IDL in case of CORBA and slice in case of ICE) and the implementations are also similar. Similarity

means same number of program steps but of course the instructions are dependent on the studied technology. The other component has two ports as well, which consume the test. After creating the two component instances, the appropriate ports were connected. The test cases were run sequentially step by step under Windows and Linux. The amount of time it took for a client to invoke the test operation in a server and to receive the results of the operation was measured. The latency and throughput of the new ICE service port was tested using the demos of the Ice distribution.

In the latency test the operation has no in-parameters, so no data is sent with the invocation. Similarly, because it does not have out-parameters and has a void return value, no data is returned by the operation.

Throughput determines how much data the middleware can move per unit of time. Throughput heavily depends on the type of data that is sent. Byte sequence was selected for transmission as it is the most efficient data type, which does not require further processing. We have sent 50 Megabytes from consumer to provider in 100 Kbytes steps.

The test function of the ICE system was used for comparison of the two service ports. Both Ice and CORBA (omniORB in case of C++, and jacORB in case of Java) are run with no special configuration, so they use the default options. The time required for 100000 function calls was measured. In case of loopback device test the CORBA is 1.5 times faster than ICE. In real usage the endpoints are on different machines therefore this test is insignificant. The results of the test for different hosts are shown in Fig. 4.

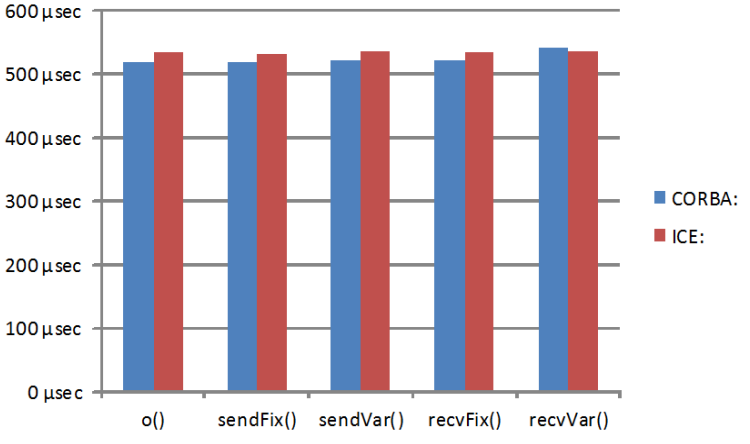


Fig. 4. The performance comparison of Iceport and CORBA port per-formance in case of C++.

According to the results presented in Fig. 4, the performance of the two technologies are almost the same. Using ICE port the features are significantly better and the performance is nearly the same as CORBA has. The two most common programming languages (JAVA and C++) were also tested, with the same results.

In spite of the similar performance the application of the ICE service port is strongly suggested instead of the CORBA port, as it has a better programming interface and the customization is also much simpler. The performance of ICE in case of different programming languages and operating systems is also different. The best performance was measured in case of C++ and Linux operating system. The worst results were produced by the Windows and Java combination. The results are summarized in Fig. 5. According to the opinions of the developers, the reasons of these performance differences are implementation issues of the ICE network layer and the java virtual machine.

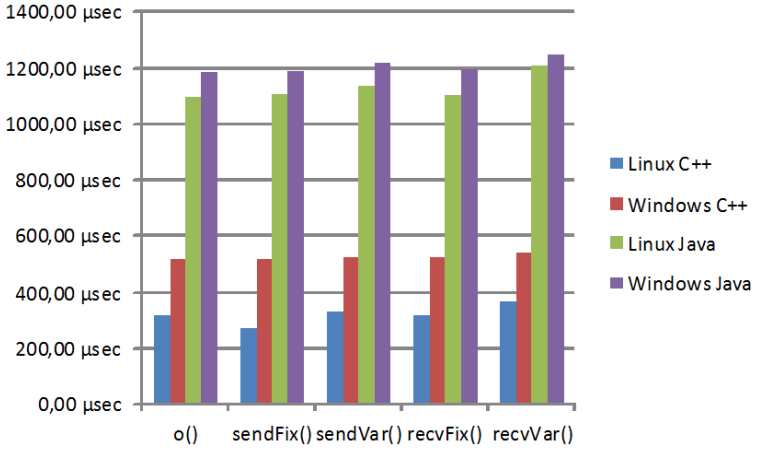


Fig. 5. The performance comparison of Iceport in case of different platform.

8 Conclusions

This chapter presents some improvement ideas for the existing framework OpenRTM-aist. The contribution introduced in this chapter is the integration of ICE into the OpenRTM-aist and the introduction of a new lightweight Web based system editor.

The ICE integration was carried out by extending the existing features, without their modification. On the other hand thanks to the IcePort extension the quantity of the available system services have been improved significantly. The extension can be easily added to the intact original OpenRTM-aist system leaving the original features unchanged in order to avoid compatibility issues. The user client of the new editor is a simple web browser which does not require the installation of any special applications such as CORBA and Eclipse which was required for the original system editor. The new editor is built upon the infrastructure of a web application and now it supports component sharing in a trusted and controlled way.

Acknowledgments. This research was partly supported by the Hungarian National Scientific Research Fund grant no. OTKA K77809, by the Hanbat National University, Korea, and by the Hungarian National Development Agency, NAP project (KCKHA005, OMFB-01137/2008). This research was carried out as part of the TAMOP-4.2.1.B-10/2/KONV-2010-0001 project with support by the European Union, co-financed by the European Social Fund.

References

[1] http://eris.liralab.it/yarpdoc/index.html
[2] http://openrdk.sourceforge.net/
[3] http://www.omg.org/robotics/
[4] Ando, N., Suehiro, T., Kitagaki, T., Kotoku, T., Yoon, W.K.: RT-Middleware: Distributed component middleware for RT (robot technology. In: Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, Canada, pp. 3555–3560 (2005)
[5] http://openrtm.org
[6] http://zeroc.com
[7] http://virca.hu/
[8] http://www.coginfo.sztaki.hu/

Decision Support at a New Global Level Definition of Products in PLM Systems

László Horváth and Imre J. Rudas

Óbuda University, John von Neumann Faculty of Informatics, Institute of Intelligent Engineering Systems, Bécsi út 96/B, H-1034 Budapest, Hungary
horvath.laszlo@nik.uni-obuda.hu, rudas@uni-obuda.hu

Abstract. Knowledge based support of decision making in product definition using modeling system may produce the advantage of coordination different opinions, attempts, and intents. In order to achieve this, special interaction between engineer and product object definition processes was proposed by the authors in order to communicate content for decision in the form of expertise and experience based knowledge. This knowledge is required to represent engineer in the course of coordinated definition of product objects. This chapter introduces a new method in order to establish a global level of the decision making on product object parameters. Global level control organizes product object generation processes on the local level of currently applied product entity definition. Local level processes should be modified in order to accept control from the global level. This control replaces direct human definition of product object parameters. Other new methodological element in the proposed modeling is that the only allowed way of product object definition is through definition of function and quality based behaviors of product. The main contribution of this chapter is introduction of the contextual chains along which communication is done from human thinking to product model entity parameter generation. The proposed method is demanded to be capable of serving product model definition for lifecycle in a product lifecycle management (PLM) system.

1 Introduction

The work of engineers has changed by wide application of product lifecycle management (PLM) systems for development, production, application, and recycling related engineering activities by using of a single but very complex product model. Model of a product may include thousands of objects representing subassemblies, parts, form features, analysis model entities, analysis results, device control programs, and many other entities. Complexity of model is not only the result of

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 301–319.
springerlink.com

high number of product related engineering objects (EO) but also the high number of unorganized contextual connections within it. Correct context definitions are essential because consistency of a product model is assured by definition of any engineering object in the context of appropriate engineering objects. In current product models, no organized means is available for tracing contextual chains in product model.

Product model is defined by authorized and responsible humans. The same engineering object may be defined by several humans acting directly or indirectly. Direct acting human communicates with engineering object generation processes while indirect acting human influences product model by higher level decision, law, standard, proven solution, or modeled knowledge.

The authors of this chapter recognized that current product modeling systems were not suitable for essential human influence coordination and the relevant model representation, and processing functions. This problem is considered as one of the most important areas of research in product modeling for the next future. In order to achieve better decision assistance including better communication between engineers using product model as a medium, the authors of this chapter proposed knowledge based modeling method as extension to current product modeling systems. Because knowledge is property of companies, implementation of the proposed modeling is planned by using of open application development surface of PLM systems. Essential issues on knowledge acquisition at human-computer interaction for product definition as it was conceptualized by the authors of this chapter were discussed in [11].

This chapter starts with an introduction of history of product modeling and the method by the authors of this chapter for the definition of consistent product model. This is followed by discussion on product definition in modeling environment and on a basic problem with product model based communication between engineers. Next, a new extended human-modeling process communication and collecting information from human thinking process in it are shown as contribution by the authors. As processing of human influence attempts, coordination of product behaviors, analysis of affect propagation, and building of extended product definition space are explained. The proposed model is introduced as extension to the conventional product definition space. Finally, few words explain implementation of the proposed modeling in product lifecycle management (PLM) environment.

2 Model Mediated Communication

Product model increasingly serves as medium amongst human and controlled equipment participants of engineering processes. Product model for lifecycle engineering is a result of long development in computer based product definition.

Brief history of this development is outlined in Fig. 1. Partial solutions in the form of CAD/CAM, CAE, etc. systems concentrated on engineering areas where computer description and processing of shape as well as essential analysis were urged by development of automation during the 70-90's. Unified representation of shape was realized by boundary representation (B-rep) where non uniform rational B-spline geometry and Eulerian topology brought a new age in 80-90's.

Task and user specific objects were introduced by using of form feature definition and recognition for part model construction during the 90's. Activities for integration of partial models into a unified and structured model of product were concentrated in the project to establish the product model standard STEP (Standard for the Exchange of Product Model Data, ISO 10303) during the 80-90's [12]. Recent paradigm of product lifecycle management (PLM) brought the product model that is suitable for the support of all engineering activities. PLM systems apply recent information technology.

Integration of product elements in a single model made it possible to integrate active knowledge for product object generation in product model. This knowledge

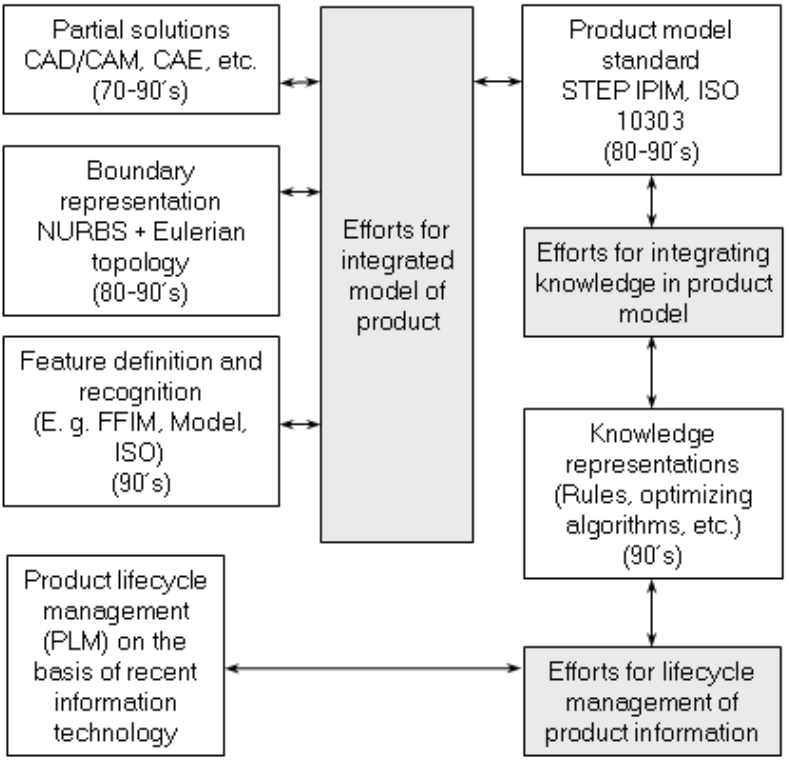


Fig. 1. Brief history of computer based product definition.

redefines engineering objects in product model at any change of the related situations and events. Engineer friendly knowledge representations such as rules, checks, reactions, optimizing algorithms, etc. are available.

Complexity of product model requires means for providing consistency. The authors of this chapter proposed product modeling with consistency control for human source, product object, and product behavior (Fig. 2). For this purpose, human influence, human intent, contextual connection, product behavior, product object, and decision entities are represented in a contextual chain. According to the utmost aim, human intent is always defined in the context of human influence, contextual connection and product behavior are always defined in the context of human intent, and so on.

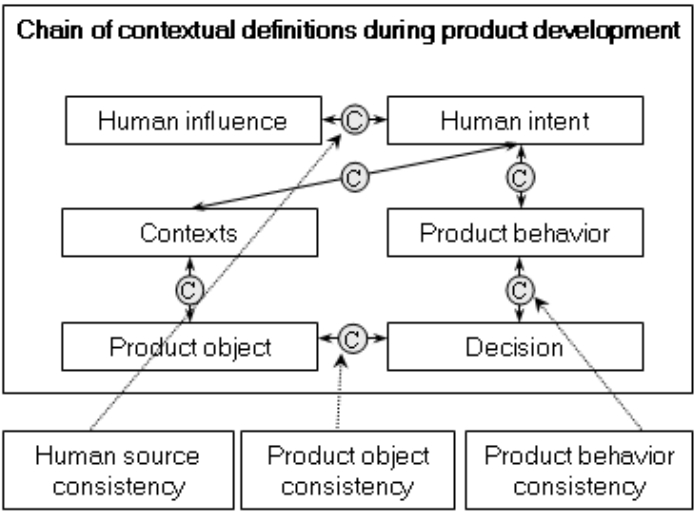


Fig. 2. Context for consistency.

3 Human Influence and Communication

Product definition or model space is the place where product is defined by direct and indirect human influences. This three dimensional space is characterized by Descartes coordination system, and it is in possession of modeling capabilities to shape centered representation of parts, their connections, and other objects necessary during lifecycle management of product information [1]. Extensive dialogue surface is available for engineer communication (Fig. 3). Engineers define product elements as objects, knowledge for the definition of object parameters and constraints, make analyses, and define control programs. Interoperability offers

interfaces and translators for the communication with other PLM systems, production equipment controls, and capturing equipment. The proposed modeling is intended as an extension to the currently applied product definition space. Capabilities for human intent based representation of engineering objectives and intended influences, and consequence changes for engineering object definitions are produced. The proposed extension is based on the concept and method of modeling by information content in [2].

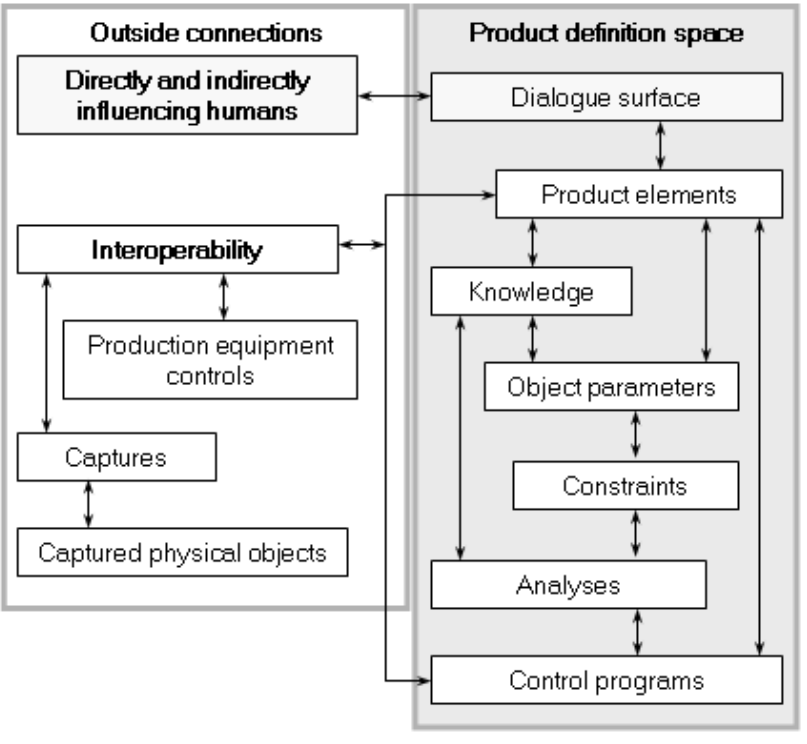


Fig. 3. Product definition in modeling environment.

Product definition space serves communication between human who defines and human who applies the engineering object A. Main problems with the current product modeling is related with this communication. Human who applies engineering object A thinks on modification of this object and definition of related objects.

One of the purposes of the proposed modeling is improving communication between users of product definition space. The currently applied communication is outlined in Fig. 4. A human defines engineering object A. This object may be anything that is needed during lifecycle of the actual product. Work of this human is affected by indirectly influencing humans restricting the solution space. Indirectly affecting humans communicate higher level decisions, standards, legal decisions,

and law. These influences determine given engineering objects and their parameters and must be considered as constraints. In other words, engineering objects are defined in the context of these influences. Authorized and responsible human thinks on object definition in product definition space and communicates results of thinking process through dialogue surface with the product definition space (also called as model space) in the form of features, their parameters, and connections of parameters.

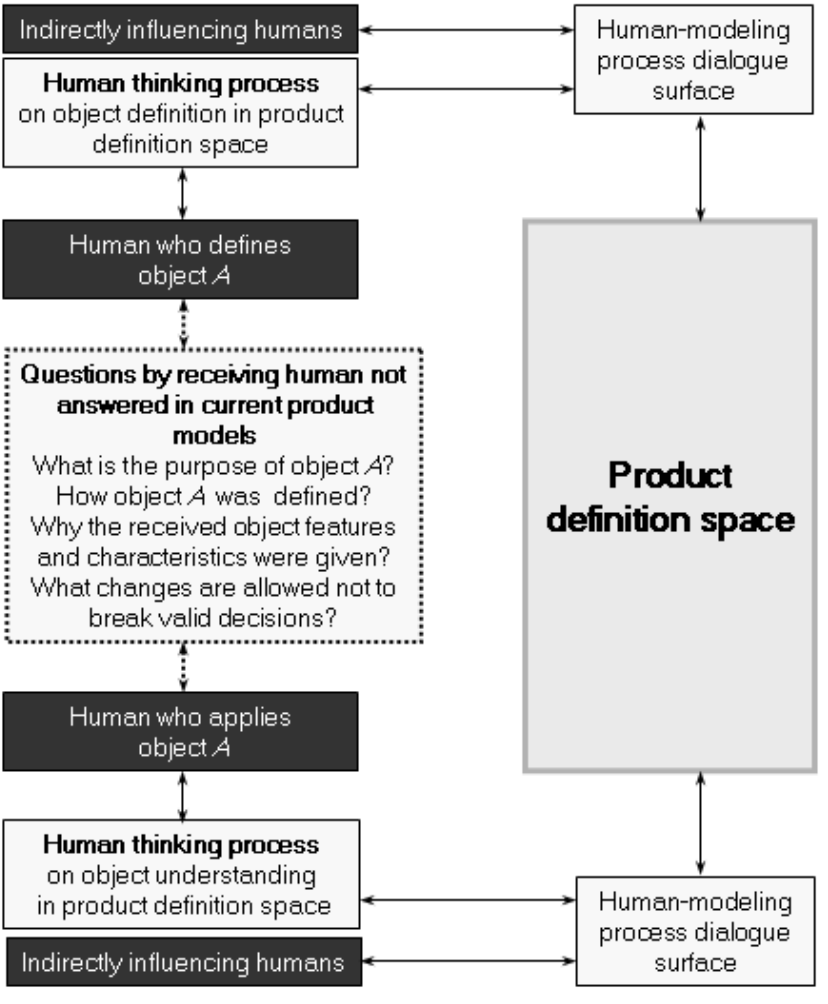


Fig. 4. Not properly informed human at model application.

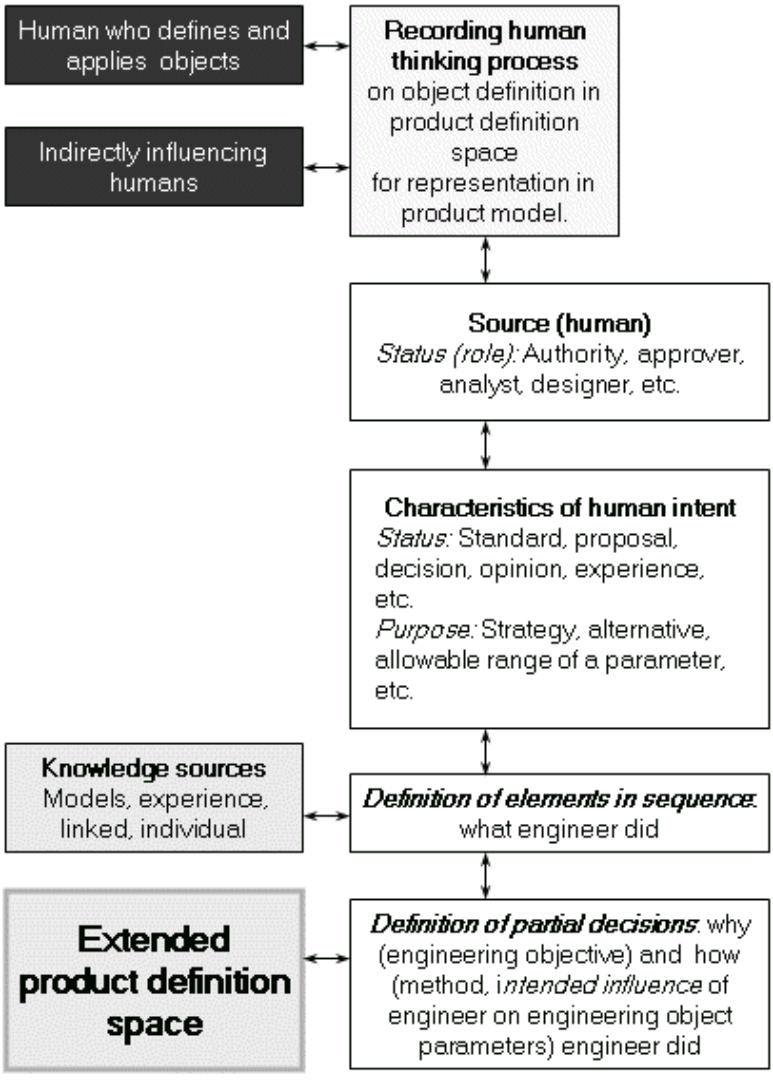


Fig. 5. Extended human communication.

Questions are emerged those can not be answered in product definition space. Main questions are for purpose of object A and the method that was applied at its definition. Background of definition of received object features and characteristics would be necessary to evaluate the model. Engineer, who receives model, would like to know that what changes are allowed without breaking of valid decisions. The modeling proposed in this chapter is intended to have capabilities to represent answers for the above questions in an extended product definition space.

4 Knowledge Definition for Product Model

In order to better communication between engineers, more information must be represented from the process during which a proposed influence is developed in the human mind. The authors of this chapter started from this recognition and elaborated a modeling called as representation of human intent (Fig. 5). Their former publication explained the basics of this method [3].

Simple communication on engineering object related results was replaced by communication of knowledge from human thinking process on object definition for the record in engineering objectives and related definition methods in product definition space. For this purpose, appropriate entities and representations were defined in product model. Relevant knowledge related research concentrates on knowledge formalization that is suitable for lifecycle management of product information [14], and representation of knowledge from multiple experts for product definition [15]. The former more general approaches have not proven in practice where quality is given for the product by human experience, expertise, and intelligence. In [17], knowledge acquisition process is applied to capture structured systematic knowledge.

In an intent model, first of all status or role of the intent source human should be recorded such as authority, approver, analyst, designer, etc. Following this, knowledge sources and general characteristics of human intent should be defined. Knowledge sources are, for example, models, experience representations, linked objects, and individuals. Characteristics of human intent are status and purpose. Status may be standard, proposal, decision, opinion, experience, etc. Examples for purposes are strategy, alternative, and allowable range of a parameter.

Human thinking process record includes definition of elements and partial decisions. Definition of elements is a sequence of what engineer did. Definition of partial decisions includes why and how engineer did engineering object definition. Answer for question why is engineering objective while for question how is method and intended influence. Method and influence are defined for engineering object parameters. Product definition space is extended by the above new entities.

Definitions of elements and partial decisions are interrelated in the human thinking process record. Fig. 6 illustrates interrelations by a simple example. Engineer observes the problem and decides engineering objectives in the form of functions and quality. For example, function of a part object is supporting other object and must keep a deformation limit under a specified load. Next element is definition of object with the decision on type and parameter set of that object. As background of these decisions, choice and method of selection are recorded as they were applied by the engineer. The subsequent element is analyzing object variants.

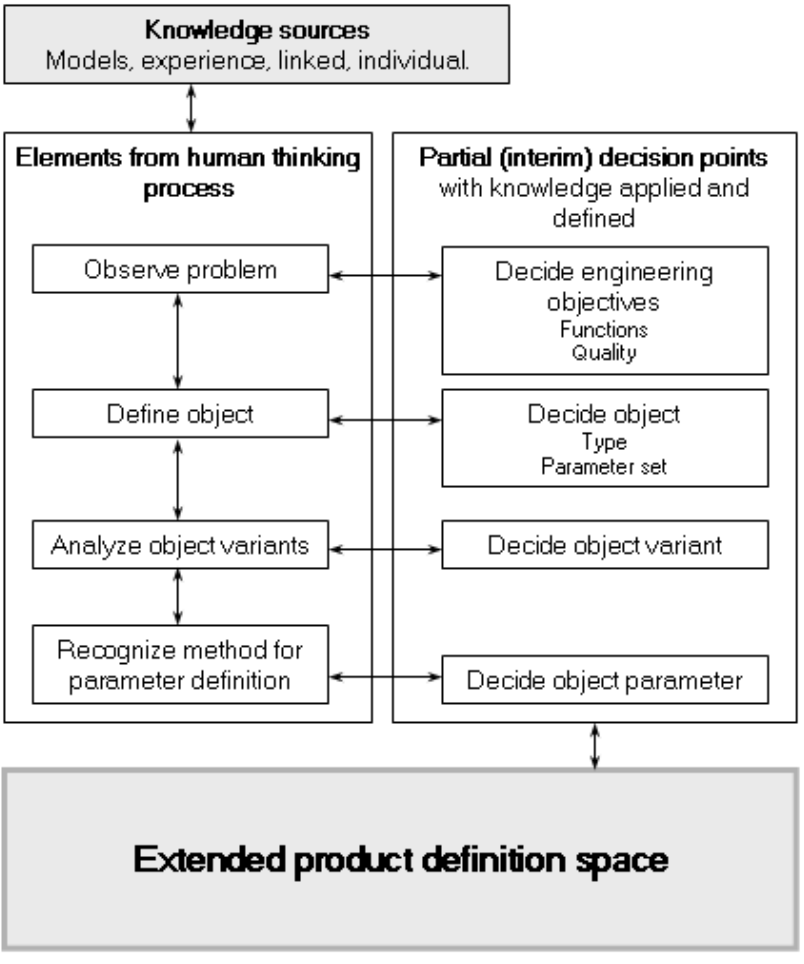


Fig. 6. Information from human thinking process.

The mating decision is about the variant of object. This decision may be temporary because variants are available for future re-evaluation under changed circumstances. Finally, method is recorded for parameter definition and object parameters are calculated by using of the recorded method. Records may be links to available knowledge or outside expert contribution.

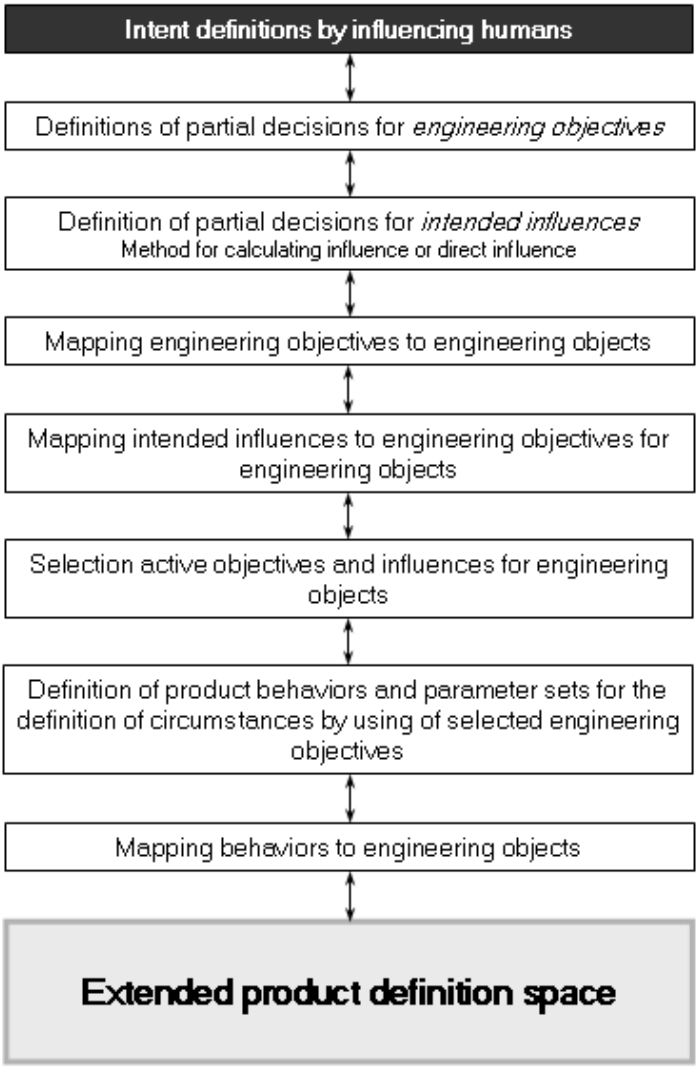


Fig. 7. Processing objectives and intended influences.

5 Coordinated Product Behaviors and Affect Propagation

The conventional product model is a structure of product objects such as parts, part connections in assemblies and mechanisms, analysis results for parts, control programs for equipment, etc. However, these entities ere defined for product

functions and qualities, in other words for engineering objectives. Behavior is the representation of engineering objectives. Definition of product objects from behaviors is inevitably a knowledge intensive task. Representation of knowledge in product model also serves modification of model in case of changed situations for behaviors. Situations are defined by circumstances both for product object and outside affect parameters.

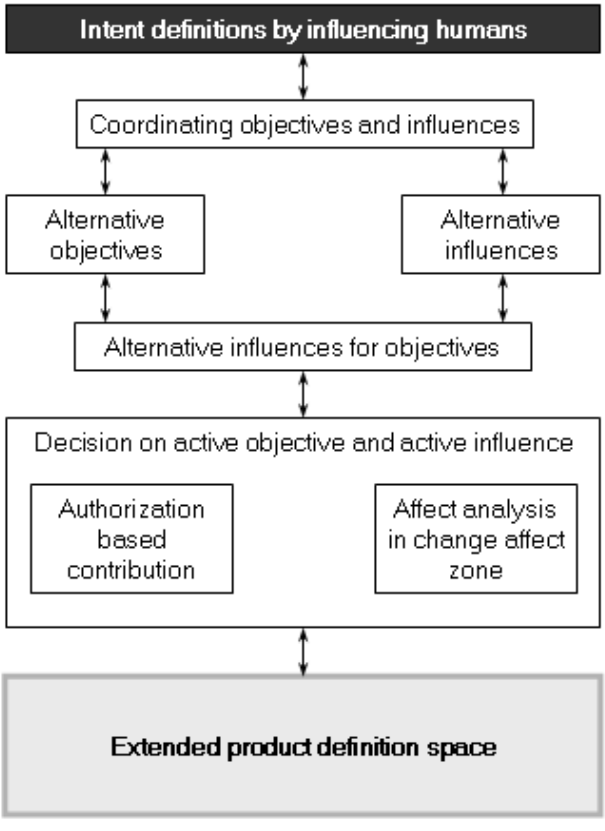


Fig. 8. Selection active objectives and influences.

The above findings by the authors of this chapter make special emphasis on corporate knowledge representation in product model. Author of [4] introduces an approach to definition and mapping of knowledge based on the point of view of expert. Personalization and codification are introduced for the development of a multidisciplinary framework in [8]. Form feature information model in the STEP standard for product modeling and other solutions use multilevel structures. Parametric and semantics feature definition supports communication between users and engineering modeling environment in function-oriented product modeling, as

it is shown in [16]. Authors of [5] recognized that, a new overall approach to leveling was necessary to be developed in order to enable interoperability of STEP Application Protocols at meta-data and knowledge levels. Similar results are important in knowledge technology, because the STEP product model standard and its application at leading industries opened the road for integrated solutions in product engineering.

In human intent record on thinking process, definitions of partial decisions for engineering objectives and for intended influences are followed by mapping of these entities to relevant engineering objectives (Fig. 7). Method is recorded for calculating object parameter values for influence or for means applied by the engineer to the definition of direct influence on object parameters. Because variants of objectives and influences are included, active objective and influence variants should be selected for engineering objects. Active variants may be decisions or only default ones those are allowed to be replaced by other recorded variants.

Following the above procedure, product object behaviors and parameter sets are defined for the definition of circumstances by using of selected engineering objectives. Behaviors are mapped to engineering objects and product definition space is extended by the new entities. Behavior definitions for engineering objects in lifecycle management of product information are explained and discussed by the authors of this chapter in [6]. Behavior definitions are based on situation modeling and simulation. This method is applied for wide applications in various areas of industry and research [7].

Some elements of selection active objectives and influences for engineering objects are shown in Fig. 8. Objectives and influences are coordinated in case of multiple intents for the same object. Alternative influences are passed to decision on active one while alternative objectives may demand definition of influences by object related procedures in the modeling system. Decision on active objective and active influence may be authorization based contribution or may require affect analysis in change affect zone. Product definition space is extended accordingly. It can be seen, that the proposed method allows for definition of the required final objectives and influences in accordance with authorization of engineer.

Engineering objectives are represented as features to give a well-defined modification of a product or its structural and other unit. Although features are defined during product definition, sometimes their recognition is necessary in an existing model. In [9], geometric feature recognition is introduced by using of graph and hint based methods, convex hull decomposition, and volume techniques.

6 Extended Product Model Description

As it was stated above, the proposed modeling is an extension to the product definition space of current product modeling (Fig. 9). Elements of extension are defined in the context of influencing humans. Human thinking process is about

object definition in product definition space. At the same time human is also thinking about representation of engineering object related information in product model.

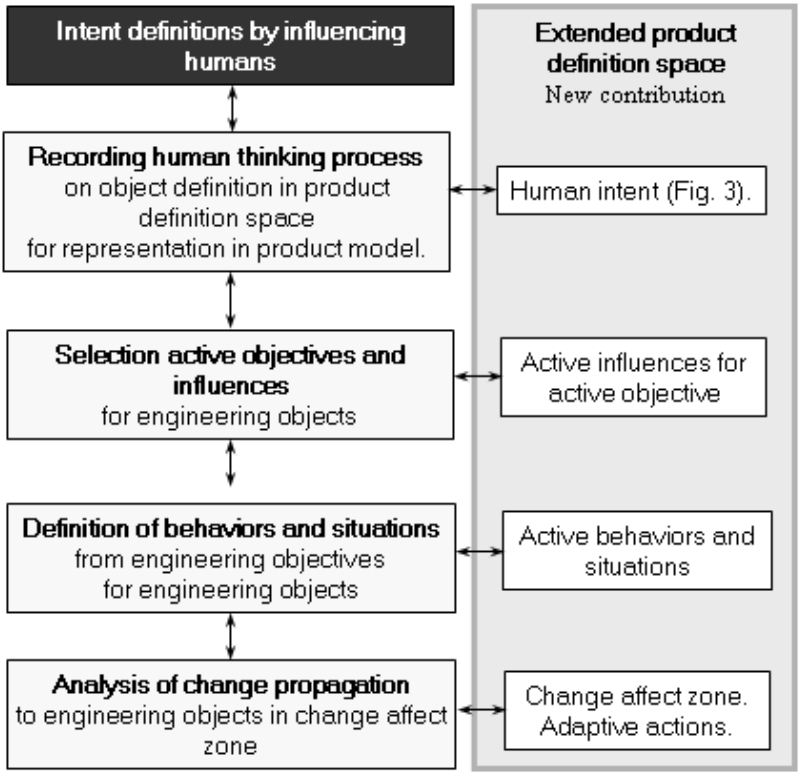


Fig. 9. Building extended product definition space.

Extensions are for record of human thinking process, selection of active objectives and influences, definition of behaviors and situations, and analysis of change propagation. New contributions to the product model in these extensions are human intent (Fig. 5), active influences for active objectives, active behaviors and situations, and change affect zone width adaptive actions, respectively. Results of decisions in product model are represented as constraints. Constraining should be organized and consistent. As an example, a theory called as generalized local propagation is proposed in [10] as a foundation of algorithms for solving constraint hierarchies.

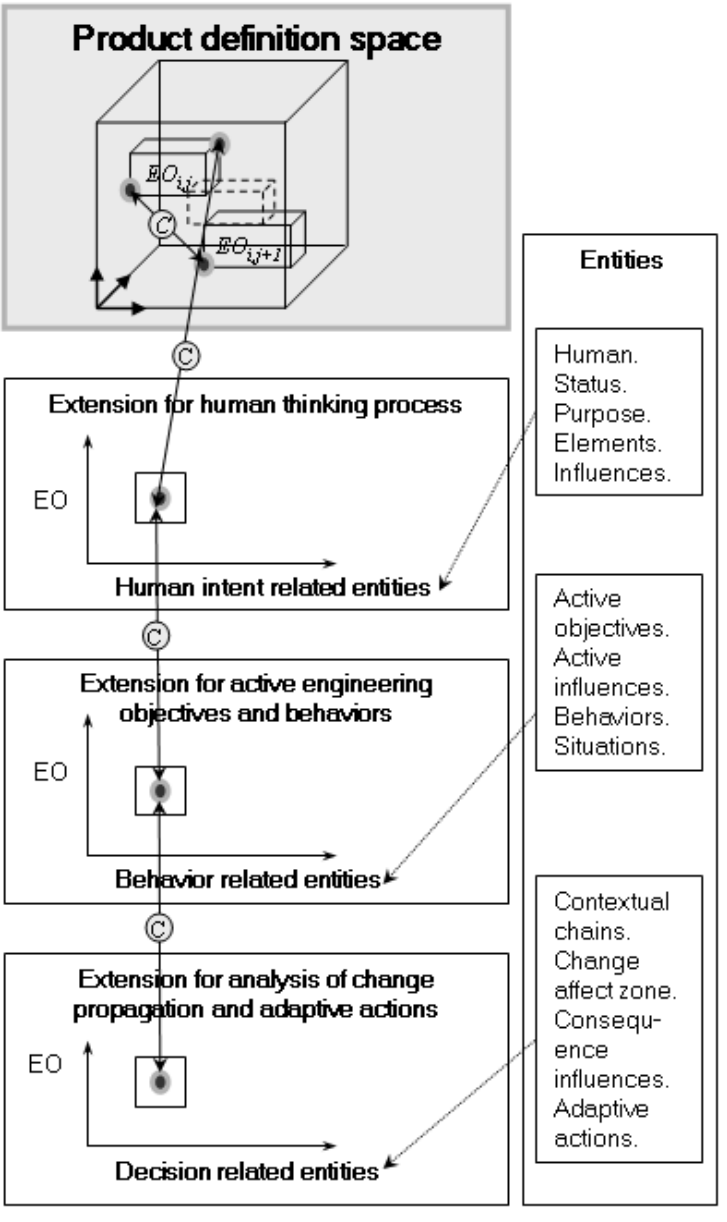


Fig. 10. Extensions to product definition space.

A new classification of solution criteria is based on their difficulties in constraint satisfaction. Other example for the related efforts was published in [13] where change propagation paths and their impact on the delivery of a product

were analyzed. In the modeling proposed by the authors, constraining is controlled by consequence analysis in change propagation.

The extended product definition space is a product definition space which is extended by three rectangular spaces for analysis of change propagation and adaptive actions, for active engineering objectives and behaviors, and for human thinking process (Fig. 10). The new spaces map decision, behavior, and human intent related entities to engineering objects.

Decision related entities are change affect zone, consequence influence, and adaptive action. Adaptive action controls well-defined engineering objects in the product definition space taking the function of direct human control [2]. Behavior related entities are active objective, active influence, behavior, and situation. Human intent related entities are human, status, purpose, element, objective, and influence.

Similarly to product and other engineering objects in product structure, objects in the product definition space and the extension spaces are contextual. Engineering objectives and the related influences are defined in the context of human thinking process. Behaviors and the related situations are defined in the context of engineering objectives. Analysis of change propagation is defined in the context of active behaviors and influences. Engineering objects are defined in the context of adaptive actions. Adaptive action carries verified and decided influence information.

7 Change Propagation and Adaptive Actions

An attempt for influencing product under construction is accepted or rejected according to its anticipated affect on existing engineering objects. Other considered parameter is the weight of the influencing human. As it was emphasized above, influence can be evaluated only in case of background knowledge is defined in the influence attempt. Evaluation of consequences of an influence attempt and decision on it is one of the future developments. The authors of this chapter conceptualized a method and elaborate it gradually. During this development, human dialogue intervention is increased but not eliminated. Decision process must be transparent for authorized engineers who are responsible for the result of relevant engineering activities.

In order to understand the main essence of the proposed method, Fig. 11 outlines change propagation analysis and decision on executable adaptive actions. This sector of product modeling starts with definition of behaviors and situations from engineering objectives for engineering objects. Tracking of change propagation in product model can be done along contextual chains. For this purpose, a contextual graph is proposed that is really a contextual map of the product model. Engineering objects in nodes and contextual connections on arches of this graph make definition of contextual chains for the propagation possible.

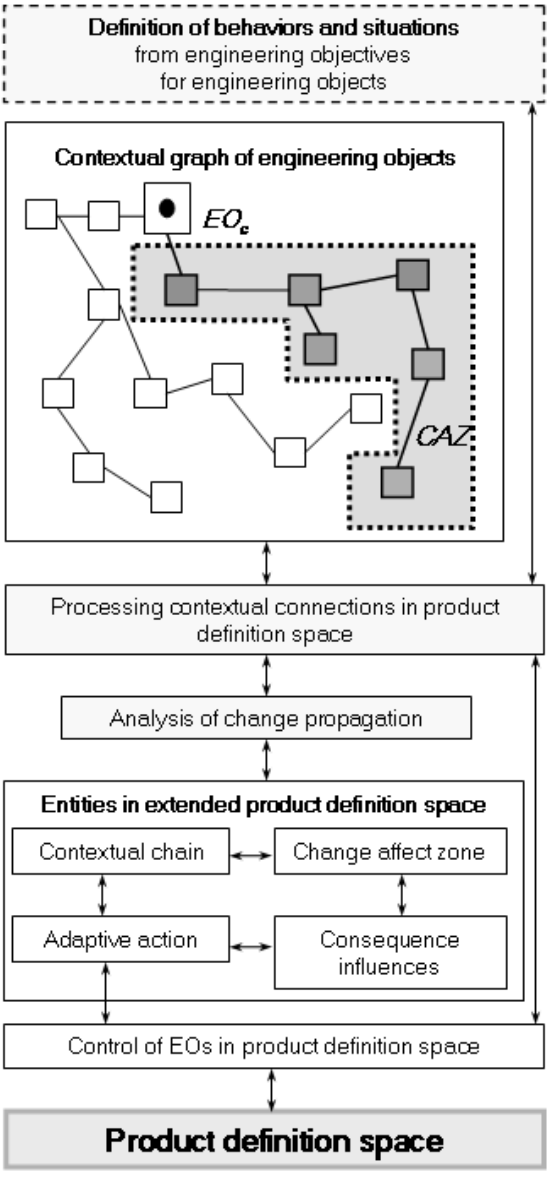


Fig. 11. Analysis of change propagation and adaptive actions.

In Fig. 11, influence attempt is placed on engineering object EO_C. Entity change affect zone (CAZ) was proposed for enclosing potentially affected engineering objects. CAZ in Fig 11 includes two partially overlapped change chains. Contextual graph is elaborated and continuously updated during processing

contextual connections in product definition space. Consequence influences of an attempted influence are revealed in CAZ which is really a view and is developed during analysis of change propagation. Beyond contextual graph, and contextual chain and change affect zone in it, new entity in extended product definition space is the above defined adaptive action.

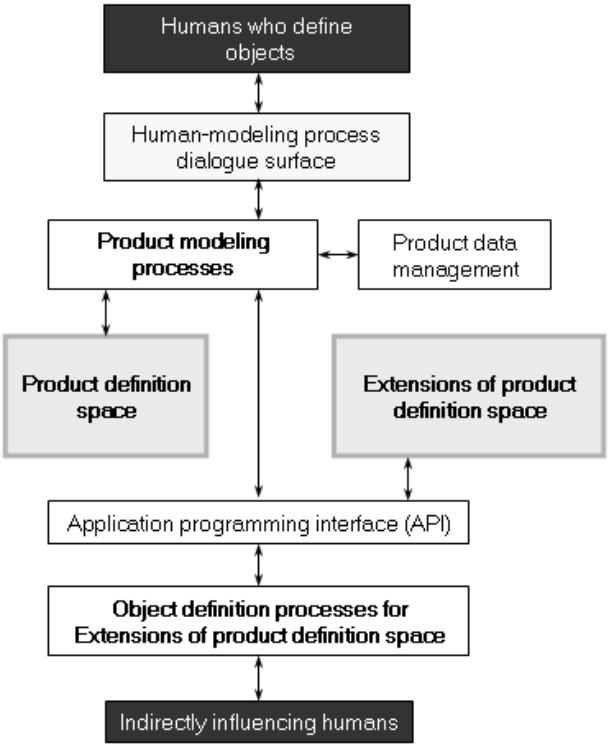


Fig. 12. Implementation in PLM system.

When an adaptive action gains the status executable, it is in possession of all necessary information for control of relevant engineering objects in product definition space. Adaptive action is a feature and can be deactivated and reordered such as any another features in the product model. Current tendency is experienced for unified feature based product model where any engineering object is handled solely as feature. Feature based modeling supports this type of solution.

8 Implementation in PLM System

The proposed modeling and model extension applies mainly locally owned knowledge at industrial companies. This knowledge and procedures of its handling best applied in application environment development of product lifecycle management (PLM) systems. Application programming interface (API) organizes application programming environment for this purpose.

Modeling as it is implemented in API environment is sketched in Fig. 12. Object definition processes for extension of product definition space access human-modeling process dialogue surface. Humans who define objects in product model work on this extended dialogues surface. Product modeling processes are served by product data management and save model entities in the product definition space. At the same time, extensions of product definition space are accessed and build through API. Indirectly influencing humans act through object definition processes for extensions of product definition space.

9 Conclusions and Future Research

This chapter introduced a new method for representation and application of authorized human defined knowledge at global product level decision on engineering objects during lifecycle of product. This knowledge is generated at the definition of engineering objects as method for object entity generation during product model construction. The method is devoted for group work of engineers where a decision may be affected by humans with the same or different role. This work was motivated by the recognition that current product modeling systems are not suitable for human influence coordination. This capability of product modeling is important to develop because an effective decision considers different aspects and a new situation may require application of earlier abandoned knowledge.

The introduced method is one of the results by the authors in order to establish adaptive control of product model. Adaptive control redefines the related engineering objects as result of a new or modified decision. The conventional human dialogue is replaced by adaptive action that modifies product model temporarily for analysis of consequences of an intended influence on product model or finally as part of model construction.

The proposed modeling is expected as a contribution to solve communication problems between engineers. At the same time, this attempt to define engineering objects in the context of engineering objectives prepares function and quality driven integrated modeling of products. Because currently applied product models can be controlled by active rules, checks, and reactions, adaptive actions will be developed to generate these knowledge entities by using of the knowledge represented and applied in the proposed modeling extension.

Acknowledgments. The authors gratefully acknowledge the grant provided by the KTIA-OTKA Fund for Research of the Hungarian Government. Project number is K 68029. The authors also acknowledge the grant for Research Groups from the Óbuda University.

References

[1] Horváth, L., Rudas, I.J.: Modeling and problem solving methods for engineers. Elsevier, Academic Press, New York (2004)

[2] Horváth, L.: A new method for enhanced information content in product model. Trans. Inf. Sci. Appl 5, 277–285 (2008)

[3] Horváth, L., Rudas, I.J.: Human intent representation in knowledge intensive product model. J. Comp. 4, 954–961 (2009)

[4] Renaud, J.: Improvement of the design process through knowledge capitalization: an approach by know-how mapping. Concurrent Eng. 12, 25–37 (2004)

[5] Jardim-Goncalves, R., Figay, N., Steiger-Garcao, A.: Enabling interoperability of STEP application protocols at meta-data and knowledge level. Int. J. Technol. Manag. 36, 402–421 (2006)

[6] Horváth, L.: Supporting lifecycle management of product data by organized descriptions and behavior definitions of engineering objects. J. Aadv. Comp. Intell. Intell. Inf. 9, 1107–1113 (2007)

[7] Andoga, R., Madarász, L., Főző, L.: Situational modeling and control of a small turbojet engine MPM 20. In: Proceedings of IEEE International Conference on Computational Cybernetics, Tallinn, Estonia, pp. 81–85 (2006)

[8] McMahon, C., Lowe, A., Culley, S.: Knowledge management in engineering design: personalization and codification. J. Eng. Des. 15, 307–325 (2004)

[9] Shah, J.J., Anderson, D., Kim, Y.S., Joshi, S.: A discourse on geometric feature recognition from CAD models. J. Comp. Inf. Sci. Eng. 1, 41–51 (2001)

[10] Hosobe, H., Matsuoka, S.: A foundation of solution methods for constraint hierarchies. Constraints 8, 41–59 (2003)

[11] Horváth, L., Rudas, I.J.: Knowledge acquisition at human-computer interaction for product definition. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisora, Romania, pp. 489–494 (2011)

[12] Gielingh, W.: An assessment of the current state of product data technologies. Comp. Aid. Des. 40, 750–759 (2008)

[13] Clarkson, P.J., Simons, C., Eckert, C.: Predicting change propagation in complex design. J. Mech. Des. 126, 788–797 (2004)

[14] Bernard, A., Laroche, F., Da Cunha, C.: Models and methods for knowledge formalisation in a PLM context. In: Proceedings of 3rd International Congress Design and Modelling of Mechanical Systems, Hammamet, Tunisia, pp. 1–9 (2009)

[15] Richardson, M., Domingos, P.: Learning with knowledge from multiple experts. In: Proceedings of 20th International Conference on Machine Learning, Washington, DC, USA, pp. 624–631 (2003)

[16] Bronsvoort, W.F., Bidarra, R., Nyirenda, P.J.: Developments in feature modeling. Comp. Aid. Des. Appl. 5, 655–664 (2006)

[17] Preece, A., Flett, A., Sleeman, D., Curry, D., Meany, N., Perry, P.: Better knowledge management through knowledge engineering. IEEE Intell. Syst. 16, 36–43 (2005)

Autocollimator Calibration Using a Tangent Bar

Gyula Hermann[1] and Kálmán Tomanyiczka[2]

[1] Óbuda University, Bécsi út 96/B, H-1034 Budapest, Hungary
hermann.gyula@nik.uni-obuda.hu
[2] Hungarian Trade Licencing Office, Metrology Department, Nemetvölgyi út 37-39, H-1124 Budapest, Hungary
tomanyiczka@mkeh.hu

Abstract. The chapter describes a new low cost angle generator intended for the calibration of autocollimator. The description of the working principle is followed by the detailed calibration procedure which is based on the comparison principle. The small angles are derived from relatively large displacements. Following the presentation of the operation of the small angle generator, the various error sources are discussed. Finally the chapter discusses the calculation of the extended calibration uncertainty in details.

1 Introduction

In professional metrology precise and contactless measurement of small angles, angle deviations and alignment of machine components are required. Autocollimator is the appropriate high accuracy apparatus, which can measure angles with an accuracy of fractions of an arcsecond.

Brucas and Giniotis give in their paper a short overview of the plane angle measurement principles. They describe a system, based on the comparator principle and intended for the calibration of geodetic measuring instruments. The system consists of an autocollimators and a rotary table. On the rotary table a mirror was placed. Therefore the rotary table acts as a small angle generator with a measuring sensibility of 0,0324" a repeatability of 0,03" and a standard deviation of less than 0,32".

Just applied a high accuracy angle comparator based on the division of a circle. The comparator's main component is a rotor with precision air bearings in which a divided circle with a radial phase grating is installed. The grating consists of 2^{17} grating lines in 360° which corresponds to a graduation interval of ~10 arcsec. Each reading head produce sinusoidal signal with twice the frequency of the graduation frequency. This signal is subdivided by a factor of $4096=2^{12}$ which results in 2^{30} measurement steps, leading to a resolution of ~0,0012 arsec. The drive

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 321–330.
springerlink.com

system of the comparator consists of a variable-speed DC motor. With a gear train, the drive allows the speed be adjusted in the range from 7,5 rev/min to 7,5 arcsec/min. For the calibration of autocollimators, a piezoelectric transducer integrated into the driving unit allows a fine positioning in a range of 6 arcsec with a reproducibility of ~0,002 arcsec.

Kim and his coauthors applied a small angle generator equipped with a laser angle interferometer to calibrate the autocollimator part of their combined miniature Michelson interferometer and autocollimator. They used a motorized micrometer to finely rotate the arm of the small angle generator.

Sohn, Kwon and Choe [14] used a reference angle generator, which employed a Hewlett-Packard laser interferometer to measure the angular displacement of a the rotational bar. The laser beam was divided by a polarizing beam splitter and the two separated beams were reflected by two corner cubes, mounted on ends of the rotational bar. The bar was rotated by a linear motor. The rotating angle was proportional to the change of optical path difference between the two corner cubes, which was determined by the laser interferometer. The angular resolution of this device was more than 0,01" over a range of 40'. It was designed to minimize the systematic error caused by the movement of the axis such as rotation, thermal expansion, pitch, and roll.

De Oliveira and his coauthors describe an autocollimator calibration system using a sine bar and a small displacement generator based small angle generator. The aluminium bar has a calibrated length established by two spheres fixed on it. The deviation of the bar's length from the nominal value and the indication errors of the small displacement generator are corrected to establish the reference small angle. With this system, it is possible to have angular increments of 0,1" and a measuring range of 42'. The authors claim to have a measuring uncertainty of 0,05"*l where l is the displacement by the small displacement generator.

Yandayan present in their paper a high precision small angle generator developed for the calibration of high precision autocollimators. The method utilizes the definition of a plane angle as a ratio of two lengths. The instrument consists of a base unit, a 200 mm long sine arm, a the flexural pivot, a piezo nano-positioner and a 50 mm diameter precise optical mirror with a flatness value of 3 nm rms. The angles are generated when the sine arm, pushed by the nano-positioner, rotates around the flexural pivot. The flexural pivot subtended at the center of rotation provides an excellent repeatable performance with frictionless and stiction-free rotation as the nano-positioner moves in nanometer steps. The piezo nano-positioner pusher equipped with a capacitive nano sensor provides 0,1 nm resolution resulting in an anguler resolution of 0,0001 arc seconds.

2 Autocollimator Working Principle

Autocollimators work in combination with a mirror or another reflecting surface. They check the angular orientation of this mirror. A light source located in the focal point of a lens of the autocollimator results in a parallel light beam at the exit of the lens. The light source enlightens a reticle in the focal plane of the objective and is reflected back into the autocollimator by the mirror.

If the mirror is slightly tilted, by an angle α, then the reflected picture of the recticle will have an angular deviation of 2*α. The image of the light source is transferred over a distance in the focal plane. For a lens with focal length *f* the following equation is valid:

$$\alpha = \frac{1}{2}\arctan\frac{d}{f}, \tag{1}$$

where d is the measured displacement and *f* is the focal length.

In a visual autocollimator the light source enlightens an autocollimator reticle in the focal plane of the objective lens. The light goes through the objective and is reflected back into the autocollimator by the mirror. A beamsplitter separates the outgoing from the reflected light. The reflected image of the autocollimator reticle is projected on the eyepiece reticle in the focal plane of the objective lens and can be seen through the eyepiece (Fig. 1).

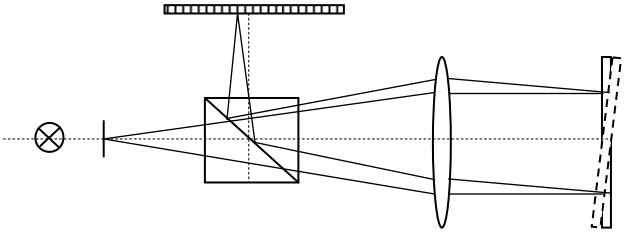


Fig. 1. Sketch of an electronic autocollimator.

In the electronic autocollimator the light source enlightens a slit in the focal plane of the objective lens. The reflected light slit is projected on a linear CCD array, located exactly in the focal plane of the objective. This CCD array measures the distance d. In the bi-directional autocollimators, there are separate measuring systems for both the X and Y directions. The measured distances are translated into an angle deviation. The electronic autocollimator enables very accurate angle measurement.

3 The Calibration Set Up

The calibration process is that the autocollimator is aligned to the mirror on the tangent bar and then the output signals of the autocollimator are compared to the values of the reference angle generator. A small angle generator was developed as the calibration tool for the Hungarian participation in the Euramet Angle Comparison Using Autocollimator project (Fig. 2 and Fig. 3).

The small angle generator consists of steel beam with I cross section supported by two high precision steel edges on adjustable beams. As the edges are placed on a length measuring machine the distance between them was measured using the microscope of the measuring machine and a laser interferometer. The distance between the supporting edges was L=2066,6999 mm. At the left end between the beam and the edge, gauge blocks of different sizes are placed. Hereby the required small angles can be generated. The vertical displacement l_y of the beam at this end is measured by a laser interferometer.

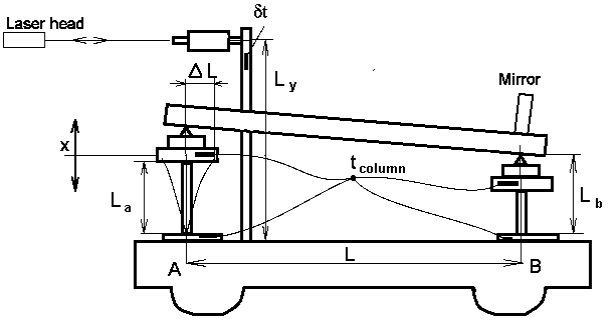


Fig. 2. The small angle generator.

Fig. 3. Supporting edge.

Theoretically this results in an angular displacement of:

$$\theta = \arctan\frac{x}{L}, \tag{2}$$

where (Fig. 4):

x is the vertical displacement of the tangent bar,

L is the distance between the supporting edges.

Fig. 4. Right end of the small angle generator with the autocollimator mirror.

4 Error Sources

The measurement equation for the autocollimator calibration is given by below:

$$\begin{aligned} h_{\varphi} &= \left[\varphi_{meas} - h_{\varphi Ao} - h_{mirror} - h_{optic}\right] - \\ &- \left[\varphi_{E} - h_{\varphi y} - h_{\varphi A} - h_{\varphi B} - h_{Ehist} - h_{\varphi f} - h_{\varphi ft} - h_{\varphi Eo}\right], \end{aligned} \tag{3}$$

where :

φ_{meas} is the measured angular orientation,

$h_{\varphi Ao}$ stands for the resolution of the autocollimator,

h_{mirror} represents the error of the mirror,

h_{optic} error of the optic,

φ_{E} angle represented by the small angle generator,

$h_{\varphi y}$ position error of the interferometer retroreflector,

$h_{\varphi A}$ position error of the A support due to thermal expansion,

$h_{\varphi B}$ error caused by the difference of the thermal expansion of the A and B supports,

h_{Ehist} hysteresis error of the small angle generator,

$h_{\varphi f}$ error due to parasitic rotation of the mirror,

$h_{\varphi ft}$ error resulting from the difference between the real and ideal axis of rotation,

$h_{\varphi Eo}$ the error of the laser interferometer.

The expressions for the main error sources are given below:

$$\varphi_E = \arctan \frac{x}{L(1+\alpha \Delta t_{mean})}, \tag{4}$$

$$h_{\varphi y} = \arctan \frac{l_y \alpha \delta t}{L}, \tag{5}$$

$$h_{\varphi A} = \arctan \frac{L_A \alpha \delta t}{L}, \tag{6}$$

$$h_{Ehist} = \arctan \frac{x}{L-\Delta L} - \arctan \frac{x}{L+\Delta L}, \tag{7}$$

$$h_{\varphi B} = \frac{L_B \alpha \Delta t_t}{L}, \tag{8}$$

$$h_{\varphi Eo} = \arctan \frac{o_x}{L}. \tag{9}$$

The sensitivity functions are listed below:

$$\frac{\partial h}{\partial x} = -\frac{1}{L\left(1+\alpha \Delta t_{mean}\right)^2 \left(1+\frac{x^2}{L^2\left(1+\alpha \Delta t_{mean}\right)^2}\right)} + \\ + \frac{1}{\left(L-\Delta L\right)\left(1+\frac{x^2}{\left(L-\Delta L\right)^2}\right)} - \frac{1}{\left(L+\Delta L\right)\left(1+\frac{x^2}{\left(L+\Delta L\right)^2}\right)}, \tag{10}$$

$$\frac{\partial h}{\partial \alpha}=\frac{x\Delta t_{mean}}{L\left(1+\alpha\Delta t_{mean}\right)^2\left(1+\dfrac{x^2}{L^2\left(1+\alpha\Delta t_{mean}\right)^2}\right)}+\frac{L_y\delta t}{L\left(1+\dfrac{L_y^2\alpha^2\Delta t^2}{L^2}\right)}+\frac{L_A\delta t}{L\left(1+\dfrac{L_A^2\alpha^2\Delta t^2}{L^2}\right)}+\frac{L_B\Delta t_{col}}{L\left(1+\dfrac{L_y^2\alpha^2\Delta t_{col}{}^2}{L^2}\right)}, \tag{11}$$

$$\frac{\partial h}{\partial L}=\frac{x}{L^2\left(1+\alpha\Delta t_{mean}\right)\left(1+\dfrac{x^2}{L^2\left(1+\alpha\Delta t_{mean}\right)^2}\right)}+\frac{L_y\alpha\delta t}{L^2\left(1+\dfrac{L_y^2\alpha^2\Delta t^2}{L^2}\right)}-\frac{L_A\alpha\delta t}{L^2\left(1+\dfrac{L_A^2\alpha^2\Delta t^2}{L^2}\right)}-\frac{L_B\alpha\delta t}{L^2\left(1+\dfrac{L_B^2\alpha^2\Delta t^2}{L^2}\right)}+\frac{1}{\left(L-\Delta L\right)\left(1+\dfrac{x^2}{\left(L-\Delta L\right)^2}\right)}-\frac{x}{\left(L+\Delta L\right)\left(1+\dfrac{x^2}{\left(L+\Delta L\right)^2}\right)}-\frac{o_x}{L^2\left(1+\dfrac{o_x^2}{l^2}\right)}, \tag{12}$$

$$\frac{\partial h}{\partial t}=\frac{\alpha x}{L\left(1+\alpha\Delta t\right)^2\left(1+\dfrac{x^2}{L^2\left(1+\alpha\Delta t\right)^2}\right)}, \tag{13}$$

$$\frac{\partial h}{\partial l_B}=\frac{\alpha\Delta t}{L\left(1+\dfrac{L_B^2\alpha^2\Delta t^2}{L^2}\right)},\ \frac{\partial h}{\partial l_y}=\frac{\alpha\delta t}{L\left(1+\dfrac{l_y^2\alpha^2\delta t^2}{L^2}\right)}, \tag{14}$$

$$\frac{\partial h}{\partial \Delta t}=\frac{L_B\alpha}{L\left(1+\dfrac{L_B^2\alpha^2\Delta t^2}{L^2}\right)}, \tag{15}$$

$$\frac{\partial h}{\partial \delta t} = \frac{l_y \alpha}{L\left(1+\frac{l_y^2 \alpha^2 \delta t^2}{L^2}\right)} + \frac{L_A \alpha}{\left(1+\frac{L_B^2 \alpha^2 \delta t^2}{L^2}\right)}, \tag{16}$$

$$\frac{\partial h}{\partial \Delta l} = \frac{x}{(L+\Delta L)^2\left(1+\frac{x^2}{(L+\Delta L)^2}\right)} - \frac{x}{(L-\Delta L)^2\left(1+\frac{x^2}{(L-\Delta L)^2}\right)}, \tag{17}$$

$$\frac{\partial h}{\partial l_A} = \frac{\alpha \delta t}{L\left(1+\frac{L_A^2 \alpha^2 \delta t^3}{L^2}\right)}, \quad \frac{\partial h}{\partial o_x} = \frac{1}{L\left(1+\frac{o x^2}{L^2}\right)}. \tag{18}$$

5 Calibration Uncertainty and Conclusions

The determination of the calibration uncertainty is based on the recommendations of the ISO "Guide to the Expression of Uncertainty of Measurement". The results are presented in Table 1. Table 1 shows extended results compared to [19].

Table 1. Calibration uncertainty results

Description of input quantity (x_i)	Symbol for x_i	Distribution	Std. meas. uncertainty $u(x_i)$ of input quantity	Dim.	Sensitivity coefficient $c_i = \partial\delta/\partial x_i$	Standard meas. uncertainty contribution $u_i(\delta$
Angle measured by the autocollimator	φ_{meas}	Normal	1,000E-01	arcsec	1	1,00E-01
Error of display of autocollimator values	$h_{\varphi Ao}$	Rectangular	2,887E-01	arcsec	1	2,89E-01
Flaws of mirror (flatness deviations, etc.)	h_{mirror}	Rectangular	5,774E-02	arcsec	1	5,77E-02
Flaws of the autocollimator's optics	h_{optic}	Rectangular	5,774E-03	arcsec	1	5,77E-03

Table 1 *(continued)*

Vertical elevation of the angle-generator	x	Rectangular	5,774E-06	mm	7,56E-10	4,37E-15
Base length of the angle-generator	L	Normal	3,599E-03	mm	5,34E-11	1,92E-13
Linear thermal expansion coefficient	α	Rectangular	5,774E-07	1/°C	9,16E-03	5,29E-09
Average body temperature of the base	Δt_{mean}	Normal	6,255E-03	°C	5,58E-10	3,49E-12
Vertical length of supporting column for the interference mirror of the laser	L_y	Rectangular	5,774E-01	mm	1,05E-10	6,04E-11
Body temperature of the supporting column for the interference mirror	δt	Normal	4,519E-03	°C	2,07E-06	9,37E-09
Initial distance of the lifter from the base	L_a	Rectangular	5,774E-06	mm	1,05E-10	6,04E-16
Difference of temperatures measured at both lifters and ends of the base	Δt_{column}	Normal	6,842E-04	°C	7,81E-07	5,34E-10
Horizontal deviations of the lifter	ΔL	Rectangular	2,887E-02	mm	2,35E-08	6,79E-10
Resolution of the laser interferometer	o_x	Rectangular	5,774E-06	mm	4,85E-04	2,80E-09
Errors of rotation of the mirror	$h_{\square f}$	Rectangular	2,887E-02	arcsec	1	2,89E-02
Errors of the axis of rotation of the mirror	$h_{\square ft}$	Rectangular	1,155E-02	arcsec	1	1,15E-02
			Standard uncertainty	arcsec		0,31
			Expanded uncertainty	arcsec		0,63

Acknowledgments. The authors thank Mr. Gy. Benkó for his valuable contribution in implementing the system.

References

[1] Adams, T.M.: G103-A2LA guide for estimation of uncertainty of dimensional calibration and testing results, The American Association for Laboratory Accreditation (2008)
[2] Brucas, D., Giniotis, V.: Calibration of precision polygon/autocollimator measurement system. In: J. Physics Conf. Ser., vol. 2388, pp. 1–7 (2010)
[3] Brucas, D., Giniotis, V.: The construction and accuracy analysis of the multireference equipment for calibration of angle measuring instruments. In: Proceedings of XIX IMEKO World Congress, Fundamental and Applied Metrology, Lisbon, Portugal, pp. 1823–1837 (2009)
[4] Eom, T., Chung, D., Kim, J.: A small angle generator based on a laser angle interferometer. Int. J. Prec. Eng. Manuf. 8, 20–23 (2007)
[5] EURAMET.L-K3a Angle Comparison Using Autocollimator
[6] ISO, Guide to the expression of uncertainty in measurement, Geneva (1993)
[7] Just, A., Krause, M., Probstan, R., Wittekopf, R.: Calibration of high-resolution autocollimators against an angle comparator. Metrologia 40, 288–294 (2003)
[8] Kim, J.W., Kang, C.S., Kim, J.A., Eom, T., Cho, M., Kong, H.J.: A compact system for simultaneous measurement of linear and angular displacement of nano-stages. Optics Express 15 (2007)
[9] Krause, M., Just, A., Geckeler, R.M., Bosse, H.: Angle metrology at the PTB: current status and development. In: Proceedings of 2011 International Dimensional Workshop, Knoxville TN, USA (2011)
[10] Kruger, O.A.: Performance evaluation of phase shifting interferometer compared to an autocollimator in the measurement of angle. In: Proceedings of 2011 International Dimensional Workshop, Knoxville TN, USA (2011)
[11] Kruger, O.A.: Methods for determining the effect of flatness deviations, eccentricity, and pyramidal error on angle measurements. Metrologia 37, 101–105 (2000)
[12] Malacara, D.: Optical shop testing, 2nd edn. Wiley (1992)
[13] Shen, M.Z., Liao, S.T.: A high precision dynamic autocollimator. In: Proceedings of 2011 International Dimensional Workshop, Knoxville TN, USA (2011)
[14] Sohn, Y.J., Kwon, J.H., Choe, O.S.: Portable autocollimator using the laser diode and the position sensitive detector. Rev. Sci. Instr. 69, 402–405 (1998)
[15] Stone, J.A., Amer, M., Faust, B., Zimmerman, J.: Uncertainties in small-angle measurement systems used to calibrate angle artifacts. J. Res. Nat. Inst. Stand. Tech. 109, 319–333 (2004)
[16] Valente de Oliveira, J.C., Brum Vieira, L.H.: Traceability chain and dissemination of angle measurement in Brazil. Prec. Eng. 22, 141–152 (1998)
[17] Vloet, R.E.J.M.: Electronic autocollimation, WPA Report 31002, Eindhoven University of Technology, Eindhoven, The Netherlands (1995)
[18] Yandayan, T., Ozgur, B., Karaboce, N., Yaman, O.: High precision small angle generator for realization of SI unit of plane angle and calibration of high precision autocollimators. In: Proceedings of Macroscale (2011)
[19] Hermann, G.: Calibration of an autocollimator (method and uncertainty). In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisoara, Romania, pp. 645–647 (2011)

Gesture Control: A New and Intelligent Man-Machine Interface

Dan Ionescu, Bogdan Ionescu, Cristian Gadea, and Shahid Islam

NCCT Laboratory, University of Ottawa, 161 Louis Pasteur Room B-306,
Ottawa ON K1N-6N5, Canada
ionescu@site.uottawa.ca, {bogdan,cgadea,sislam}@ncct.uottawa.ca

Abstract. Although investigated from early days of research in the domain of human-computer interfaces, gesture-based control of computer application entered in the everyday life of computer users with the advent of 3D infrared cameras. The usage of real-time depth-mapping cameras and of robust image processing applied the flow of depth map streams triggered the production of a plethora of applications ranging from controlling electronic games and electronic devices such as TV sets or set-top boxes, to e-learning, and sterile environments such as operating rooms. Gesture and motion-based control of computer applications received increased attention from both academic and industrial research groups for the unique interactive experiences it offers. Of particular research interest has been the control of games through gesture-based interface. In this chapter after a brief survey of the methods and technologies used in the gesture control area, a new and intelligent user interface based on a sequence of gestures linked in a gesture language through a sign grammar will be introduced and described. The gesture recognition language is based on image processing functions which are grouped in a combination of algorithmic and learning approaches. Applications of the gesture language in gaming, TV and set-top box control, e-learning and virtual reality-reality interaction illustrate the validity of the approach.

1 Introduction

Since the beginning of the penetration of computers in the everyday life of the mankind, gestures have long been considered as an alternative to the classical computer interaction pair keyboard-mouse. Although the first attempt [1] was far from what it is expected today from such an interaction, it demonstrated that there creating computer commands using other methods then the keyboard, or keyboard and mouse is possible. The gesture sensor used in [1] was a light pen with which graphical onscreen objects were manipulated through pen moves and strokes. A

R.-E. Precup et al. (Eds.): Applied Computational Intelligence in Engineering, TIEI 1, pp. 331–354.
springerlink.com

series of researchers followed up on this open road investigating a series of aspects of this new and promising approach to enabling a natural and intuitive human-computer interaction in order to naturally control with movements of the hand, fingers and body parts a myriad computer applications in various domains in which computers were embedded or applied [2], [3], [4], [5].

Gesture control has since received widespread acceptance in the human-computer interaction (HCI) community and gained popularity not only among researchers seeking to implement novel interactions with computers but rather with the large segment of electronic gamers.

As sensors for gesture control were embodied by many types of transducers, from light pens, to gloves, and to sophisticated infrared (IR) devices which culminated with cameras capable of producing a three dimensional map of the user present in its Field of View (FOV), so were the variety of fields in which the gesture control has been applied. Naturally, industrial implementations appeared and a series of patents have been applied boldly, even if the principle has been publicly disclosed.

Gloves augmented with electronic motion and position sensors [6], [7], [10], or markers sending their spatial position (polhemus) [8], [9] were developed and thus research fields called haptic perception, haptic control, haptic technology or haptics, etc. have mushroomed. The aim of the research in these domains was to enhance the human-computer interaction with computer applications in general but merely with those on virtual reality where users were capable to manipulate digital objects using hand motions [6], [7].

As computer technology brought more computational power to researchers, computer vision technology started to be used in the ’90s and the research and technical literature recorded a series of attempts demonstrating that a vision-based system can be used to control various applications or electronic devices such as the volume and channel tuning of a television set.

A series of review of the domain have been published recording progresses made along the years [11], [12], [13].

The efforts made in this domain of hand gesture recognition and control were motivated by the richness of the information contained in an image without intruding in the environment like using gloves, markers, etc. and at a low cost, which makes it very attractive for the consumer segment of the market.

As vision based gesture recognition involves assembling together image processing functions such as filters, segmentation, feature extraction, and others for separating objects in the image which can be identified as being the body parts of interest which define the gesture of interest. These objects are further processed in order to track them in their movement in the reality space and then recognizing a sequence of such gesture to determine which meaning the gesture has [14].

From a high level point of view the image processing functions can be grouped in functions which implement the detection of the body parts, functions which track the body parts and functions which perform the recognition of the gesture

from a sequence of positions of the body parts in their move in the space delimited by the FOV [15], [16].

To make the gesture detection even more difficult to implement, while applying gesture control to electronic gaming environment, or to augmented virtual reality applications, the results of the gesture recognition process have to be provided at the rhythm of image acquisition, in other words in real-time.

It is well-known fact that image processing functions are all error prone, and as such the detection, tracking, and recognition of gestures is not precise and gesture misses might occur.

One of the main causes for the presence of errors in the results produced by image processing functions is the variations in the lighting which affects quite heavily the segmentation and therefore the feature extraction.

To remedy the situations cameras operating beyond the visible spectrum mainly in the infrared domain have been produced and used [17]. It is worth mentioning that even structured light methods obtained by projecting a pattern on the object present in the FOV have been devised also in IR.

In what follows, this chapter will investigate gesture based control where the sensor is a special IR camera capable of producing a depth map of the object present in the field of view.

In Section 2, a background of gesture definition and gesture control is provided. Section 3 discusses the use of gestures in controlling virtual environments through the use of a novel 3D camera. Section 4 introduces an architecture for the control of TVs and set-top boxes using a 3D camera and the corresponding software. Section 5 presents a multimodal architecture that combines gestures with physical controllers, allowing for six-degrees-of-freedom. The chapter ends with conclusions about gesture control and its applications.

2 Background

With the advent of sensors and systems for electronic gaming, vision based gesture control has seen a strong resurgence. From and industrial and commercially sustained business model the application of gesture control has expanded in electronic gaming. Besides this market area there is the domain of electronic devices such as TV sets, set-top boxes, laptops, mobile phones and robotics.

There is a potential of applying it in medicine everywhere the sterile environment prevents touching the keyboard or the mouse of the computer, in the military, although there is not yet a proof of such possible applications, in distance learning, and teaching, automobile industry where some controls of the navigation devices, or mobile phones can be built, monitoring the driver and many others.

The limitations of gesture control applications stems in the fact that gestures are ambiguous and incompletely specified as there is always a many-to-one mapping from what it expresses and the form it embraces.

Another difficulty to make gestures unique is due to the fact that similar to the speech and handwriting every individual has its own way of making them, and it makes to task more difficult it can vary from gesture to gesture in different moments for the same individual.

To comply with the above difficulties of gesture detection, tracking, and recognition various approaches such as mathematical models [18], [19] statistical pattern recognition, learning systems [20], [21], soft computing [23], algorithmic, and/or a combination of the above have been proposed and experimented with.

As of now, the problem of detecting, tracking, and recognizing gestures either based on hand and finger, and or body part movements in real-time, i.e., at a speed of 30 frames per second (fps) is still lacking a robust and complete solution.

As such, in addition to the theoretical aspects mentioned above, practical implementations typically use complementary information obtained by different imaging and/or tracking devices in order to make the gesture recognition process more stable.

Thus devices that sense body (e.g., hand, head) position and orientation, facial features or expressions, and other aspects of human behavior obtained through experiments and recorded in databases are used to make the human computer interface more robust.

There is a large variety of gestures which encompass gestures in which the user assumes a certain pose or configuration known as static gestures, gestures which evolve known as dynamic gestures which can be decomposed in three phases a prestroke, stroke, and poststroke phase, and combined gestures which contain both static and dynamic components. The literature classifies them in distinct classes [12]:

1. hand and arm gestures entailing recognizing hand poses, finger position and or combination in a specified sequence, or of both hands,
2. head and face gestures: some examples are: a) nodding or shaking of head; b) direction of eye gaze; c) raising the eyebrows; d) opening the mouth to speak; e) winking, f) flaring the nostrils; and g) looks of surprise, happiness, disgust, fear, anger, sadness, contempt, etc.,
3. body gestures: involvement of full body motion, as in driving, sports, or manipulation gestures with applications to tracking movements of two people interacting outdoors or analyzing movements of a dancer for matching them to music and graphics or recognizing human gaits for medical rehabilitation and athletic training.

Following the above classification the dominant application of gesture control uses body gestures, arms and legs to interact with the gaming environment. Other applications of gestures can be considered sporadic and with sporadic lifes.

In what follows, the chapter concentrates on the gesture recognition and control system comprising both the hardware and the software, built by the authors.

3 Gesture Recognition and Control for Virtual Environments

3.1 Acquiring 3D Data

The new 3D camera used in the work reported in this chapter is based on producing a NIR cone of light that is projected onto the object in order to generate grey level images in which the grey level is proportional to the depth of the object points with respect to the origin of the Euclidean axis placed in the center of the camera. The precision is therefore automatically limited by the distance used and by the fact that there are only 256 levels of grey used to encode the depth. The precision can be augmented if subpixel accuracy methods are used. The improvement in precision can be increased from 10 to 20 times by proper application of the subpixel calculation techniques or pseudocoloring the depth image. The total distance to which the 3D camera can detect the reflected NIR light is also a factor in the total precision obtained. For example, if the range of the 3D camera is 4 meters, the maximum precision is limited to increments of 1.5625 cm. As mentioned above, by using subpixel accuracy methods, the precision can easily be increased to up to 0.125625 cm, which is more than sufficient for applications such as telerobotics and even telemedicine in most cases. However, experimental results show that for computer games, the grey precision is more than enough, as detecting moves of the hands, fingers, head, or the body of the user does not require precision lower than, or even near, 1.5 cm.

The depth measurement principle is based on using pulsed IR light with a high frequency (tens of MHz) which is tuned based on the desired precision and sensor sensitivity. The camera used for the work reported in this chapter used LED generated NIR light in the 950 nm range, while the total power consumption based on the distances to be measured was determined to be 200 mW. Under these conditions, the camera is capable of providing data in regards to the depth of objects situated up to 5 m. The light source is generated as a square pulse of short and variable duration (from 30 ns up). The infrared light produced is an expanding spherical surface of a finite width determined by the square pulse of the diode. In literature, this has been called a “light wall”. The wall is also controlled in intensity and a special intensity function is generated based on the controller program implemented in the controller of the illuminator and the image acquisition shutter. The infrared light is reflected back to the camera by the real 3D scene, and provides the information needed to decode when measuring the depth. The principle of this new device is shown in Fig. 1.

The 3D camera is an embedded system made up of an electronic board for the image sensor, a controller board, an illuminator board, and an image processing board. The controller and the image processing are synchronized to produce a depth map from the image sensor output. The camera produces 3D images at the speed of 30 frames per second, with almost no delay in regards to the moves captured.

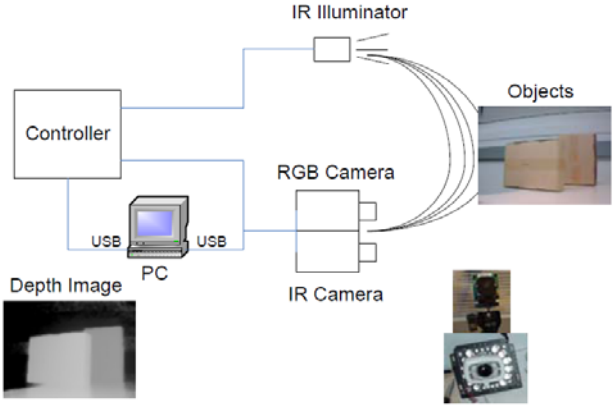


Fig. 1. 3D camera diagram.

The reflected light wall is adjusted by the controller which, synchronously with the laser pulses, opens the shutter of the camera for only the time the wave needs to imprint an image for the duration determined by the distance and the speed of the NIR wave. This results in an integrated electric charge loaded in the sensor cells. By repeating this process a variable number of times and by properly controlling the camera parameters in real-time, an image of the object is produced. By controlling the image parameters in the opposite direction, a second image is obtained after the proper number of iterations. The number of iterations depends on the sensitivity of the image sensor and is selected based on the application domain and its specific requirements.

The captured gray level image has to be normalized to account for effects such as changes in the scene's reflection coefficient, non uniformity of the illumination or the detector, and for the slope of the wall edge. Normalization is carried out by capturing a second un-gated image that contains all these effects without the effect of the gating.

One of the most important features of the depth camera is the ability to change the parameters of the depth measurements on an as-needed basis. It can therefore read gestures ranging from small fingers to whole body movements as needed by the application.

Due to this dynamic feature, it can also include certain objects, while disregarding others. In this way, the image sensor features are used dynamically for the measurement of the space between the camera and depth limits programmed by the user on the host computer interface and transferred by a USB connection to the configuration interface of the camera hardware.

Fig. 2 shows an example of the 3D camera being used. The depth of the hand is translated to a grey level which is lighter on the palm and fingers and darker on the body of the user.

Fig. 2. Getting depth image information with the 3D camera.

3.2 Depth Map and Applications

As explained above, the camera builds a depth map transforming the image of the user in a grey or colored scheme of the depth of the (x,y) points pairs. This is shown in Fig. 3 where a parallelepiped – a box – is held at an angle for the clarity of the explanations. Fig. 3a and Fig. 3b show the image obtained by a camera CMOS sensor at two different settings. Fig. 3c shows the final depth image. The corners of the box that are further from the camera have a different grayscale value than those closer to the camera. Fig. 3d is a typical RGB image of the scenario. Fig. 3e and Fig. 3f show two other frames in the processing of the raw IR images used to obtain 3D data from the environment with less precision if the application so requires.

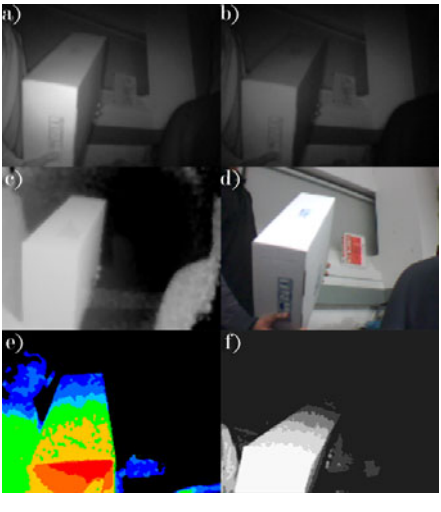


Fig. 3. Depth results.

The depth map is then used to build proper actions by detecting gestures or body movements based on the application requirements.

3.3 Applications of 3D Camera to Virtual Environments

One assists recently to a growing research and experimental works in the field of new or next generation human computer interfaces. The sounding success obtained by the 3D approaches of the movie industries motivated the consumer industry to move in full force towards new 3D control of computer applications and of electronic devices for the consumers' market.

Stagnant since the introduction of the mouse and the keyboard, the Man-Machine Interaction and thus the Human-Computer Interfaces (HCI) have seen more recently an evolution towards the use of touch-screen, and eventually gestures, to control various applications. Hand gestures were always used by humans to illustrate and give more weight and meaning to what they want to express in a more powerful way. However, the expressiveness of hand gestures has not been fully explored for HCI applications. This was due to the fact that the translation of gestures in computer commands is difficult due to the ambiguity of the gestures. At some moment in time a gesture might mean a command which can also be a transition state in another gesture. This can automatically trigger false commands to the computer as the meaning of gestures is contextual as the natural language itself.

3.3.1 Gesture Recognition

The human hand is a complex articulated object consisting of many connected parts and joints which, in motion, can be characterized by approximately 27 degrees of freedom [23]. This makes the recognition problem complex enough to make place for errors. If one thinks that image processing is by definition error prone, the difficulty of controlling computers or computer applications by gestures is immediate.

The problem is alleviated if three dimensions of the hand gesture are acquired and measured correctly. The information contained in a 3D image might have enough power of separation such that the computer can clearly distinguish the context in which the gesture means a command.

The new 3D camera brings the advantage of a considerable high resolution when it comes to the data it has to provide. This helps to make the result of gesture understanding stable enough. It is worth noting that the NIR light has also its own disadvantages of being used, and the images obtained are quite unstable, presenting a flickering effect due to scattered reflections.

The principle of detecting hand gestures is given in Fig. 4. The hand and fingers are captured by the 3D camera and a 3D image is formed. Special algorithms which detect the convex hull of the hand and the irregularities present when fingers are apart are used to detect one, two, and more fingers as they are used.

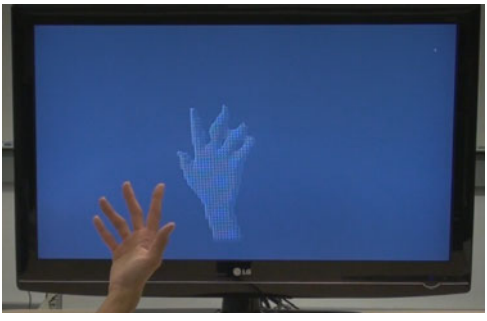


Fig. 4. Understanding gestures.

3.3.2 Gesture Recognition

Gesture Recognition Interface (GRI) as a mean for computer and computer application control has received a great deal of attention from the gaming industry more recently. The gesture and body parts movement recognition are natural to the game and gamer, as it is immersive and with appropriate tools in hand can create a superior experience. There is already a competition among console gaming manufacturers. Each of them promises to come with new GRIs.

They are two distinct applications of the 3D camera in a virtual reality environment and thus two types of new Human Computer Interactions paradigms. The first application relates to the use of hand and fingers to trigger commands such as shooting, driving, moving objects on the screen, etc.

The second group of interactions with a Virtual Reality Environment is related to the full body immersion into the Virtual Reality Space.

In Fig. 5 a set of complex hand and finger gestures are shown as the basis for interacting with a game-based virtual reality environment.

Fig. 5. Complex gestures as recognized by the 3D camera.

Since the introduction of 2D GRIs, the devices gained popularity and are sold in large quantities since gamers appreciate the convenience and thrill associated with GRI.

The current generation of 2D GRI has a few serious drawbacks which each and every vendor promises to overcome soon. The difficulty of providing a good GRI stems from the fact that 2D images cannot provide depth perception. If two cameras are used for depth generation, a series of camera calibration processes hinder the acceptance of the market of such methods. Moreover, image processing functions needed to get the gestures require robust segmentation and stable lighting conditions besides the heavy calibration. In addition, the 2D GRI is limited in the amount of information that can be delivered to the computer. For example, pushing forwards or backwards has no meaning in a 2D image, yet it is rather important to the gamer. Changes in background may affect the 2D GRI and result in a poor gaming experience.

As image processing is computer intensive, interpreting gestures might introduce unacceptable delays which spoil the game experience.

In this section, we introduce the idea of incorporating the new 3D camera in the game, thereby making the gamer part of the game world (Fig. 6).

Fig. 6. GRI for playing games.

3.3.3 3D Immersion of the User in the Virtual Reality Environment

The 3D camera can be used in any computer application whose control by gestures makes sense.

In what follows, the use of the 3D camera to create a total 3D immersion of the user in the Virtual Reality space is introduced.

The technique relies on fusing the real data obtained by the camera with the physical constraints of the Virtual Reality space.

The 3D camera produces the 3D data as shown in Fig. 7. The image processing algorithm used will detect the main points of the users' body by using anthropomorphic measurements [24] of various body parts such that finally various points of the human body parts can be connected to the virtual reality points which correspond to the same points of the user.

Fig. 7. 3D image of the user.

In order to obtain the 3D body points of the user the raw 3D data obtained from the 3D camera shown in Fig. 7 is transformed in a series of connected points and lines as shown in Fig. 8.

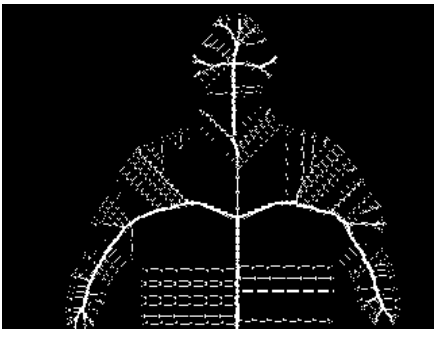


Fig. 8. Points and lines representing the 3D image of the user as shown in Fig. 7.

Eventually, a network of connected points is generated as shown in Fig. 9. The anthropomorphic graph obtained is then mapped on the Virtual Reality space where the 3D image of the user is also reconstructed. Fig. 10 shows a user manipulating a virtual ball in a 3D Virtual Reality Space with his real hands. Additional examples are provided in [25].

4 Gesture Recognition and Control for Consumer Electronics

Fig. 11 shows the main architecture required for the 3D camera to be used to control TVs and set-top boxes [26]. The *Controller* synchronizes the *Illuminator* and *3D Camera* to capture 3D depth data which the *PC* can access. The PC then processes this depth data and checks the XML-based configuration file for matching gesture conditions. If the required gesture is triggered, the PC uses the two attached infrared blasters to emit the appropriate command to the nearby TV or set-top box.

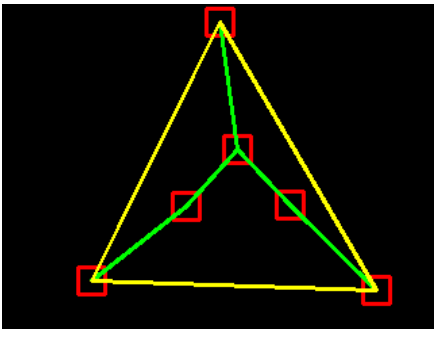


Fig. 9. An anthropomorphic graph of the user is generated.

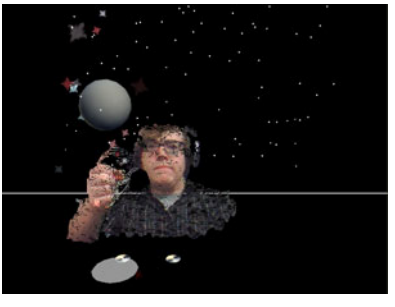


Fig. 10. An example of augmented virtual reality where the user plays in a virtual world with a virtual ball him being in a real one.

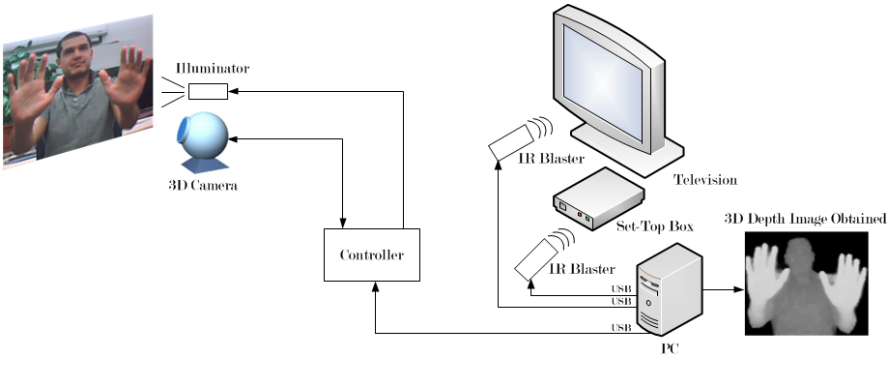


Fig. 11. Architecture for controlling a TV and set-top box with gestures.

To learn and emit the necessary remote control signals from a PC, two Windows Media Center USB infrared blasters and a modified driver are used. They provide access to the sending and receiving capabilities of the blasters through an API provided by the manufacturer. By pointing the existing TV and receiver

remotes to the receiving end of the infrared blasters, the infrared signals emitted by the physical remotes were recorded for all of the necessary commands. These infrared signals could then be retransmitted by accessing the infrared blasters from the gesture-processing code and playing back the appropriate recording based on the gesture that the user performed. Since any remote's infrared signal can be recorded and replayed in this way, the system allows for the control of practically any television set and set-top box on the market today, as well as devices such as AV receivers and Blu-ray players.

Since such electronic devices have numerous features accessible via their remote controls, a simple method is needed to map a large number of possible hand gestures to device functions. An XML file is therefore used to describe these gestures and allow users to form a gesture language. The basic format of the XML-based configuration file that allows defining which gesture should trigger which infrared signals is shown in Fig. 12.

```
<appConfig>
  <gesture>
    <righthand>
      <finger>
        <min_angle>A</min_angle>
        <max_angle>B</max_angle>
      </finger>
      <finger>
          .
          .
      </finger>
    </righthand>
    <lefthand>
        .
        .
    </lefthand>
    <remote_ir_code>C</remote_ir_code>
  </gesture>

  <gesture>
      .
      .
  </gesture>

</appConfig>
```

Fig. 12. Gesture mapping XML.

Note how the syntax from Fig. 12 allows two-handed gestures to be defined. The required number of fingers, as well as the angles of those fingers, are specified for each hand. The values for A and B are defined in degrees based on A being between -180 and 180 degrees. In addition B is greater than or equal to A, and both should be left as zero (or omitted entirely) when a specific angle is not required. If the <lefthand> property is left empty, a fist (no fingers) is expected for

the left hand, while if the property is omitted entirely, a left hand must not be present for the gesture to be triggered. Finally, C is the infrared code recorded from the physical remote control.

For example, Fig. 13 shows the XML code to detect one finger pointing to the right from the right hand. A 45 degree range is allotted to make it easier for the user to trigger. The actual infrared code is omitted due to length.

```
<gesture>
  <righthand>
    <finger>
      <min_angle>0</min_angle>
      <max_angle>45</max_angle>
    </finger>
  </righthand>
  <remote_ir_code>...</remote_ir_code>
</gesture>
```

Fig. 13. Detecting a finger pointing right.

Fig. 5 of Section 3.3.2 contains some of the possible gestures that have been defined using this language. This flexibility in defining gestures is now made possible due to the gesture detection capabilities enabled by the 3D depth camera. The gestures defined in the XML-based configuration file can be performed at any time to cause the mapped infrared commands to be emitted. TVs and set-top boxes are therefore given an intelligent interface for reliable and convenient control.

The objective in this case was to control a LG television and a Bell Expressvu Satellite set-top box through an intelligent gesture interface. The TV was to receive commands such as turn on and off, volume up and down, and mute based on user-defined gestures. The set-top box would need to receive the channel up and down commands and always remain turned on.

Before performing the user tests, the XML file was used to define a gesture language that was considered to be appropriate for users. It was decided that the user can turn on the TV by showing five fingers on their right hand (and no left hand), as shown in Fig. 14.

To increase the volume, the user should show two fingers on the left hand and one on the right hand. Fig. 15 shows the XML used to emit the increase volume command (in this case specific angles are not required). The gesture to decrease the volume is shown in Fig. 16 and requires hiding the fingers on the left hand (while still showing a finger on the right hand). Note that the volume bar has appeared at the center-bottom of the television screen. To change the channel on the set-top box, the gestures are similar to changing the volume, except the left and right hands are reversed (two fingers on the right hand increases the channel number). Finally, the user can also mute the audio from the television. The gesture is a simple thumbs down, as shown in Fig. 17 (this gesture makes use of the angle property defined through the XML file).

Fig. 14. Turning the TV on.

```
<gesture>
  <righthand>
    <finger></finger>
  </righthand>
  <lefthand>
    <finger></finger>
    <finger></finger>
  </lefthand>
  <remote_ir_code>...</remote_ir_code>
</gesture>
```

Fig. 15. Gesture for increasing the TV volume.

Fig. 16. Decreasing the TV volume.

Fig. 17. Muting the sound of the TV.

Several friends and colleagues were asked to test the system and they were generally surprised by its accuracy in detecting their hand gestures. One user experienced difficulties showing five fingers for the on/off gesture as they could not properly separate their fingers. Through a quick modification of the XML file, the required gesture was remapped to just two fingers on the right hand, which the user found much easier. The ease by which the gesture language can be modified allows for much experimentation to take place to find convenient and reliable gestures, or even customization of gestures based on what is preferred by a specific user.

The tiny infrared emitters of the infrared blaster were observed to be very sensitive to how they were positioned in front of the television or set-top box. They had to be positioned very close to the devices for optimal results, and their appearance was somewhat obtrusive. Once the initial setup was completed, however, the signals were transmitted quickly and reliably to the devices. While there exist televisions and set-top boxes that can be controlled through serial ports or USB, these are still very rare and do not apply to the average user. The system in this section instead uses a more flexible solution that can be reproduced with a large variety of users as 3D cameras become more available. The delay between the moment the gesture was performed to the moment the on-screen result appeared was measured to be around 0.09 seconds.

It should also be noted that, while both the 3D camera illumination system and the infrared blasters rely on emitting pulsed near-infrared light, they were not found to interfere with each other, as they are emitting in opposite directions and use different wavelengths.

The ability to reliably determine if a user is showing one or more fingers has not been possible with non-depth-based camera technology. Eliminating the background data from the depth data returned by the camera offers several advantages over existing solutions. For example, users do not need to perform gestures against

a white or stable background for gestures to be detected correctly. In fact, people can be moving around behind the user without affecting the accuracy of the gesture processing.

The amount of data processed by the gesture detection algorithms is also minimized since the algorithms only need to run on a small subset of the data returned by the camera. This allows the system to run in real-time with very little computational burden on the user's PC and with practically no lag or delay between the user's actions and the on-screen response. Additionally, the "gorilla arm"' effect is diminished as users can perform gestures quickly.

Since the implementation reliably returns if the user is showing one or more fingers, gestures that were previously not possible to be determined in a robust way can finally be used as part of a flexible gesture language targeted at the typical home user for interacting with their home electronics.

It was therefore found that depth-based cameras are optimal for hand gesture processing. The recent drop in the price of 3D cameras to the level where they are comparable with high-end webcams indicates that there will be an increasing number of people using gesture control in the future. The convenience of having to position just one camera instead of needing a sophisticated laboratory setup is also ideal for consumers and for use in their homes. By using the depth data from a 3D camera, gesture detection algorithms are far less error prone. As the XML-based configuration has shown, it also enables the ability of defining a flexible gesture language. The far improved reliability of such processing means that gestures can be easier to learn, easier to remember, less awkward to use, and generally more immersive and natural for the user. There are no special gloves or markers for the user to wear; users can simply walk up to their TV and instantly take control by holding up their hands.

5 Multimodal Gesture Fusion

Fig. 18 shows an architecture for multimodal interaction that combines hand gestures and a physical controller [27]. It consists of two concurrent threads that make up the game: a *Sensor Interface Thread* for dealing with the data from the 3D camera and physical controller, and the main *Game Update Thread* for updating the display and game state. The functionality of the *3D Camera* and *Hand Gesture Processing* stages has been described and involves the triggering of in-game actions once specific gestures have been identified.

Once the 3D data has been processed for hand gestures, the location of the hand holding the controller in 3D space is determined as part of the *Physical Controller Position Processing* stage. When the user holds out their hands to perform a gesture and use a controller, the two closest objects to the 3D camera are the user's hands. For the purpose of this exposure, the left hand is to be used for gestures, while the right hand is assumed to be holding the physical controller. Therefore,

by tracking the brightest point in the right side of the depth data, a three-dimensional value for the location of the controller is obtained. By representing the value as the vector *(x, y ,z)*, and given unit vector ($\hat{i}, \hat{j}, \hat{k}$), the translation $\vec{T}_2$ is obtained as follows:

$$, \vec{T}_2 = x\hat{i} + y\hat{j} + z\hat{k} . \quad (1)$$

Therefore, based on this determined absolute position of the controller, a virtual 3D object can be placed within the game world.

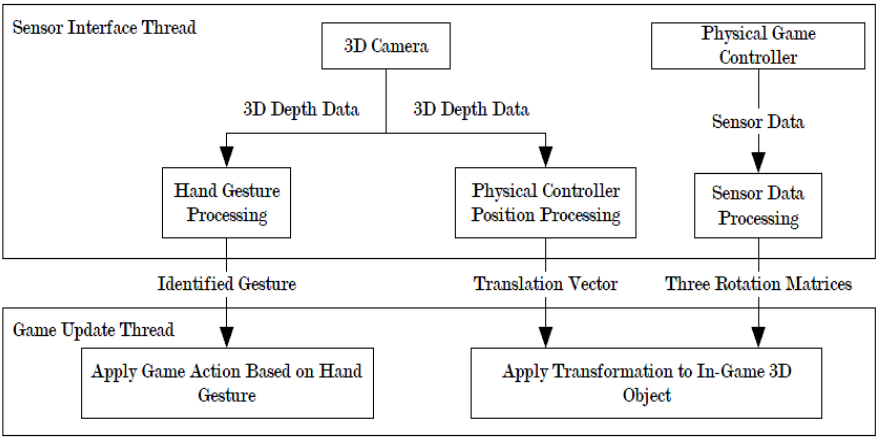


Fig. 18. Multimodal interaction architecture.

The *Sensor Interface Thread* is also responsible for sampling the accelerometer and gyroscope data of the physical game controller. As shown in Fig. 19, the local coordinate system of the game controller $X_1Y_1Z_1$ needs to be combined with the local coordinate system of the hand seen by the 3D camera $X_2Y_2Z_2$. R_{x2}, R_{y2} and R_{z2} are zero since no rotation data is obtained from the 3D camera. By integrating the data returned from the gyroscopes and using the accelerometer data to complement the resulting angles, three 3x3 rotation matrices are obtained: R_{x1}, R_{y1} and R_{z1}. When combining these matrices with $\vec{T}_2$, a 4x4 composite transformation matrix results, which is applied to the game object in its homogeneous coordinate system *X Y Z*. As the game's *Sensor Interface Thread* continues to update, the virtual 3D object should therefore update in real-time and provide a one-to-one correspondence to the real-world controller.

The 3D camera connected to a PC using USB and provided access to its 3D data through a C-based driver and API. Image processing for gesture detection and controller positioning was done using OpenCV, while OpenGL was used for all 2D and 3D display requirements. Fig. 20 shows a test application that was developed to

plot the sensor data of a physical controller that contains three accelerometers and three gyroscopes obtained wirelessly through a PC-based API. Based on the latest angular transformation matrices, the 3D block on the right updates in real-time to reflect the real-world position of the physical controller.

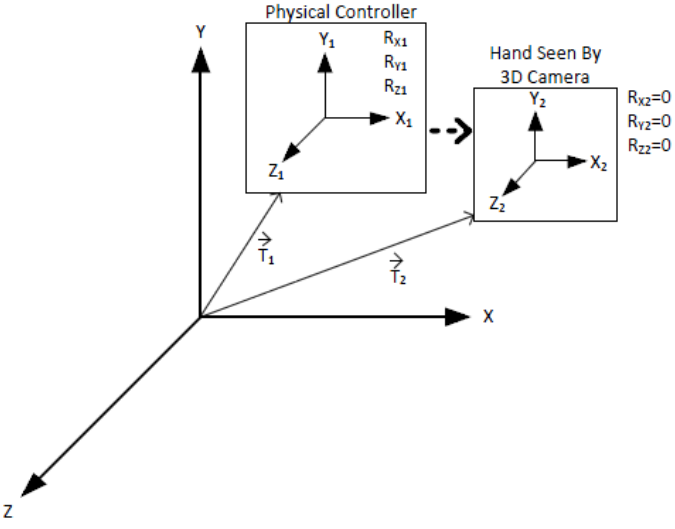


Fig. 19. Combining data from two different coordinate systems.

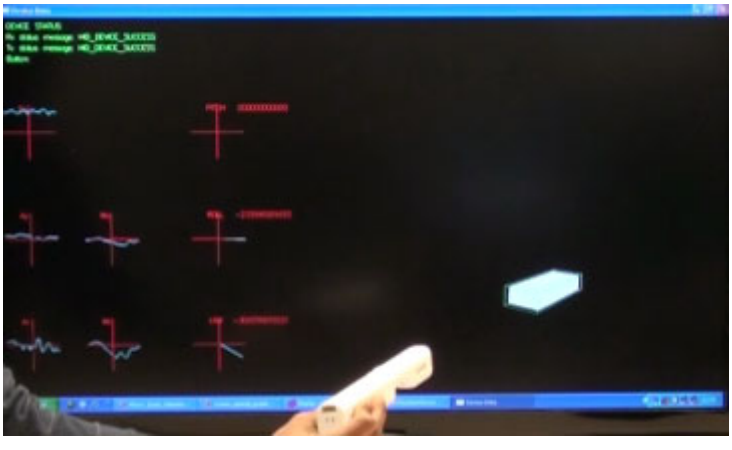


Fig. 20. Monitoring accelerometer and gyroscope data.

Low-pass filtering had to be applied to reduce the jitter that results when scaling the resolution of the 3D data from the camera to that of the game world. A running average of the translation vector and rotation matrices is kept before the transformations are applied to the in-game objects. With a target game frame rate of 30 frames per second, it was discovered that this allowed for four samples to be

collected in the 33 milliseconds between frames. Thus, the 3D object remained stable and transitioned smoothly between updates.

Two video game prototypes were implemented to evaluate the architecture. They are presented in Section 5.1 and in Section 5.2.

5.1 Laser Sword Game

Fig. 21 shows the initial version of a laser sword fighting game that uses gestures and a physical controller. The physical controller is wielded as a sword and its position in 3D space is observed by the 3D camera. In this case, the 3D object receiving the real-time updates from the controller and camera is the hilt of a sword. Fig. 21 shows the player performing a "force push"' gesture, which sends the on-screen enemies flying away from the player. The gesture causes an instant in-game action and can be used at the same time as the physical controller is being moved.

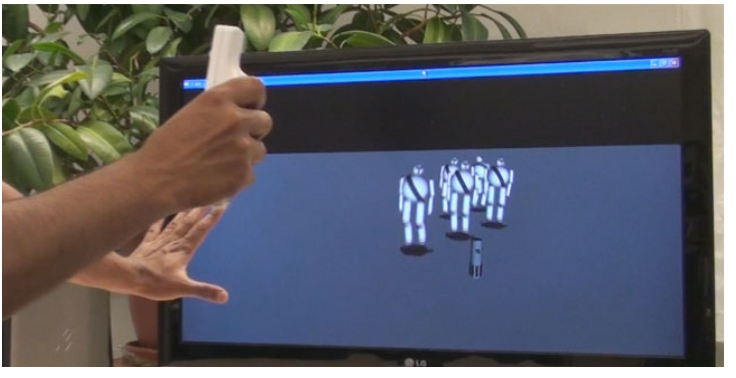


Fig. 21. Gesture and controller combination prototype.

When demonstrating the game to real users at a local technology trade show, users instantly knew how to play the game since they observed the one-to-one movements of the on-screen sword to the controller. Without requiring specific instructions, users realized that by pushing the large button on the controller, they could cause the laser blade of the sword to appear. The button is naturally expected to affect the sword and it would make little sense to map this action to a gesture. In addition, the force feedback feature of the physical controller provided the player with further haptic feedback when the blade of the sword made contact with another sword.

A later version of the game is shown in Fig. 22. The three accelerometers and three gyroscopes of the physical controller, when combined with the 3D tracking of the depth camera, allow the position and orientation of the 3D sword to be controlled with great accuracy. For example, a player can move the sword around the

on-screen enemy by moving closer to the camera. They player can also target areas on the enemy's body with great precision. It was discovered that this allows for an entertaining gameplay experience when combined with a physics engine for rag doll physics for the on-screen enemies.

Fig. 22. Laser sword swing tracked with one-to-one correspondence.

5.2 Third-Person Action Game

A custom third-person action game was developed using Microsoft XNA to further evaluate the viability of such a multimodal interface. In this game, the player controls a female heroine who must reach her home by shooting at enemies attempting to block her path. The heroine can cast a magic shield that traps enemy bullets when activated. When deactivated, all bullets caught in the shield are fired back towards the enemies. An open palm gesture from the left hand was assigned to cast the shield spell, as shown in Fig. 23. By closing one's hand as in Fig. 24, the bullets are returned to the enemies. The movement controls were placed on the physical controller, as movement greatly benefits from the additional precision offered by buttons. Shooting is also performed by using a button on the physical controller. By timing the use of the shield and moving to attack enemies, players are given a novel gameplay experience.

Several friends and co-workers were asked to test and provide feedback on the game prototype. While it took time for some players to get used to the unique control scheme, all testers generally found the controls responsive and fun to use since both hands were actively engaged in the gameplay. Based on recorded video footage of these experiments, the delay between the moment the shield gesture was performed to the moment the on-screen result appeared was observed to be around 0.09 seconds, which is more than acceptable.

Fig. 23. Casting a spell to block enemy fire using gestures.

Fig. 24. Releasing the bullets back at the enemy.

6 Conclusions

The recent growth in the popularity of motion-based control for video games triggered a recrudescence in the research related to gesture control of computer applications or of consumer electronic devices, enabling at the same time new frontiers for the intelligent human-computer interfaces.

This chapter presented the next step in gesture-based interfaces where the convenience of hand gestures is used either alone or in combined with the tactile feel of a physical controller. A custom 3D IR camera is used to obtain accurate depth data which is further processed for hand gestures detection, tracking and recognition in order control games or electronic devices such as TV sets and set-top boxes. It was also shown how the depth data can be used to detect the position of a

physical game controller in 3D space. By using a composite transformation matrix this 3D data obtained from the special camera is fused with the accelerometer and gyroscope data produced by a game controller. This combination provides a true six degrees of freedom (SDF) and a one-to-one correspondence with the real and virtual reality objects. A prototype game was shown where a player performs a "force push" gesture by quickly moving their hand outward while controlling a 3D laser sword with the physical controller to defeat on-screen enemies. A third-person action game was also shown where gestures are used to trap and release enemy bullets by "casting a spell" while the movement of the player's character is controlled using a physical controller. Users found such multimodal interaction methods to be natural and immersive, thereby offering numerous unique gaming experiences that are attractive and accessible to people of all ages.

References

[1] Sutherland, I.E.: Sketchpad: A man-machine graphical communication system. In: Proceedings of the AFIPS Spring Joint Computer Conference, vol. 23, pp. 329–346 (1963)

[2] Teitelman, W.: Real time recognition of hand-drawn characters. In: AFIPS Proceedings of Fall Joint Computer Conference, pp. 559–575. Spartan Books (1964)

[3] Buxton, W., Fiume, E., Hill, R., Lee, A., Woo, C.: Continuous hand-gesture driven input. In: Proceedings of Graphics Interface 83rd Conference of the Canadian Man-Computer Communications Society, pp. 191–195 (1983)

[4] Ou, J., Fussell, S.R., Chen, X., Setlock, L.D., Yang, J.: Gestural communication over video stream: supporting multimodal interaction for remote collaborative physical tasks. In: Proceedings of 5th International Conference on Multimodal interfaces (ICMI 2003), Vancouver, BC, Canada, pp. 242–249 (2003)

[5] Grossman, T., Wigdor, D., Balakrishnan, R.: Multi-finger gestural interaction with 3d volumetric displays. In: Proceedings of 17th Annual ACM Symposium on User Interface Software and Technology, Santa Fe, NM, USA, pp. 61–70 (2004)

[6] Sturman, D.J., Zeltzer, D., Pieper, S.: Hands-on interaction with virtual environments. In: Proceedings of 2nd Annual ACM SIGGRAPH Symposium on User Interface Software and Technology, Williamsburg, West Virginia, USA, pp. 19–24 (1989)

[7] Wexelblat, A.: An approach to natural gesture in virtual environments. ACM Trans. Comput. Hum. Interact. 2, 179–200 (1995)

[8] Bolt, R.A.: Put that there - voice and gesture at the graphics interface. In: Proceedings of 7th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1980), Seattle, WA, USA, pp. 262–270 (1980)

[9] Bolt, R.A., Herranz, E.: Two-handed gesture in multi-modal natural dialog. In: Proceedings of 5th Annual ACM Symposium on User Interface Software and Technology, Monteray, CA, USA, pp. 7–14 (1992)

[10] Goza, S.M., Ambrose, R.O., Diftler, M.A., Spain, I.M.: Telepresence control of the nasa/darpa robonaut on a mobility platform. In: Proceedings of 2004 Conference on Human Factors in Computing Systems, pp. 623–629 (2004)

[11] Derpanis, K.G.: A review of vision-based hand gestures, Technical Report. Department of Computer Science, York University, Toronto, ON, Canada (2004)

[12] Mitra, S., Acharya, T.: Gesture recognition: A survey. IEEE Trans. Syst., Man, Cybern. C, Appl. Rev. 37, 311–324 (2007)
[13] Murthy, G.R.S., Jadon, R.S.: A review of vision based hand gesture recognition. Int. J. Inf. Tech. Knowl. Manag. 2, 405–410 (2009)
[14] Wachs, J.P., Kolsch, M., Stern, H., Edan, Y.: Vision-based hand-gesture applications. Mag. ACM Commun. 54, 60–71 (2011)
[15] Suma, E.A., Lange, B., Rizzo, A.S., Krum, D.M., Bolas, M.: FAAST: The Flexible Action and Articulated Skeleton Toolkit. In: Proceedings of IEEE Conference on Virtual Reality, Singapore, pp. 247–248 (2011)
[16] Ramey, A., Gonzalez-Pachec, V., Salichs, M.A.: FAAST: The Flexible Action and Articulated Skeleton Toolkit, Research report, USC Institute for Creative Technologies, University of Southern California, Playa Vista, USA (2011)
[17] Varona, J., Jaume-i Capó, A., González, J., Perales, F.J.: Toward natural interaction through visual recognition of body gestures in real-time. Interact Comp. 21, 3–10 (2009)
[18] Ju, Z., Liu, H., Zhu, X., Xiong, Y.L.: Dynamic Grasp Recognition Using Time Clustering, Gaussian Mixture Models and Hidden Markov Models. In: Xiong, C.-H., Liu, H., Huang, Y., Xiong, Y.L. (eds.) ICIRA 2008. LNCS (LNAI), vol. 5314, pp. 669–678. Springer, Heidelberg (2008)
[19] Ramamoorthy, A., Vaswani, X., Chaudhury, S., Banerjee, S.: Recognition of dynamic hand gestures. Pattern Recogn. 36, 2069–2081 (2003)
[20] Quek, F.K.H., Zhao, M.: Inductive learning in hand pose recognition. In: Proceedings of IEEE Conference on Automatic Face and Gesture Recognition, Killington, Vermont, USA, pp. 78–87 (1996)
[21] Zhao, M., Quek, F.K.H.: RIEVL: Recursive Induction Learning in hand gesture recognition. IEEE Trans. Pattern Anal. Mach. Intell. 20, 1174–1185 (1998)
[22] Mitra, S., Acharya, T.: Data mining: Multimedia, soft computing, and bioinformatics. Wiley, New York (2003)
[23] Wu, Y., Huang, T.S.: Hand modeling analysis and recognition for vision-based human computer interaction. IEEE Signal Process. Mag. 18, 51–60 (2001)
[24] Yao, J., Zhang, H., Zhang, H., Chen, Q.: R&D of a parameterized method for 3D virtual human body based on anthropometry. Int. J. Virtual Reality 3, 9–12 (2008)
[25] Ionescu, D., Ionescu, B., Islam, S., Gadea, C., McQuiggan, E.: Using depth measuring cameras for a new human computer interaction in augmented virtual reality environments. In: Proceedings of International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems (VECIMS 2010), Taranto, Italy, pp. 114–119 (2010)
[26] Ionescu, D., Ionescu, B., Gadea, C., Islam, S.: An intelligent gesture interface for controlling TV sets and set-top boxes. In: Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisoara, Romania, pp. 159–165 (2011)
[27] Ionescu, D., Ionescu, B., Gadea, C., Islam, S.: A multimodal interaction method that combines gestures and physical game controllers. In: Proceedings of 20th IEEE International Conference on Computer Communication Networks (ICCCN 2011), Maui, Hawaii, USA, pp. 1–6 (2011); doi:10.1109/ICCCN.2011.6006079

Author Index