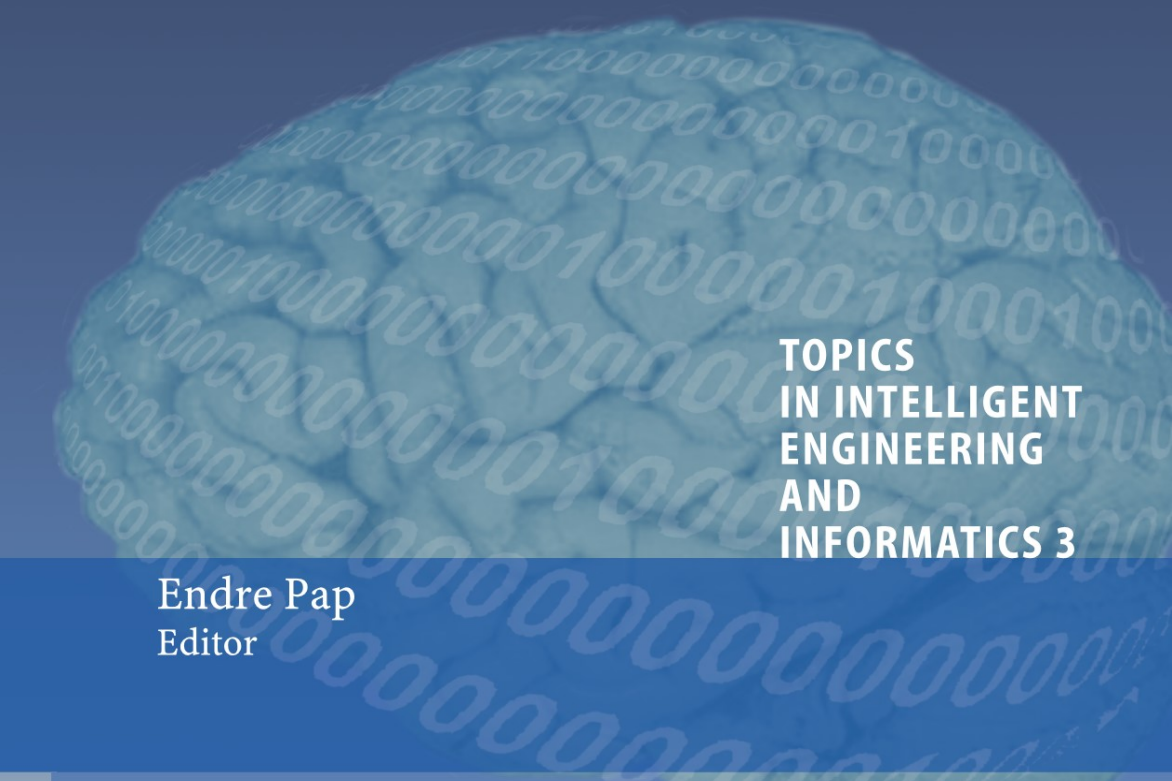


TOPICS IN INTELLIGENT ENGINEERING AND INFORMATICS 3

Endre Pap
Editor

Intelligent Systems: Models and Applications

Revised and Selected Papers from the 9th IEEE International Symposium on Intelligent Systems and Informatics SISY 2011

Springer

Topics in Intelligent Engineering and Informatics 3

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.

Endre Pap (Ed.)

Intelligent Systems: Models and Applications

Revised and Selected Papers from the 9th IEEE International Symposium on Intelligent Systems and Informatics SISY 2011

Springer

Editor
Prof. Endre Pap
Faculty of Sciences and Mathematics
Department of Mathematics and Informatics
University of Novi Sad
Novi Sad
Serbia

and

Educons University
21208 Sremska Kamenica
Serbia

and

Óbuda University
Budapest
Hungary

ISSN 2193-9411
e-ISSN 2193-942X
ISBN 978-3-642-33958-5
e-ISBN 978-3-642-33959-2
DOI 10.1007/978-3-642-33959-2
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012949576

Preface

The theory and applications of intelligent system is today an important field of research of scientists and engineers, doctors, economists, in many different fields, which include learning, adaptability, management of uncertain information, decision. It is hard to define precisely uniquely the notion of intelligent system, including the intelligence of humans, which strongly depends on the field of research and applications.

The present edited book is a collection of 17 chapters, written by recognized experts in the fields. The origin of papers are based on the invitation of authors initiated by selected papers from the 9th IEEE International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, September 8-10, 2011. Chapters are contributions to different aspects and applications of intelligent systems. I believe that the state-of-the art investigations presented in this book will be very useful not only for experts in the field but to broad readership. The volume is organized in five parts according to the considered subjects.

The first part of the book is devoted to the mathematical base. First, it is given an overview of generalizations of the integral inequalities for (Choquet, Sugeno, universal, pseudo) integrals based on nonadditive measures as Hölder, Minkowski, Jensen, Chebishev and Berwald inequalities, with their applications in pseudo-probability and decision making. Second chapter in this part is focused on two approaches to interval-valued measures and corresponding integrals with generalizations of the Jensen and the Chebyshev integral inequalities. The third chapter is devoted to a construction of the General Prioritized Fuzzy Satisfaction Problem (GPFCSP) that can handle any logical expression whose atomic symbols are prioritized constraints with applications in Fuzzy Relational Databases, multilateral negotiations, decision making, etc. In the fourth chapter there are presented new results about T-supermodularity for Choquet integral as generalization of the concept of supermodularity for Choquet integral which was based on Frank t-norms, now extending on general membership functions using properties and links among belief measures, Möbius transform and Choquet integrals, in order to present the general case studied over a finite set.

The second part of the book concerns different aspects of robotics. The first chapter in this part emphasizes some of the differences between human beings and robots, and also some differences between computers and their programming, and robots and programming of robots, and it recognize programming as an integral part of the overall human culture, and formulates complementing the usual Turing hypothesis, another Turing hypothesis rooted in the famous Turing test. The second chapters demonstrates the possibility of using primitives to generate complex movements that ensure motion of bipedal humanoid robots in unstructured environments, and it is pointed out that for the robot's motion in an unstructured environment an on-line generation of motion is required. Third chapter deals with benchmarking and qualitative evaluation of different autonomous quadrotor flight controllers using three characteristic representatives of frequently used flight control techniques: PID, backstepping and fuzzy.

The third part of the book presents applications of fuzzy systems in different fields. The first chapter of this part presents a new mathematical model of basic planar imprecise geometric objects (fuzzy line, fuzzy triangle and fuzzy circle), basic measurement functions (distance between fuzzy point and fuzzy line, fuzzy point and fuzzy triangle, two fuzzy lines and two fuzzy triangles), spatial operation (linear combination of two fuzzy points) and main spatial relations (coincidence, between and collinear) allowing various applications in image analysis (imprecise feature extraction), GIS (imprecise spatial object modeling), robotics (environment models). The second chapter deals with basic concepts of type-2 fuzzy logic, computing with words and perceptual computing, and then architectural details of an object-oriented realization of a software library for developing perceptual computers are exposed and explained. The third chapter addresses issues of two–degree–of–freedom (2–DOF) speed control solutions for brushless Direct Current motor (BLDC–m) drives with focus on design methodologies: classical 2–DOF structure, 2–DOF proportional-integral (PI) and proportional-integral-derivative (PID) structures and 2–DOF fuzzy control solutions.

The forth part of the book is devoted to applications of intelligent systems in medicine. The first chapter is devoted to Virtual Doctor System (VDS) which is built as intelligent thinking support for assisting medical doctor in a hospital to do medical diagnosis based on the avatar of that doctor, where the medical knowledge is collected from the doctor based on his/her experience in diagnosis, and the avatar interacts with patients through their voices, and other sensors to read patient mental state and physical state that are used in aligned manner to assess the patient sickness states through Bayesian network. The second chapter deals with methods, which are used for an individual adaption of a dialog system, an automatic real-time capable visual user attention estimation for a face to face human machine interaction, and an emotion estimation is presented, which combines a visual and an acoustic method, specially onto the current head pose. The third chapter presents a mathematical model construction approach for functioning systems for heterogeneous systems when the available information is insufficient, with applications for multiple diseases and the effect of the applied therapy. The fourth chapter presents the concepts of knowledge similarity and approximation, together with appropriate

scaling - degrees of similarity/approximation, in metric spaces for continuous case and the solution for discrete information spaces-spaces for the later case, with different applications as for example signals with speech, arterial pressure, electroencephalogram, heart rate, and then the beginning of the famous music performance, intraocular pressure, submarine echogram, and encephalogram of acoustic stimulus.

The last part of the book covers different applications in transportation, network monitoring, and localization of pedestrians in images. The first chapter presents an overview of existing optimization problems in container terminal operations, with the goal of defining a model for quay crane operations in river container terminals giving a scheduling of quay cranes in river terminals for barge reloading. The second chapter describes implementation of the agent-based system for network components and services monitoring, and it is designed in a modular fashion to provide easy and efficient inclusion of diverse monitoring objects. The third chapter presents two different approaches for pedestrian localization using neural networks with local receptive fields, where the first approach uses a trained ranking classifier to determine the relative order of image windows in regard to their localization quality (coverage) of the pedestrian, and the second approach uses a binary classifier which is trained to stepwise move an initial window towards the optimal position.

The editor is grateful to the authors for their effort in bringing their papers in nice form and in time. Thanks due to the editors of the series Imre Rudas and Janos Fodor to accept this edited book. I want to thank Faculty of Sciences, Department of Mathematics and Informatics in Novi Sad, and Óbuda university in Budapest, for working conditions in the preparation of this edited book. The book is supported in part by the Project MPNRS 174009, and by the project *Mathematical models of intelligent systems and their applications* of Academy of Sciences and Arts of Vojvodina supported by Provincial Secretariat for Science and Technological Development of Vojvodina.

July 12, 2012

Endre Pap
Novi Sad and Budapest

Contents

Part IV: Applications in Medicine

Part I
Mathematical Base

Generalizations of Integral Inequalities for Integrals Based on Nonadditive Measures

Endre Pap and Mirjana Štrboja

Abstract. There is given an overview of generalizations of the integral inequalities for integrals based on nonadditive measures. The Hölder, Minkowski, Jensen, Chebishev and Berwald inequalities are generalized to the Choquet and Sugeno integrals. A general inequality which cover Hölder and Minkowski type inequalities is considered for the universal integral. The corresponding inequalities for important cases of the pseudo-integral and applications of these inequalities in pseudo-probability are also given.

Keywords: Nonadditive measure, Choquet integral, Sugeno integral, pseudo-integral, Jensen inequality, Hölder inequality, Minkowski inequality, Chebishev inequality, Berwald inequality.

1 Introduction

The Hölder, Minkowski, Jensen and Chebyshev inequalities for Lebesgue integral, see [12], play an important role in mathematical analysis and in other areas of mathematics, especially in theory of probability, differential equations, geometry, and wider, e.g., information sciences, economics, engineering.

Endre Pap · Mirjana Štrboja
Faculty of Sciences and Mathematics,
University of Novi Sad,
Trg Dositeja Obradovića 4,
21000 Novi Sad, Serbia
e-mail: {pap,mirjana.strboja}@dmi.uns.ac.rs

Endre Pap
Óbuda university,
H-1034 Budapest,
Becsi út 96/B, Hungary,
Educons University, 21208 Sremska Kamenica, Serbia

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 3–22.
DOI: 10.1007/978-3-642-33959-2_1

The classical measure theory is based on countable additive measures and the Lebesgue integral. However, additivity measures do not allow modeling many phenomena. For this reason, nonadditive measure, called also fuzzy measure or capacity, and the corresponding integrals, e.g., Choquet, Sugeno, are introduced, see [15, 18, 36, 45]. The Choquet and Sugeno integrals have important applications as aggregation functions in decision theory (multiple criteria, multiple attributes, multiperson decision making, multiobjective optimization), information or data fusion, artificial intelligence and fuzzy logic, see [17, 44, 47, 48].

The pseudo-analysis as a generalization of the classical analysis is based on a special nonadditive measures, called pseudo-additive measures, and one of its tools is the pseudo-integral. There we consider instead of the field of real numbers a semiring, i.e., a real interval $[a,b] \subseteq [-\infty,\infty]$ with pseudo-addition $\oplus$ and with pseudo-multiplication $\odot$, see [34, 35, 36, 37, 39, 40, 46]. On this structure the notions of $\oplus$-measure (pseudo-additive measure) and corresponding integral (pseudo-integral) are introduced. Methods of the pseudo-analysis can be applied for solving problems in many different fields such as system theory, optimization, decision making, control theory, differential equations, difference equations, etc. Similar approach was introduced independently by Maslov and his collaborators in the framework of idempotent measures theory, see [20, 22, 25]. The theory of cost measures based on idempotent measures and integrals of Maslov was delevoped, see [7, 8, 9].

Considering the wide applications of integrals based on nonadditive measure, there is a need for the study of inequalities for those integrals. The study of inequalities for Choquet and Sugeno integral, were given in [1, 2, 16, 27, 28, 30, 31, 43]. The first of all Jensen type inequality for Sugeno integral was obtained by Román-Flores *et al.* [43]. A fuzzy Chebyshev type inequalty has been considered by a several authors, see [2, 16, 28, 30, 32]. Inequalities with respect to the Choquet integral is observed by Wang [31] and Mesiar, Li, and Pap [27]. The generalizations of the classical integral inequalities for the universal integral (introduced in [19]) were investigated in [6]. In [1, 5, 41, 42] inequalities with respect to pseudo-integrals were obtained.

In the Section 2 an overview on generalizations of the Jensen, Hölder, Minkowski, Chebishev and Berwald type inequalities for Choquet and Sugeno integrals are given. In Section 3, we review results related to the universal integral, as generalization of Choquet and Sugeno integral. There are given a general inequality which cover Hölder and Minkowski type inequalities. Generalizations of the Hölder, Minkowski, Jensen and Chebyshev type inequalities for important cases of the pseudo-integral are presented in Section 4. Inspired by applications integral inequalities in the probability theory, the pseudo-probability is introduced and the inequalities valid for the pseudo-integral is applied in this theory in Section 4.6.1. Finally, using the notions of the cost measure we review these inequalities related to the value of a cost variables.

2 General Nonadditive Measures and Integrals Based on Them

Let X be a non-empty set and $\mathscr{A}$ a σ-algebra of subsets of X. Then $(X,\mathscr{A})$ is measurable space and a function $f : X \to [0,\infty]$ is called $\mathscr{A}$-measurable if, for each $B \in \mathscr{B}([0,\infty])$, the σ -algebra of Borel subsets of $[0,\infty]$, the preimage $f^{-1}(B)$ is an element of $\mathscr{A}$.

Definition 1. ([36, 48]) A monotone measure m on a measurable space $(X,\mathscr{A})$ is a function $m : \mathscr{A} \to [0,\infty]$ satisfying
(i) $m(\varnothing) = 0$,
(ii) $m(X) > 0$,
(iii) $m(A) \leqslant m(B)$ whenever $A \subseteq B$.

Normed monotone measures on $(X,\mathscr{A})$, i.e., monotone measures satisfying $m(X) = 1$, are also called fuzzy measures (see [36, 48]).

The Choquet, Sugeno and Shilkret integrals (see [10, 36, 38, 48]) are based on monotone measure and they are defined, respectively, for any measurable space $(X,\mathscr{A})$, for any measurable function f and for any monotone measure m, by

$$\begin{aligned}
\mathbf{Ch}(m,f) &= \int_0^\infty m(\{f \geqslant t\})\,dt,\\
\mathbf{Su}(m,f) &= \sup\{\min(t, m(\{f \geqslant t\})) \mid t \in]0,\infty])\},\\
\mathbf{Sh}(m,f) &= \sup\{t \cdot m(\{f \geqslant t\}) \mid t \in]0,\infty])\},
\end{aligned}$$

where the convention $0 \cdot \infty = 0$ is used.

Now we give a short overview on results related to the generalizations of the classical integral inequalities for Choquet and Sugeno integrals.

Jensen inequality and reverse Jensen inequality for Sugeno integral is obtained in [43]. Jensen, Chebyshev, Hölder and Minkowski type inequalities for Choquet integral and several convergence concepts as applications of these inequalities are observed in [31]. An approach to the Choquet integral as Lebesgue integral is given in [27] and in this way there are obtained the related inequalities.

Chebyshev type inequality for Sugeno integral based on Lebesgue measure are obtained in [16]. Previous results from [16] are generalised in [30]. Namely, there is presented a Chebyshev type inequality for Sugeno integral based on an arbitrary fuzzy measure. This inequality for comonotone functions and arbitrary fuzzy measure-base Sugeno integral were given in [28]

A general Minkowski type inequality for Sugeno integral is obtained in [1].

Berwald inequality for Sugeno integral is studied in [4]. This inequality holds in the following form:

Theorem 1. *Let* $0 < r < s < \infty$, $f : [a,b] \to [0,\infty[$ *be a concave function and* m *be the Lebesgue measure on* $\mathbb{R}$. *Then*
(a) if $f(a) < f(b)$, *then*

$$(\mathbf{Su}(m,f^r))^{\frac{1}{r}} \geqslant \min\left\{ \begin{array}{l} \frac{(b-a)^{\frac{1}{r}}(1+s)^{\frac{1}{s}}}{(1+r)^{\frac{1}{r}}}\left(\frac{\mathbf{Su}(m,f^s)}{b-a}\right)^{\frac{1}{s}}, \\ \left(b-\frac{\frac{(b-a)^{\frac{1+r}{r}}(1+s)^{\frac{1}{s}}}{(1+r)^{\frac{1}{r}}}\left(\frac{\mathbf{Su}(m,f^s)}{b-a}\right)^{\frac{1}{s}}+af(b)-bf(a)}{f(b)-f(a)}\right)^{\frac{1}{r}} \end{array} \right\},$$

(b) if $f(a)=f(b)$, *then*

$$(\mathbf{Su}(m,f^r))^{\frac{1}{r}} \geqslant \min\left\{f(a),(b-a)^{\frac{1}{r}}\right\},$$

(c) if $f(a)>f(b)$, *then*

$$(\mathbf{Su}(m,f^r))^{\frac{1}{r}} \geqslant \min\left\{ \begin{array}{l} \frac{(b-a)^{\frac{1}{r}}(1+s)^{\frac{1}{s}}}{(1+r)^{\frac{1}{r}}}\left(\frac{\mathbf{Su}(m,f^s)}{b-a}\right)^{\frac{1}{s}}, \\ \left(\frac{\frac{(b-a)^{\frac{1+r}{r}}(1+s)^{\frac{1}{s}}}{(1+r)^{\frac{1}{r}}}\left(\frac{\mathbf{Su}(m,f^s)}{b-a}\right)^{\frac{1}{s}}+af(b)-bf(a)}{f(b)-f(a)}-a\right)^{\frac{1}{r}} \end{array} \right\}.$$

3 Universal Integral

In order to define the notion of the universal integral the following notions are needed.

Definition 2. ([19]) Let $(X,\mathscr{A})$ be a measurable space.
(i) $\mathscr{F}^{(X,\mathscr{A})}$ is the set of all $\mathscr{A}$ -measurable functions $f: X \to [0,\infty]$;
(ii) For each number $a \in]0,\infty]$, $\mathscr{M}_a^{(X,\mathscr{A})}$ is the set of all monotone measures (in the sense of Definition 1) satisfying $m(X)=a$; and we take

$$\mathscr{M}^{(X,\mathscr{A})} = \bigcup_{a\in]0,\infty]} \mathscr{M}_a^{(X,\mathscr{A})}.$$

An equivalence relation between pairs of measures and functions was introduced in [19].

Definition 3. Two pairs $(m_1,f_1) \in \mathscr{M}^{(X_1,\mathscr{A}_1)} \times \mathscr{F}^{(X_1,\mathscr{A}_1)}$ and $(m_2,f_2) \in \mathscr{M}^{(X_2,\mathscr{A}_2)} \times \mathscr{F}^{(X_2,\mathscr{A}_2)}$ satisfying

$$m_1(\{f_1 \geqslant t\}) = m_2(\{f_2 \geqslant t\}) \text{ for all } t \in]0,\infty],$$

will be called integral equivalent, in symbols

$$(m_1, f_1) \sim (m_2, f_2).$$

Notion of the pseudo-multiplication is necessary to introduce the universal integral.

Definition 4. ([36, 46]) A function $\otimes\colon [0,\infty]^2 \to [0,\infty]$ is called a pseudo-multiplication if it satisfies the following properties:

(i) it is non-decreasing in each component, i.e., for all $a_1, a_2, b_1, b_2 \in [0,\infty]$ with $a_1 \leqslant a_2$ and $b_1 \leqslant b_2$ we have $a_1 \otimes b_1 \leqslant a_2 \otimes b_2$;
(ii) 0 is an annihilator of , i.e., for all $a \in [0,\infty]$ we have $a \otimes 0 = 0 \otimes a = 0$;
(iii) has a neutral element different from 0, i.e., there exists an $e \in]0,\infty]$ such that, for all $a \in [0,\infty]$, we have $a \otimes e = e \otimes a = a$.

Let $\mathscr{S}$ be the class of all measurable spaces, and take

$$\mathscr{D}_{[0,\infty]} = \bigcup_{(X,\mathscr{A})\in\mathscr{S}} \mathscr{M}^{(X,\mathscr{A})} \times \mathscr{F}^{(X,\mathscr{A})}.$$

The Choquet, Sugeno and Shilkret integrals are particular cases of the following integral given in [19].

Definition 5. A function $\mathbf{I}\colon \mathscr{D}_{[0,\infty]} \to [0,\infty]$ is called a universal integral if the following axioms hold:

(I1) For any measurable space $(X,\mathscr{A})$, the restriction of the function $\mathbf{I}$ to $\mathscr{M}^{(X,\mathscr{A})} \times \mathscr{F}^{(X,\mathscr{A})}$ is non-decreasing in each coordinate;
(I2) there exists a pseudo-multiplication $\otimes\colon [0,\infty]^2 \to [0,\infty]$ such that for all pairs $(m, c\cdot \mathbf{1}_A) \in \mathscr{D}_{[0,\infty]}$ (where $\mathbf{1}_A$ is the characteristic function of the set A)

$$\mathbf{I}(m, c\cdot \mathbf{1}_A) = c \otimes m(A);$$

(I3) for all integral equivalent pairs $(m_1, f_1), (m_2, f_2) \in \mathscr{D}_{[0,\infty]}$ we have

$$\mathbf{I}(m_1, f_1) = \mathbf{I}(m_2, f_2).$$

By Proposition 3.1 from [19] we have the following important characterization.

Theorem 2. *Let $\otimes\colon [0,\infty]^2 \to [0,\infty]$ be a pseudo-multiplication on $[0,\infty]$. Then the smallest universal integral $\mathbf{I}$ and the greatest universal integral $\mathbf{I}$ based on $\otimes$ are given by*

$$\begin{aligned}\mathbf{I}_{\otimes}(m,f) &= \sup\{t \otimes m(\{f \geqslant t\}) \mid t \in]0,\infty]\},\\ \mathbf{I}^{\otimes}(m,f) &= \operatorname{essup}_m f \otimes \sup\{m(\{f \geqslant t\}) \mid t \in]0,\infty]\},\end{aligned}$$

where $\operatorname{essup}_m f = \sup\{t \in [0,\infty] \mid m(\{f \geqslant t\}) > 0\}$.

Notice that $\mathbf{Su} = \mathbf{I}_{Min}$ and $\mathbf{Sh} = \mathbf{I}_{Prod}$, where the pseudo-multiplications Min and $Prod$ are given by $Min(a,b) = \min(a,b)$ and $Prod(a,b) = a\cdot b$.

There is neither a smallest nor a greatest pseudo-multiplication on $[0,\infty]$. But, if we fix the neutral element $e \in]0,\infty]$, then the smallest pseudo-multiplication $\otimes_e$ with neutral element e is given by

$$a \otimes_e b = \begin{cases} 0 & \text{if } (a,b) \in [0,e[^2, \\ \max(a,b) & \text{if } (a,b) \in [e,\infty]^2, \\ \min(a,b) & \text{otherwise.} \end{cases}$$

Then by Proposition 3.2 from [19] there exists the smallest universal integral $\mathbf{I}_{\otimes_e}$ among all universal integrals given by

$$\mathbf{I}_{\otimes_e}(m,f) = \max(m(\{f \geqslant e\}), \operatorname{essinf}_m f),$$

where $\operatorname{essinf}_m f = \sup\{t \in [0,\infty] \mid m(\{f \geqslant t\}) = e\}$.

3.1 A General Inequality for the Universal Integral

We will give first a main inequality, see [6], and then the Minkowski and Chebyshev type inequalities appear as special cases.

The following important property of a pair of functions is needed, see [15, 36]. Functions $f,g\colon X \to \mathbb{R}$ are said to be comonotone if for all $x,y \in X$,

$$(f(x)-f(y))(g(x)-g(y)) \geqslant 0.$$

The comonotonicity of functions f and g is equivalent to the nonexistence of points $x,y \in X$ such that $f(x) < f(y)$ and $g(x) > g(y)$.

Theorem 3. *Let $\star\colon [0,\infty[^2 \to [0,\infty[$ be continuous and nondecreasing in both arguments and $\varphi\colon [0,\infty[\to [0,\infty[$ be continuous and strictly increasing function. Let $f,g \in \mathscr{F}^{(X,\mathscr{A})}$ be two comonotone measurable functions and $\otimes_e\colon [0,\infty]^2 \to [0,\infty]$ be a smallest pseudo-multiplication on $[0,\infty]$ with neutral element $e \in]0,\infty]$ and $m \in \mathscr{M}^{(X,\mathscr{A})}$ be a monotone measure such that $\mathbf{I}_{\otimes_e}(m,\varphi(f))$ and $\mathbf{I}_{\otimes_e}(m,\varphi(g))$ are finite. If*

$$\varphi^{-1}((\varphi(a \star b) \otimes_e c)) \geqslant \left(\varphi^{-1}((\varphi(a) \otimes_e c)) \star b\right) \vee \left(a \star \varphi^{-1}((\varphi(b) \otimes_e c))\right),$$

then the inequality

$$\varphi^{-1}(\mathbf{I}_{\otimes_e}(m,\varphi(f \star g))) \geqslant \varphi^{-1}(\mathbf{I}_{\otimes_e}(m,\varphi(f))) \star \varphi^{-1}(\mathbf{I}_{\otimes_e}(m,\varphi(g)))$$

holds.

3.2 Minkowski's Inequality for Universal Integral

As a corollary of Theorem 3 we obtain an inequality related to Minkowski type for universal integral. Hence, if $\varphi(x) = x^s$ for all $s > 0$, the following holds:

Corollary 1. *Let $f, g \in \mathscr{F}^{(X,\mathscr{A})}$ be two comonotone measurable functions and $\otimes_e : [0,\infty]^2 \to [0,\infty]$ be a smallest pseudo-multiplication on $[0,\infty]$ with neutral element $e \in]0,\infty]$ and $m \in \mathscr{M}^{(X,\mathscr{A})}$ be a monotone measure such that $\mathbf{I}_{\otimes_e}(m, f^s)$ and $\mathbf{I}_{\otimes_e}(m, g^s)$ are finite. Let $\star : [0,\infty[^2 \to [0,\infty[$ be continuous and nondecreasing in both arguments. If*

$$((a \star b)^s \otimes_e c)^{\frac{1}{s}} \geqslant \left((a^s \otimes_e c)^{\frac{1}{s}} \star b\right) \vee \left(a \star (b^s \otimes_e c)^{\frac{1}{s}}\right),$$

then the inequality

$$(\mathbf{I}_{\otimes_e}(m, (f \star g)^s))^{\frac{1}{s}} \geqslant (\mathbf{I}_{\otimes_e}(m, f^s))^{\frac{1}{s}} \star (\mathbf{I}_{\otimes_e}(m, g^s))^{\frac{1}{s}}$$

holds for all $s > 0$.

3.3 Chebyshev's Inequality for Universal Integral

Due to Theorem 3, if $s = 1$ we have the Chebyshev type inequality.

Corollary 2. *Let $f, g \in \mathscr{F}^{(X,\mathscr{A})}$ be two comonotone measurable functions and $\otimes_e : [0,\infty]^2 \to [0,\infty]$ be a smallest pseudo-multiplication on $[0,\infty]$ with neutral element $e \in]0,\infty]$ and $m \in \mathscr{M}^{(X,\mathscr{A})}$ be a monotone measure such that $\mathbf{I}_{\otimes_e}(m, f)$ and $\mathbf{I}_{\otimes_e}(m, g)$ are finite. Let $\star : [0,\infty[^2 \to [0,\infty[$ be continuous and nondecreasing in both arguments. If*

$$(a \star b) \otimes_e c) \geqslant [(a \otimes_e c) \star b] \vee [a \star (b \otimes_e c)],$$

then the inequality

$$\mathbf{I}_{\otimes_e}(m, (f \star g)) \geqslant \mathbf{I}_{\otimes_e}(m, f) \star \mathbf{I}_{\otimes_e}(m, g)$$

holds.

4 Pseudo-Integral

Let $[a,b]$ be a closed (in some cases semiclosed) subinterval of $[-\infty,\infty]$. The full order on $[a,b]$ will be denoted by $\preceq$. This can be the usual order of the real line, but it can be another order. The operation $\oplus$ (pseudo-addition) is a commutative, non-decreasing (with respect to $\preceq$), associative function $\oplus : [a,b] \times [a,b] \to [a,b]$ with a zero (neutral) element denoted by $\mathbf{0}$. Denote $[a,b]_+ = \{x : x \in [a,b], \mathbf{0} \preceq x\}$. The operation $\odot$ (pseudo-multiplication) is a function $\odot : [a,b] \times [a,b] \to [a,b]$ which is commutative, positively non-decreasing, i.e., $x \preceq y$ implies $x \odot z \preceq y \odot z$, $z \in [a,b]_+$, associative and for which there exist a unit element $\mathbf{1} \in [a,b]$, i.e., for each $x \in [a,b]$, $1 \odot x = x$. We assume $\mathbf{0} \odot x = \mathbf{0}$ and that $\odot$ is distributive over $\oplus$, i.e.,

$$x \odot (y \oplus z) = (x \odot y) \oplus (x \odot z)$$

The structure $([a,b], \oplus, \odot)$ is called a *semiring* (see [23, 36]). We will consider only semirings with the following continuous operations:

Case I: The pseudo-addition is idempotent operation and the pseudo-multiplication is not.

(a) $x \oplus y = \sup(x,y)$, $\odot$ is arbitrary not idempotent pseudo-multiplication on the interval $[a,b]$ (or $[a,b)$). We have $\mathbf{0} = a$ and the idempotent operation sup induces a full order in the following way: $x \preceq y$ if and only if $\sup(x,y) = y$.

(b) $x \oplus y = \sup(x,y)$, $\odot$ is arbitrary not idempotent pseudo-multiplication on the interval $[a,b]$ (or $(a,b]$). We have $\mathbf{0} = b$ and the idempotent operation inf induces a full order in the following way: $x \preceq y$ if and only if $\inf(x,y) = y$.

Case II: The pseudo-operations are defined by a monotone and continuous function $g : [a,b] \to [0,\infty]$, i.e., pseudo-operations are given with

$$x \oplus y = g^{-1}(g(x) + g(y)) \text{ and } x \odot y = g^{-1}(g(x) \cdot g(y)).$$

If the zero element for the pseudo-addition is a, we will consider increasing generators. Then $g(a) = 0$ and $g(b) = \infty$. If the zero element for the pseudo-addition is b, we will consider decreasing generators. Then $g(b) = 0$ and $g(a) = \infty$.

If the generator g is increasing (decreasing), the operation $\oplus$ induces the usual order (opposite to the usual order) on the interval $[a,b]$ in the following way: $x \preceq y$ if and only if $g(x) \leq g(y)$.

Case III: Both operations are idempotent. We have

(a) $x \oplus y = \sup(x,y)$, $x \odot y = \inf(x,y)$, on the interval $[a,b]$. We have $\mathbf{0} = a$ and $\mathbf{1} = b$. The idempotent operation sup induces the usual order ($x \preceq y$ if and only if $\sup(x,y) = y$).

(b) $x \oplus y = \inf(x,y)$, $x \odot y = \sup(x,y)$, on the interval $[a,b]$. We have $\mathbf{0} = b$ and $\mathbf{1} = a$. The idempotent operation inf induces an order opposite to the usual order ($x \preceq y$ if and only if $\inf(x,y) = y$).

In order to present the Hölder and Minkowski integral inequalities for the pseudo-integral, it is necessary to introduce the pseudo-power. For $x \in [a,b]_+$ and $p \in]0,\infty[$, the pseudo-power $x_{\odot}^{(p)}$ is defined in the following way. If $p = n$ is an integer then $x_{\odot}^{(n)} = \underbrace{x \odot x \odot \cdots \odot x}_{n}$. Moreover, $x_{\odot}^{(\frac{1}{n})} = \sup\left\{y \mid y_{\odot}^{(n)} \leq x\right\}$. Then $x_{\odot}^{(\frac{m}{n})} = x_{\odot}^{(r)}$ is well defined for any rational $r \in]0,\infty[$, independently of representation $r = \frac{m}{n} = \frac{m_1}{n_1}$, m, n, m_1, n_1 being positive integers (the result follows from the continuity and monotonicity of $\odot$). Due to continuity of $\odot$, if $p \in]0,\infty[$ is not rational, then

$$x_{\odot}^{(p)} = \sup\left\{x_{\odot}^{(r)} \mid r \in]0,p[,\ r \text{ is rational}\right\}.$$

Evidently, if $x \odot y = g^{-1}(g(x) \cdot g(y))$, then $x_{\odot}^{(p)} = g^{-1}(g^p(x))$. On the other hand, if $\odot$ is idempotent, then $x_{\odot}^{(p)} = x$ for any $x \in [a,b]$ and $p \in]0,\infty[$.

Let $(X, \mathscr{A})$ be a measurable space. A set function $m : \mathscr{A} \to [a,b]$ is a σ-$\oplus$-measure if there hold

(i) $m(\varnothing) = \mathbf{0}$ (if $\oplus$ is not idempotent),
(ii)$m\left(\bigcup_{i=1}^{\infty} A_i\right) = \bigoplus_{i=1}^{\infty} m(A_i)$ holds for any sequence $(A_i)_{i\in\mathbb{N}}$ of pairwise disjoint sets from $\mathscr{A}$.

We suppose that $([a,b], \oplus)$ and $([a,b], \odot)$ are complete lattice ordered semigrups. We suppose that $[a,b]$ is endowed with a metric d compatible with sup and inf, i.e. $\limsup x_n = x$ and $\liminf x_n = x$, imply $\lim_{n\to\infty} d(x_n, x) = 0$, and which satisfies at least one of the following conditions:

(a) $d(x \oplus y, x' \oplus y') \leq d(x,x') + d(y,y')$
(b) $d(x \oplus y, x' \oplus y') \leq \max\{d(x,x'), d(y,y')\}$.

Let f and h be two functions defined on X and with values in $[a,b]$. We define for any $x \in X$ for functions f and g that $(f \oplus g)(x) = f(x) \oplus g(x)$ and $(f \odot g)(x) = f(x) \odot g(x)$, and for any $\lambda \in [a,b]$ $(\lambda \odot f)(x) = \lambda \odot f(x)$.

The pseudo-characteristic function with values in a semiring is defined with

$$\chi_A(x) = \begin{cases} \mathbf{0}, & x \notin A \\ \mathbf{1}, & x \in A \end{cases}.$$

A step (measurable) function is a mapping $e : X \to [a,b]$ that has the following representation $e = \bigoplus_{i=1}^{n} a_i \odot \chi_{A_i}$ for $a_i \in [a,b]$ and sets $A_i \in \mathscr{A}$ are pairwise disjoint if $\oplus$ is nonidempotent.

Let ε be a positive real number, and $B \subset [a,b]$. A subset $\{l_i^{\varepsilon}\}_{n\in\mathbb{N}}$ of B is a ε-net in B if for each $x \in B$ there exists l_i^{ε} such that $d(l_i^{\varepsilon}, x) \leqslant \varepsilon$. If we have $l_i^{\varepsilon} \preceq x$, than we call $\{l_i^{\varepsilon}\}$ a lower ε-net. If $l_i^{\varepsilon} \preceq l_{i+1}^{\varepsilon}$ holds, than $\{l_i^{\varepsilon}\}$ is monotone, for more details see [33, 36].

Let $m : \mathscr{A} \to [a,b]$ be a $\oplus$-measure.

(i) The pseudo-integral of a step function $e : X \to [a,b]$ is defined by

$$\int_X^{\oplus} e \odot dm = \bigoplus_{i=1}^{n} a_i \odot m(A_i).$$

(ii)The pseudo-integral of a measurable function $f : X \to [a,b]$, (if $\oplus$ is not idempotent we suppose that for each $\varepsilon > 0$ there exists a monotone ε-net in $f(X)$) is defined by

$$\int_X^{\oplus} f \odot dm = \lim_{n\to\infty} \int_X^{\oplus} e_n(x) \odot dm,$$

where $(e_n)_{n\in\mathbb{N}}$ is a sequence of step functions such that $d(e_n(x), f(x)) \to 0$ uniformly as $n \to \infty$.

4.1 Two Important Special Cases

If the pseudo-operations are defined by a monotone and continuous function $g:[a,b]\to[0,\infty]$, the pseudo-integral for a measurable function $f:X\to[a,b]$ is given by,

$$\int_X^{\oplus} f\odot dm = g^{-1}\left(\int_X (g\circ f)\,d\,(g\circ m)\right), \tag{1}$$

where the integral applied on the right side is the standard Lebesgue integral. In special case, when $X=[c,d]$, $\mathscr{A}=\mathscr{B}(X)$ and $m=g^{-1}\circ\lambda$, λ the standard Lebesgue measure on $[c,d]$, then the pseudo-integral reduces on g -integral. Therefore, due to (1) we have

$$\int_{[c,d]}^{\oplus} f dx = g^{-1}\left(\int_c^d g\,(f\,(x))\,dx\right).$$

When the semiring is of the form $([a,b],\sup,\odot)$, cases I(a) and III(a) from section 4 we shall consider complete sup-measure m only and $\mathscr{A}=2^X$, i.e., for any system $(A_i)_{i\in I}$ of measurable sets, $m\left(\bigcup_{i\in I}A\right)=\sup_{i\in I}m\,(A_i)$. Recall that if X is countable (especially, if X is finite) then any σ-sup-measure m is complete and, moreover, $m\,(A)=\sup_{x\in A}\psi\,(x)$, where $\psi:X\to[a,b]$ is a density function given by $\psi\,(x)=m\,(\{x\})$. Then the pseudo-integral for a function $f:X\to[a,b]$ is given by

$$\int_X^{\oplus} f\odot dm = \sup_{x\in X}\,(f\,(x)\odot\psi\,(x)),$$

where function ψ defines sup-measure m.

In [29] is shown that any sup-measure generated as essential supremum of a continuous density can be obtained as a limit of pseudo-additive measures with respect to generated pseudo-addition.

Theorem 4. *Let m be a* sup*-measure on $([0,\infty],\mathscr{B}([0,\infty]))$, where $\mathscr{B}([0,\infty])$ is the Borel σ-algebra on $[0,\infty]$,*

$$m\,(A)=\operatorname{essup}_{\mu}\,(\psi\,(x):x\in A), \tag{2}$$

and $\psi:[0,\infty]\to[0,\infty]$ is a continuous density. Then for any pseudo-addition $\oplus$ with a generator g there exists a family $\{m_\lambda\}$ of $\oplus_\lambda$-measure on $([0,\infty[,\mathscr{B})$, where $\oplus_\lambda$ is generated by g^λ (the function g of the power λ), $\lambda\in\,]0,\infty[$, such that

$$\lim_{\lambda\to\infty} m_\lambda = m.$$

For any continuous function $f:[0,\infty]\to[0,\infty]$ the integral $\int^{\oplus} f\odot dm$ can be obtained as a limit of g-integrals, ([29]).

Theorem 5. *Let* $([0,\infty],\sup,\odot)$ *be a semiring with* $\odot$ *generated by the generator* g. *Let* m *be the same as in Theorem 4. Then exists a family* $\{m_\lambda\}$ *of* $\oplus_\lambda$*-measures, where* $\oplus_\lambda$ *is generated by* g^λ, $\lambda \in]0,\infty[$, *such that for every continuous function* $f:[0,\infty]\to[0,\infty]$

$$\int^{\sup} f \odot dm = \lim_{\lambda\to\infty} \int^{\oplus_\lambda} f \odot_\lambda dm_\lambda .$$

In the following we will give the Hölder, Minkowski, Jensen and Chebyshev type inequaities for important cases of the pseudo-integral.

Since the cases I(b) and III(b) are linked to the cases I(a) and III(a) by duality, all inequaities for the pseudo-integral related to the cases I(a) and III(a) can be transformed into inequaities for pseudo-integral related to the cases I(b) and III(b).

4.2 Hölder's Inequality for Pseudo-integral

Now we shall give a generalization of the classical Hölder inequality on the semiring with generated pseudo-operations based on [5].

Recall that if p and q are positive real number such that $\frac{1}{p}+\frac{1}{q}=1$, then p and q is a pair of conjugate exponents.

Theorem 6. *Let* p *and* q *be conjugate exponents,* $1<p<\infty$. *For a given measurable space* $(X,\mathscr{A})$ *let* $u,v:X\to[a,b]$ *be two measurable functions and let a generator* $g:[a,b]\to[0,\infty]$ *of the pseudo-addition* $\oplus$ *and the pseudo-multiplication* $\odot$ *be an increasing function. Then for any* σ*-*$\oplus$*-measure* m *it holds:*

$$\int_X^{\oplus} (u\odot v)\odot dm \leqslant \left(\int_X^{\oplus} u_{\odot}^{(p)}\odot dm\right)_{\odot}^{\left(\frac{1}{p}\right)} \odot \left(\int_X^{\oplus} v_{\odot}^{(q)}\odot dm\right)_{\odot}^{\left(\frac{1}{q}\right)} .$$

Example 1. (i) Let $[a,b]=[0,\infty]$ and $g(x)=x^\alpha$ for some $\alpha\in[1,\infty[$. The corresponding pseudo-operations are $x\oplus y=\sqrt[\alpha]{x^\alpha+y^\alpha}$ and $x\odot y=xy$. Then the inequality from *Theorem 6* reduces on the following inequality

$$\sqrt[\alpha]{\int_{[c,d]} u(x)^\alpha v(x)^\alpha\,dx} \leqslant \sqrt[p\alpha]{\int_{[c,d]} u(x)^{p\alpha}\,dx}\ \sqrt[q\alpha]{\int_{[c,d]} v(x)^{q\alpha}\,dx}.$$

(ii) Let $[a,b]=[0,\infty]$ and $g(x)=x^\alpha$ for some $\alpha\in[1,\infty[$. The corresponding pseudo-operations are $x\oplus y=\sqrt[\alpha]{x^\alpha+y^\alpha}$ and $x\odot y=xy$. Then the inequality from *Theorem 6* reduces on the following inequality

$$\sqrt[\alpha]{\int_{[c,d]} u(x)^\alpha v(x)^\alpha\,dx} \leqslant \sqrt[p\alpha]{\int_{[c,d]} u(x)^{p\alpha}\,dx}\ \sqrt[q\alpha]{\int_{[c,d]} v(x)^{q\alpha}\,dx}.$$

Let $x \oplus y = \sup(x,y)$ and $x \odot y = g^{-1}(g(x)g(y))$. As a consequence of the Hölder type theorem for a complete sup-measure the following result holds for general σ-sup-measure by [5].

Theorem 7. *Let $\odot$ be represented by an increasing generator g and m be σ-sup-measure. Let p and q be conjugate exponents with $1 < p < \infty$. Then for any measurable functions $u, v : X \to [a,b]$, it holds:*

$$\int_X^{\sup} (u \odot v) \odot dm \leq \left[\int_X^{\sup} u_{\odot}^{(p)} \odot dm\right]_{\odot}^{\left(\frac{1}{p}\right)} \odot \left[\int_X^{\sup} v_{\odot}^{(q)} \odot dm\right]_{\odot}^{\left(\frac{1}{q}\right)}.$$

In the case III(a) and $p > 0$, $x_{\odot}^{(p)} = x$, the Hölder inequality reduces to the inequality

$$\int_X^{\sup} (u \odot v) \odot dm \leq \left(\int_X^{\sup} u \odot dm\right) \odot \left(\int_X^{\sup} v \odot dm\right),$$

which trivially holds because of distributivity of sup and inf.

Example 2. Let $[a,b] = [-\infty, \infty]$ and g generating $\odot$ be given by $g(x) = e^x$. Then

$$x \odot y = x + y,$$

and Hölder type inequality from *Theorem 7* reduces on

$$\sup_{x \in X} (u(x) + v(x) + \psi(x))$$

$$\leqslant \frac{1}{p} \sup_{x \in X} (p \cdot u(x) + \psi(x)) + \frac{1}{q} \sup_{x \in X} (q \cdot v(x) + \psi(x))$$

where u, v, ψ are arbitrary real function on X.

4.3 Minkowski's Inequality for Pseudo-integral

In [5] is given Minkowski's inequality for three cases of the pseudo-integrals. The following inequality holds for the case II and corresponding pseudo-integral.

Theorem 8. *Let $u, v : X \to [a,b]$ be two measurable functions and $p \in [1, \infty[$. If an additive generator $g : [a,b] \to [0, \infty]$ of the pseudo-addition $\oplus$ and the pseudo-multiplication $\odot$ are increasing. Then for any σ-$\oplus$-measure m it holds:*

$$\left(\int_X^{\oplus} (u \oplus v)_{\odot}^{(p)} \odot dm\right)_{\odot}^{\left(\frac{1}{p}\right)}$$

$$\leqslant \left(\int_X^{\oplus} u_{\odot}^{(p)} \odot dm\right)_{\odot}^{\left(\frac{1}{p}\right)} \oplus \left(\int_X^{\oplus} v_{\odot}^{(p)} \odot dm\right)_{\odot}^{\left(\frac{1}{p}\right)}.$$

If we observe semiring $([a,b],\sup,\inf)$, then the corresponding inequality means (recall that $x_{\odot}^{(p)} = x$ for each $x \in [a,b]$, $p > 0$)

$$\int_X^{\oplus} (u \oplus v) \odot dm \leqslant \sup\left(\int_X^{\oplus} u \odot dm, \int_X^{\oplus} v \odot dm\right).$$

In the case I(a) also Minkowski type inequality holds.

Theorem 9. *Let $\odot$ be represented by an increasing generator g, m be a complete* sup*-measure and $p \in]0,\infty[$. Then for any functions $u,v : X \to [a,b]$, it holds:*

$$\left(\int_X^{\sup} (\sup(u,v))_{\odot}^{(p)} \odot dm\right)_{\odot}^{\left(\frac{1}{p}\right)}$$

$$= \sup\left(\left(\int_X^{\sup} u_{\odot}^{(p)} \odot dm\right)_{\odot}^{\left(\frac{1}{p}\right)}, \left(\int_X^{\sup} v_{\odot}^{(p)} \odot dm\right)_{\odot}^{\left(\frac{1}{p}\right)}\right).$$

4.4 Jensen Inequality for Pseudo-integral

Due to [42] we have the following generalization Jensen inequality for two cases of the pseudo-integral.

Theorem 10. *Let $(X,\mathscr{A})$ be a measurable space, m be a σ-$\oplus$-measure and $m(X) =$* **1**. *Let a generator g of the pseudo-addition $\oplus$ and the pseudo-multiplication $\odot$ is a convex and increasing function. If $f : X \to [a,b]$ is a measurable function such that $\int_X^{\oplus} f \odot dm < b$ and Φ is a convex and nonincreasing function on a subinterval of $[a,b]$ containing the range of f, with values in $[a,b]$, then we have*

$$\Phi\left(\int_X^{\oplus} f \odot dm\right) \leqslant \int_X^{\oplus} (\Phi \circ f) \odot dm.$$

Example 3. (i) Let $g(x) = x^{\alpha}$ for some $\alpha \in [1,\infty[$. The corresponding pseudo-operations are $x \oplus y = \sqrt[\alpha]{x^{\alpha} + y^{\alpha}}$ and $x \odot y = xy$. Then the inequality from *Theorem* 10 has the following form

$$\Phi\left(\sqrt[\alpha]{\int_{[0,1]} f(x)^{\alpha}\, dx}\right) \leqslant \sqrt[\alpha]{\int_{[0,1]} \Phi(f(x))^{\alpha}\, dx}.$$

(ii)Let $g(x) = e^x$. The corresponding pseudo-operations are $x \oplus y = \ln(e^x + e^y)$ and $x \odot y = x + y$. Then we have the following inequality

$$\Phi\left(\ln \int_{[0,1]} e^{f(x)}\, dx\right) \leqslant \ln\left(\int_{[0,1]} e^{\Phi(f(x))}\, dx\right).$$

Now we consider the case when $\oplus = \max$, and $\odot = g^{-1}(g(x)g(y))$. From Theorem 5 and the previous theorem it follows the next theorem.

Theorem 11. *Let the pseudo-multiplication $\odot$ is represented by a convex and increasing generator g, μ be the usual Lebesgue measure on $\mathbb{R}$ and m be a* sup-*measure on $([c,d],\mathscr{B}([c,d]))$ defined by (2) where $\psi : [c,d] \to [0,\infty]$ is a continuous density and $m([c,d]) = \mathbf{1}$. If $f : [c,d] \to [0,\infty]$ is continuous function such that pseudo-integral $\int_{[c,d]}^{\sup} f \odot dm$ is finite and Φ is a convex and non-increasing function on the range of f, then it holds:*

$$\Phi\left(\int_{[c,d]}^{\sup} f \odot dm\right) \leqslant \int_{[c,d]}^{\sup} (\Phi \circ f) \odot dm.$$

Example 4. Using Example 3(ii) we have that $g^{\lambda}(x) = e^{\lambda x}$. Then

$$\lim_{\lambda\to\infty} \frac{1}{\lambda} \ln\left(e^{\lambda x} + e^{\lambda y}\right) = \max(x,y),$$

and

$$x \odot_{\lambda} y = x + y.$$

Therefore Jensen type inequality from *Theorem* 11 reduces on

$$\Phi\left(\sup(f(x) + \psi(x))\right) \leqslant \sup\left(\Phi(f(x)) + \psi(x)\right),$$

where ψ is a density function related to m..

4.5 Chebyshev's Inequality for Pseudo-integral

The Chebyshev type inequality for pseudo-integral is studied in [1, 41]. Let in the following theorems $([c,d],\mathscr{A})$ be a measure space and m be σ-$\oplus$ -measure such that $m([c,d]) = \mathbf{1}$.

Theorem 12. *Let $f_1, f_2 : [c,d] \to [a,b]$ be measurable functions. If an additive generator g of the pseudo-addition $\oplus$ and pseudo-multiplication $\odot$ is an increasing function and f_1, f_2 are either both increasing or both decreasing, then*

$$\int_{[c,d]}^{\oplus} f_1 \odot dm \odot \int_{[c,d]}^{\oplus} f_2 \odot dm \leqslant \int_{[c,d]}^{\oplus} (f_1 \odot f_2) \odot dm.$$

In special case, when $[c,d] = [0,1]$, $\mathscr{A} = \mathscr{B}([0,1])$ and $m = g^{-1} \circ \lambda$, λ the standard Lebesgue measure on $[0,1]$, we have the inequality from [41].

Example 5. (i) Let $g(x) = x^\alpha$ for some $\alpha \in [1,\infty)$. The corresponding pseudo-operations are $x \oplus y = \sqrt[\alpha]{x^\alpha + y^\alpha}$ and $x \odot y = xy$. Then by the previous theorem we obtain the following inequality

$$\sqrt[\alpha]{\int_{[0,1]} f_1(x)^\alpha \, dx} \sqrt[\alpha]{\int_{[0,1]} f_2(x)^\alpha \, dx} \leqslant \sqrt[\alpha]{\int_{[0,1]} f_1(x)^\alpha f_2(x)^\alpha dx}.$$

(ii) Let $g(x) = e^x$. The corresponding pseudo-operations are $x \oplus y = \ln(e^x + e^y)$ and $x \odot y = x + y$. Then by the previous theorem we obtain the following inequality

$$\ln \int_{[0,1]} e^{f_1(x)} \, dx + \ln \int_{[0,1]} e^{f_2(x)} \, dx \leqslant \ln \left(\int_{[0,1]} e^{f_1(x)+f_2(x)} dx \right).$$

Now, we consider the second case, when $\oplus = \max$, and $\odot = g^{-1}(g(x)g(y))$. Due to Theorem 5 and the previous theorem the following holds:

Theorem 13. *Let $\odot$ is represented by an increasing multiplicative generator g and m be the same as in Theorem 11. Then for any continuous functions $f_1, f_2 : [c,d] \to [a,b]$, which are either both increasing or both decreasing, holds:*

$$\int_{[c,d]}^{\sup} f_1 \odot dm \odot \int_{[c,d]}^{\sup} f_2 \odot dm \leqslant \int_{[c,d]}^{\sup} (f_1 \odot f_2) \odot dm.$$

Example 6. Using Example 5(ii) we have that $g^\lambda(x) = e^{\lambda x}$. Then

$$\lim_{\lambda \to \infty} \frac{1}{\lambda} \ln \left(e^{\lambda x} + e^{\lambda y} \right) = \max(x, y),$$

and

$$x \odot_\lambda y = x + y.$$

Therefore, Chebyshev type inequality from *Theorem* 13 reduces on

$$\sup(f_1(x) + \psi(x)) + \sup(f_2(x) + \psi(x)) \leqslant \sup(f_1(x) + f_2(x) + \psi(x)),$$

where ψ is a density function related to m.

4.6 Applications

4.6.1 Pseudo-probability

The pseudo-probability is a generalization of the classical probability. In an analogous way as in the probability theory, see [11], we will introduced the corresponding notions in the framework of the σ-$\oplus$-measure and the pseudo-integral.

Let $([a,b], \oplus, \odot)$ be a semiring and $(X, \mathscr{A})$ be a measurable space.

The pseudo-probability $\mathbb{P}$ is σ-$\oplus$-measure on $(X, \mathscr{A})$ satisfying $\mathbb{P}(X) = \mathbf{1}$.

Specially, if we observe case II and P is a classical probability then $\mathbb{P} = g^{-1} \circ P$ is distorted probability, see [13].

The function $Y : X \to [a,b]$ is a pseudo-random variable if for any $y \in [a,b]$ it holds

$$\{\omega \in X \mid Y(\omega) \prec y\} = \{Y \prec y\} \in \mathscr{A}.$$

The pseudo-expectation of the pseudo-variable Y is introduced by

$$\mathbb{E}(Y) = \int_X^{\oplus} Y \odot d\mathbb{P}.$$

If the pseudo-expectation of the pseudo-random variable Y has a finite value in the sense of a given semiring, i.e., if the operation $\oplus$ induces the usual order (opposite to the usual order) on the interval $[a,b]$ it means that $\mathbb{E}(Y) < b$, $(\mathbb{E}(Y) > a)$, then Y is integrable.

Due to definition of the pseudo-expectation, we have

$$\mathbb{E}(f(Y)) = \int_X^{\oplus} f(Y) \odot d\mathbb{P} \tag{3}$$

for any measurable function $f : [a,b] \to [a,b]$.

As consequences of (3) and the inequalities valid for pseudo-integral, with the same assumptions as in the corresponding inequalities type theorems the following hold:

(i) By Hölder's inequality we have

$$E(Y \odot Z) \preceq \left(E\left((Y)_{\odot}^{(p)}\right)\right)_{\odot}^{\left(\frac{1}{p}\right)} \odot \left(E(Z)_{\odot}^{(q)}\right)_{\odot}^{\left(\frac{1}{q}\right)}, \tag{4}$$

(ii) By Minkowski's inequality we have

$$\left(E\left((Y \oplus Z)_{\odot}^{(p)}\right)\right)_{\odot}^{\left(\frac{1}{p}\right)} \preceq \left(E\left((Y)_{\odot}^{(p)}\right)\right)_{\odot}^{\left(\frac{1}{p}\right)} \oplus \left(E\left((Z)_{\odot}^{(p)}\right)\right)_{\odot}^{\left(\frac{1}{p}\right)}, \tag{5}$$

(iii) By Jensen's inequality we have

$$\Phi(\mathbf{E}(Y)) \preceq \mathbf{E}(\Phi(Y)),$$

(iv) By Chebyshev's inequality we have

$$\mathbf{E}(f_1(Y)) \odot \mathbf{E}(f_2(Y)) \preceq \mathbf{E}(f_1(Y) \odot f_2(Y)).$$

4.6.2 Cost Measure and Decision Variable

The duality between probability and optimization was considered in [9]. Hence, there $(\mathbb{R}^+, +, \times)$ is replaced by the semiring $(]-\infty, \infty], \min, +)$. By analogy with

probability theory there cost mesure, decision variable and related notions were introduced. Let the semiring $(]-\infty,\infty], \min, +)$ is denoted by $\mathbb{R}_{\min}$.

Let U be a topological spaces, $\mathscr{U}$ be the set of open sets of U. A set function $\mathbb{K}: \mathscr{U} \to \mathbb{R}_{\min}$ is a cost measure if there hold

(i) $\mathbb{K}(\emptyset) = \infty$,
(ii) $\mathbb{K}(U) = 0$,
(iii) $\mathbb{K}(\bigcup_n A_n) = \inf_n \mathbb{K}(A_n)$ za sve A_n iz $\mathscr{U}$.

The triplet $(U, \mathscr{U}, \mathbb{K})$ is called a decision space.

A function $c: U \to \mathbb{R}_{\min}$ such that $\mathbb{K}(A) = \inf_{u \in A} c(u)$ for any $A \in \mathscr{U}$ is a cost density of the cost measure $\mathbb{K}$.

A l.s.c. function $c: U \to \mathbb{R}_{\min}$ such that $\inf_u c(u) = 0$ defines a cost measure on $(U, \mathscr{U})$ by $\mathbb{K}(A) = \inf_{u \in A} c(u)$ ([8]). Also, for any cost measure $\mathbb{K}$ defined on open sets of a topological space with a countable basis of open sets there exists a unique minimal extension $\mathbb{K}_*$ to $\mathscr{P}(U)$ ([21, 26]).

Now we recall the definitions of the decision variables and related notions [9].

The mapping $Y: U \to E$ is a decision variables on $(U, \mathscr{U}, \mathbb{K})$, where E is a topological space with a countable basis of open set. It induces a cost measure $\mathbb{K}_Y$ on $(E, \mathscr{B})$ (where $\mathscr{B}$ denotes the set of open sets of E) defined by $\mathbb{K}_Y(A) = \mathbb{K}_*\left(Y^{-1}(A)\right)$ for all $A \in \mathscr{B}$. The cost measure $\mathbb{K}_Y$ has a l.s.c. density denoted c_Y. When $E = \mathbb{R}_{\min}$ a decision variable Y is called a cost variable.

The value of a cost variable Y is defined by

$$\mathbb{V}(Y) = \inf_x (x + c_Y(x)).$$

The convergence of decision variables, law of large numbers, Bellman chains and processes are also considered in [9].

Remark 1. The cost measure, decision variables and related notions can also be defined in a general idempotent semiring, see [8].

Notice that the value of a cost variable Y is defined by the pseudo-integral with respect to σ-$\oplus$-measure $\mathbb{K}_*$, i.e.,

$$\mathbb{V}(Y) = \int_U^{\inf} Y \odot d\mathbb{K}_*.$$

Let Y and Z be decision variables on $(U, \mathscr{U}, \mathbb{K})$. Due to the previous notations the inequalities (4) and (5) have the following forms:

(i) For Hölder's inequality

$$\mathbb{V}(Y + Z) \geq \frac{1}{p}\mathbb{V}(pY) + \frac{1}{q}\mathbb{V}(qZ),$$

where p and q are conjugate exponents, $1 < p < \infty$,

(ii) For Minkowski's inequality

$$\frac{1}{p}\mathbb{V}(p\inf(Y,Z)) = \inf\left(\frac{1}{p}\mathbb{V}(pY), \frac{1}{p}\mathbb{V}(pZ)\right),$$

where $p \in]0,\infty[$.

In [14] the semiring $([-\infty,\infty), \sup, +)$ are considered and there are given the corresponding inequalities related to idempotent integral introduced in [24]. These inequalities also have applications in decision theory. Hence, in this case Hölder's and Minkowski's inequalities have the following forms:

$$\mathbb{E}(Y+Z) \leq \frac{1}{p}\mathbb{E}(pY) + \frac{1}{q}\mathbb{E}(qZ),$$

where p and q are conjugate exponents, $0 < p \leq q < \infty$ and

$$\frac{1}{p}\mathbb{E}(p\sup(Y,Z)) = \sup\left(\frac{1}{p}\mathbb{E}(pY), \frac{1}{p}\mathbb{E}(pZ)\right),$$

where $p \in (0,\infty)$ and $\mathbb{E}(Y)$ is the value of a decision variables (in the sense of a given semiring) defined by sup-integral (see [24]).

5 Conclusion

We have given the generalizations of the mostly used integral inequalities for non-additive integrals (Choquet integral, Sugeno integral, universal integral, pseudo integral). The future work will be the investigation of new applications of obtained results in many fields.

Acknowledgements. The authors are supported in part by the Project MPNRS 174009, and by the project *Mathematical models of intelligent systems and their applications* of Academy of Sciences and Arts of Vojvodina supported by Provincial Secretariat for Science and Technological Development of Vojvodina.

References

1. Agahi, H., Mesiar, R., Ouyang, Y.: General Minkowski type inequalities for Sugeno integrals. Fuzzy Sets and Systems 161, 708–715 (2010)
2. Agahi, H., Mesiar, R., Ouyang, Y.: New general extensions of Chebyshev type inequalities for Sugeno integrals. Int. J. of Approximate Reasoning 51, 135–140 (2009)
3. Agahi, H., Mesiar, R., Ouyang, Y.: Chebyshev type inequalities for pseudo-integrals. Nonlinear Analysis: Theory, Methods and Applications 72, 2737–2743 (2010)
4. Agahi, H., Mesiar, R., Ouyang, Y., Pap, E., Štrboja, M.: Berwald type inequality for Sugeno integral. Applied Mathematics and Computation 217, 4100–4108 (2010)

5. Agahi, H., Mesiar, R., Ouyang, Y., Pap, E., Štrboja, M.: Hölder and Minkowski type inequalities for pseudo-integral. Applied Mathematics and Computation 217(21), 8630–8639 (2011)
6. Agahi, H., Mesiar, R., Ouyang, Y., Pap, E., Štrboja, M.: General Chebyshev type inequalities for universal integral. Information Sciences 107, 171–178 (2012)
7. Akian, M.: Theory of cost measures: convergence of decision variables. INRIA Report (1995)
8. Akian, M.: Densities of idempotent measures and large deviations. Transactions of the American Mathematical Society 351(11), 4515–4543 (1999)
9. Akian, M., Quadrat, J.P., Viot, M.: Duality between Probability and Optimization. In: Gunawardena, J. (ed.) Idempotency. Publication of the Isaac Newton Institute. Cambridge University Press (1998)
10. Benvenuti, P., Mesiar, R., Vivona, D.: Monotone set functions-based integrals. In: Pap, E. (ed.) Handbook of Measure Theory, vol. II, pp. 1329–1379. Elsevier (2002)
11. Billingsley, P.: Probability and Measure. John Wiley & Sons, New York (1995)
12. Bullen, P.S.: Handbook of Means and Their Inequaliies. Kluwer Academic Publishers, Dordrecht (2003)
13. Chateauneuf, A.: Decomposable Measures, Distorted Probabilities and Concave Capacities. Mathematical Social Sciences 31, 19–37 (1996)
14. Del Moral, P.: Résolution particulaire des problèmes d'estimation et d'optimisation nonlinéaires. Thesis dissertation, Université Paul Sabatier, Toulouse (1994)
15. Denneberg, D.: Non-additive measure and integral. Kluwer Academic Publishers, Dordrecht (1994)
16. Flores-Franulič, A., Román-Flores, H.: A Chebyshev type inequality for fuzzy integrals. Applied Mathematics and Computation 190, 1178–1184 (2007)
17. Grabisch, M., Marichal, J.L., Mesiar, R., Pap, E.: Aggregation Functions. Encyclopedia of Mathematics and Its Applications, vol. 127. Cambridge University Press (2009)
18. Klement, E.P., Mesiar, R., Pap, E.: Triangular norms. Trends in Logic. Studia Logica Library, vol. 8. Kluwer Academic Publishers, Dodrecht (2000)
19. Klement, E.P., Mesiar, R., Pap, E.: A universal integral as Common Frame for Choquet and Sugeno Integral. IEEE Trans. on Fuzzy Systems 18(1), 178–187 (2010)
20. Kolokoltsov, V.N., Maslov, V.P.: Idempotent calculus as the apparatus of optimization theory. I, Functional. Anal. i Prilozhen 23(1), 1–14 (1989)
21. Kolokoltsov, V.N., Maslov, V.P.: The general form of the endomorphisms in the space of continuous functions with values in a numerical commutative semiring (with the operation $\oplus$ = max). Soviet Math. Dokl. 36(1), 55–59 (1988)
22. Kolokoltsov, V.N., Maslov, V.P.: Idempotent Analysis and Its Applications. Kluwer Academic Publishers, Dordrecht (1997)
23. Kuich, W.: Semirings, Automata, Languages. Springer, Berlin (1986)
24. Maslov, V.: Méthodes Opératorielles. Editions MIR, Moscow (1987)
25. Maslov, V.P., Samborskij, S.N. (eds.): Idempotent Analysis. Advances in Soviet Mathematics, vol. 13. Amer. Math. Soc., Providence (1992)
26. Maslov, V.P., Kolokoltsov, V.N.: Idempotent analysis and its applications to optimal control theory, Moscow, Nauka (1994) (in Russian)
27. Mesiar, R., Li, J., Pap, E.: The Choquet integral as Lebesgue integral and related inequalities. Kybernetika 46, 1098–1107 (2010)
28. Mesiar, R., Ouyang, Y.: General Chebyshev type inequalities for Sugeno integrals. Fuzzy Sets and Systems 160, 58–64 (2009)
29. Mesiar, R., Pap, E.: Idempotent integral as limit of g -integrals. Fuzzy Sets and Systems 102, 385–392 (1999)

30. Ouyang, Y., Fang, J., Wang, L.: Fuzzy Chebyshev type inequality. Int. J. of Approximate Reasoning 48, 829–835 (2008)
31. Ouyang, Y., Mesiar, R., Agahi, H.: An inequality related to Minkowski type for Sugeno integrals. Information Sciences 180(14), 2793–2801 (2010)
32. Ouyang, Y., Mesiar, R.: On the Chebyshev type inequality for seminormed fuzzy integral. Applied Mathematics Letters 22, 1810–1815 (2009)
33. Pap, E.: An integral generated by decomposable measure. Univ. u Novom Sadu Zb. Rad. Prirod.-Mat. Fak. Ser. Mat. 20(1), 135–144 (1990)
34. Pap, E.: g-calculus. Univ. u Novom Sadu Zb. Rad. Prirod.-Mat. Fak. Ser. Mat. 23(1), 145–156 (1993)
35. Pap, E.: Applications of decomposable measures. In: Höhle, U., Rodabaugh, R.S. (eds.) Handbook Mathematics of Fuzzy Sets-Logic, Topology and Measure Theory, pp. 675–700. Kluwer Academic Publishers (1999)
36. Pap, E.: Null-Additive Set Functions. Kluwer Academic Publishers, Dordrecht (1995)
37. Pap, E.: Decomposable measures and nonlinear equations. Fuzzy Sets and Systems 92, 205–222 (1997)
38. Pap, E. (ed.): Handbook of Measure Theory. Elsevier Science, Amsterdam (2002)
39. Pap, E.: Pseudo-Additive Measures and Their Aplications. In: Pap, E. (ed.) Handbook of Measure Theory, vol. II, pp. 1403–1465. Elsevier (2002)
40. Pap, E.: Pseudo-analysis approach to nonlinear partial differential equations. Acta Polytechnica Hungarica 5, 31–45 (2008)
41. Pap, E., Štrboja, M.: Generalization of the Chebishev Inequality for Pseudo-Integral. In: Proceedings of the 7th International Symposium on Intelligent Systems and Informatics SISY Subotica, IEEE Catalog Number: CFP0984C-CDR, pp. 123–126 (2009)
42. Pap, E., Štrboja, M.: Generalization of the Jensen inequality for pseudo-integral. Information Sciences 180, 543–548 (2010)
43. Román-Flores, H., Flores-Franulič, A., Chalco-Cano, Y.: A Jensen type inequality for fuzzy integrals. Information Sciences 177, 3192–3201 (2007)
44. Sirbiladze, G., Gachechiladze, T.: Restored fuzzy measures in expert decision-making. Information Sciences 169, 71–95 (2005)
45. Sugeno, M.: Theory of fuzzy integrals and its applications. Ph.D. Thesis, Tokyo Institute of Technology (1974)
46. Sugeno, M., Murofushi, T.: Pseudo-additive measures and integrals. J. Math. Anal. Appl. 122, 197–222 (1987)
47. Tsai, H.-H., Lu, I.-Y.: The evalution of service quality using generalized Choquet integral. Information Sciences 176, 640–663 (2006)
48. Wang, Z., Klir, G.J.: Generalized measure theory. Springer, Boston (2009)
49. Wang, R. S.: Some inequalities and convergence theorems for Choquet integrals. J. Appl. Comput. 35(1-2), 305-321 (2011).

Inequalities of Jensen and Chebyshev Type for Interval-Valued Measures Based on Pseudo-integrals

Tatjana Grbić, Slavica Medić, Ivana Štajner-Papuga, and Tatjana Došenović

Abstract. Since interval-valued measures have applications in number of practical areas, this paper is focused on two approaches to this problem as well as on the corresponding generalizations of the Jensen and the Chebyshev integral inequalities. The first approach is based on an interval-valued measure defined by the pseudo-integral of interval-valued function, while the second approach considers an interval-valued measure obtained through pseudo-integrals of real-valued functions.

Keywords: Pseudo-operations, Pseudo-integral, Interval-valued measure, Jensen type inequality, Chebyshev type inequality.

1 Introduction

The interval-valued measures, as the natural generalization of the classical (single-valued) measures, have applications in number of practical areas. Investigation of these specific measures, due to the possible representation of uncertainty by intervals ([10, 13, 17, 28, 30, 35]), can lead in the direction of the economic uncertainty, interval-valued probability, multi-valued fuzzy entropy, fuzzy random variables, fixed point theory, etc.

Tatjana Grbić · Slavica Medić
Faculty of Technical Sciences,
University of Novi Sad, Serbia
e-mail: {tatjana,slavicam}@uns.ac.rs

Ivana Štajner-Papuga
Department of Mathematics and Informatics,
University of Novi Sad, Serbia
e-mail: ivana.stajner-papuga@dmi.uns.ac.rs

Tatjana Došenović
Faculty of Technology,
University of Novi Sad, Serbia
e-mail: tatjanad@tf.uns.ac.rs

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 23–41.
DOI: 10.1007/978-3-642-33959-2_2

Since the problem of generalization of different classical integral inequalities is also an contemporary issue ([1, 2, 3, 4, 6, 11, 25, 29], etc.), the main topic of this paper are two approaches to the interval-valued measures and the corresponding extensions of two well known classical integral inequalities to the interval-valued case. Pseudo-analysis is chosen as the background of this paper due to its applicability in many different areas of mathematics as well as in various practical problems (see [14, 15, 18, 23, 24, 25, 26, 34], etc.).

The first approach is based on the pseudo-integral of interval-valued function that is generalization of the Aumann's integral ([7, 9]). The second approach as a base has an interval-valued measure obtained through pseudo-integrals of real-valued functions. The proposed approach has been investigated in [12] for Choquet integral.

The paper is organized as follows. Section 2 contains preliminary notions, such as pseudo-operations, semiring, pseudo-integral and interval-valued functions. The short overview of results from [1] and [25] is also given in the second section. The first approach to the interval-valued measure is given in the third section. Also, Section 3 contains results concerning the Jensen type and the Chebyshev type integral inequalities from [8, 32, 33] that are now given in the form which corresponds to the presented interval-valued measure. The second approach and corresponding inequalities are given in Section 4. Some concluding remarks are given in the fifth section.

2 Preliminary Notions

The basic preliminary notions needed in this paper are notions of the pseudo-operations and the semiring.

Let $[a,b]$ be a closed subinterval of $[-\infty,+\infty]$ (in some cases semiclosed subintervals will be considered) and let $\preceq$ be a total order on $[a,b]$.

A *semiring* is a structure $([a,b],\oplus,\odot)$ such that the following holds:

- $\oplus$ is a *pseudo-addition*, i.e., a function $\oplus:[a,b]\times[a,b]\to[a,b]$ which is commutative, non-decreasing (with respect to $\preceq$), associative and with a zero element, denoted by $\mathbf{0}$;
- $\odot$ is a *pseudo-multiplication*, i.e., a function $\odot:[a,b]\times[a,b]\to[a,b]$ which is commutative, positively non-decreasing ($x\preceq y$ implies $x\odot z\preceq y\odot z$, where $z\in[a,b]_+=\{x:x\in[a,b],\ \mathbf{0}\preceq x\}$), associative and for which there exists a unit element denoted by $\mathbf{1}$;
- $\mathbf{0}\odot x=\mathbf{0}$;
- $x\odot(y\oplus z)=(x\odot y)\oplus(x\odot z)$.

There are three basic classes of semirings with continuous (up to some points) pseudo-operations. The first class contains semirings with idempotent pseudo-addition and non idempotent pseudo-multiplication. Semirings with strict pseudo-operations defined by the monotone and continuous generator function

$g : [a,b] \to [0,+\infty]$, i.e. g-semirings ([22, 23]), form the second class, and semirings with both idempotent operations belong to the third class.

Total order $\preceq$ is closely connected to the choice of the pseudo-addition.

For the semirings of the first and the third class, i.e., for the semirings with idempotent pseudo-addition, the total order is induced by the following

$$x \preceq y \text{ if and only if } x \oplus y = y.$$

For the semiring of the second class given by a generator g, total order is given by

$$x \preceq y \text{ if and only if } g(x) \leq g(y).$$

Further on, let $([a,b], \oplus, \odot)$ be a semiring and let $([a,b], \oplus)$ and $([a,b], \odot)$ be complete lattice ordered semigrups. Also, let us suppose that the interval $[a,b]$ is endowed with a specific metric d (see [23]).

Of the special interest for this paper are the following semirings of the first and the second class:

Example 1. (a) $([a,b], \sup, \odot)$ and $([a,b], \inf, \odot)$ where $\odot$ is an operation given by a strictly monotone bijection $g : [a,b] \to [0,+\infty]$ as

$$x \odot y = g^{-1}(g(x)g(y));$$

(b) g-semiring, i.e., $([a,b], \oplus, \odot)$ where $\oplus$ and $\odot$ are operations given by a strictly monotone bijection $g : [a,b] \to [0,+\infty]$ as

$$x \oplus y = g^{-1}(g(x) + g(y)) \quad \text{and} \quad x \odot y = g^{-1}(g(x)g(y)).$$

Remark 1. It was proven in [19] that a semiring of the first class, provided that pseudo-multiplication is of the form $x \odot y = g^{-1}(g(x)g(y))$ where g is a generator, can be obtained as a limit of a family of semirings of the second class generated by g^{λ}. The connection between semirings of the second and the third class was also established in [19].

2.1 Pseudo-integral

The construction of the pseudo-integral is similar to the construction of the Lebesgue integral, therefore it is necessary to define a σ-$\oplus$-decomposable set function ([23]).

Definition 1. Let $\mathscr{A}$ be a σ-algebra of subsets of an arbitrary nonempty set X. The $\oplus$-*measure* is a set function $\mu : \mathscr{A} \to [a,b]_+$ if it holds

i) $\mu(\emptyset) = \mathbf{0}$,

ii) $\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \bigoplus_{i=1}^{\infty} \mu(A_i) = \lim_{n\to\infty} \bigoplus_{i=1}^{n} \mu(A_i)$ (σ-$\oplus$-decomposable)

where $(A_i)_{i\in N}$ is a sequence of pairwise disjoint sets from $\mathscr{A}$. If $\oplus$ is an idempotent operation, then the disjointness of sets and the condition i) can be omitted.

This construction starts with pseudo-integration of an elementary function ([23]).

If $e : X \to [a,b]$ is an *elementary function* of the following representation

$$e = \bigoplus_{i=1}^{\infty} a_i \odot \chi_{A_i}, \tag{1}$$

where $a_i \in [a,b]$, $A_i \in \mathscr{A}$ and χ_A is *the pseudo-characteristic function* of a set A (see [23]), *the pseudo-integral* of an elementary function e with respect to the $\oplus$-measure μ is

$$\int_X^{\oplus} e \odot d\mu = \bigoplus_{i=1}^{\infty} a_i \odot \mu(A_i). \tag{2}$$

The pseudo-integral of a measurable function $f : X \to [a,b]$ (with respect to σ-algebra $\mathscr{A}$ and $\oplus$-measure μ, see [23]) is

$$\int_X^{\oplus} f \odot d\mu = \lim_{n\to\infty} \int_X^{\oplus} \varphi_n \odot d\mu, \tag{3}$$

where $(\varphi_n)_{n\in\mathbb{N}}$ is a sequence of elementary functions chosen in such manner that $d(\varphi_n(x), f(x)) \to 0$ uniformly while $n \to \infty$ and d is the previously mentioned metric. The previous limit does not dependent on the choice of a sequence $(\varphi_n)_{n\in\mathbb{N}}$ (see [23]). The pseudo-integral of a function f on some arbitrary subset A of X is given by

$$\int_A^{\oplus} f \odot d\mu = \int_X^{\oplus} (f \odot \chi_A) \odot d\mu,$$

where χ_A is the pseudo-characteristic function.

Forms of the pseudo-integral that are essential for this paper are given by the following example.

Example 2. (a) For the semiring of the first class such that $x \oplus y = \sup\{x,y\}$ and $x \odot y = g^{-1}(g(x)\, g(y))$, the pseudo-integral of a function $f : X \to [a,b]$ is given by

$$\int_X^{\oplus} f \odot d\mu = \sup_{x\in X}(f(x) \odot \psi(x))$$

where function $\psi : X \to [a,b]$ defines sup-measure μ. Analogous form is obtained when $x \oplus y = \inf\{x,y\}$ (see [23]).

(b) For the semiring of the second class, i.e., for pseudo-operations generated by a strictly monotone bijection $g : [a,b] \to [0,\infty]$, the pseudo-integral of a measurable function $f : X \to [a,b]$ has the following form ([22, 23]):

$$\int_X^{\oplus} f \odot d\mu = g^{-1}\left(\int_X (g\circ f)\odot d(g\circ\mu)\right),$$

where integral on the right-hand side is the Lebesgue integral.

Remark 2. Both sup-measure and inf-measure with continuous density and the corresponding pseudo-integral based on a semiring of the first class, that had been previously obtained as a limit of semirings of the second class generated by g^λ, are the limits of families of decomposable measure generated by g^λ and corresponding g^λ-integrals (see [19]). That is, based on [19], it holds:

- Let $\mathscr{B}([c,d])$ be Borel σ-algebra in an interval $[c,d]$. Let ν be the usual Lebesgue measure on R and μ be a sup-measure on $([c,d],\mathscr{B}([c,d]))$ such that

$$\mu(A) = esssup_\nu(\psi(x) \mid x \in A), \tag{4}$$

where $\psi : [c,d] \to [a,b]$ is a continuous density. Let $([a,b],\sup,\odot)$ be a semiring with $\odot$ generated by a strictly increasing bijection g.
Then, there exists a family $\{\mu_\lambda\}$ of $\oplus_\lambda$-measures, where $\oplus_\lambda$ is generated by g^λ, $\lambda \in (0,\infty)$, such that for every continuous function $f : [c,d] \to [a,b]$ it holds

$$\int^{\sup} f \odot d\mu = \lim_{\lambda\to\infty} \int^{\oplus_\lambda} f \odot d\mu_\lambda.$$

It should be stressed that boundness that is required through this paper is considered in the sense of a given semiring. The boundness issues that are relevant for the presented results are given by the following example (see [9]).

Example 3. (a)If the given semiring is $([-\infty,\infty],\sup,\odot)$, it is obvious that $\mathbf{0} = -\infty$, $\mathbf{1} = 0$, $[a,b] = [a,b]_+ = [-\infty,\infty]$ and that the convention $-\infty+\infty = -\infty$ has to be accepted. For this example the function $f : X \to [a,b]_+$ is bounded if $f(x) \le M < \infty$ for all $x \in X$ and some $M \in [a,b)$, i.e., if $f(x) \preceq M \prec b$ for all $x \in X$.
Another possibility, in order to avoid unpleasant conventions, is to observe the semiring $([-\infty,\infty),\max,\odot)$ and functions that map X to $[-\infty,\infty)$.
The pseudo-integral of a function f is given by (3). However, we say that the function f is pseudo-integrable if $\int_X^{\oplus} f \odot d\mu$ exists as a finite value in the sense of a given semiring, which in this case means that $\int_X^{\oplus} f \odot d\mu \prec b$.
(b)Let $([a,b],\oplus,\odot)$ be a semiring of the second class given by a generator $g : [a,b] \to [0,\infty]$. In this case it holds that $\mathbf{0} = g^{-1}(0)$, $\mathbf{1} = g^{-1}(1)$ and $[a,b]_+ = [a,b]$ and the needed convention is $0\cdot\infty = 0$. If g is a strictly increasing bijection, then $\mathbf{0} = a$ and some function $f : X \to [a,b]_+$ is bounded if $f(x) \le M < b$ for all $x \in X$ and some $M \in [a,b)$, i.e., if $f(x) \preceq M \prec b$ for all $x \in X$.
The function f is pseudo-integrable if $\int_X^{\oplus} f \odot d\mu \prec b$.

2.1.1 Jensen Type Inequality for Pseudo-integrals

Pseudo-analysis' form of the well known Jensen integral inequality has been studied in [25]. The authors have proved the Jensen type inequality for pseudo-integrals based on two classes of semirings, i.e., on g-semiring and on $([a,b],\sup,\odot)$, where $\odot$ is given by some generating function.

Let $[a,b]\subseteq[0,\infty]$, $([a,b],\oplus,\odot)$ be a semiring of the second class given by some generator g, $[c,d]$ an interval and $(X,\mathscr{A},\mu)$ a measure space where μ is $\oplus$-measure such that $\mu(X)=\mathbf{1}$. Let $f:X\to(a,b)$ be an arbitrary pseudo-integrable function, i.e., pseudo-integral $\int_X^{\oplus} f\odot d\mu$ exist as a finite value in the sense of the observed semiring, and Φ a function on (a,b). Now, based on [25], we have the following:

- If g is a convex and increasing and Φ a convex and non-increasing, then it holds

$$\Phi\left(\int_X^{\oplus} f\odot d\mu\right)\preceq\int_X^{\oplus}(\Phi\circ f)\odot d\mu. \tag{5}$$

- If g is a convex and decreasing and Φ a concave and non-decreasing, then (5) holds.

Also, based on results from [19], the following has been proved in [25]:

- Let $([a,b],\sup,\odot)$ be a semiring where $\odot$ is generated by a strictly increasing, convex bijection. If μ is given by (4), then for some continuous function $f:[c,d]\to(a,b)$, the corresponding form of inequality (5) holds.

2.1.2 Chebyshev Type Inequality for Pseudo-integrals

The Chebyshev integral inequality in the pseudo-analysis' surrounding has been studied in [1]. For this type of inequality some additional restrictions for integrable functions are needed, i.e., condition of comonotonicity is unavoidable.

Definition 2. A function $f,h:X\to R$ are *comonotone* if for all $x,y\in X$ it holds

$$(f(x)-f(y))(h(x)-h(y))\geq 0.$$

Results from [1] can be easily extended from the unit interval to an arbitrary interval $[c,d]$, $\mu([c,d])=\mathbf{1}$, that is, the following hold.

- Let $([a,b],\oplus,\odot)$ be a semiring of the second class given by an increasing generator g, $[c,d]$ an interval and $([c,d],\mathscr{A},\mu)$ a measure space where μ is $\oplus$-measure such that $\mu([c,d])=\mathbf{1}$. Let $f,h:[c,d]\to[a,b]$ be two comonotone measurable functions. Then, it holds:

$$\left(\int_{[c,d]}^{\oplus} f\odot d\mu\right)\odot\left(\int_{[c,d]}^{\oplus} h\odot d\mu\right)\preceq\int_{[c,d]}^{\oplus}(f\odot h)\odot d\mu. \tag{6}$$

- Let $([0,\infty],\sup,\odot)$ be a semiring with $\odot$ generated by some increasing generator g, $[c,d]$ an interval and $([c,d],\mathscr{A},\mu)$ a measure space where μ is given by (4). Let $\mu([c,d])=\mathbf{1}$. If $f,h:[c,d]\to[0,\infty]$ are continuous comonotone functions, then the corresponding form of inequality (6) holds.

2.2 *Interval-Valued Functions on Semiring* $([a,b],\oplus,\odot)$

Let $([a,b],\oplus,\odot)$ be a semiring that belongs to one of the three basic classes. Let X be a nonempty set, $\mathscr{A}=\mathscr{P}(X)$ and $(X,\mathscr{A},\mu)$ measure space where μ is a $\oplus$-measure.

Let $\mathscr{I}$ be the class of all closed subintervals of $[a,b]_+$, i.e.,

$$\mathscr{I}=\{[u,v]\mid u\le v \text{ and } [u,v]\subseteq[a,b]_+\}.$$

Definition 3. An *interval-valued function* is a function from X to $\mathscr{I}$.

Interval-valued functions that are being observed, with respect to the previously given measure space, are measurable ([9]).

Remark 3. A set-valued functions $F:X\to\mathscr{F}\setminus\{\emptyset\}$, where $\mathscr{F}$ is the class of all closed subsets of $[a,b]_+$, is measurable if its graph is measurable, that is if

$$Gr(F)=\{(x,r)\in X\times[a,b]_+\mid r\in F(x)\}\in\mathscr{A}\times\mathscr{B}([a,b]_+),$$

where $\mathscr{B}([a,b]_+)$ is the Borel field of $[a,b]_+$.

Due to theirs specific range, interval-valued functions F that are being studied through this paper, are represented by *the border functions* $l,r:X\to[a,b]_+$ in the following way

$$F(x)=[l(x),r(x)]. \tag{7}$$

While working with sets, by the pseudo-sum and the pseudo-product of sets we are considering the following.

Definition 4. Let $([a,b],\oplus,\odot)$ be a semiring from one of three basic classes. Let A and B be two arbitrary subsets of $[a,b]$ and let α be a constant from $[a,b]_+$. Then

i) $A\oplus B=\{x\oplus y\mid x\in A \text{ and } y\in B\}$;
ii) $A\odot B=\{x\odot y\mid x\in A \text{ and } y\in B\}$;
iii) $\alpha\odot A=\{\alpha\odot x\mid x\in A\}$.

It has been shown (see [31]) that for arbitrary interval-valued functions F_1 and F_2 represented by their border functions l_1, r_1, l_2 and r_2, the following holds

$$(F_1\oplus F_2)(x)=F_1(x)\oplus F_2(x)=[l_1(x)\oplus l_2(x),r_1(x)\oplus r_2(x)] \tag{8}$$

and

$$(\alpha \odot F_1)(x) = \alpha \odot F_1(x) = [\alpha \odot l_1(x), \alpha \odot r_1(x)],$$

for some $\alpha \in [a,b]_+$.

Also, it can be easily shown that, if $\odot$ is given by a strictly monotone generator g, for interval-valued functions F_1 and F_2 with border functions l_1, r_1, l_2 and r_2, it holds

$$(F_1 \odot F_2)(x) = F_1(x) \odot F_2(x) = [l_1(x) \odot l_2(x), r_1(x) \odot r_2(x)].$$

Remark 4. The pseudo-sum (8) can be extended to countable case by the following definition

$$\lim_{n\to\infty}[a_n, b_n] = [\lim_{n\to\infty} a_n, \lim_{n\to\infty} b_n],$$

when the right-hand side exists.

Remark 5. Constructions and results presented in [8, 9, 31, 32, 33] are given for set-valued functions in general, however presented restriction to the interval-valued functions is being additionally considered due to the specific form which has the high impact to the possible applications in the field of convex fuzzy random variables and random sets ([16, 20, 21, 27]).

Another notion necessary for the presented research is the notion of an interval-valued set-function on a semiring $([a,b], \oplus, \odot)$.

Definition 5. Let $\mathscr{A}$ be a σ-algebra of subsets of X. A function $\Pi : \mathscr{A} \to \mathscr{I}$ is the *interval-valued* $\sigma - \oplus$*-measure* if

i) $\Pi(\emptyset) = \{\mathbf{0}\} = [\mathbf{0}, \mathbf{0}]$,

ii) $\Pi(\bigcup_{i=1}^{\infty} A_i) = \bigoplus_{i=1}^{\infty} \Pi(A_i) = \lim_{n\to\infty} \bigoplus_{i=1}^{n} \Pi(A_i)$,

where $(A_i)_{i\in N}$ is a sequence of pairwise disjoint sets from $\mathscr{A}$. If $\oplus$ is an idempotent operation, the disjointness of sets can be omitted.

Example 4. (a)If $\mu : \mathscr{A} \to [a,b]_+$ is a $\oplus$-measure then for all $x \in X$

$$\Pi(A) = [\mu(A), \mu(A)] = \{\mu(A)\}$$

is an interval-valued $\sigma - \oplus$-measure.

(b)If $\mu : \mathscr{A} \to [a,b]_+$ is a $\oplus$-measure then for all $x \in X$

$$\Pi(A) = [\mathbf{0}, \mu(A)]$$

is an interval-valued $\sigma - \oplus$-measure.

In order to investigate monotonicity of this specific type of measure, it is necessary to introduce the relation "less or equal" applied on sets from $\mathscr{I}$:

for $C, D \in \mathscr{I}$ holds $C \preceq_S D$ if for all $x \in C$ there exists $y \in D$ such that $x \preceq y$ and for all $y \in D$ there exists $x \in C$ such that $x \preceq y$.

Now, it is possible to formulate the following definition.

Definition 6. An interval-valued $\sigma-\oplus$-measure $\Pi : \mathscr{A} \to \mathscr{I}$ is *monotone with respect to* $\preceq_S$ if for $A \subseteq B, A, B \in \mathscr{A}$, holds $\Pi(A) \preceq_S \Pi(B)$.

Remark 6. Necessity for introduction of relation $\preceq_S$ can be illustrated by the previous example. If the usual subset $\subseteq$ is used instead of $\preceq_S$, monotonicity for interval-valued $\sigma-\oplus$-measure from Example 4(a) would not hold. However, for some special cases, as in Example 4(b), $\preceq_S$ and $\subseteq$ can coincide.

3 Case I: Interval-Valued Measure via Pseudo-integral of Interval-Valued Function

The first approach to the interval-valued σ-$\oplus$-measures presented in this paper is baseb on a generalization of the pseudo-integral done in the Aumann's style (see [7, 9]).

Through this section let $([a,b], \oplus, \odot)$ be a semiring that belongs to one of the three basic classes. Let X be a nonempty set, $\mathscr{A} = \mathscr{P}(X)$ and $(X, \mathscr{A}, \mu)$ measure space where μ is a $\oplus$-measure. Let $f : X \to [a,b]_+$ be a measurable real-valued function. Furthermore, let us suppose that the function f is integrable with respect to the pseudo-integral given by (3), i.e., pseudo-integral $\int_X^{\oplus} f \odot d\mu$ exist as a finite value in the sense of the observed semiring. The family of all such functions f is denoted with $L^1_{\oplus}(\mu)$.

Definition 7. Let F be an interval-valued function. *The pseudo-integral* of F on $A \in \mathscr{A}$ is

$$\int_A^{\oplus} F \odot d\mu = \left\{ \int_A^{\oplus} f \odot d\mu \,\middle|\, f \in S(F) \right\}, \tag{9}$$

where

$$S(F) = \left\{ f \in L^1_{\oplus}(\mu) \,|\, f(x) \in F(x) \text{ on } X \ \mu - a.e. \right\}. \tag{10}$$

Specially, for $\int^{\oplus}$ being the Lebesgue integral, (9) is the classical Aumann's integral ([5]).

An interval-valued function $F : X \to \mathscr{I}$ is *pseudo-integrable* on some $A \in \mathscr{A}$ if

$$\int_A^{\oplus} F \odot d\mu \neq \emptyset.$$

As in the classical case, the following property is closely connected with pseudo-integration of interval(set)-valued functions (see [7, 9]).

Definition 8. An interval-valued function F is *pseudo-integrably bounded* if there is a function $h \in L^1_{\oplus}(\mu)$ such that:

i) $\bigoplus_{\alpha\in F(x)} \alpha \preceq h(x)$, for the idempotent pseudo-addition,
ii) $\sup_{\alpha\in F(x)} \alpha \preceq h(x)$, for the pseudo-addition given by an increasing generator g,
iii) $\inf_{\alpha\in F(x)} \alpha \preceq h(x)$, for the pseudo-addition given by a decreasing generator g.

It has been proved in [9] that border functions of integrably bounded interval(set)-valued functions are from class $S(F)$ and that integrably bounded interval(set)-valued functions are pseudo-integrable. Now, for a pseudo-integrably bounded interval-valued function F with border functions l and r the following holds

$$\int_X^{\oplus} F \odot d\mu = \left[\int_X^{\oplus} l \odot d\mu, \int_X^{\oplus} r \odot d\mu\right] \tag{11}$$

(see [9]).

The interval-valued σ-$\oplus$-measure that is being considered through this section is given by the following theorem from [9, 31].

Theorem 1. *Let $F : X \to \mathscr{I}$ be a pseudo-integrably bounded interval-valued function with border functions l and r, an interval-valued set-function μ_F based on interval-valued pseudo-integral of F given by*

$$\mu_F(A) = \int_A^{\oplus} F \odot d\mu = \left[\int_A^{\oplus} l \odot d\mu, \int_A^{\oplus} r \odot d\mu\right], \tag{12}$$

where $A \subseteq X$, has the following properties:

i) $\mu_F(\emptyset) = \{\mathbf{0}\} = [\mathbf{0},\mathbf{0}]$;
ii) μ_F *is* monotone with respect to $\preceq_S$, *i.e., if* $A \subseteq B$, $A,B \in \mathscr{A}$, *then* $\mu_F(A) \preceq_S \mu_F(B)$;
iii) μ_F *is* additive, *i.e.,* $\mu_F(A \cup B) = \mu_F(A) \oplus \mu_F(B)$, *where A and B are pairwise disjoint sets from $\mathscr{A}$ (if $\oplus$ is an idempotent operation, then disjointness of sets can be omitted).*
iv) μ_F *is* $\sigma-\oplus$-additive, *i.e.,* $\mu_F(\cup_{i=1}^{\infty} A_i) = \bigoplus_{i=1}^{\infty} \mu_F(A_i)$, *where $(A_i)_{i\in N}$ is a sequence of pairwise disjoint sets from $\mathscr{A}$ (if $\oplus$ is an idempotent operation, then disjointness of sets can be omitted).*

Obviously, (12) is a monotone (with respect to $\preceq_S$) interval-valued σ-$\oplus$-measure.

3.1 Integral Inequalities for Case I

The following results concerning Jensen type and Chebyshev type inequalities are from [8, 32, 33] and are now, for the sake of completeness, given in the new form, that is, in the form of the interval-valued measure (12).

3.1.1 Jensen Type Inequality for Case I

Let $([a,b],\oplus,\odot)$, $[a,b]\subseteq[0,\infty]$, be a semiring of the second class given by some generator g and μ the corresponding $\oplus$-measure.

Theorem 2. *Let $\mu(X)=\mathbf{1}$ and let $\Phi:[a,b]_+\to[a,b]_+$ be a convex, decreasing and bounded function. If the generator g is a convex and increasing function, and if F is a pseudo-integrably bounded interval-valued function on X, then it holds:*

$$\Phi(\mu_F(X))\preceq_S\mu_{\Phi\circ F}(X). \tag{13}$$

Remark 7. If Φ is a function from U to V, where U and V are some arbitrary sets, and $A\subset U$, then the following notation is used

$$\Phi(A)=\{\Phi(x)\,|x\in A\}. \tag{14}$$

As seen in [8], the analogous claim holds for a decreasing generator g. Additionally, based on results from [19, 25], the previous can be extended to the semiring of the first class as follows (see [8]).

Theorem 3. *Let $([a,b],\sup,\odot)$ be a semiring where $\odot$ is given by some generator g. Let $\mathscr{B}([c,d])$ be Borel σ-algebra on an interval $[c,d]$ such that $\mu([c,d])=\mathbf{1}$, where μ is given by (4). Let Φ be a convex, decreasing and bounded function. If F is a pseudo-integrably bounded interval-valued function on $[c,d]$, with continuous $S(F)$, then it holds:*

$$\Phi(\mu_F([c,d]))\preceq_S\mu_{\Phi\circ F}([c,d]).$$

3.1.2 Chebyshev Type Integral Inequalities for Case I

Let $([a,b],\oplus,\odot)$ be an arbitrary semiring of the first class with $\oplus=\sup$ and pseudo-multiplication given by some generator g, or a g-semiring.

Theorem 4. *Let $([a,b],\oplus,\odot)$ be an arbitrary g-semiring given by an increasing generator g. Let $F_1,F_2:[c,d]\to\mathscr{I}$ be pseudo-integrably bounded interval-valued functions represented by their border functions l_1, r_1, l_2 and r_2, where $\mu([c,d])=\mathbf{1}$. Let $F_1\odot F_2$ be a pseudo-integrably bounded interval-valued function. If l_1 and l_2 are comonotone and r_1 and r_2 are comonotone, then it holds:*

$$\mu_{F_1}([c,d])\odot\mu_{F_2}([c,d])\preceq_S\mu_{F_1\odot F_2}([c,d]). \tag{15}$$

Theorem 5. *Let $([a,b],\sup,\odot)$ be a semiring where $\odot$ is given by an increasing generator g. Let $\mathscr{B}([c,d])$ be Borel σ-algebra on an interval $[c,d]$ such that $\mu([c,d])=\mathbf{1}$, where μ is given by (4). Let $F_1,F_2:[c,d]\to\mathscr{I}$ be pseudo-integrably bounded interval-valued functions represented by their continuous border functions l_1, r_1, l_2 and r_2, such that $F_1\odot F_2$ is also a pseudo-integrably bounded interval-valued function. If l_1 and l_2 are comonotone and r_1 and r_2 are comonotone, then the corresponding form of (15) holds.*

4 Case II: Interval-Valued Measure via Pseudo-integrals of Real Valued Functions

This section follows the approach to interval-valued measures from [12] that is now considered in the pseudo-analysis' framework.

Through this section let $([a,b],\oplus,\odot)$ be a semiring that belongs to one of the three basic classes. Let X be a nonempty set, $\mathscr{A}$ a σ-algebra of its subsets and $(X,\mathscr{A},\mu)$ measure space where μ is a $\oplus$-measure. Let $f:X\to[a,b]$ be a measurable real-valued function.

The interval-valued σ-$\oplus$-measure that is used as the base for construction of a new type of interval-valued pseudo-integral is given by the following definition.

Let $\mathscr{M}$ be an arbitrary nonempty family of $\oplus$-measures μ given on $\mathscr{A}$.

Definition 9. The interval-valued set-function $\overline{\mu}_{\mathscr{M}}:\mathscr{A}\to\mathscr{I}$ for the family $\mathscr{M}$ is

$$\overline{\mu}_{\mathscr{M}}=[\mu_l,\mu_r],\quad \mu_l,\mu_r\in\mathscr{M},$$

where

$$\mu_l(A)\preceq\mu(A)\preceq\mu_r(A)$$

for all μ from $\mathscr{M}$ and all A from $\mathscr{A}$, if such μ_l and μ_r exist.

Proposition 1. *If exists, $\overline{\mu}_{\mathscr{M}}$ is an interval-valued σ-$\oplus$-measure.*

The previous proposition follows directly from the definitions of $\oplus$-measures and pseudo-addition of intervals.

Remark 8. Let $\mathscr{M}_0$ be a family of $\oplus$-measures that includes the trivial $\oplus$-measures μ_0 of the form $\mu_0(A)=\mathbf{0}$ for all $A\in\mathscr{A}$. If μ_r from the previous definition for family $\mathscr{M}_0$ exists, the interval-valued set-function $\overline{\mu}_{\mathscr{M}_0}$ is

$$\overline{\mu}_{\mathscr{M}_0}=[\mathbf{0},\mu_r].$$

Further on, for two $\oplus$-measures μ_1 and μ_2 from $\mathscr{M}$ the shorter notation $\mu_1\preceq\mu_2$ will be used if for all sets A from $\mathscr{A}$ holds $\mu_1(A)\preceq\mu_2(A)$.

4.1 Construction of Interval-Valued Pseudo-integral Based on $\overline{\mu}_{\mathscr{M}}$

The proposed construction is an extension of the construction from [23] to the interval-valued case. The first step is to define the interval-valued pseudo-integral of an elementary function.

Let $\mathscr{M}$ be an arbitrary nonempty family of $\oplus$-measures μ such that μ_l and μ_r from Definition 9 exist.

Definition 10. *The interval-valued pseudo-integral* of an elementary function e given by (1), with respect to the interval-valued σ-$\oplus$-measure $\overline{\mu}_{\mathscr{M}}$ is

$$\int_X^{\oplus} e \odot d\overline{\mu}_{\mathscr{M}} = \bigoplus_{i=1}^{\infty} a_i \odot \overline{\mu}_{\mathscr{M}}(A_i).$$

Proposition 2. *If e is an elementary function given by (1), then it holds*

$$\int_X^{\oplus} e \odot d\overline{\mu}_{\mathscr{M}} = \left[\int_X^{\oplus} e \odot d\mu_l, \int_X^{\oplus} e \odot d\mu_r \right],$$

where integrals of the right are the pseudo-integrals from Section 2.1.

Proof. Proof follows from definition of the pseudo-integral of elementary function given by (2) and extension of the pseudo-sum to countable case from Remark 4:

$$\begin{aligned}
\int_X^{\oplus} e \odot d\overline{\mu}_{\mathscr{M}} &= \bigoplus_{i=1}^{\infty} a_i \odot \overline{\mu}_{\mathscr{M}}(A_i) \\
&= \lim_{n\to\infty} \bigoplus_{i=1}^{n} a_i \odot \overline{\mu}_{\mathscr{M}}(A_i) \\
&= \lim_{n\to\infty} \bigoplus_{i=1}^{n} a_i \odot [\mu_l(A_i), \mu_r(A_i)] \\
&= \lim_{n\to\infty} \bigoplus_{i=1}^{n} [a_i \odot \mu_l(A_i), a_i \odot \mu_r(A_i)] \\
&= \lim_{n\to\infty} \left[\bigoplus_{i=1}^{n} a_i \odot \mu_l(A_i), \bigoplus_{i=1}^{n} a_i \odot \mu_r(A_i) \right] \\
&= \left[\lim_{n\to\infty} \bigoplus_{i=1}^{n} a_i \odot \mu_l(A_i), \lim_{n\to\infty} \bigoplus_{i=1}^{n} a_i \odot \mu_r(A_i) \right] \\
&= \left[\int_X^{\oplus} e \odot d\mu_l, \int_X^{\oplus} e \odot d\mu_r \right].
\end{aligned}$$

□

Now is possible to introduce the corresponding integral of an arbitrary measurable function.

Definition 11. *The interval-valued pseudo-integral* of a measurable function $f : X \to [a,b]$ with respect to the interval-valued σ-$\oplus$-measure $\overline{\mu}_{\mathscr{M}}$ is

$$\int_X^{\oplus} f \odot d\overline{\mu}_{\mathscr{M}} = \lim_{n\to\infty} \int_X^{\oplus} \varphi_n \odot d\overline{\mu}_{\mathscr{M}}, \tag{16}$$

where $(\varphi_n)_{n\in N}$ is a sequence of elementary functions chosen in such manner that $d(\varphi_n(x), f(x)) \to 0$ uniformly while $n \to \infty$.

As in the case of the pseudo-integral from [23], the interval-valued pseudo-integral based $\overline{\mu}_{\mathscr{M}}$ of a function f on some arbitrary subset A of X is given by

$$\overset{\oplus}{\int_A} f \odot d\overline{\mu}_{\mathscr{M}} = \overset{\oplus}{\int_X} (f \odot \chi_A) \odot d\overline{\mu}_{\mathscr{M}}, \tag{17}$$

where χ_A is the pseudo-characteristic function.

Connection between interval valued pseudo-integral (16) and the pseudo-integral (3) is given in the next theorem.

Theorem 6. *If $f : X \to [a,b]$ is a measurable function, then*

$$\overset{\oplus}{\int_X} f \odot d\overline{\mu}_{\mathscr{M}} = \left[\overset{\oplus}{\int_X} f \odot d\mu_l, \overset{\oplus}{\int_X} f \odot d\mu_r \right]. \tag{18}$$

Proof. Proof is based on Definition 11, Proposition 2 and extension of the pseudo-sum to the countable case:

$$\begin{aligned}
\overset{\oplus}{\int_X} f \odot d\overline{\mu}_{\mathscr{M}} &= \lim_{n\to\infty} \overset{\oplus}{\int_X} \varphi_n \odot d\overline{\mu}_{\mathscr{M}} \\
&= \lim_{n\to\infty} \left[\overset{\oplus}{\int_X} \varphi_n \odot d\mu_l, \overset{\oplus}{\int_X} \varphi_n \odot d\mu_r \right] \\
&= \left[\lim_{n\to\infty} \overset{\oplus}{\int_X} \varphi_n \odot d\mu_l, \lim_{n\to\infty} \overset{\oplus}{\int_X} \varphi_n \odot d\mu_r \right] \\
&= \left[\overset{\oplus}{\int_X} f \odot d\mu_l, \overset{\oplus}{\int_X} f \odot d\mu_r \right]. \qquad \square
\end{aligned}$$

Some of the basic properties of this new integral are given by the following proposition.

Proposition 3. *Let $f, f_1, f_2 : X \to [a,b]$ be measurable functions, α a value from $[a,b]_+$ and A a subset of X from $\mathscr{A}$. Then the following holds*

i) $\overset{\oplus}{\int_X} (f_1 \oplus f_2) \odot d\overline{\mu}_{\mathscr{M}} = \overset{\oplus}{\int_X} f_1 \odot d\overline{\mu}_{\mathscr{M}} \oplus \overset{\oplus}{\int_X} f_2 \odot d\overline{\mu}_{\mathscr{M}}$,

ii) $\overset{\oplus}{\int_X} (\alpha \odot f) \odot d\overline{\mu}_{\mathscr{M}} = \alpha \odot \overset{\oplus}{\int_X} f \odot d\overline{\mu}_{\mathscr{M}}$,

iii) $\overset{\oplus}{\int_A} \alpha \odot d\overline{\mu}_{\mathscr{M}} = \alpha \odot \overline{\mu}_{\mathscr{M}}(A)$,

iv) if $f_1 \preceq f_2$ *then* $\overset{\oplus}{\int_X} f_1 \odot d\overline{\mu}_{\mathscr{M}} \preceq_S \overset{\oplus}{\int_X} f_2 \odot d\overline{\mu}_{\mathscr{M}}$.

The claims of the previous proposition easily follow from Theorem 6, properties of the pseudo-integral (3) and pseudo-addition of intervals.

Remark 9. It is easy to see that Theorem 6 and Proposition 3 also hold for integration over arbitrary set A from $\mathscr{A}$.

Remark 10. If the family $\mathscr{M}_0$ is used for construction of the interval-valued set-function $\overline{\mu}_{\mathscr{M}_l}$, integral (18) is of the form

$$\int_X^{\oplus} f \odot d\overline{\mu}_{\mathscr{M}_0} = \left[\mathbf{0}, \int_X^{\oplus} f \odot d\mu_r \right].$$

4.2 Interval-Valued Measure Based on Pseudo-integration with Respect to $\overline{\mu}_{\mathscr{M}}$

The construction of new interval-valued measure that is the core of this section is presented by the next theorem. Now we are observing functions with range in $[a,b]_+$.

Let $\mathscr{M}$ be an arbitrary nonempty family of $\oplus$-measures μ such that μ_l and μ_r from Definition 9 exist.

Theorem 7. *Let $f : X \to [a,b]_+$ be a measurable function. An interval-valued set-function $\overline{\mu}^f_{\mathscr{M}}$ based on interval-valued pseudo-integral of f given by*

$$\overline{\mu}^f_{\mathscr{M}}(A) = \int_A^{\oplus} f \odot d\overline{\mu}_{\mathscr{M}} = \left[\int_A^{\oplus} f \odot d\mu_l, \int_A^{\oplus} f \odot d\mu_r \right], \tag{19}$$

where $A \subseteq X$, has the following properties:

i) $\overline{\mu}^f_{\mathscr{M}}(\emptyset) = \{\mathbf{0}\} = [\mathbf{0},\mathbf{0}]$;

ii) $\overline{\mu}^f_{\mathscr{M}}$ *is* monotone with respect to $\preceq_S$, *i.e., if* $A \subseteq B$, $A,B \in \mathscr{A}$, *then* $\overline{\mu}^f_{\mathscr{M}}(A) \preceq_S \overline{\mu}^f_{\mathscr{M}}(B)$;

iii) $\overline{\mu}^f_{\mathscr{M}}$ *is* additive, *i.e.,* $\overline{\mu}^f_{\mathscr{M}}(A \cup B) = \overline{\mu}^f_{\mathscr{M}}(A) \oplus \overline{\mu}^f_{\mathscr{M}}(B)$, *where A and B are pairwise disjoint sets from $\mathscr{A}$ (if $\oplus$ is an idempotent operation, then disjointness of sets can be omitted).*

iv) $\overline{\mu}^f_{\mathscr{M}}$ *is* $\sigma - \oplus$-additive, *i.e.,* $\overline{\mu}^f_{\mathscr{M}}(\cup_{i=1}^{\infty} A_i) = \bigoplus_{i=1}^{\infty} \overline{\mu}^f_{\mathscr{M}}(A_i)$, *where $(A_i)_{i\in N}$ is a sequence of pairwise disjoint sets from $\mathscr{A}$ (if $\oplus$ is an idempotent operation, then disjointness of sets can be omitted).*

Proof. Proofs of claims i), iii) and iv) follow from Theorem 6, Proposition 3 and well known properties of the pseudo-integral (3) (see [23]).

Proof of claim ii) is base on Theorem 6. Let A and B two sets from $\mathscr{A}$ such that $A \subseteq B$, and let us use the following notation

$$A_l = \int_A^{\oplus} f \odot d\mu_l, \quad A_r = \int_A^{\oplus} f \odot d\mu_r, \quad B_l = \int_B^{\oplus} f \odot d\mu_l \quad \text{and} \quad B_r = \int_B^{\oplus} f \odot d\mu_r.$$

Now,

$$\overline{\mu}^{f}_{\mathscr{M}}(A) = [A_l, A_r] \quad \text{and} \quad \overline{\mu}^{f}_{\mathscr{M}}(B) = [B_l, B_r].$$

Since, based on properties of the pseudo-integral (3), it holds

$$A_l \preceq B_l \quad \text{and} \quad A_r \preceq B_r,$$

there are only two following possibilities for intervals $[A_l, A_r]$ and $[B_l, B_r]$:

$$A_l \preceq A_r \preceq B_l \preceq B_r \quad \text{or} \quad A_l \preceq B_l \preceq A_r \preceq B_r.$$

For both possibilities it is obvious that for all $x \in [A_l, A_r]$ there exists $y \in [B_l, B_r]$ such that $x \preceq y$ and that for all $y \in [B_l, B_r]$ there exists $x \in [A_l, A_r]$ such that $x \preceq y$, i.e., $[A_l, A_r] \preceq_S [B_l, B_r]$. □

Obviously, (19) is a monotone (with respect to $\preceq_S$) interval-valued σ-$\oplus$-measure.

4.3 Integral Inequalities for Case II

This section contains Jensen type and Chebyshev type inequalities based on the new interval-valued measure (19). Through this section let us suppose that $\mathscr{M}_0$ is a family of $\oplus$-measures such that it includes μ_0 and μ_r from Definition 9 exists.

Due to the specific interval forms that are being observed, the Jensen type inequality is now of the reverse form while the Chebyshev type inequality remain as in the previous case.

4.3.1 Jensen Type Inequality for Case II

Let $([a,b], \oplus, \odot)$, $[a,b] \subseteq [0,\infty]$, be a semiring of the second class given by some generator g. Let $f : X \to [a,b]_+$ be a measurable pseudo-integrable function, i.e., pseudo-integral $\int_X^{\oplus} f \odot d\mu$ exist as a finite value in the sense of the observed semiring for all $\mu \in \mathscr{M}_0$. Let $\overline{\mu}_{\mathscr{M}_0}(X) = [\mathbf{0}, \mathbf{1}]$.

Theorem 8. *Let $\Phi : [a,b]_+ \to [a,b]_+$ be a convex, decreasing and bounded function. If the generator g is a convex and increasing function, then it holds:*

$$\overline{\mu}^{\Phi \circ f}_{\mathscr{M}_0}(X) \preceq_S \Phi\left(\overline{\mu}^{f}_{\mathscr{M}_0}(X)\right). \tag{20}$$

Proof. From (19) and properties of family $\mathscr{M}_0$ follow

$$\overline{\mu}^{\Phi \circ f}_{\mathscr{M}_0}(X) = \left[\mathbf{0}, \int_X^{\oplus} (\Phi \circ f) \odot d\mu_r\right]$$

and

$$\Phi\left(\overline{\mu}^{f}_{\mathscr{M}_0}(X)\right) = \Phi\left(\left[\mathbf{0}, \int_X^{\oplus} f \odot d\mu_r\right]\right).$$

Now, due to the monotonicity of Φ, for an arbitrary value ω from $\Phi\left(\overline{\mu}^{f}_{\mathscr{M}_0}(X)\right)$ it holds

$$\Phi\left(\int_X^{\oplus} f \odot d\mu_r\right) \preceq \omega \preceq \Phi(\mathbf{0}).$$

If the following notation is used

$$A = \int_X^{\oplus} (\Phi \circ f) \odot d\mu_r \quad \text{and} \quad B = \Phi\left(\int_X^{\oplus} f \odot d\mu_r\right)$$

from (5) it follows $B \preceq A$. Also, based on properties of the pseudo-integral, nonnegativity of function f and monotonicity of Φ it can be easily shown that $A \preceq \Phi(\mathbf{0})$. Now, the following order is obtained

$$\mathbf{0} \preceq B \preceq A \preceq \Phi(\mathbf{0})$$

from which easily follows the claim. □

As in the case I, the analogous claim holds for a decreasing generator g. Also, again based on results from [19, 25], the previous can be extended to the semiring of the first class.

Theorem 9. *Let $([a,b], \sup, \odot)$ be a semiring where $\odot$ is given by some generator g. Let $\mathscr{B}([c,d])$ be Borel σ-algebra on an interval $[c,d]$ such that $\overline{\mu}_{\mathscr{M}_0}([c,d]) = [\mathbf{0},\mathbf{1}]$, where μ from $\mathscr{M}_0$ are of the form (4). Let Φ be a convex, decreasing and bounded function and let f be continuous. Then, it holds:*

$$\overline{\mu}^{\Phi\circ f}_{\mathscr{M}_0}([c,d]) \preceq_S \Phi\left(\overline{\mu}^{f}_{\mathscr{M}_0}([c,d])\right).$$

4.3.2 Chebyshev Type Integral Inequalities for Case II

Let $([a,b], \oplus, \odot)$ be an arbitrary semiring of the first class with $\oplus = \sup$ and pseudo-multiplication given by some generator g, or a g-semiring.

Theorem 10. *Let $([a,b], \oplus, \odot)$ be an arbitrary g-semiring given by an increasing generator g. Let $f_1, f_2 : [c,d] \to [a,b]_+$ be measurable functions where $\overline{\mu}_{\mathscr{M}_0}([c,d]) = [\mathbf{0},\mathbf{1}]$. If f_1 and f_2 are comonotone, then it holds:*

$$\overline{\mu}^{f_1}_{\mathscr{M}_0}([c,d]) \odot \overline{\mu}^{f_2}_{\mathscr{M}_0}([c,d]) \preceq_S \overline{\mu}^{f_1\odot f_2}_{\mathscr{M}_0}([c,d]). \tag{21}$$

The proof of the previous theorem can be done in the same manner as the proof of Theorem 8. It is based on results known for the pseudo-integral [1], the construction of measure $\overline{\mu}^{f}_{\mathscr{M}}$ and the pseudo-multiplication of intervals.

Also, the corresponding claims can be easily shown for a g-semiring given by decreasing generator g and for the semiring of the first class. Form of this inequality for the semiring of the first class is given by the following theorem.

Theorem 11. *Let* $([a,b],\sup,\odot)$ *be a semiring where* $\odot$ *is given by an increasing generator* g. *Let* $\mathscr{B}([c,d])$ *be Borel* σ*-algebra on an interval* $[c,d]$ *such that* $\overline{\mu}_{\mathscr{M}_0}([c,d]) = [\mathbf{0},\mathbf{1}]$, *where* μ *from* $\mathscr{M}_0$ *are of the form (4). Let* $f_1, f_2 : [c,d] \to [a,b]_+$ *be continuous functions . If* f_1 *and* f_2 *are comonotone, then the corresponding form of (21) holds.*

5 Conclusion

Some of the recent research by the various authors is focused on generalizations of the classical integral inequalities for the integrals based on non-additive measures (see [1, 2, 3, 4, 6, 11, 25, 29], etc.). The presented results are incorporating this direction of research with the problem of the interval-valued measures. Therefore, a highly interesting possibility for the future lies in the field of random sets and fuzzy random variables.

Acknowledgements. The authors are very grateful to the anonymous reviewer for highly helpful observations and suggestions. This paper has been supported by the Ministry of Science and Technological Development of Republic of Serbia 174009 and TR32035 and by the Provincial Secretariat for Science and Technological Development of Vojvodina.

References

1. Agahi, H., Mesiar, R., Ouyang, Y.: Chebyshev type inequalities for pseudo-integrals. Nonlinear Analysis: Theory, Methods and Applications 72, 2737–2743 (2010)
2. Agahi, H., Mesiar, R., Ouyang, Y.: General Minkowski type inequalities for Sugeno integrals. Fuzzy Sets and Systems 161, 708–715 (2010)
3. Agahi, H., Mesiar, R., Ouyang, Y., Pap, E., Štrboja, M.: Hölder and Minkowski type inequalities for pseudo-integral. Applied Mathematics and Computation 217, 8630–8639 (2011)
4. Agahi, H., Mesiar, R., Ouyang, Y., Pap, E., Štrboja, M.: Berwald type inequality for Sugeno integral. Applied Mathematics and Computation 217, 4100–4108 (2010)
5. Aumann, R.J.: Integrals of set-valued functions. Journal of Mathematical Analysis and Applictions 12, 1–12 (1965)
6. Caballero, J., Sadarangani, K.: A CauchySchwarz type inequality for fuzzy integrals. Nonlinear Analysis: Theory, Methods and Applications 73, 3329–3335 (2010)
7. Grbić, T., Štajner-Papuga, I., Nedović, L.: Pseudo-integral of set-valued funtions. In: Proc. of EUSFLAT 2007, vol. I, pp. 221–225 (2007)
8. Grbić, T., Štrboja, M., Štajner-Papuga, I., Vivona, D., Mihailović, C.B.: Jensen type inequalities for pseudo-integrals of set-valued functions. In: Proc. of SISY 2010, Subotica, pp. 978–971 (2010) ISBN 978-1-4244-2407-8
9. Grbić, T., Štajner-Papuga, I., Štrboja, M.: An approach to pseudo-integration of set-valued functions. Inf. Sci. 181(11), 2278–2292 (2011)
10. Hiai, F., Umegaki, H.: Integrals, conditional expectations, and martingales of multivalued functions. J. Multivariate Anal. 7, 149–182 (1977)
11. Hong, D.H.: A Liapunov type inequality for Sugeno integrals. Nonlinear Analysis: Theory, Methods and Applications 74, 7296–7303 (2011)

12. Jang, L.C.: A note on convergence properties of interval-valued capacity functionals and Choquet integrals. Information Sciences 183, 151–158 (2012)
13. Klein, E., Thompson, A.C.: Theory of Correspondences. Wiley-Interscience Publication, New York (1984)
14. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer Academic Publishers, Dordrecht (2000)
15. Kolokoltsov, V.N., Maslov, V.P.: Idempotent Analysis and Its Applications. Kluwer Academic Publishers, Dordrecht (1997)
16. Krätschmer, V.: Limit theorems for fuzzy-random variables. Fuzzy Sets and Systems 126, 256–263 (2002)
17. Li, L.S., Sheng, Z.: The fuzzy set-valued measures generated by random variables. Fuzzy Sets Systems 97, 203–209 (1998)
18. Maslov, V.P., Samborskij, S.N.: Idempotent Analysis. American Mathematical Society, Providence (1992)
19. Mesiar, R., Pap, E.: Idempotent integral as limit of g -integrals. Fuzzy Sets and Systems 102, 385–392 (1999)
20. Michta, M.: On set-valued stochastic integrals and fuzzy stochastic equations. Fuzzy Sets and Systems 177, 1–19 (2011)
21. Molchanov, I.: Theory of Random Sets. Springer (2005)
22. Pap, E.: g-calculus. Univ. u Novom Sadu Zb. Rad. Prirod.-Mat. Fak. Ser. Mat. 23, 145–156 (1993)
23. Pap, E.: Null-Additive Set Functions. Kluwer Academic Publishers, Dordrecht (1995)
24. Pap, E.: Generalized real analysis and its applications. International Journal of Approximate Reasoning 47, 368–386 (2008)
25. Pap, E., Štrboja, M.: Generalization of the Jensen inequality for pseudo-integral. Information Sciences 180, 543–548 (2010)
26. Perović, A., Ognjanović, Z., Rašković, M., Radojević, R.: Finitely additive probability measures on classical propositional formulas definable by Gödel's t-norm and product t-norm. Fuzzy Sets and Systems 169, 65–90 (2011)
27. Puri, M.L., Ralescu, D.A.: Fuzzy random variables. Journal of Mathematical Analysis and Applications 114, 409–422 (1986)
28. Román-Flores, H., Bassanezi, R.C.: On multivalued fuzzy entropies. Fuzzy Sets Syst. 86, 169–177 (1997)
29. Román-Flores, H., Flores-Franulič, A., Chalco-Cano, Y.: A Jensen type inequality for fuzzy integrals. Inform. Sci. 177, 3192–3201 (2007)
30. Schjaer-Jacobsen, H.: Representation and calculation of economic ucertainties: intervals, fuzzy numbers, and probabilities. Int. J. Prod. Econ. 78, 91–98 (2002)
31. Štajner-Papuga, I., Grbić, T., Štrboja, M.: A note on absolute continuity for the interval-valued measures based on pseudo-integral of interval-valued functions. In: Proc. of SISY 2009, Subotica, pp. 279–284 (2009) ISBN 978-1-4244-5349-8
32. Štrboja, M., Grbić, T., Grujić, G., Mihailović, B., Medić, S.: Chebyshev integral type inequalities for pseudo-integrals of set-valued functions. In: Proc. of SISY 2011, Subotica, pp. 978–971 (2011) ISBN 978-1-4244-2407-8
33. Štrboja, M., Grbić, T., Štajner-Papuga, I., Grujić, G., Medić, S.: Jensen and Chebyshev type inequalities for pseudo-integrals of set-valued functions (submitted)
34. Wang, Z., Klir, G.J.: Fuzzy Measure Theory. Plenum Press, New York (1992)
35. Weichselberger, K.: The theory of interval-probability as a unifying concept for uncertainty. Int. J. Approx. Reas. 24, 149–170 (2000)

GPFCSP Systems: Definition and Formalization

Aleksandar Takači, Aleksandar Perović, and Srdjan Škrbić

Abstract. The aim of this paper is to construct a General Prioritized Fuzzy Satisfaction Problem (GPFCSP) that can handle any logical expression whose atomic symbols are prioritized constraints. GPFCSP can be applied in the field of Fuzzy Relational Databases, multilateral negotiations, decision making etc. The interpretation method is used in order to obtain a complete axiomatization.

Keywords: Decision making, constraint satisfaction problem, priority, fuzzy set, fuzzy constraint satisfaction problem.

1 Introduction

The aim of this paper is to formalize Generalized Constraint Satisfaction Problems(GPFCSP) using the $L\Pi\frac{1}{2}$ logic. GPFCSP system incorporates priorities into constraints allowing calculations of any logical formula whose atomic elements are prioritized constraints. The main contribution of this paper is the definition of GPFCSP and their formalizatiion.

GPFCSP problems have evolved from the concept of constraint satisfaction problem (CSP). The aim of CSP is to find a solution that satisfies all the constraints in optimal time, see [3]. If the satisfaction of the constraint is not a Boolean value, i.e., if there can be many levels of constraint satisfaction, it is clear that there is room for

Aleksandar Takači
Faculty of Technology, University of Novi Sad, Serbia
e-mail: stakaci@tf.uns.ac.rs

Aleksandar Perović
Faculty of Transportation and Traffic Engineering University of Belgrade, Serbia
e-mail: pera@sf.bg.ac.rs

Srdjan Skrbić
Faculty of Natural Sciences, University of Novi Sad Serbia
e-mail: shkrba@uns.ac.rs

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 43–60.
DOI: 10.1007/978-3-642-33959-2_3

inserting fuzzy values and fuzzy logic into CSP. We can model constraints as fuzzy sets over a particular domain. This leads to the fuzzy constraint satisfaction problem (FCSP). Fargier, Lang in [6] interpret the degree of satisfaction of a constraint as the membership degree of its domain value on the fuzzy set that represents it. In order to obtain the global satisfaction degree, we need to aggregate the values of each constraint. For the aggregation operator we use fuzzy logic operators: t-norms (see [16]), t-conorms and fuzzy negation.

Priority is generally viewed as the importance level of an object among others and it is often used in real time systems. PFCSP is actually a fuzzy constraint satisfaction problem (FCSP) in which the notion of priority is introduced. Dubois, Fargier, Prade [4] propose the idea on how to handle priority in decision making. Later, Luo, Jennings, Shadbolt, Leung, Lee [17] and Luo, Lee, Leung, Jennings [18] develop the idea and axiomatize PFSCP. Finally, Takači [35] and Takači, Škrbić [37] generalize PFCSP and obtain Generalized Priority Fuzzy Constraint Satisfaction Problem.

It is not difficult to see that formalization of the reasoning about GPFCSP can be carried out in several well known Hilbert style systems such as ZFC (Zermelo-Fraenkel set theory with the axiom of choice), RCF (the first order theory of real closed fields), $L\Pi\frac{1}{2}$ (a fuzzy logic that combine product logic and Łukasiewicz logic with all rational numbers from the real unit interval as truth constants) etc. However, due to decidability and complexity issues, it is important to identify formalizations that are as simplest as possible in terms of complexity. Here we will propose one such system - the L_{GPFCSP}-logic. The most important properties of L_{GPFCSP}-are:

- The SAT problem for L_{GPFCSP}-formulas is NP-complete;
- L_{GPFCSP} is expressible enough to formally capture decision making according to prioritized queries;
- Strong completeness (every finite satisfiable L_{GPFCSP}-theory is satisfiable).

In section 2 some preliminaries that are needed for better understanding of the paper will be given. In section 3 GPFCSP systems are defined. In section 4 the logic L_{GPFCSP} is defined. Finally, section 5 concludes the chapter.

2 Preliminaries

Definition 1. A fuzzy subset A of a universe X is defined by its membership function $\mu_A : X \to [0,1]$, where $\mu_A(x)$ is the membership degree of element x in fuzzy set A.

Let us recall the definition of t-norms. For more details on t-norms, see [16].

Definition 2. A mapping $T : [0,1]^2 \to [0,1]$ is called a t-norm if the following conditions are satisfied for all $x, y, z \in [0,1]$:

(T1) $T(x,y) = T(y,x)$,
(T2) $T(x,T(y,z)) = T(T(x,y),z)$,
(T3) if $y \leqslant z$ then $T(x,y) \leqslant T(x,z)$,
(T4) $T(x,1) = x$.

The four basic t-norms are:

$$T_M(x,y) = \min(x,y),$$
$$T_P(x,y) = x \cdot y,$$
$$T_L(x,y) = \max(x+y-1,0),$$
$$T_D(x,y) = \begin{cases} 0, & \text{if}(x,y) \in [0,1)^2; \\ \min(x,y), & \text{otherwise.} \end{cases}$$

In fuzzy logic, t-norms are a generalization of the conjunction operator. In order to generalize disjunction, t-conorms or s-norms are used.

Definition 3. A mapping $S : [0,1]^2 \to [0,1]$ is called a s-norm or a t-conorm if the following conditions (T1), (T2), (T3) from the previous definition and the condition (S4) are satisfied for all $x,y,z \in [0,1]$.
(S4) $S(x,0) = x$.

The four basic t-conorms are:

$$S_M(x,y) = \max(x,y),$$
$$S_P(x,y) = x+y-x \cdot y,$$
$$S_L(x,y) = \min(x+y,1),$$
$$S_D(x,y) = \begin{cases} 1, & \text{if}(x,y) \in [0,1)^2; \\ \max(x,y), & \text{otherwise.} \end{cases}$$

3 Generalized Priority Fuzzy Constraint Satisfaction Problem

Constraint satisfaction problems (CSP) have been developed over a long period of time. Basically, they deal with a set of constraints and the means to find a solution, i.e., an evaluation of variables that satisfies all the constraints. Applications of CSP problems are found mostly in scheduling problems, e.g. bus/plain schedules, school timetables, etc. If we cannot precisely determine whether a constraint is satisfied, i.e., if there can be many levels of constraint satisfaction, we can expand CSP allowing a constraint to have a satisfaction degree from the unit interval. In practice, most of constraints have inherited fuzziness (tall, bald, strong, young, age around 24, good stamina, etc.) and they are more naturally represented as fuzzy sets. We can model constraints as fuzzy sets over a particular domain. This leads to the fuzzy constraint satisfaction problems (FCSP). Obviously, the degree of satisfaction of a constraint is the membership degree of its domain value on the fuzzy set that represents it. In order to obtain the global satisfaction degree, we need to aggregate the values of each constraint. For the aggregation operator we can use operators from fuzzy logic: t-norms, t-conorms and fuzzy negation.

Further on, we can add an importance degree to each of the constraints. Besides a satisfaction degree each constraint has its importance value, i.e., priority. Many concepts of priority have been studied. In our paper, we consider priority of constraints as the global importance of a constraint among other ones. The more important the constraint is, the more impact it has on the aggregated output of the PFCSP. If the value of a more important constraint is increased by some value, then the aggregated output should be greater than in the case when the value of the less important constraint is increased by the same. If we try to interpret this concept, we can view constraints as two investments where one of the investments results in a bigger profit margin, i.e., constraint with a larger priority. Obviously, if we have more money to invest, we will invest it in the better earning investment, i.e., increase the value of constraint with the larger priority, which would result in a bigger profit margin. We are dealing with a very strict notion of priority which favors the constraint with the larger priority regardless of its value. Systems that satisfy and implement this notion of priority is found. These systems use well known t-norms and t-conorms (see [35]).

Priority Fuzzy Constraint Satisfaction Problem (PFCSP) make decisions that depend not only on the satisfaction degree of each constraint (which is the case in FCSP), but also on the priority that each constraint has. PFCSP are introduced by an axiomatic framework.

The problem is that FCSP only deals with the conjunction of the constraints. Obviously, PFCSP need to be generalized in order to handle disjunction and negation. The result is that PFCSP systems evolve into a GPFCSP that can handle priorities which are incorporated into each atomic formula (see [37]).

Definition 4. (see [17]) A fuzzy constraint satisfaction problem (FCSP) is a triple $\langle X, D, C^f \rangle$ such that:

- $X = \{x_1, \ldots, x_n\}$ is a set of variables.
- $D = \{d_1, \ldots, d_n\}$ is a set of domains. Each domain d_i is a finite set of possible values for the corresponding variable x_i.
- C^f is a finite nonempty set of elements called fuzzy constraints, where each constraint $C_i \in C^f, i \in \{1, \ldots, m\}$ has a form:

$$C_i : d_{i_1} \times \cdots \times d_{i_k} \longrightarrow [0,1],\ 1 \leq k \leq n.$$

□

The membership degree of each constraint indicates the local degree to which the constraint is satisfied. In order to obtain the global satisfaction degree, local degrees are aggregated using certain t–norm. Adding priorities to the FCSP and allowing constraints to be aggregated by any logical formula produces the GPFCSP.

Before we define GPFCSP, let us recall the well known notion of the compound label. A label of a variable x is an assignment of a value of the variable, denoted as v_x. A compound label v_X of all variables in the set $X = \{x_1, x_2, \ldots, x_n\}$ is a simultaneous assignment i.e. an evaluation of all variables in the set X, that is,

$$v_X = (v_{x_1}, v_{x_2}, \ldots, v_{x_n}).$$

For each assignment we obtain a local satisfaction degree s_i for every constraint. For example, if a constraint $C_i : d_j \longrightarrow [0,1]$ is a fuzzy subset of a domain d_j, $(\mathrm{dom}(C_i) = d_j)$ then we have $s_i = \mu_{C_i}(v_{x_j}),\ v_{x_j} \in d_j$.

Definition 5. Let (X, D, C^f) be a FCSP defined as in definition 4, and let $\rho : C^f \rightarrow [0,\infty)$ and a compound label v_X of all variables in X, and , $g : [0,1) \times [0,\infty] \rightarrow [0,1]$.

Generalized PFCSP is defined as a tuple $(X, D, C^f, \rho, g, \wedge, \vee, \neg)$.

An elementary formula in generalized PFCSP is a pair $(s_i, \rho(C_i))$ where $C_i \in C^f$, s_i represents the local satisfaction degree of a constraint C_i and $p_i = \rho(C_i)$ represents its priority.

A formula in GPFCSP is defined in the following way:

(i) An elementary formula is a formula.
(ii) If f_1 and f_2 are formulas then also $\wedge(f_1, f_2)$, $\vee(f_1, f_2)$ and $\neg(f_1)$ are formulas.

For each valuation v_X a satisfaction degree $\alpha_F(v_X)$ of a formula F is calculated depending on the interpretation of connectives.

A system is a GPFCSP if

1. Let $F = \wedge_{i\in\{1,\ldots,n\}} f_i$ be a formula in GPFCSP where f_i, $i \in \{1,\ldots,n\}$ are elementary formulas and let R^f be a set of constrains that appear in the formula. Let R^f_{max} be a constraint with the highest priority i.e it satisfies the following condition:

$$\rho_{max} = \rho(R^f_{max}) = \max\{\rho(R^f) \mid R^f \in C^f\}.$$

Then for each formula F we have:

$$\mu_{R^f_{max}}(v_X) = 0 \Rightarrow \alpha_F(v_X) = 0.$$

2. If $\forall R^f \in C^f$, $\rho(R^f) = \rho_0$ where $\rho_0 \in \mathbb{R}^+$, then for each formula F it holds:

$$\alpha_F(v_X) = F_{\mathscr{L}}(v_X)$$

where $F_{\mathscr{L}}$ is the interpretation of the logical formula F in fuzzy logic $\mathscr{L}(\wedge, \vee, \neg))$.

3. For R^f_i, $R^f_j \in C^f$, assume $\rho(R^f_i) \geq \rho(R^f_j)$, $\delta > 0$, a small positive number δ and assume that there are two different compound labels v_X and v'_X such that:

- if $\forall R^f \neq R^f_i$ and $\forall R^f \neq R^f_j$, then $\mu_{R^f}(v_X) = \mu_{R^f}(v'_X)$,
- if $R^f = R^f_i$, then $\mu_{R^f}(v_X) = \mu_{R^f}(v'_X) + \delta$,
- if $R^f = R^f_j$, then $\mu_{R^f}(v'_X) = \mu_{R^f}(v_X) + \delta$.

Then the following properties hold for

$$F = \bigwedge_{k=1}^{n} (x_k, \rho(R_k)),\ x_k \in \mathrm{Dom}(R_k)$$

or $F = \bigvee_{k=1}^{n} (x_k, \rho(R_k)),\ x_k \in \mathrm{Dom}(R_k)$:

$$\alpha_F(v_X) \geq \alpha_F(v'_X).$$

4. Assume that two different compound labels v_X and v'_X such that $\forall R^f \in C^f$ satisfy

$$\mu_{R^f}(v_X) \geqslant \mu_{R^f}(v'_X).$$

If formula F that has no negation connective, then it holds

$$\alpha_F(v_X) \geqslant \alpha_F(v'_X).$$

5. Let there be a compound label such that $\forall R^f \in C^f,\ \mu_{R^f}(v_X) = 1$.

If F is a formula $F = \bigwedge_{i=1}^{n} f_i$, where f_i are elementary formulas then

$$\alpha_F(v_X) = 1.$$

The function ρ represents the priority of each constraint. Greater value of $\rho(R^f)$ means that the constraint R^f has larger priority.

On the other hand, the function g aggregates priority of each constraint with the value of that constraint. These aggregated values of each constraint are then aggregated by the operator $\bigoplus$, which results in the satisfaction degree of that valuation.

The first axiom states that, if the constraint with the maximum priority has a zero value of the local satisfaction degree, then the global satisfaction degree should be also zero.

The second axiom states that, if all priorities are equal, the satisfaction degree is the evaluation of the logical formula F.

The third axiom is the most important one. It captures the notion of priority, i.e., if one constraint has a larger priority then, the increase of the value on that constraint should result in a bigger increase of the global satisfaction degree than when the value with the smaller priority has the same increase.

The fourth axiom is the monotonicity property, and finally the fifth is the upper boundary condition.

One of the problems is to find a triple $\wedge, \vee, \neg$ that satisfies the previous properties. The following theorem gives the answer to this question.

Theorem 1. *The following system is a GPFCSP, where* $(X, D, C^f, \rho,\ g, \wedge, \vee, \neg)$ *where* $\wedge = T_L$, $\vee = S_L$, $\neg = N_S$, $\rho : C^f \to [0, \infty)$, $\rho_{\max} = \max\{\rho(C_i), C_i \in C^f\}$ *and finally* $g(x_i, \rho(C_i)) = S_P(x_i, 1 - \frac{\rho(C_i)}{\rho_{max}})$ *. The global satisfaction degree of a valuation* v_X *for a formula* F *is obtained in the following way:*

$$\alpha_F(v_X) = \mathscr{F}\{g(v_{x_i}, \rho(C_i)) | C_i \in C^f\},$$

where T_L is the Lukasiewicz t-norm, S_L is the Lukashiewicz t-conorm, N_S is the standard involutive negation, C^f is the set of constraints of formula F , and $\mathscr{F}$ is the interpretation of formula F in GPFCSP.

Proof. The proof is similar to the proof of Theorem 7 in [35]. Axioms 1,2,4,5 are trivially satisfied.

Now, we will prove that this system satisfies Axiom 3. Let $p_i > p_j,\ \delta > 0$. Then the following inequalitie need to be proven:

$$A(S_P(1-p_i, a_i+\delta), S_P(1-p_j, a_j)) \geq A(S_P(1-p_i, a_i), S_P(1-p_j, a_j+\delta))$$

where $A = T_L$ and $A = S_L$. Actually, it is enough to prove the following inequality:

$$p'_i + a_i + \delta - p'_i * (a_i + \delta) + p'_j + a_j - p_j * a_j \geq p'_i + a_i - p'_i * a_i + p'_j + a_j + \delta - p_j * (a_j + \delta).$$

In fact, after short calculation we obtain that it is equivalent to:

$$-p'_i \delta \geq -p'_j \delta.$$

This is a true statement since $-p'_i \geq -p'_j$ and $\delta \geq 0$ which finally proves our theorem. □

The previous theorem gives us a concrete set of operators whose combination gives us a concrete GPFCSP. We illustrate on an example how a GPFCSP works.

Example 1. Let us give an example of a GPFCSP. Suppose that we require students to promote the University. We rank our candidates based on 3 variables `Age`, `testScore` and `PhE` (physical exam). We have formulated the following grading criteria:

(`Age`="around 24 years" PRIORITY 0.8 OR `testScore`="excellent test results" PRIORITY 0.8) AND (`PhE`="good fitness" PRIORITY 1)

So, we would have the following set of variables and domains:

- $x_1 =$ `Age`, $d_1 = [0, 150]$,
- $x_2 =$ `testScore`, $d_1 = [0, 10]$,
- $x_3 =$ `PhE`, $d_1 = [0, 5]$.

Constraints would be:

- $c_1 =$"around 24 years",
- $c_2 =$"excellent test results",
- $c_3 =$"good fitness".

Constraints c_i are given by fuzzy subsets of the domains d_i. Constraint c_1 is a linear triangular fuzzy number whose center is at the point 24, with left and right tolerance of 2.0. The constraint c_2 is a left shoulder fuzzy set with an increasing membership

function starting at the point 8.0 and increasing linearly up to the point 10.0. Similarly, c_3 is aleft shoulder fuzzy set with an increasing membership function starting at the point 2.0 increasing up to the point 5.0.

The priorities of the constraints are represented by $\rho(c_1) = 8$, $\rho(c_2) = 8$ and $\rho(c_3) = 10$ yielding that physical fitness is the most important, age and test score are less important. Each candidate is represented by an evaluation e which consists of the values of his `Age`, `testScore` and `PhE` respectively. Assume we have data available for 4 students which is given in table 1.

Table 1 Sample test data for testing GPFCSP

Age	testScore	PhE
25	8.2	4
23	8.4	3
24	6.4	4
28	8.6	5

We will explain in detail how the value of the satisfaction degree α is calculated for student 1. First, we normalize the priorities of constraints c_1, c_2 and c_3 in order to obtain normalized priority values for each $\rho_{norm}(c_i) = \frac{\rho(c_i)}{\rho_{max}}$. In this case, $\rho_{max} = \rho(c_3) = 10$. This leads to the following normalized priority values $\rho_{norm}(c_1) = 0.8$, $\rho_{norm}(c_2) = 0.8$ and $\rho_{norm}(c_3) = 1$. Next, the satisfaction degree for each condition (constraint) should be calculated. Since the data values of constraints are all exact, the satisfaction degree for each constraint is obtained as the value of the membership function of each constraint at a particular point. Thus, for the first student we obtain $v_1 = \mu_{c_1}(25) = 0.5$, $v_2 = \mu_{c_2}(8.4) = 0.2$ and $v_3 = \mu_{c_3}(4) = 0.\overline{6}$. Now, we calculate the satisfaction degree, denoted by $\alpha_i, i \in 1,\ldots,4$ for each student using GPFCSP. We will show in detail how it is done for student 1.

$$\begin{aligned}\alpha_1 &= T_L(S_L(S_P(v_1, 1-\rho_{norm}(c_1)), S_P(v_2, 1-\rho_{norm}(c_2))), S_P(v_2, 1-\rho_{norm}(c_3)))\\ &= T_L(S_L(S_P(0.5, 0.2), S_P(0.1, 0.2)), S_P(0.\bar{6}, 0))\\ &= 0,54\bar{6}\end{aligned}$$

Similarly, $\alpha_2 = 0,29\bar{3}$, $\alpha_3 = 0.\bar{6}$ and $\alpha_4 = 0.72$.

4 L_{GPFCSP} Logic

Before we start to formalize reasoning about general prioritized fuzzy constraint satisfaction problem, we should specify what exactly we want to reason about.

Our aim is to capture certain type of classification problems with the classification criteria expressible as prioritized query with fuzzy constraints, i.e. something that looks like the example (3.1.).

To be more precise, we want to develop a logic that can yield formal classification (partial ordering) of any finite or countably infinite set E of $[0,1]$-evaluations of the set X of variables according to the classification criteria which is expressible as a formula of the Łukasiewicz logic L, where each propositional letter is a triple of the form

$$\langle x, c, p\rangle;$$

x is a variable, c is a fuzzy set (fuzzy constraint) and $p \in [0,1] \cap \mathbb{Q}$ is a priority. More on fuzzy logic can be found in [8, 9, 10, 11, 12, 13, 14, 19, 20, 21, 23, 24]. Here we have restricted fuzzy constraints to the following four types: triangular, left shoulder, right shoulder and trapezoidal. For the sake of simplicity, all of them are in normalized form.

Definition 6. Suppose that X is a nonempty set of syntactical objects called variables, E is a nonempty set of $[0,1]$-evaluations of X and that C is a nonempty set of fuzzy sets, where each of them is either triangular, left shoulder, right shoulder or trapezoidal. Furthermore, let L be the Łukasiewicz logic with propositional letters of the form $\langle x, c, p\rangle$, where $x \in X$, $c \in C$ and $p \in [0,1] \cap \mathbb{Q}$.

For each L-formula ϕ we define a function $[\phi] : E \longrightarrow [0,1]$ recursively on complexity of formulas as follows:

- $[\langle x, c, p\rangle](e) = p \cdot c(e(x))$;
- $[\phi \& \psi](e) = \max(0, [\phi](e) + [\psi](e) - 1)$;
- $[\phi \rightarrow_L \psi](e) = \min(1, 1 - [\phi](e) + [\psi](e))$;
- $[\underline{0}](e) = 0$. □

With the notation as in the previous definition, a classification of E according to ϕ is a partial ordering $<_\phi$ defined by

$$e <_\phi e' \text{ iff } [\phi](e) < [\phi](e').$$

Clearly, there are many formal systems that are expressible enough for the computation of $<_\phi$. The most obvious example is ZFC (Zermelo-Fraenkel set theory with axiom of choice); the same can be achieved in RCF (first order theory of the real closed fields) and $L\Pi\frac{1}{2}$ (fuzzy logic that combines Łukasiewicz logic, product logic and rational numbers from the real unit interval as truth constants).

It is well known that ZFC is undecidable, while RCF and $L\Pi\frac{1}{2}$ have far too complex decision procedures (EXPSPACE and PSPACE respectively). We believe it would be of interest to develop a simpler formalism in terms of the complexity of decision procedure that is expressible enough for the formalization of $<_\phi$. In the development of a suitable propositional formal system for our purposes we will closely follow the research presented for various probability logics, see for instance [7, 15, 25, 26, 27, 29, 38].

4.1 Syntax

Let $X = \{x_n \mid n \in \mathbb{N}\}$ be the set of variables for individuals, $\mathtt{E} = \{\mathtt{e}_n \mid n \in \mathbb{N}\}$ be the set of variables for evaluations of X and let $\mathtt{C}$ be the set of variables for fuzzy constraints, where each element of $\mathtt{C}$ has exactly one of the following forms:

- $\mathtt{TRI}(a,b,c)$, where a,b and c are rational numbers from the real unit interval and $0 \leqslant a < b < c \leqslant 1$;
- $\mathtt{LS}(a,b)$, where a and b are rational numbers from the real unit interval and $0 \leqslant a < b \leqslant 1$;
- $\mathtt{RS}(c,d)$, where c and d are rational numbers from the real unit interval and $0 \leqslant c < d \leqslant 1$;
- $\mathtt{TRA}(a,b,c,d)$, where a,b,c and d are rational numbers from the real unit interval and $0 \leqslant a < b < c < d \leqslant 1$.

Let L be the Łukasiewicz logic built over the set

$$\{\langle x, \mathtt{c}, p\rangle \mid x \in X, \mathtt{c} \in \mathtt{C}, p \in [0,1] \cap \mathbb{Q}\}$$

of propositional letters. L-formulas will be denoted by ϕ, ψ and θ, indexed or primed if necessary.

A basic L_{GPFCSP}-term is any syntactical object that has one of the following forms:

- $\underline{q}$, $q \in \mathbb{Q}$;
- $\mathtt{e}_n(x_m)$;
- $\phi(\mathtt{e}_n)$.

The set of all L_{GPFCSP}-terms is the smallest superset of the set of all basic L_{GPFCSP}-terms that is closed under the following formation rules:

- $\langle \underline{p}, f\rangle \mapsto \underline{p}\, f$;
- $\langle f, g\rangle \mapsto (f+g)$;
- $f \mapsto (-f)$.

Variables for L_{GPFCSP}-terms are f, g and h, indexed or primed if necessary. In order to simplify notation we will use the usual convention for the association and omission of parentheses. For instance,

$$3f - g$$

is the L_{GPFCSP}-term

$$(((f+f)+f)+(-g)).$$

The set of all L_{GPFCSP}-formulas is the set of classical propositional formulas built over the set of propositional letters

$$\{f \leqslant \underline{0} \mid f \text{ is an } L_{\mathrm{GPFCSP}} - \text{term}\}.$$

Variables for L_{GPFCSP}-formulas are Φ, Ψ and Θ, indexed or primed if necessary. In order to simplify notation we will use the usual abbreviations: for instance, $f \leqslant g$ is $f - g \leqslant \underline{0}$

4.2 Semantics

Recall that triangular fuzzy number $\mathrm{tri}(a,b,c)$, left shoulder $\mathrm{ls}(a,b)$, right shoulder $\mathrm{rs}(a,b)$ and trapezoidal fuzzy number $\mathrm{tra}(a,b,c,d)$ are defined as follows:

- $\mathrm{tri}(a,b,c)(x) = \begin{cases} 0 & , x \leqslant a \vee x \geqslant c \\ \frac{1}{b-a}x - \frac{a}{b-a} & , \quad a < x \leqslant b \\ \frac{-1}{c-b}x + \frac{c}{c-b} & , \quad b < x \leqslant c, \end{cases}$
- $\mathrm{ls}(a,b)(x) = \begin{cases} 0 & , \quad x \leqslant a \\ 1 & , \quad x \geqslant b \\ \frac{1}{b-a}x - \frac{a}{b-a} & , a < x < b, \end{cases}$
- $\mathrm{rs}(a,b)(x) = \begin{cases} 0 & , \quad x \geqslant b \\ 1 & , \quad x \leqslant a \\ \frac{-1}{b-a}x + \frac{b}{b-a} & , a < x < b, \end{cases}$
- $\mathrm{tra}(a,b,c,d)(x) = \begin{cases} 0 & , x \leqslant a \vee x \geqslant d \\ 1 & , \quad b \leqslant x \leqslant c \\ \frac{1}{b-a}x - \frac{a}{b-a} & , \quad a < x < b \\ \frac{-1}{d-c}x + \frac{d}{d-c} & , \quad c < x < d. \end{cases}$

Here x ranges over the unit real interval $[0,1]$.

A model is any set $E = \{e_n \mid n \in \mathbb{N}\}$ of evaluations $e_n : X \longrightarrow [0,1]$, where $X = \{x_n \mid n \in \mathbb{N}\}$. For each L_{GPFCSP}-term f we define its E-interpretation f^E as follows:

- $\underline{q}^E = q$;
- $\mathtt{e}_n(x_m)^E = e_n(x_m)$;
- $\langle x_m, \mathtt{TRI}(a,b,c), p\rangle(\mathtt{e}_n)^E = 1 - p + p\,\mathrm{tri}(a,b,c)(e_n(x_m))$;
- $\langle x_m, \mathtt{LS}(a,b), p\rangle(\mathtt{e}_n)^E = 1 - p + p\,\mathrm{ls}(a,b)(e_n(x_m))$;
- $\langle x_m, \mathtt{RS}(a,b), p\rangle(\mathtt{e}_n)^E = 1 - p + p\,\mathrm{rs}(a,b)(e_n(x_m))$;
- $\langle x_m, \mathtt{TRA}(a,b,c,d), p\rangle(\mathtt{e}_n)^E = 1 - p + p\,\mathrm{tra}(a,b,c,d)(e_n(x_m))$;
- $(\phi \& \psi)(\mathtt{e}_n)^E = \max\left(0, \phi(\mathtt{e}_n)^E + \psi(\mathtt{e}_n)^E - 1\right)$;
- $(\phi \to_L \psi)(\mathtt{e}_n)^E = \min\left(1, 1 - \phi(\mathtt{e}_n)^E + \psi(\mathtt{e}_n)^E\right)$;
- $(f+g)^E = f^E + g^E$;
- $(\underline{p} f)^E = p f^E$;
- $(-f)^E = -f^E$.

The satisfiability relation $\models$ is defined recursively on complexity of formulas in the following way:

- $E \models f \leqslant \underline{0}$ iff $f^E \leqslant 0$;
- $E \models \Phi \wedge \Psi$ iff $E \models \Phi$ and $E \models \Psi$;
- $E \models \neg\Phi$ iff $E \not\models \Phi$.

Satisfiability, validity and finite satisfiability are defined as usual.

Remark 1. Similarly as in the case of probability logics (see [27]), the compactness theorem (every finite satisfiable theory is satisfiable) fails for L_{GPFCSP} logic as well: it is easy to see that theory

$$T = \{\mathtt{e}_1(x_1) > \underline{0}\} \cup \{\mathtt{e}_1(x_1) < \underline{10^{-n}} \mid n \in \mathbb{N}\}$$

is finitely satisfiable and unsatisfiable. □

4.3 Axiomatization

L_{GPFCSP} logic is a Hilbert style formal system with the following axioms and inference rules:

Propositional axioms

A1 Substitutional instances of classical tautologies;

Bookkeeping axioms

A2 $\underline{p}+\underline{q} = \underline{p+q}$;
A3 $\underline{p}\,\underline{q} = \underline{pq}$;
A4 $\underline{p} \leqslant \underline{q}$ whenever $p \leqslant q$;
A5 $\neg(\underline{p} \leqslant \underline{q})$ whenever $p > q$;

Algebraic axioms

A6 $f+g = g+f$;
A7 $f+(g+h) = (f+g)+h$;
A8 $f+\underline{0} = f$;
A9 $f-f = \underline{0}$;
A10 $\underline{0}f = \underline{0}$;
A11 $\underline{1}f = f$;
A12 $\underline{p}f+\underline{q}f = \underline{p+q}f$;

Equality axioms

A13 $f = f$;
A14 $f = g \to g = f$;
A15 $f = g \wedge g = h \to f = h$;
A16 (substitution) $f = g \to (\Phi(\dots,f,\dots) \to \Phi(\dots,g,\dots))$;

Ordering axioms

A17 $f \leqslant f$;
A18 $f \leqslant g \vee g \leqslant f$;
A19 $f \leqslant g \wedge g \leqslant h \to f \leqslant h$;
A20 $f \leqslant g \to f + h \leqslant g + h$;
A21 $f \leqslant g \wedge \underline{p} > \underline{0} \to \underline{p} f \leqslant \underline{p} g$;

Triangular axioms

A22 $(\mathsf{e}_n(x_m) \leqslant \underline{a} \vee \mathsf{e}_n(x_m) \geqslant \underline{c}) \to$
$\langle x_m, \mathtt{TRI}(a,b,c), p\rangle(\mathsf{e}_n) = \underline{1-p}$;
A23 $\underline{a} < \mathsf{e}_n(x_m) \leqslant \underline{b} \to$
$\langle x_m, \mathtt{TRI}(a,b,c), p\rangle(\mathsf{e}_n) = \underline{p\frac{1}{b-a}}\mathsf{e}_n(x_m) + \underline{1-p-p\frac{a}{b-a}}$;
A24 $\underline{b} < \mathsf{e}_n(x_m) \leqslant \underline{c} \to$
$\langle x_m, \mathtt{TRI}(a,b,c), p\rangle(\mathsf{e}_n) = \underline{-p\frac{1}{c-b}}\mathsf{e}_n(x_m) + \underline{1-p+p\frac{c}{c-b}}$;

Left shoulder axioms

A25 $\mathsf{e}_n(x_m) \leqslant \underline{a} \to \langle x_m, \mathtt{LS}(a,b), p\rangle(\mathsf{e}_n(x_m)) = \underline{1-p}$;
A26 $\mathsf{e}_n(x_m) \geqslant \underline{b} \to \langle x_m, \mathtt{LS}(a,b), p\rangle(\mathsf{e}_n(x_m)) = \underline{1}$;
A27 $\underline{a} < \mathsf{e}_n(x_m) < \underline{b} \to$
$\langle x_m, \mathtt{LS}(a,b), p\rangle(\mathsf{e}_n(x_m)) = \underline{p\frac{1}{b-a}}\mathsf{e}_n(x_m) + \underline{1-p-p\frac{a}{b-a}}$;

Right shoulder axioms

A28 $\mathsf{e}_n(x_m) \leqslant \underline{a} \to \langle x_m, \mathtt{RS}(a,b), p\rangle(\mathsf{e}_n(x_m)) = \underline{1}$;
A29 $\mathsf{e}_n(x_m) \geqslant \underline{b} \to \langle x_m, \mathtt{RS}(a,b), p\rangle(\mathsf{e}_n(x_m)) = \underline{1-p}$;
A30 $\underline{a} < \mathsf{e}_n(x_m) < \underline{b} \to$
$\langle x_m, \mathtt{RS}(a,b), p\rangle(\mathsf{e}_n(x_m)) = \underline{-p\frac{1}{b-a}}\mathsf{e}_n(x_m) + \underline{1-p+p\frac{b}{b-a}}$;

Trapezoidal axioms

A31 $(\mathsf{e}_n(x_m) \leqslant \underline{a} \vee \mathsf{e}_n(x_m) \geqslant \underline{d}) \to \langle x_m, \mathtt{TRA}(a,b,c,d), p\rangle(\mathsf{e}_n) = \underline{1-p}$;
A32 $\underline{b} \leqslant \mathsf{e}_n(x_m) \leqslant \underline{c} \to \langle x_m, \mathtt{TRA}(a,b,c,d), p\rangle(\mathsf{e}_n) = \underline{1}$;
A33 $\underline{a} < \mathsf{e}_n(x_m) < \underline{b} \to$
$\langle x_m, \mathtt{TRA}(a,b,c,d), p\rangle(\mathsf{e}_n) = \underline{p\frac{1}{b-a}}\mathsf{e}_n(x_m) + \underline{1-p-p\frac{a}{b-a}}$;
A34 $\underline{c} < \mathsf{e}_n(x_m) < \underline{d} \to$
$\langle x_m, \mathtt{TRA}(a,b,c,d), p\rangle(\mathsf{e}_n) = \underline{-p\frac{1}{d-c}}\mathsf{e}_n(x_m) + \underline{1-p+p\frac{d}{d-c}}$;

Conjunction axioms

A35 $\phi(\mathsf{e}_n) + \psi(\mathsf{e}_n) < \underline{1} \to (\phi \& \psi)(\mathsf{e}_n) = \underline{0}$;
A37 $\phi(\mathsf{e}_n) + \psi(\mathsf{e}_n) \geqslant \underline{1} \to (\phi \& \psi)(\mathsf{e}_n) = \phi(\mathsf{e}_n) + \phi(\mathsf{e}_n) - \underline{1}$;

Implication axioms

A38 $\phi(\mathsf{e}_n) \leqslant \psi(\mathsf{e}_n) \to (\phi \& \psi)(\mathsf{e}_n) = \underline{1}$;
A39 $\phi(\mathsf{e}_n) > \psi(\mathsf{e}_n) \to (\phi \to_L \psi)(\mathsf{e}_n) = 1 - \phi(\mathsf{e}_n) + \phi(\mathsf{e}_n)$.

Modus ponens

R1 From Φ and $\Phi \to \Psi$ infer Ψ;

Archimedean rule

R2 From the set of premises

$$\{\Phi \to f \leqslant \underline{p + (n+1)^{-1}} \mid n \in \mathbb{N}\}$$

infer $\Phi \to f \leqslant \underline{p}$.

Remark 2. The present form of the Archimedean rule is well known to the researches in probability logics, since it is a standard tool for the achievement of strong completeness in real valued probability logics, see for instance [26, 27, 29]. The Archimedean rule should be understood as follows: if the value of f is infinitely close to the rational number a, than it must be equal to a. □

Remark 3. The proof of the strong completeness theorem for L_{GPFCSP} is a straightforward modification of the corresponding proofs for real valued probability logics presented in [26, 27, 29]. It is based on the following steps:

1. Completion technique. Using the pseudo code, we can describe it as follows: input = consistent T; output = its completion T^*;

   $T \mapsto T^*$;
   $For \mapsto F$;
   while $F \neq \emptyset$ do
   chose $\Phi \in F$;
   $F \setminus \{\Phi\} \mapsto F$;
   if $T^* \cup \{\Phi\}$ is consistent, then $T^* \cup \{\Phi\} \mapsto T^*$;
   if $T^* \cup \{\Phi\}$ is inconsistent and Φ cannot be derived by the application of Archimedean rule, then $T^* \cup \{\neg\Phi\} \mapsto T^*$;
   else, find a premise Ψ in derivation of Φ by Archimedean rule so that $T^* \cup \{\neg\Psi\}$ is consistent; $T^* \cup \{\neg\Phi, \neg\Psi\} \mapsto T^*$; $F \setminus \{\Psi\} \mapsto F$;
   return T^*;

2. The canonical model $E = \{e_n \mid n \in \omega\}$ is defined by

$$e_n(x_m) = \sup\{a \in [0,1]_{\mathbb{Q}} \mid T^* \vdash \mathsf{e}_n(x_m) \geqslant \underline{a}\};$$

3. $E \models \Phi$ iff $T^* \vdash \Phi$. The proof can be carried out by induction on complexity of Φ. It is reasonably a straightforward consequence of the following two facts:

   - Triangular, left and right shoulder and trapezoidal fuzzy sets are continuous functions;
   - For each positive integer k, $\mathbb{Q}^k$ is dense in $\mathbb{R}^k$.

Any eventual doubt should be easily dispelled once the reader becomes familiar with at least one completeness proof for probability logics with Archimedean rule, see for instance [27]. □

Remark 4. Concerning decidability, it is easy to see that basic L_{GPFCSP}-formulas are actually linear inequalities with rational coefficients in variables $\mathsf{e}_n(x_m)$. As a consequence, the satisfiability problem for the L_{GPFCSP}-formulas is NP-complete, since it can be reduced to the satisfiability of systems of linear inequalities with rational coefficients. □

4.4 Formalization of GPFCSP in L_{GPFCSP}

As we have mentioned earlier, the structure of each query in prioritized fuzzy relational database has the following form:

SELECT from TABLE by CRITERION

In normalized form, each table t, which is defined by

Table 2 Definition of data table

	x_0	...	x_k
e_0	a_{00}	...	a_{0k}
⋮			
e_l	a_{l0}	...	a_{lk}

can be represented by an L_{GPFCSP}-formula Φ_t defined by

$$\bigwedge_{i=0}^{l}\bigwedge_{j=0}^{k}\mathsf{e}_i(x_j)=\underline{a_{ij}}.$$

Moreover, each criterion c is expressible as an L-formula ϕ_c, so the resulting classification (partial ordering) $\langle\{e_0,\ldots,e_l\},\prec\rangle$ is expressible by an L_{GPFCSP}-formula $\Phi_{\langle\{e_0,\ldots,e_l\},\prec\rangle}$ defined by

$$\bigwedge_{e_i\prec e_j}\phi_c(\mathsf{e}_i)<\phi_c(\mathsf{e}_j).$$

Finally, $\Phi_{\langle\{e_0,\ldots,e_l\},\prec\rangle}$ is derivable from Φ_t. We will illustrate this by the explicit L_{GPFCSP}-formalization of the example 2.1.

The corresponding L_{GPFCSP}-formula Φ_t is formed as it is described above. Furthermore, the corresponding criterion in the example 2.1 can be formally represented by the L-formula ϕ defined by

$$(\langle x_0,\mathtt{TRI}(22/150,24/150,26/150),0.8\rangle\oplus\langle x_1,\mathtt{LS}(0.8,1),0.8\rangle)\&\langle x_2,\mathtt{LS}(2/5,1),1\rangle,$$

where $\oplus$ is the Łukasiewicz disjunction

Table 3 Normalized version of the corresponding table t

	x_0	x_1	x_2
e_0	$\frac{25}{150}$	0.82	$2/3$
e_1	$\frac{23}{150}$	0.84	$1/3$
e_2	$\frac{24}{150}$	0.64	$2/3$
e_3	$\frac{28}{150}$	0.86	1

$$\psi \oplus \theta \Leftrightarrow_{\text{def}} (\psi \to_L \underline{0}) \& (\theta \to_L \underline{0}) \to_L \underline{0}.$$

Now we can easily derive the following:

- $\vdash \Phi_t \to \underline{0.5466} \leqslant \phi(\mathtt{e}_0) \leqslant \underline{0.5467}$;
- $\vdash \Phi_t \to \underline{0.2933} \leqslant \phi(\mathtt{e}_1) \leqslant \underline{0.2934}$;
- $\vdash \Phi_t \to \underline{0.6666} \leqslant \phi(\mathtt{e}_2) \leqslant \underline{0.6667}$;
- $\vdash \Phi_t \to \underline{0.7200} \leqslant \phi(\mathtt{e}_3) \leqslant \underline{0.7201}$.

Finally, the above enable us to formally derive in L_{GPFCSP} the corresponding classification:

- $\vdash \Phi_t \to \phi(\mathtt{e}_0) > \phi(\mathtt{e}_1)$;
- $\vdash \Phi_t \to \phi(\mathtt{e}_0) < \phi(\mathtt{e}_2)$;
- $\vdash \Phi_t \to \phi(\mathtt{e}_0) < \phi(\mathtt{e}_3)$;
- $\vdash \Phi_t \to \phi(\mathtt{e}_1) < \phi(\mathtt{e}_2)$;
- $\vdash \Phi_t \to \phi(\mathtt{e}_1) < \phi(\mathtt{e}_3)$;
- $\vdash \Phi_t \to \phi(\mathtt{e}_2) < \phi(\mathtt{e}_3)$.

5 Concluding Remarks

In this paper, we have presented a logic based on a concrete set of operators that satisfies GPFCSP (Theorem 1), and applied the interpretation method in order to obtain a complete and decidable axiomatization.

Some of the results presented here were announced in ([28], [37]). We are aware of only a few papers that address the similar problems. Usually, problems concerning the priority of constraints are studied without a proper logic formalization ([2, 17, 18]). Interpretation method given here is one of the possible syntactical approaches. A part of the future work will include the formalization of certain aspects of GPFCSP within the framework of the $L\Pi\frac{1}{2}$ logic.

We also plan to expand our research towards the applications of GPFCSP, especially in the field of fuzzy relational databases. In particular, we intend to formalize fuzzy relational databases in such a way that will allow the application of the methodology presented here.

Acknowledgements. This paper was written with the support of the The Serbian Ministry of Sciences and Environmental Protection Projects No 174009 and The Academy of Sciences and Arts of Vojvodina. The authors would like to thank the anonymous referees whose remarks have subsequently improved the quality of the paper.

References

1. Canny, J.: Some algebraic and geometric computations in PSPACE. In: Proc. of XX ACM Symposium on Theory of Computing, pp. 460–467 (1987)
2. Dubois, D., Fargier, H., Prade, H.: Possibility theory in constraint satisfaction problems: handling priority, preference and uncertanty. In: Yager, R., Zadeh, L. (eds.) Fuzzy Sets, Neural Networks and Soft Computing, pp. 166–187 (1994)
3. Dubois, D., Fortemps, P.: Computing improved optimal solutions to max-min flexible constraint satisfaction problems. European Journal of Operational Research 118, 95–126 (1999)
4. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In: Gabbay, D.M., Hogger, C.J., Robinson, J.A., Nute, D. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming. Nonmonotonic Reasoning and Uncertain Reasoning, vol. 3, pp. 439–513. Clarendon Press, Oxford (1994)
5. Dubois, D., Prade, H.: Possibilistic logic: a retrospective and prospective view. Fuzzy Sets and Systems 144(1), 3–23 (2004)
6. Fargier, H., Lang, J.: Uncertainty in Constraint Satisfaction Problems: a Probalistic Approach. ECSQARU, pp. 97–104 (1993)
7. Fagin, R., Halpern, J., Megiddo, N.: A logic for reasoning about probabilities. Information and Computation 87(1-2), 78–128 (1990)
8. Godo, L., Marchioni, E.: Coherent conditional probability in a fuzzy logic setting. Logic Journal of the IGPL 14(3), 457–481 (2006)
9. Hájek, P.: Metamathematics of fuzzy logic. Kluwer Academic Publishers (1998)
10. Horcik, R., Cintula, P.: Product Łukasiewicz logic. Arch. Math. Log. 43(4), 477–504 (2004)
11. Esteva, F., Godo, L., Montagna, F.: The $L\Pi$ and $L\Pi\frac{1}{2}$ logics: two complete fuzzy systems joining Lukasiewicz and Product Logics. ARCH 40(1), 39–67 (2001)
12. Flaminio, T.: NP-containment for the coherence test of assessments of conditional probability: a fuzzy logic approach. Arch. Math. Log. 46(3-4), 39–67 (2007)
13. Flaminio, T.: Strong non-standard completeness for fuzzy logics. Soft Comput. 12(4), 321–333 (2008)
14. Gerla, B., Rational, B.: Łukasiewicz logic and DMV-algebras. Neural Networks World 11, 579–584 (2011)
15. Keisler, H.J.: Probability quantifiers. In: Barwise, J., Feferman, S. (eds.) Model–Theoretic Logics. Perspectives in Mathematical Logic. Springer (1985)
16. Klement, E., Mesiar, R., Pap, E.: Triangular norms, Series: Trends in Logic, vol. 8. Kluwer Academic Publishers, Dordrecht (2000)
17. Luo, X., Jennings, N.R., Shadbolt, N., Leung, H., Lee, J.H.: A fuzzy constraint based model for bilateral multi-isssue negotiations in semi competitive enviroments. Artificial Intelligence 148, 53–102 (2003)
18. Luo, X., Lee, J.H., Leung, H., Jennings, N.R.: Prioritised fuzzy constraint satisfaction problems: axioms, instantiation and validation. Fuzzy Sets and Systems 136, 151–188 (2003)

19. Marchioni, E., Godo, L.: A Logic for Reasoning About Coherent Conditional Probability: A Modal Fuzzy Logic Approach. In: Alferes, J.J., Leite, J. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 213–225. Springer, Heidelberg (2004)
20. Marchioni, E., Montagna, F.: Complexity and Definability issues in $L\Pi\frac{1}{2}$. JLC 17(2), 311–331 (2007)
21. Marchioni, E., Montagna, F.: On triangular norms and uninorms definable in $L\Pi\frac{1}{2}$. Int. J. Approx. Reasoning 47(2), 179–201 (2008)
22. Marker, D.: Model Theory. Springer (2002)
23. Montagna, F.: An algebraic approach to propositional fuzzy logic. JLLI 9(1), 91–124 (2000)
24. Montagna, F., Subreducts, F.: of MV-algebras with product and product residuation. Algebra Universalis 53, 109–137 (2005)
25. Nilsson, N.: Probabilistic logic. Artif. Intell. 28, 71–78 (1986)
26. Ognjanović, Z., Perović, A., Rašković, M.: Logics with the qualitative probabilty operator. Logic Journal of the IGPL 16(2), 105–120 (2008)
27. Ognjanović, Z., Rašković, M., Marković, Z.: Probability Logics. In: Ognjanović, Z. (ed.) Logic in Computer Science. Mathematical Institute of Serbian Academy of Sciences and Arts, pp. 35–111 (2009)
28. Perović, A., Jovanović, M., Jovanović, A.: An application of Interpretation Method in the Axiomatization of the Lukasziewicz Logic. Acta Polytechnica Hungarica, 95–104 (2008)
29. Perović, A., Ognjanović, Z., Rašković, M., Marković, Z.: A Probabilistic Logic with Polynomial Weight Formulas. In: Hartmann, S., Kern-Isberner, G. (eds.) FoIKS 2008. LNCS, vol. 4932, pp. 239–252. Springer, Heidelberg (2008)
30. Perović, A., Takači, A., Škrbić, S.: Formalising PFSQL queries using 23 LP 1/2 fuzzy logic. Mathematical Structures in Computer Science, Available on CJO (2011), doi:10.1017/S0960129511000673
31. Perović, A., Škrbić, S., Takači, A.: Towards the Formalization of Fuzzy Relational DatabaseQueries. Acta Politechnica Hungarica 6(1), 185–193 (2009)
32. Schweizer, B., Sklar, A.: Probabilistic metric spaces. North Holand, New York (1983)
33. Škrbić, S., Racković, M., Takači, A.: Prioritized fuzzy logic based information processing in relational databases. Knowledge-Based Systems (in press, 2012), http://dx.doi.org/10.1016/j.knosys.2012.01.017
34. Škrbić, S., Racković, M., Takači, A.: Development of fuzzy relational database applications. Computer Scienece and Information Systems 8(1), 27–40 (2011)
35. Takači, A.: Schur-concave triangular norms: Characterization and application in pFCSP. Fuzzy Sets and Systems 155, 50–64 (2005)
36. Takači, A.: Handling Priority Within a Database Scenario. ETF Journal of Electrical Engineering, 130–134 (2006)
37. Takači, A., Škrbić, S.: Data Model of FRDB with Different Data Types and PFSQL. In: Handbook of Research on Fuzzy Information Processing in Databases, Hershey, PA, USA. Information Science Reference (2008) (in print)
38. van der Hoek, W.: Some considerations on the logic P_FD: a logic combining modality and probability. Journal of Applied Non-Classical Logics 7(3), 287–307 (1997)
39. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)

Choquet Integrals and T-Supermodularity

Martin Kalina, Maddalena Manzi, and Biljana Mihailović

Abstract. This paper presents new results about T-supermodularity for Choquet integral. This property was introduced in [17] in order to extend the concept of supermodularity for Choquet integral, by using Frank t-norms and considering the particular case of two membership functions such that their minimum is zero. Now we consider general membership functions and we use properties and links among belief measures, Möbius transform and Choquet integrals, in order to present the general case studied over a finite set.

Keywords: Triangular norms, fuzzy measures, fuzzy sets, belief measure, necessity measure, Möbius transform, Choquet integral, T-supermodularity.

1 Introduction

Nonlinear integrals defined with respect to a fuzzy measure are often used in economics, engineering, pattern recognition and decision analysis as nonlinear aggregation tools [13, 16, 26, 31]. In the literature a set function which is monotone and such that its value at the empty set equals zero is known by various names, such as capacity, monotonic cooperative game, non-additive measure, fuzzy measure. In this

Martin Kalina
Department of Mathematics, Faculty of Civil Engineering,
Slovak University of Technology, Radlinského 11, 813 68 Bratislava, Slovakia
e-mail: kalina@math.sk

Maddalena Manzi
Deams "Bruno de Finetti", University of Trieste,
Piazzale Europa 1, 34127 Trieste, Italy
e-mail: maddalena.manzi@econ.units.it

Biljana Mihailović
Faculty of Technical Sciences, University of Novi Sad,
Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia
e-mail: lica@uns.ac.rs

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 61–75.
DOI: 10.1007/978-3-642-33959-2_4

paper we will call it a *fuzzy measure*. The fuzzy measure is a non-negative set function, obviously the requirement $\mu(\emptyset) = 0$ ensures its non-negativity. To any fuzzy measure μ we can link its Möbius transform m, vanishing at the empty set, which can take negative values. A characterization of Möbius transform of fuzzy measures was given in [9], and some of its properties were studied in [9, 12]. The application in combinatorics was presented in [28].

The most important integral defined with respect to fuzzy measures is the Choquet integral, introduced by G. Choquet in [10]. The main properties of Choquet integral are monotonicity and comonotonic additivity, but it is not additive, in general ([11, 26]). If the underlying fuzzy measure is supermodular (resp. submodular), then the Choquet integral is superadditive (resp. subadditive) and, in the case of additive measures on the finite set, the Choquet integral is equal to the Lebesgue integral ([5, 11, 26, 30]).

In the fuzzy set theory, introduced by L. Zadeh in [33], the two-place functions on the unit interval, triangular norms (t-norms, for short) and triangular conorms (t-conorms) were used for defining intersection and union of fuzzy sets. We recall also that t-norms are bounded from above with the minimum, and t-conorms are bounded from below with the maximum and all of them are used in fuzzy logic, with wide applications in many fields of science and real life. In [3] evaluators with respect to t-norms T, respectively t-conorms S, were introduced and their integral representations were studied in [4].

In [17], the authors introduced T-supermodularity of Choquet integral in order to generalize the property of supermodularity.

In this article we continue to study T-supermodularity for Choquet integral in the general case of membership functions of all fuzzy sets on a finite set $\mathbf{X}_n \neq \emptyset$. The paper is organized as follows. In Section 2 we recall t-norms and t-conorms, and in the next two sections we recall fuzzy sets and fuzzy measures, in particular belief and necessity measures. In Section 5 we present Choquet integral. In Section 6 we define T-supermodularity, provide some examples and prove that the Choquet integral with respect to belief measures is T-supermodular for an arbitrary Frank t-norm T. Finally, the S-superadditivity of Choquet integral is considered for Frank t-conorms and some concluding remarks are given.

Throughout the paper, we assume a finite space $\mathbf{X}_n$ with n elements, i.e. $\mathbf{X}_n = \{1, 2, \ldots, n\}$. By $|A|$ we will denote the cardinality of a subset $A \subset \mathbf{X}_n$. Moreover, we denote by $\wedge$, $\vee$ the minimum and maximum operators on the real line.

2 Triangular Norms

A triangular norm (briefly t-norm) T is defined to be a two-place function

$$T : [0,1] \times [0,1] \to [0,1]$$

fulfilling the following properties:

1. $T(1,y) = y$, for each $y \in [0,1]$;
2. $T(x,y_1) \leq T(x,y_2)$, for all $x,y_1,y_2 \in [0,1]$, if $y_1 \leq y_2$;
3. $T(x,y) = T(y,x)$, for all $x,y \in [0,1]$;
4. $T(x,T(y,z)) = T(T(x,y),z)$, for all $x,y,z \in [0,1]$.

Note that a t-norm defines an Abelian monoid on $[0,1]$ with unit 1 and annihilator 0 and where the semigroup operation is order-preserving.
Given a t-norm T, the two-place function

$$S : [0,1] \times [0,1] \to [0,1],$$

defined by

$$S(x,y) = 1 - T(1-x, 1-y)$$

is called a t-conorm (or the dual of T). Obviously S fulfills monotonicity, commutativity, associativity and

$$S(x,0) = x \quad \forall x \in [0,1], \qquad \textit{Boundary} \quad \textit{Condition.}$$

Here we deal with the Frank family of t-norms T_λ, $\lambda \in [0,\infty]$.
For each $\lambda \in [0,\infty]$, the *Frank* t-norms are defined by the following formulas

- the minimum t-norm, $T_0(x,y) := \min\{x,y\}$,
- the product t-norm, $T_1(x,y) := x \cdot y$,
- the Łukasiewicz t-norm $T_\infty(x,y) := \max\{0, x+y-1\}$,
- if $\lambda \in (0,\infty) \setminus \{1\}$,

$$T_\lambda(x,y) := \log_\lambda \left[1 + \frac{(\lambda^x - 1)(\lambda^y - 1)}{\lambda - 1} \right], \tag{1}$$

where usually we denote $T_0 = T_M$, $T_1 = T_P$ and $T_\infty = T_L$.
The basic t-conorms (dual to four basic t-norms) are:

- the maximum t-conorm, $S_0(x,y) := \max\{x,y\}$,
- the probabilistic sum, $S_1(x,y) := x + y - x \cdot y$,
- the Łukasiewicz t-conorm $S_\infty(x,y) := \min\{1, x+y\}$,
- if $\lambda \in (0,\infty) \setminus \{1\}$,

$$S_\lambda(x,y) := 1 - \log_\lambda \left[1 + \frac{(\lambda^{1-x} - 1)(\lambda^{1-y} - 1)}{\lambda - 1} \right] \tag{2}$$

and similarly we denote $S_0 = S_M$, $S_1 = S_P$, $S_\infty = S_L$.
The family $\{T_\lambda | \lambda \in [0,\infty]\}$ appeared first in Frank's [14] investigation of the functional equation

$$x + y = T(x,y) + S(x,y), \quad \forall x,y \in [0,1], \tag{3}$$

where T is a triangular norm and S is an associative function on the unit square. Note that the only strict solutions of (3) are just t-norms T_λ for $\lambda \in (0,\infty)$ and the corresponding S_λ are just the dual t-conorms, i.e., $S_\lambda(x,y) = 1 - T_\lambda(1-x, 1-y)$.

In particular Frank [14] showed that the t-norms T_λ, $0 \leq \lambda \leq \infty$, form a parametric family in the sense that T_M, T_P and T_L are the limits of T_λ corresponding to their subscripts. The system of all Frank t-norms (i.e., all solutions of (3)) is the set $\{T_\lambda | \lambda \in [0,\infty]\}$ and ordinal sums of t-norms T_λ for $\lambda \in [0,\infty]$.

3 Fuzzy Sets

Following [32, 34], we recall the definition of a fuzzy set.

Let $n \in \mathbb{N}$ and $\mathbf{X}_n$ be a nonempty finite set $\mathbf{X}_n = \{1,\ldots,n\}$, a fuzzy subset F of $\mathbf{X}_n$ is defined by a characteristic function $f : \mathbf{X}_n \to [0,1]$ which associates with each i in $\mathbf{X}_n$ its "grade of membership", $f(i)$ in $[0,1]$. To distinguish between the characteristic function of a nonfuzzy set and the characteristic function of a fuzzy set, the latter will be referred to as a *membership* function.

A fuzzy set F is *normal* if

$$\sup_{i \in \mathbf{X}_n} f(i) = \max_{i \in \mathbf{X}_n} f(i) = 1.$$

Since any crisp set A can be defined by its characteristic function $\chi_A : \mathbf{X}_n \to \{0,1\}$, it is a special fuzzy set, which is normal if $A \neq \emptyset$.

Note that the vector $\mathbf{x} = (x_1,\ldots,x_n) \in [0,1]^n$ corresponds to the membership function f defined on $\mathbf{X}_n$, i.e. $\mathbf{x}$ can be understood as a membership function $f : \mathbf{X}_n \to [0,1]$, $f(i) = x_i$, $i \in \mathbf{X}_n$. So, $f \equiv \mathbf{x}$ and from now on we will use the vector symbol to denote the membership function of fuzzy subsets of $\mathbf{X}_n$.

Let $\mathscr{F}(\mathbf{X}_n)$ denote the family of all membership functions of all fuzzy sets on $\mathbf{X}_n$. For $\mathbf{x},\mathbf{y} \in \mathscr{F}(\mathbf{X}_n)$, such that $\mathbf{x} : \mathbf{X}_n \to [0,1]$ and $\mathbf{y} : \mathbf{X}_n \to [0,1]$, $\mathbf{x} \leq \mathbf{y}$ means that $\mathbf{x}(i) \leq \mathbf{y}(i)$ $\forall i \in \mathbf{X}_n$. Further, $(\mathbf{x} \vee \mathbf{y})(i) = \max\{\mathbf{x}(i),\mathbf{y}(i)\}$ and $(\mathbf{x} \wedge \mathbf{y})(i) = \min\{\mathbf{x}(i),\mathbf{y}(i)\}$.

Definition 1. Let F and G be fuzzy subsets of $\mathbf{X}_n$ whose membership functions are $\mathbf{x}$ and $\mathbf{y}$, respectively. If $\mathbf{x}(i) \leq \mathbf{y}(i)$ for any $i \in \mathbf{X}_n$, we say that the fuzzy set F is *included* in the fuzzy set G and we write $F \subset G$. If $F \subset G$ and $G \subset F$, we say that F and G are *equal*, which we write as $F = G$.

Definition 2. Let F and G be fuzzy sets. The *standard union* of F and G, $F \cup G$, is defined by

$$(\mathbf{x} \vee \mathbf{y})(i) = \mathbf{x}(i) \vee \mathbf{y}(i), \qquad \forall i \in \mathbf{X}_n.$$

Definition 3. Let F and G be fuzzy sets. The *standard intersection* of F and G, $F \cap G$, is defined by

$$(\mathbf{x} \wedge \mathbf{y})(i) = \mathbf{x}(i) \wedge \mathbf{y}(i), \qquad \forall i \in \mathbf{X}_n.$$

Similar to the way operations on ordinary sets are treated, we can generalize the standard union and the standard intersection for an arbitrary class of fuzzy sets: if $\{F_r | r \in R\}$ is a class of fuzzy sets, where R is an arbitrary index set, then $\cup_{r \in R} F_r$

is the fuzzy set having membership function $\sup_{r\in R}\mathbf{x}_r(i)$, $i\in\mathbf{X}_n$, and $\cap_{r\in R}F_r$ is the fuzzy set having membership function $\inf_{r\in R}\mathbf{x}_r(i)$, $i\in\mathbf{X}_n$.

Definition 4. Let F be a fuzzy set. The *standard complement* of F, $\bar{F}$, is defined by the membership function

$$\mathbf{x}^c(i) := 1-\mathbf{x}(i), \qquad \forall i\in\mathbf{X}_n.$$

Two or more of the three basic operations can also be combined. For example, concerning the difference $F-G$ of fuzzy sets F and G, we have the following membership function

$$(\mathbf{x}-\mathbf{y})(i) := \max\{0,\mathbf{x}(i)-\mathbf{y}(i)\},$$

for all $i\in\mathbf{X}_n$.

Alsina et al. [1] and Prade [27] suggested to use a t-norm for intersection and its dual t-conorm for union of fuzzy sets. The extension of the operations intersection, union and complementation in ordinary set theory to fuzzy sets was always done pointwise: one considered two two-place functions $T:[0,1]\times[0,1]\to[0,1]$, $S:[0,1]\times[0,1]\to[0,1]$ and one-place function $(\cdot)^c:[0,1]\to[0,1]$ and extended them in the usual way: if $\mathbf{x}$, $\mathbf{y}$ are two membership functions of fuzzy sets F and G, then

$$T(\mathbf{x},\mathbf{y})(i) = T(\mathbf{x}(i),\mathbf{y}(i)), \tag{4}$$
$$S(\mathbf{x},\mathbf{y})(i) = S(\mathbf{x}(i),\mathbf{y}(i)), \tag{5}$$
$$(\mathbf{x})^c(i) = (\mathbf{x}(i))^c. \tag{6}$$

4 Fuzzy Measures and Möbius Transform

Each integration method on $\mathbf{X}_n$ is based on a set function $\mu:\mathscr{P}(\mathbf{X}_n)\to[0,1]$ (fulfilling certain properties), which can be understood as a system of weights of the relevant set of criteria $\mathbf{X}_n$.

Definition 5. Let $\mathbf{X}_n=\{1,2,\ldots,n\}$ be a fixed set of criteria. A mapping $\mu:\mathscr{P}(\mathbf{X}_n)\to[0,1]$ is called a fuzzy measure, whenever $\mu(\emptyset)=0$, $\mu(\mathbf{X}_n)=1$ and for all $A\subset B\subset\mathbf{X}_n$, it holds $\mu(A)\le\mu(B)$.

The second requirement means that a greater set of criteria B cannot have the weight which is less than the weight of a smaller set of criteria A. As we have mentioned already in Introduction, set functions with these properties are known under several names. For example, these functions are called pre-measures in [30], capacities in [10], but the most popular seems to be the name *fuzzy measures*, introduced by Sugeno and therefore in what follows we will adopt this terminology.

A fuzzy measure μ^d is said to be *dual* to μ if for all $A\subset\mathbf{X}_n$ we have

$$\mu^d(A) = 1-\mu(A^c).$$

For a fixed $k \geq 2$, μ is called *k-monotone* if for all $E_1, \ldots, E_k \in \mathscr{P}(\mathbf{X}_n)$ we have

$$\mu\left(\bigcup_{i=1}^{k} E_i\right) \geq \sum_{\emptyset \neq J \subseteq \{1,\ldots,k\}} (-1)^{|J|+1} \mu\left(\bigcap_{j \in J} E_j\right). \tag{7}$$

Moreover, if a fuzzy measure μ satisfies (7) for all $k \geq 2$ then μ is called *totally monotone* or a *belief measure*. For more details see [26, 32].
2-monotonicity is known also as *supermodularity*, a widely used concept in the theory of non-additive measures.
With the inequality reversed we have *submodularity* and when $A \cap B = \emptyset$, we have respectively *super/subadditivity*.

Remark 1. (i) If a fuzzy measure is both submodular and supermodular, it is modular and thus a probability measure on $\mathbf{X}_n$. Evidently, each supermodular fuzzy measure is also superadditive and similarly, each submodular fuzzy measure is subadditive.

(ii) It is evident that $(\mu^d)^d = \mu$ and that μ is submodular if and only if μ^d is supermodular. Concerning subadditivity and superadditivity, they are not dual properties, i.e., a dual μ^d of a subadditive fuzzy measure μ need not be superadditive, and vice versa.

For an arbitrary finite set A $|A|$ will denote its cardinality. Any fuzzy measure μ can be uniquely expressed in terms of its *Möbius transform* or *Möbius representation* given by $m^\mu : \mathscr{P}(\mathbf{X}_n) \to \mathbb{R}$, such that

$$m^\mu(A) = \sum_{B \subseteq A} (-1)^{|A \setminus B|} \mu(B), \quad \forall A \subset \mathbf{X}_n, \tag{8}$$

so that we have

$$\mu(A) = \sum_{B \subseteq A} m^\mu(B), \quad \forall A \subset \mathbf{X}_n. \tag{9}$$

In a more general way, the Möbius transform provides an inversion formula useful in combinatorics (see [9] and [28]). In the following, the superscripts μ will be omitted.

Of course, a function $m : \mathscr{P}(\mathbf{X}_n) \to \mathbb{R}$ does not necessarily correspond to the Möbius transform of a fuzzy measure on $\mathbf{X}_n$. The boundary and monotonicity conditions must be ensured, as established in the following result (see Proposition 2 in [9]).

Proposition 1 *μ is a fuzzy measure if and only if its Möbius transform satisfies*

$$m(\emptyset) = 0, \quad \sum_{B \subset \mathbf{X}_n} m(B) = 1, \quad \sum_{B \subset A} m(B) \geq 0, \quad A \subset \mathbf{X}_n.$$

In particular, it is necessary that $m(\{i\}) \geq 0$, $\forall i \in \mathbf{X}_n$.

Since $\mathbf{X}_n$ is finite, A is also finite and if we denote $A_i = A \setminus \{i\}$, then $A = \cup_{i=1}^{n} A_i$, so that the previous expression (8) becomes

$$\begin{aligned} m(A) &= \sum_{B\subset A} (-1)^{|A\setminus B|}\mu(B) \\ &= \mu(A) - \sum_{\emptyset\neq I\subset\{1,\ldots,n\}} (-1)^{|I|+1}\mu(\cap_{i\in I}A_i) \\ &= \mu(\cup_{i=1}^{n}A_i) - \sum_{\emptyset\neq I\subset\{1,\ldots,n\}} (-1)^{|I|+1}\mu(\cap_{i\in I}A_i). \end{aligned} \tag{10}$$

Hence, we can conclude that for a belief measure we obtain $m(A)\geq 0, \forall A\subset \mathbf{X}_n$ (see also Corollary 1 in [9]).

Definition 6. Let T be a t-norm. A fuzzy measure $\mu : \mathscr{P}(\mathbf{X}_n)\to[0,1]$ is called a T-measure if for all $A,B\in\mathscr{P}(\mathbf{X}_n)$, such that $A\cup B = X$, we have

$$\mu(A\cap B) = T(\mu(A),\mu(B)). \tag{11}$$

A special type of T-measures, when $T = T_M$ is called a *necessity measure* [13].

For $\mathbf{X}_n$ finite, i.e. our case, any necessity measure is a special example of a belief measure (see Theorem 4.26 in [32]).

5 Choquet Integral for Membership Functions with Respect to Belief Measures

Now we are focusing our attention to the integral with respect to *nonadditive fuzzy measures*, known as the *Choquet integral*, which is useful in many fields such as *mathematical economics* and *multicriteria decision making*.
Let $\mathscr{F}(\mathbf{X}_n)$ denote the family of all membership functions of all fuzzy sets on a finite set $\mathbf{X}_n\neq\emptyset$, and $\mu : \mathscr{P}(\mathbf{X}_n)\to[0,1]$ a fuzzy measure. Then we can give the following definition of the Choquet integral for a membership function $\mathbf{x}\in\mathscr{F}(\mathbf{X}_n)$ with respect to a fuzzy measure μ.

Definition 7. Let μ be a fuzzy measure defined on $\mathbf{X}_n$. The Choquet integral of a membership function $\mathbf{x}$ is the operator $\mathrm{Ch}_\mu : [0,1]^n\to[0,1]$ given by

$$\mathrm{Ch}_\mu(\mathbf{x}) = (C)\int \mathbf{x}d\mu = \sum_{i=1}^{n} x_{(i)}[\mu(X_{(i)}) - \mu(X_{(i+1)})], \tag{12}$$

where $(\cdot)$ indicates a permutation on $\mathbf{X}_n$ such that $x_{(1)}\leq\cdots\leq x_{(n)}$. Furthermore $X_{(i)} := \{(i),\ldots,(n)\}$ and $X_{(n+1)} := \emptyset$.

Observe that if we fix the order of input arguments, e.g. $x_1\leq\cdots\leq x_n$, then (12) becomes a weighted mean $\sum_{i=1}^{n} w_i x_i$ with weights

$$w_i = \mu(X_i) - \mu(X_{i+1}), \quad i = 1,\ldots,n \quad \text{and} \quad \sum_{i=1}^{n} w_i = 1.$$

If we consider the space of all fuzzy measures $V := (\mathbf{X}_n, \mathscr{P}(\mathbf{X}_n), \mu)$ and we define $\forall A \subset \mathbf{X}_n$

$$\mu_B(A) = \begin{cases} 1 & B \subset A \\ 0 & otherwise \end{cases}$$

then the set $\{\mu_B\}_{B \subset \mathbb{X}_n \setminus \{\emptyset\}}$ is a linear basis for V, such that

$$\mu(A) = \sum_{B \subset \mathbf{X}_n \setminus \{\emptyset\}} m(A)\mu_B(A) \tag{13}$$

and the function $m : \mathscr{P}(\mathbf{X}_n) \to \mathbb{R}$ given by (10) is the unique system of coefficients satisfying (13). This result is proved for the general case of real-valued functions defined on arbitrary finite lattices in [15] and is used in [16] in order to express Choquet integral with respect to the Möbius transform, i.e. in Theorem 4.3 that now we recall.

Theorem 1. *For every $\mu \in V$ and $\mathbf{x} \in \mathscr{F}(\mathbf{X}_n)$, the following result holds:*

$$\mathrm{Ch}_\mu(\mathbf{x}) = \sum_{A \subset \mathbf{X}_n} m(A) \bigwedge_{i \in A} x_i. \tag{14}$$

6 T-Supermodular Choquet Integrals

Now we recall the definition of *T-supermodularity* for Choquet integrals (see Definition 8 in [17]).

Definition 8. Let T be a Frank t-norm. For a fixed fuzzy measure μ, the corresponding Choquet integral is said to be *T-supermodular* if it satisfies the following relation:

$$\mathrm{Ch}_\mu(T(\mathbf{x},\mathbf{y})) + \mathrm{Ch}_\mu(S(\mathbf{x},\mathbf{y})) \geq \mathrm{Ch}_\mu(\mathbf{x}) + \mathrm{Ch}_\mu(\mathbf{y}), \tag{15}$$

$\forall \mathbf{x}, \mathbf{y} \in \mathscr{F}(\mathbf{X}_n)$. If the equality holds, we say that this Choquet integral is *T-modular*.

Remark 2. (i) Note that the T-modularity was exactly examined by Butnariu and Klement with the name T-measures, introduced in [8] as a natural generalization of $\sigma-$additive measures on $\sigma-$algebras (see also [2], [5], [6], [7], [18], [19], [20], [23] and [25]).

(ii) When we use the minimum t-norm and its dual t-conorm, we have $\wedge$-supermodular Choquet integral, i.e. standard supermodularity.

(iii) In the case of a finite set $\mathbf{X}_n$ and $T = T_L$, the T-modular Choquet integral is the Lebesgue integral.

Based on the previous definition, the following proposition holds:

Proposition 2 *Consider a finite set* $\mathbf{X}_n$ *and arbitrary two membership functions* $\mathbf{x}$ *and* $\mathbf{y}$. *If* μ *is a belief measure on* $\mathscr{P}(\mathbf{X}_n)$, *then the Choquet integral is* T*-supermodular on* $\mathscr{F}(\mathbf{X}_n)$ *for all Frank t-norms* T.

Proof. We have the following membership functions: $\mathbf{x} = (x_1,\ldots,x_n)$, $\mathbf{y} = (y_1,\ldots,y_n)$, $T(\mathbf{x},\mathbf{y}) = (t_1,\ldots,t_n)$ and $S(\mathbf{x},\mathbf{y}) = (s_1,\ldots,s_n)$. We observe that $t_i = T(\mathbf{x},\mathbf{y})(i) = T(\mathbf{x}(i),\mathbf{y}(i)) = T(x_i,y_i)$ and $s_i = S(\mathbf{x},\mathbf{y})(i) = S(\mathbf{x}(i),\mathbf{y}(i)) = S(x_i,y_i)$, $\forall i = 1,\ldots,n$.
So, thanks to (14), we can write:

$$\mathrm{Ch}_\mu(T(\mathbf{x},\mathbf{y})) + \mathrm{Ch}_\mu(S(\mathbf{x},\mathbf{y})) = \sum_{A\subset\mathbf{X}_n} m(A)\left[\bigwedge_{i\in A} T(x_i,y_i) + \bigwedge_{i\in A} S(x_i,y_i)\right]. \tag{16}$$

Moreover, thanks to non-decreasingness of t-norms with respect to each argument, we have

$$T(x_i,y_i) \geq T\big(\bigwedge_{i\in A} x_i, \bigwedge_{i\in A} y_i\big), \quad \forall i = 1,\ldots,n$$

and hence

$$\bigwedge_{i\in A} T(x_i,y_i) \geqslant T\big(\bigwedge_{i\in A} x_i, \bigwedge_{i\in A} y_i\big).$$

Similarly, the same inequality holds for t-conorms. Hence, we have

$$\mathrm{Ch}_\mu(T(\mathbf{x},\mathbf{y})) + \mathrm{Ch}_\mu(S(\mathbf{x},\mathbf{y})) \geqslant \sum_{A\subset\mathbf{X}_n} m(A)\left[T\big(\bigwedge_{i\in A} x_i, \bigwedge_{i\in A} y_i\big) + S\big(\bigwedge_{i\in A} x_i, \bigwedge_{i\in A} y_i\big)\right]$$

and for the right side of the last inequality we can use equation (3), i.e.

$$T\big(\bigwedge_{i\in A} x_i, \bigwedge_{i\in A} y_i\big) + S\big(\bigwedge_{i\in A} x_i, \bigwedge_{i\in A} y_i\big) = \bigwedge_{i\in A} x_i + \bigwedge_{i\in A} y_i. \tag{17}$$

So, Choquet integral is T-supermodular with T being an arbitrary Frank t-norm whenever μ is a belief measure. □

Remark 3. We have shown that the Choquet integral with respect to belief measures is T-supermodular for an arbitrary Frank t-norm T. In the next example we show a measure that is not a belief one, but the Choquet integral with respect to this measure is supermodular. But it remains an open problem if there exists a measure μ that is not a belief one and such that the Choquet integral is T-supermodular with respect to μ for all Frank t-norms T.

Example 1. Consider $\mathbf{X}_n = \{1,2,3\}$ and a fuzzy measure $\mu : \mathscr{P}(\mathbf{X}_n) \to [0,1]$, given by its Möbius transform:

$$m(A) = \begin{cases} 0, & \text{if } A = \emptyset, \\ 0.2, & \text{if } |A| = 1, \\ 0.15, & \text{if } |A| = 2, \\ -0.05, & \text{if } A = \mathbf{X}_n. \end{cases}$$

We show that the Choquet integral with respect to μ is supermodular. For arbitrary membership functions $\mathbf{x},\mathbf{y}$ (fuzzy subsets of $\mathbf{X}_n$) we have on the one hand

$$\begin{aligned}&(C)\int \min\{\mathbf{x},\mathbf{y}\}\,d\mu + (C)\int \max\{\mathbf{x},\mathbf{y}\}\,d\mu = 0.2\sum_{i=1}^{3}(\min\{x_i,y_i\}+\max\{x_i,y_i\})\\ &+\ 0.15\sum_{i<j}\left(\min\{x_i,y_i,x_j,y_j\}+\min\left\{\max\{x_i,y_i\},\max\{x_j,y_j\}\right\}\right) \qquad (18)\\ &-\ 0.05\left(\min\{x_1,y_1,x_2,y_2,x_3,y_3\}+\min\{\max\{x_1,y_1\},\max\{x_2,y_2\},\max\{x_3,y_3\}\}\right),\end{aligned}$$

and on the other hand

$$\begin{aligned}&(C)\int \mathbf{x}\,d\mu + (C)\int \mathbf{y}\,d\mu = 0.2\sum_{i=1}^{3}(x_i+y_i)+0.15\sum_{i<j}(\min\{x_i,x_j\}+\min\{y_i,y_j\})\\ &-\ 0.05\left(\min\{x_1,x_2,x_3\}+\min\{y_1,y_2,y_3\}\right). \qquad (19)\end{aligned}$$

Obviously, we have $0.2\sum_{i=1}^{3}(\min\{x_i,y_i\}+\max\{x_i,y_i\}) = 0.2\sum_{i=1}^{3}(x_i+y_i)$. Without loss of generality we can assume that $x_1 = \min\{x_1,x_2,x_3,y_1,y_2,y_3\}$. Then, excluding equal expressions of (18) and (19), we get that the Choquet integral with respect to μ is supermodular iff the following holds

$$\begin{aligned}&0.15\left(\min\{x_2,x_3,y_2,y_3\}+\min\{y_1,\max\{x_2,y_2\}\}+\min\{y_1,\max\{x_3,y_3\}\}\right)\\ &+\ 0.15\min\{\max\{x_2,y_2\},\max\{x_3,y_3\}\}-0.05\min\{y_1,\max\{x_2,y_2\},\max\{x_3,y_3\}\}\\ &\geq\ 0.15\left(\min\{x_2,x_3\}+\min\{y_1,y_2\}+\min\{y_1,y_3\}+\min\{y_2,y_3\}\right) \qquad (20)\\ &-\ 0.05\min\{y_1,y_2,y_3\}.\end{aligned}$$

Now we will consider two cases.

- First, assume that $y_1 = \min\{y_1,y_2,y_3\}$. Then from the left-hand side of (20) we have

$$\begin{aligned}&0.15\left(\min\{x_2,x_3,y_2,y_3\}+\min\{y_1,\max\{x_2,y_2\}\}+\min\{y_1,\max\{x_3,y_3\}\}\right)\\ &+0.15\min\{\max\{x_2,y_2\},\max\{x_3,y_3\}\}-0.05\min\{y_1,\max\{x_2,y_2\},\max\{x_3,y_3\}\}\\ &=0.15\left(\min\{x_2,x_3,y_2,y_3\}+\min\{\max\{x_2,y_2\},\max\{x_3,y_3\}\}\right)+0.25y_1\\ &\geq 0.15\left(\min\{x_2,x_3\}+\min\{y_2,y_3\}\right)+0.25y_1,\end{aligned}$$

  where we have used (17). The right-hand side of the last inequality is equal to the right-hand side of (20).
- Assume that $y_2 = \min\{y_1,y_2,y_3\}$. Further, let

$$\min\{y_1,\max\{x_2,y_2\},\max\{x_3,y_3\}\} = \min\{y_1,\max\{x_2,y_2\}\}.$$

  Then from the left-hand side of (20) we get

$$
\begin{aligned}
&0.15\,(\min\{x_2,x_3,y_2,y_3\}+\min\{y_1,\max\{x_2,y_2\}\}+\min\{y_1,\max\{x_3,y_3\}\})\\
&+0.15\min\{\max\{x_2,y_2\},\max\{x_3,y_3\}\}-0.05\min\{y_1,\max\{x_2,y_2\},\max\{x_3,y_3\}\}\\
&=0.15\,(\min\{x_2,x_3,y_2\}+\min\{\max\{x_2,y_2\},\max\{x_3,y_3\}\}+\min\{y_1,\max\{x_3,y_3\}\})\\
&+0.1\min\{y_1,\max\{x_2,y_2\}\}\geq 0.15\,(\min\{x_2,x_3\}+y_2+\min\{y_1,y_3\})+0.1y_2.
\end{aligned}
$$

The case $\min\{y_1,\max\{x_2,y_2\},\max\{x_3,y_3\}\}=\min\{\max\{x_2,y_2\},\max\{x_3,y_3\}\}$ would be treated similarly.

We have shown that the Choquet integral with respect to μ, when we integrate functions with range from $[0,1]$, is supermodular.

We give also an example of belief fuzzy measure and its respective Choquet integral. As proved in [22], we observe that this fuzzy measure is supermodular (see also [3]). Now we check if this is a necessity measure and thus a belief measure, because $\mathbf{X}_n$ is finite.

Example 2. Let $\mathbf{X}_n=\{1,2,3\}$ and a fuzzy measure $\mu:\mathscr{P}(\mathbf{X}_n)\to[0,1]$ be given for all $A\in\mathscr{P}(\mathbf{X}_n)$ by

$$\mu(A)=\begin{cases}\frac{1}{4-|A|} & \text{if} \quad 1\in A,\\ 0 & \text{otherwise.}\end{cases}$$

First of all we check that this is a necessity measure. We note that we can exchange A with B and also 2 with 3, so that we consider the following two cases:

1. $A=\{1\}$ and $B=\{2,3\}$;
2. $A=\{1,2\}$ and $B=\{3\}$.

In both of these situations we have $A\cup B=X$ and $A\cap B=\emptyset$, so that $\mu(A\cap B)=0=\mu(A)\wedge\mu(B)$ and we can conclude that μ is a necessity measure and thus a belief measure, because $\mathbf{X}_n$ is finite.

Now we construct two membership functions $\mathbf{x}$ and $\mathbf{y}$.

Table 1 Membership functions $\mathbf{x}$ and $\mathbf{y}$

	1	2	3
$\mathbf{x}$	0.1	0.2	0.5
$\mathbf{y}$	0.8	0.3	0.4

Computing the Choquet integrals of $\mathbf{x}$ and $\mathbf{y}$ with respect tu μ we get

1. (C) $\int\mathbf{x}d\mu=\int_0^1\mu(\{i\in\mathbf{X}_n:\mathbf{x}(i)\geq t\})dt=\int_0^{0.1}\mu(\{\mathbf{X}_n\})dt=0.1$;
2. (C) $\int\mathbf{y}d\mu=\int_0^1\mu(\{i\in\mathbf{X}_n:\mathbf{y}(i)\geq t\})dt=\int_0^{0.3}\mu(\{\mathbf{X}_n\})dt+\int_{0.3}^{0.4}\mu(\{1,3\})dt+$ $+\int_{0.4}^{0.8}\mu(\{1\})dt=0.3+0.05+0.13=0.48.$

We consider the fuzzy sets $T_P(\mathbf{x},\mathbf{y})$ and $S_P(\mathbf{x},\mathbf{y})$ and their respective Choquet integrals:

1. (C) $\int T_P(\mathbf{x},\mathbf{y})d\mu=\int_0^1\mu(\{i\in\mathbf{X}_n:(\mathbf{x}\cdot\mathbf{y})(i)\geq t\})dt=\int_0^{0.06}\mu(\{\mathbf{X}_n\})dt+$ $+\int_{0.06}^{0.08}\mu(\{1,3\})dt=0.06+0.01=0.07$;

Table 2 Membership functions of $T_P(\mathbf{x},\mathbf{y})$ and $S_P(\mathbf{x},\mathbf{y})$

	1	2	3
$T_P(\mathbf{x},\mathbf{y})$	0.08	0.06	0.2
$S_P(\mathbf{x},\mathbf{y})$	0.82	0.44	0.7

2. (C) $\int S_P(\mathbf{x},\mathbf{y})d\mu = \int_0^1 \mu(\{i \in \mathbf{X}_n : (\mathbf{x}+\mathbf{y}-\mathbf{x}\cdot\mathbf{y})(i) \geq t\})dt = \int_0^{0.44} \mu(\{\mathbf{X}_n\})dt +$
$+ \int_{0.44}^{0.7} \mu(\{1,3\})dt + \int_{0.7}^{0.82} \mu(\{1\})dt = 0.44 + 0.13 + 0.04 = 0.61.$

Finally we have that $0.61 + 0.07 = 0.68 \geq 0.58$ and so the Choquet integral is T-supermodular on $\mathscr{F}(\mathbf{X}_n)$.

Remark 4. We know already that, for $\mathbf{x},\mathbf{y}$ the Choquet integral is S-subadditive in the following sense

$$(C)\int (\mathbf{x}\vee\mathbf{y})\,d\mu \leq S\left((C)\int \mathbf{x}\,d\mu,(C)\int \mathbf{y}\,d\mu\right)$$

if $\mu(A\cup B) \leq S(\mu(A),\mu(B))$ for an arbitrary t-conorm S that is dominated by weighted arithmetic means (see [4]). Further, it is known that if μ is supermodular then the Choquet integral is superadditive (see [11]), i.e.

$$(C)\int (\mathbf{x}+\mathbf{y})\,d\mu \geq (C)\int \mathbf{x}\,d\mu + (C)\int \mathbf{y}\,d\mu. \tag{21}$$

Now, we would like to answer the question whether formula (21) can be generalized in the following sense

$$(C)\int S(\mathbf{x},\mathbf{y})\,d\mu \geq S\left((C)\int \mathbf{x}\,d\mu,(C)\int \mathbf{y}\,d\mu\right), \tag{22}$$

whenever $\mathbf{y}$ is S-residual to $\mathbf{x}$, i.e.,

$$\mathbf{y} \leq \inf\{\mathbf{z} : \mathbf{X}_n \to [0,1]; S_\lambda(\mathbf{x},\mathbf{z}) \leq 1\}, \tag{23}$$

if S is a Frank t-norm and μ is a T-supermodular measure for T being dual to S. Of course, for the case when $T = \min$ and $S = \max$ the inequality (22) is trivially fulfilled due to the monotonicity of the Choquet integral. We show that $S = S_M$ or $S = S_L$ are the only cases out of the family of Frank t-conorms $\{S_\lambda\}_{\lambda\in[0,\infty]}$ for which the S-superadditivity is fulfilled.

Theorem 2. *Let S_λ be a Frank t-conorm for $\lambda \in [0,\infty]$ and $\mathbf{x},\mathbf{y}$ be such that $\mathbf{y}$ is S_λ-residual to $\mathbf{x}$. Further, let $\mu : \mathscr{P}(\mathbf{X}_n) \to [0,1]$ be a T_λ-supermodular measure, possessing at least 3 values. Then inequality (22) is fulfilled for S_λ if and only if $\lambda \in \{0,\infty\}$, i.e., iff $S_\lambda = S_M$ or $S_\lambda = S_L$.*

Proof. For $S_\lambda = S_M$ the inequality (22) is trivially fulfilled due to the monotonicity of Choquet integral. For $S_\lambda = S_L$ and $\mathbf{x}, \mathbf{y}$ such that $\mathbf{y} \leq 1 - \mathbf{x}$ inequalities (21) and (22) are equivalent (see [11]).

Consider an arbitrary set $A \subset \mathbf{X}_n$ and let $\lambda \in [0, \infty]$. Denote $S_{\lambda,R} = \sup\{z \in [0,1]; S_\lambda(z,z) < 1\}$ and let χ_A be the characteristic function of A and $b \in [0, S_{\lambda,R}[$. Define $\mathbf{x} = \mathbf{y} = b\chi_A$. Then inequality (22) gives

$$(C)\int S_\lambda(\mathbf{x},\mathbf{y})\,d\mu = \mu(A)\,S_\lambda(b,b) \geq S_\lambda(b\,\mu(A), b\,\mu(A)) \tag{24}$$
$$= S_\lambda\left((C)\int \mathbf{x}\,d\mu, (C)\int \mathbf{y}\,d\mu\right),$$

which is a necessary condition for S_λ to yield the S_λ-superadditivity of the Choquet integral. This condition is fulfilled for $S_\lambda = S_M$ and $S_\lambda = S_L$. Inequality (24) means that, for $x \in [0, S_{\lambda,R}[$ the diagonal section $S_\lambda(x,x)$ of the t-conorm S_λ has to be a convex function. On the other hand we have $S_\lambda(x,x) \geq x$ for an arbitrary t-conorm. For all Frank t-conorms S_λ, except for the Łukasiewicz one, $S_{\lambda,R} = 1$. This gives $S_\lambda(x,x) = x$ for all $x \in [0,1]$ and arbitrary $S_\lambda \neq S_L$, and hence $S_\lambda = S_M$ or $S_\lambda = S_L$. □

7 Concluding Remarks

In this paper we have analyzed T-supermodularity of Choquet integrals and showed that the Choquet integral with respect to belief measures is T-supermodular for the whole family of Frank t-norms. However, the question whether this is also a necessary condition under which the Choquet integral is T-supermodular for all Frank t-norms T, is still open. In Example 1 we have shown a measure that is not a belief one and such that the corresponding Choquet integral is T-supermodular for $T = \min$.

Moreover, we have given also an example of T-supermodular Choquet integral in the case of a necessity measure which is a particular belief measure for a finite space $\mathbf{X}_n$. As the next step we want to continue and study T-super- or submodularity also for Sugeno and Shilkret integrals.

Acknowledgements. The first author was supported by the Science and Technology Assistance Agency under the contract No. APVV-0073-10 and by VEGA grant agency, grant No. 1/0143/11. The second author wishes to thank Professor Silvano Holzer for comments and discussions. The third author was supported by the project MNTRS 174009 and by Vojvodina Provincial Secretariat for Science and Technological Development.

References

1. Alsina, C., Trillas, E., Valverde, L.: On non-distributive logical connectives for fuzzy sets theory. Busefal 3, 18–29 (1980)

2. Barbieri, G., Weber, H.: A representation theorem and a Lyapunov theorem for T_s-measures: the solution of two problems of Butnariu and Klement. J. Math. Anal. Appl. 244, 408–424 (2000)
3. Bodjanova, S., Kalina, M.: T-evaluators and S-evaluators. Fuzzy Sets and Syst. 160(14), 1965–1983 (2009)
4. Bodjanova, S., Kalina, M.: Fuzzy integral-based T- and S-evaluators. Shilkret and Choquet integrals. Integr. Math. Theory and Appl. 2(4), 3–26 (2012)
5. Butnariu, D.: Additive fuzzy measures and integrals I. J. Math. Anal. Appl. 93, 436–452 (1983)
6. Butnariu, D.: Values and cores of fuzzy games with infinitely many players. Int. J. Game Theory 16, 43–68 (1987)
7. Butnariu, D., Klement, E.P.: Triangular norm-based measures and their Markov kernel representation. J. Math. Anal. Appl. 162, 111–143 (1991)
8. Butnariu, D., Klement, E.P.: Triangular norm-based measures and games with fuzzy coalitions. Kluwer, Dordrecht (1993)
9. Chateauneuf, A., Jaffray, J.Y.: Some characterizations of lower probabilities and other monotone capacities through the use of Möbius inversion. Math. Soc. Sci. 17, 263–283 (1989)
10. Choquet, G.: Theory of capacities. Ann. Inst. Fourier (Grenoble) 5, 131–295 (1955)
11. Denneberg, D.: Non-additive measure and integral. Kluwer, Dordrecht (1994)
12. Denneberg, D., Grabisch, M.: Interaction transform of set functions over a finite set. Inf. Sci. 121, 149–170 (1999)
13. Dubois, D., Prade, H.: Fuzzy sets and systems: theory and applications. Academic Press, New York (1980)
14. Frank, M.J.: On the simoultaneous associativity of $f(x,y)$ and $x+y-f(x,y)$. Aequ. Math. 19, 194–226 (1979)
15. Gilboa, I., Lehrer, E.: Global games. Int. J. of Game Theory 20, 129–147 (1991)
16. Gilboa, I., Schmeidler, D.: Additive representations of non-additive measures and the Choquet integral. Ann. of Oper. Res. 52, 43-65 (1994)
17. Kalina, M., Manzi, M., Mihailovic, B.: T-supermodularity of Choquet integrals. In: Proceedings of SISY 2011, Subotica, pp. 71–74 (2011)
18. Klement, E.P.: Characterization of finite fuzzy measures using Markoff-kernels. J. Math. Anal. Appl. 75, 330–339 (1980)
19. Klement, E.P.: Characterization of fuzzy measures constructed by means of triangular norms. J. Math. Anal. Appl. 86, 345–358 (1982)
20. Klement, E.P.: Construction of fuzzy σ-algebras using triangular norms. J. Math. Anal. Appl. 85, 543–565 (1982)
21. Klement, E.P., Mesiar, R., Pap, E.: Measure-based aggregation operators. Fuzzy Sets and Syst. 142, 3–14 (2004)
22. Manzi, M.: New construction methods for copulas and the multivariate case. PhD thesis, Department of Pure and Applied Mathematics, University of Padova (2011)
23. Mesiar, R., Navara, M.: T_s-tribes and T_s-measures. J. Math. Anal. Appl. 201, 91–102 (1996)
24. Murofushi, T., Soneda, S.: Techniques for reading fuzzy measures (III): interaction index. In: Proceedings of 9th Fuzzy System Symposium, Sapporo, pp. 693–696 (1993) (in Japanese)
25. Navara, M.: Characterization of measures based on strict triangular norms. J. Math. Anal. Appl. 236, 370–383 (1999)
26. Pap, E.: Null-additive set functions. Kluwer, Dordrecht (1995)

27. Prade, H.: Unions et intersections d'ensembles flous. Busefal 3, 58–62 (1980)
28. Rota, G.C.: On the foundations of combinatorial theory I. Theory of Möbius Functions. Z. Für Wahrscheinlichkeitstheorie und Verwandte Geb. 2, 340–368 (1964)
29. Shapley, L.S.: A value for n-person games. In: Kuhn, H.W., Tucker, A.W. (eds.) Contributions to the Theory of Games II. Ann. of Math. Stud, vol. 28, pp. 307–317. Princeton University Press, Princeton (1953)
30. Šipoš, J.: Integral with respect to a pre-measure. Math. Slovaca 29, 141–145 (1979)
31. Wang, Z., Klir, G.J.: Fuzzy measure theory. Plenum Press, New York (1992)
32. Wang, Z., Klir, G.J.: Generalized measure theory. Springer, New York (2009)
33. Zadeh, L.A.: Fuzzy sets. Inform. Control. 8, 338–353 (1965)
34. Zadeh, L.A.: Probability measures of fuzzy events. J. Math. Anal. Appl. 23, 421–427 (1968)

Part II
Robots

Two Particularities Concerning Robots

Jozef Kelemen

Dedicated to the Alan M. Turing Centenary.

Abstract. The contribution, in its first part (chapters 1-4), emphasizes some of the differences between human beings and robots, and in its second part (chapters 5-7) also some differences between computers and their programming, and robots and programming of robots. It recognize programming as an integral part of the overall human culture, and formulates complementing the usual Turing hypothesis, another Turing hypothesis rooted in the famous Turing test. As an example of focusing students attention to the differences in programming computers and robots the article provides some examples of the teaching experiences of the Institute of Computer Science of the Silesian University in Opava, Czech Republic.

Keywords: Robot, mechanical lady bug, Koala robot, Khepera robot.

1 Introduction - From Traditional Views towards the Baudrillad's Question

The traditional and broadly accepted definition of the machine has been and still up to today is related with physics. Machines of the previous centuries have been considered as man-made physical systems working deterministically in some physically well-defined cycles, and intended to concentrate the dispersion of the energy in order to do some economically meaningful (valuable) physical work. A famous example of such a type of machine is the steam-engine which predestined the revolution of the whole industry during the 19^{th} century.

Jozef Kelemen
Silesian University, Institute of Computer Science and IT4I Institute,
Opava, Czech Republic
e-mail: kelemen@fpf.slu.cz

and

VSM School of Management, Bratislava, Slovakia

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 79–91.
DOI: 10.1007/978-3-642-33959-2_5

The industrial revolution of the 19^{th} century accelerated during the 20^{th} century owing to the machines intended for information processing. This technical development brings about dramatic scientific as well as social and cultural changes, and considerably influenced also the self-image of the man of the West. Karel Čapek as one among the first realized the social transformation of the human being to the form of the machine of some specific kind - of a dehumanized being with only one reason - to participate in mass production in a role of certain kind of machine called in Čapek's famous theatre play R. U. R. (first on the stage in January 1921 in the Czech National Theatre in Prague). However, Čapek's imagination gives to his robots another view which becomes to be usual in the sci-fi literature and in the movies after the first presentation of robots look (Figure 1 provides the costume of robots from the first night presentation of the R. U. R.). Might be because of the rapid development of the mechanization of the mass production, the general picture of the robots have been changed, and e.g. the robot from the movie Metropolis (directed by Fitz Lang) from 1927 presents an (electro)mechanical albeit visually a human-like device (see the right part of the Fig. 1).

The above mentioned but only very shortly illustrated developmental line of the culture of West lead us to completely new and fundamental problems of our present critical thinking, technical creativity, and to different new forms of our artistic expressions, too. *Am I a man or am I a machine?* Asked the philosopher Jean Baudrillard [3] his colleagues in his speech at the *Ars Electronica* Festival in Linz (Austria), September 14, 1988, and he answered immediately to himself: *Virtually and physically we are approaching machines*. Where has the Baudrillard's idea (and many similar ideas that appear so often in contemplations of so many intellectuals of the West during the 20^{th} century) its roots?

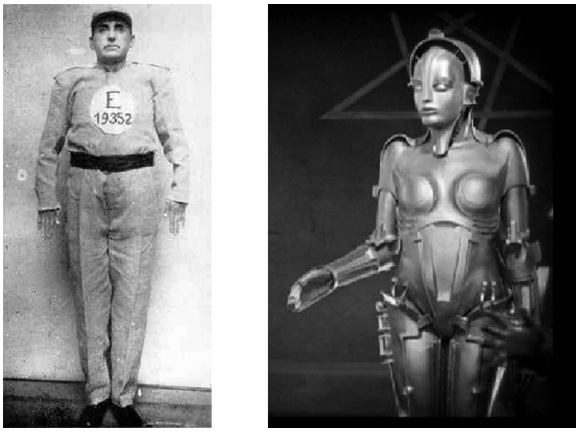


Fig. 1 The picture of the first robot (the costume designed by Josef Čapek for the first night of the Karel Čapek's R. U. R., 1921), and the robot from Fritz Lang's movie Metropolis (1927)

2 Roots and Shifts

In some of our previous publications - in [12], [11], and [10] for instance - we have analyzed from a specific cultural perspectives the common features and principal differences which robots and human beings form. First, let us sketch in very short some of these differences and similarities. Then, in the second part of this contribution, we will discuss some among the particularities of programming robots in comparison with programming "ordinary" computers.

The considerable paradigmatic shift in our cultural, scientific and technical comprehension of machines consists in the shifting from view of machines as physical systems intended to perform physical actions in physical world towards their comprehension as universal symbol-manipulating systems intended for storing and retrieving information coded (represented) symbolically in certain suitable ways. Hodges ([9], Chapter 2) informs us about the first steps towards this new picture of the machine proposed by A. M. Turing (1912-1954).

Rather surprisingly another conceptual shift has been accomplished in the first half of the past century - the shift from human beings towards "organic" robots appearing in the Karel Čapek's play R. U. R. in 1921 - for more details about this shift see e.g. in [11] - and then, during a very short period of the time towards the "mechanized" robots of the next decades (visualized manifoldly e.g. in lot of B-movies).

3 Machines of the Turing's Age

Alan Turing in spring of 1935, when his dissertation went the rounds of the King's Fellow at Cambridge University, he went to a course on foundations of mathematics delivered by M. H. A. Newman. Newman finished his course with outlining the Gödel's proof of his famous undecidability theorem, which did not rule out of the possibility that there was some way of distinguishing the provable from the non-provable statements. Newman put a question to his students: Was there a *mechanical* process which could be applied to a mathematical statement, and which would come up with the answer as to whether it was provable? The phrase "mechanical process" revolved in Turing's mind and leads him to a challenging question: What would by the most general kind of machine that deals with symbols? Inspired by a mechanical typewriter Turing invented and described with all of necessary mathematical rigor an idea of such machine called *automatic* (or *a-*) *machine* by him [23], and now generally known (according Alonzo Church's suggestion) as the *Turing Machine*.

Informally speaking, the Turing Machine consists in two formally well definable basic parts: the control engine, and the writing hand. The writing hand is able to read and (re)write symbols appearing in the bi-directionally potentially infinite tape. It is able to write a specific symbol (say the symbol 1) into a square on which it just staying on or to withdraw the symbol form this square. The control engine governs the writing hand actions by four commands: write, erase, move one square to the right,

and move one square to the left. Turing invented this abstract ”mechanical” (mathematically well-defined and formally rigorous constructively proposed) method (an abstract ”machine”), and he mathematically proved - roughly speaking - that this ”machine” is the most universal one dealing with symbols in order to provide computation (to perform algorithms). Moreover, he proved that form the perspective of this ”machine” the answer to the above mentioned Newman’s question is definitely ”no”. In other words, he gave the exact mathematical proof of the statement that there exist mathematically well-defined functions for which their values cannot be effectively (algorithmically) computed from the values of their variables.

The invention and further study of the universal Turing machine provide the basis for formulation of at least two fundamental hypotheses related with our understanding of machines and their capabilities.

The *First Turing’s Hypothesis* is about the capabilities of machines [23], and is known as the *Church-Turing Hypothesis* in the literature; for more details on it see e.g. [22]. It states, roughly speaking, that all what is intuitively computable in any realistic sense, is computed by the universal Turing machine.

The *Second Turing’s Hypothesis* is, mainly in the literature on Artificial Intelligence, known as the *Turing test*; see e.g. in [21]. The test consists in comparing the capabilities of machines (computers) and human beings with respect to their capabilities to perform tasks associated in human beings with their intelligence. If the test proves that the behavior of the human beings and the computers are unrecognizable for a human observer, then the computer might be considered as intelligent one. So, in other word, the test allows us to hypothesizes, roughly speaking, that *the human intelligence is expressible as a collection of computable tasks*. In such a way the original Turing hypothesis is conceptually connected with the Turing test, and makes clearer the relation between - build a bridge between the two in certain sense - efforts in Artificial Intelligence and the theory of abstract computation.

4 Robots

However, robots, at least in certain aspects, fundamentally differ also from other technical devices. This article as a free continuation of the matter presented in [15] focuses to another aspect of the large context of robotics - to robots connections with human beings, esp. with those who program robots to do what they are able to do. In this context we will understand robots in the usual sense formulated precisely enough in [18] as ”. . . autonomous system which exists in the physical world, can sense its environment, and can act on it to achieve some goals”, however we add - in order to exclude human beings form the scope of the definition, - the continuation of the Matarić’s definition the part *which exists as results of engineering efforts a human beings*.

We will focus towards some of the similarities and differences of programming robots and computers, and to some of our experiences with introducing these

similarities and differences to curricula of programming for students of informatics and computer science in order to prepare them to the similarities as well as to the differences, and in order to become deserving when meet them in their practice. Of course, there are numerous similarities, but also numerous differences between the two mentioned above types of programmable technical systems - the advanced computer controlled robotic systems and the usual computers. Moreover, the differences are reflected not only in the architectural structures and functional behavior of these types of systems but also in their programming.

From certain traditional perspectives the robotic platforms are at least in some extent similar devices as the traditional personal computers. The traditional computers have, similarly as the robotic platforms, a central processor, some input devices (reminding robots sensors, e.g. the ”mouse” or the keyboard, and also some output devices (reminding robots actuators), and some possibilities to program the processor for providing the required relations between the input data and the output data. In the case of the traditional computers we have keyboards, mice, monitors, printers, etc.

Similarly, in the case of robotic platforms we have different sensors which provide input data from observations of the robot sensors, and different actuators of robots which provide output performances in the robots external environments on the base of data computed by the programs in the robots. From this aspect robots and computers are similar programmable computing devices. However, there are some differences in programming traditional computers and robots. The aim of the present article is to comment some among the specificities appearing in robot programming in a significantly clearer manner as in the case of programming ordinary computers.

5 Programming Robots

If expression of ideas in the form which makes intercommunication of ideas possible represents the part of the culture, then writing programs is in certain sense an important enlargement of this ability because of communicating products of intellectual efforts not exclusively between human beings but also between human beings and the artifacts they are constructing and producing as part of their material culture - between computers and robots. So, our technical creativity and our generally comprehended culture meet in a specific discipline we will focus to it in the following sections.

We will focus to the activity consisting in programming our technical devices, esp. robots. We will discuss some among the common feature of traditional computer programming, and to compare them with the programming of robots. We will try to emphasize that robot programming should be used as an alternative approach to the computer programming. On the other side we will try to emphasize also the opposite side of the same coin, resp. that robot programming opens many problems which appear not so crucially in traditional programming as in the contexts of programming robots.

A very substantial difference of the computer programming and the robot programming consists in the ability of programmers to deal with in fact real-time programming. The robots environments are usually dynamic, end the dynamics of changes are usually only hardly predictable for physically real environments. Because of the requirement to act under such conditions it seems to be required to provide robots' processors with programs which are able to *connect the input data from the real environment very effectively* with the required output data managing the robots' actuators performance. This is one among the crucial specificity of robot programming in comparison with programming usual computers having usually, at least in the cases forming the basics of general programming training - almost nothing common with coping with programming real time processes. In the consequence of that, one among the important differences lies, for instance, in the *necessity to comprehend complexity in different ways*. In the case of programming traditional problem solving procedures by traditional computers, the traditional space and time measures play an important role from well known reasons.

However, in the case of programming robots working under the *real time conditions*, the principal requirement for having a sufficiently complex program might consist in having quick enough connection between sensed data and the output data without the necessity to pay distinguished attention to the computational complexity of the program behind the computation of the required output managing the actuators functioning from the input provided by sensors.

Another type of problem arise from the behavior of robots in physically real environments full of different types of *noises and uncertainties*, unexpected obstacles and other physical barriers appearing in their physically changing environments. These noises must be in certain sense presupposed during the process of constructing the programs for robots. The programs must be robust enough for copying with such types of complications which might appear during the particular runs of programs.

Keep in the mind the above mentioned and some similar requirements specific for robot programming seems to be important, and should be emphasized in robot programming curricula.

Few information on the above mentioned matter are contained e.g. in [2] where the languages Lisp and C are mentioned, and in [24] which emphasized the traditional *lingua franca* of artificial intelligence experimentations - different dialects of the language Lisp (the Scheme, for instance). Good inspiration might come also from [18] and from [4]. More specialized problems and their possible solutions are included in [8]. Some inspiration for programming can be found also in [19]. In the case of [2] the short notes follows from programming reactive robots, in the second one the main emphasis is put to traditional artificial intelligence programming which is not identical with the matter of robot programming, however, Lisp or its derivates like Scheme, for instance, could be effectively used for purposes of robotics. Other publications are focused to general settings of problems without a special attention devoted to programming languages.

6 Few Types of Problems

Now, based on [5] and [15], we will focus to some of the types of problems specific for programming robots in comparison with programming usual computers. The first class of such problems is the problems concerning the realization of the possibility of *emergent* behavior of robots - it means the behavior which comes as an ”unexpected” result of interactions of programmed behavior.

The primary requirement putted to robots behavior is the high level capacity of *obstacle avoidance*. The programs making possible for a robot to avoid obstacles must be quite effective and must work with high efficiency.

The problem of obstacle avoidance is in certain specific situations solvable using the principle of *reactivity*, moreover by mechanical devices having no electronic information processing parts. So, it is solvable *without any programming*, realizing only the advantage which follows form the mechanical construction of the mechanism.

A good example of the solution of this kind of problem provides the mechanical ladybug depicted in the figure 3 with its two parts arranged according the simple architectural principle of so called *subsumption architecture* usually used in the present so called *fully reactive robotics*. The application of the principle of this idea for the case of the mechanical ladybug is depicted in the figure 4. For more details on different types of autonomous agents including robots see e.g. [13].

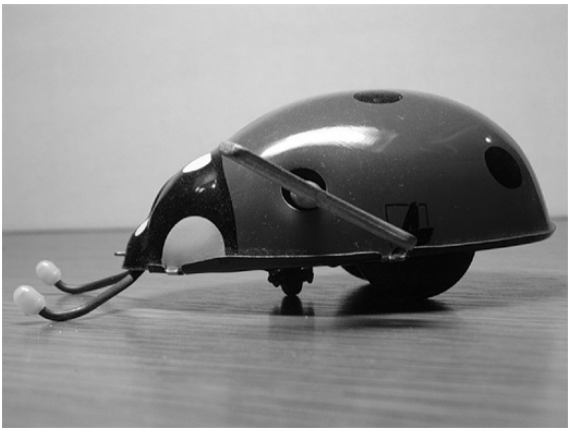


Fig. 2 The low level rational mechanical ladybug

Fig. 3 The ”ladybug”, and its two crutial parts

As presented in ([14], two non-rational parts subsumed in a right way result in a low level rational behavior consisting in the avoidance of vertical obstacles (like a table edges) of the whole ladybug; see figure 4.

A specific type of the just mentioned problem of obstacle avoidance is the problem of *collision free moving*, i.e. moving of the robot without touching the obstacles appearing in its route. As it was demonstrated e. g. in [6] in experiments with a Khepera robot (with eliminated functioning of its built-in collision avoidance program) at the Institute of Computer Science of the Silesian University during his PhD study an almost perfect collision avoiding program could be generated rather quickly (8 minutes of the run of Khepera inboard processor) using the technique of genetic algorithms, and a suitable programming tool provided for the use of the programming language Lisp. In [7] this original tool - an interface language KHLI providing the possibility to program Khepera robots in Lisp by ”translating” Lisp programs into C language programs is reported.

Another type of problem is the *wall-following* capacity appearing in certain missiles of robots. In ([18], p. 215-216) a good example of the wall-following effect which emerges from two ”simplest” programs, one which provides the straightforward movement of the robot, and the second one providing the obstacle avoidance capacity of the robot. The main idea is surprisingly simple and is depicted in the figure 5.

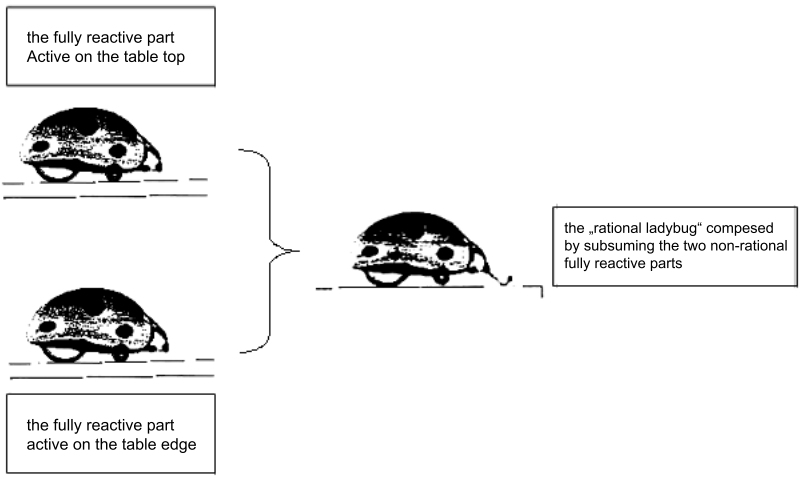


Fig. 4 The way how two well subsumed fully reactive parts result in the low level rational of the mechanical ”ladybug”

7 Some Experiments

First, we mention two traditional navigation problems which are used in order to test our students of computer science at the Silesian University in Opava. These problems involve the above mentioned problems of obstacle avoidance and line following, and integrate the solution of both of them into the program which solves a more complex navigation missile of the robot. The first one is the so called mark identification and following problem, the second one is the so called red line on top floor following problem. These experiments are provided by students as their regular part of examination process after the lectures on robotics and pattern recognition. Usually it has the form of a competition between small gross of students providing their own programs each. To solve the problems students must have preliminary knowledge from several areas of their previous study. From basics of robotics and robotic hardware, embedded programming in languages C and C++, embedded OS and Linux platforms, computer vision, and the theory of automata, for instance. For more details on the experiments see in [5]. The experiments are executed at the Silesian University Institute of Computer Science robotic laboratory.

The first example is the experiment consisting in solving a package of problems, some of them connected with patter recognition and image processing problems, other with the problem of autonomous navigation according the ”road signs”, and the line-following problem. The experiments are executed using a Koala robot produced by K-Team. The Koala robot in the configuration consisting of the autonomous automotive part with six wheels, six boards, maximally two video cams, and a WiFi connection provider; see figure 6.

The writing the problem for the robot is the part of the exam on pattern recognition and introductory robotics. Students form several small (per two to three member in each) groups, and each group proposes at the end of the exam its own program which integrates the programs for solving the parts of the robot missile. Then the programs are started, and the behavior of the Koala is evaluated for each group. The results of evaluation then form the parts of students' individual evaluations. Figure 5 takes a snapshot on the environment and the robot during the run of the experiment. More details can be found in [5].

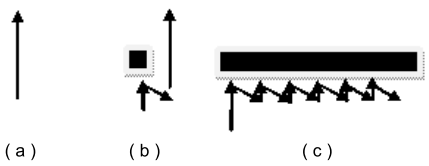


Fig. 5 Emergence of the wall-following behavior (c) from the straightforward movement (a), and the obstacle avoidance (b)

The second example is the subject of the diploma project of Michal at the ICS of Silesian University then published as [20]. As it was mentioned above, the second type of robots used in our robotic laboratory are approximately eight actually working small mobile robots Khepera produced by the K-Team. These robots are due to their small size and various extensions as ideal for a number of experiments in teaching basic programming in the field of mobile robotic. They are equipped with two independent wheels - each wheel is powered by its own motor - and with eight infra red sensors, and by some additional sensors and actuator (see figure 2). By its infrared sensors the Khepera can measure both light intensity and distance to obstacles and walls.

The main goal of the above mentioned diploma thesis was to develop a program for controlling the society, in this case particularly a two member society, of robots to solve cooperatively the so called box-pushing problem. This problem defines one among the basic and quite favorite tasks in the cooperative robotic.

The goal is to push selected object (the box) to a pre-specified goal-place in the robots environment. In order to perform this task it is necessary to design and implement number of subtasks and their composition to get the requested behavior and the solution of the problem.

Variant approaches and techniques of the solving this task can produce many results and experiences which can be used for solving similar tasks in the real world. In the thesis the so called master and slave concept was used, and programmed successfully. The original Kheperas sensors have been modified in certain extent during the experiment in order to reduce the sensors usual angle of sensitivity, as the figure 7 shows.

Figure 8 illustrates the beginning of the mission of the collective of two robots. For technical details see [20].

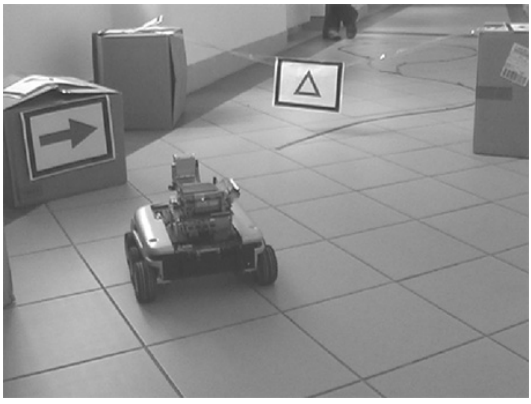


Fig. 6 A Koala robot during the missile using camera impute for obstacle-avoidance, marker-respecting motion, and line-following motion.

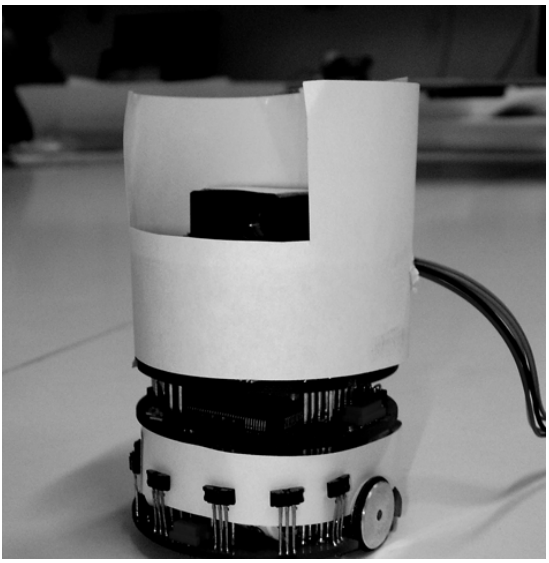


Fig. 7 A Khepera robot prepared for the multi-robot solution of the box-pushing missile

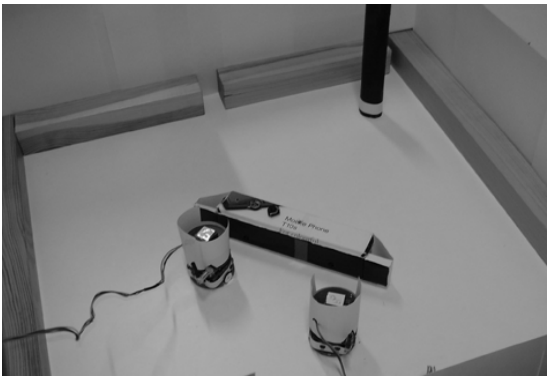


Fig. 8 Two Khepera robots in the process of the box-pushing missile

8 Conclusions

The presented contribution might be comprehended in different ways. It might be comprehended as a contribution to the didactics of teaching robot programming or as a contribution to the diverse discussion on the present influences of technologies to culture, for instance. However, from our point of view, the best way to comprehend it starts from the question of how our present culture enables to integrate different parts of the human creativity, not only the pure and traditional artistic creativity, but also other type of creative acts like technical creativity, as a part of itself.

The growing of this ability to integrate might mean that the chasm between the ”technicians” and the ”artists” opinions will be step by step eliminated, and we will be closer to certain ”general” understanding of the culture as the sphere created by the human creativity in general.

So, we are ready to conclude that the concept of robot with its long and contradictory cultural as well as technological (pre-)history together with new technologies that enable artists to create unusual interactive communication scripts in physical or virtual worlds, possibly in some of new kind of spaces is not exclusively in the context of the contemporary new branches of arts connected with a new aesthetic dimension. This dimension prefers modeling of behavior (when the author creates the possibilities for actions and reactions of the created system according to the inner or outer stimuli sensed from by it in its environment) over creation of static objects. Preference of the behavior over the form, and the system over the object is then understood as a general and characteristic feature of robotic and cybernetic art today, ninety years after placing robots into the theater stages. In the other hand, the creativity of programmers, the purely ”technologically” oriented professionals, enables to presuppose the origin of new ways how also the artistic works might be ”materialized” in working, functioning artifacts. This will be also a contribution to the unification of different parts of the widely comprehended culture.

Acknowledgements. The author thanks to Jana Horáková from the Masaryk Universitty, Brno for productive cooperation and discussions on some parts of the subject of this contribution, to the former and present members of the ”robotic staff” of the Silesian University Institute of Computer Science, namely to Michaela Ačová, Petr Čermák, Martin Dostál, Aleš Kubík, Michal Nemrava, and David Novák for providing matters for this contribution, to Lucie Ciencialová from the same Institute for her technical help, and to Gratex International for continuous support of his work. All photos occurring in this contribution expect the two appearing in Fig. 1 are courtesy of the Institute of Computer Science, Silesian University, Opava, Czech Republic.

This contribution has been supported by the project IT4Innovations Centre of Excellence, reg. no. CZ.1.05/1.1.00/02.0070, and sponsored by the Research and Development for Innovations Operational Program from the Structural Funds of the European Union, and from the state budget of the Czech Republic.

References

1. Ačová, M., Kelemen, J., Kubík, A.: Embodied hypotheses - preliminary notes and case studies. In: Kozlowski, K. (ed.) Proc. 4th International Workshop on Robot Motion and Control, pp. 35–40. Poznan University of Technology, Poznan (2004)
2. Arkin, R.C.: Behavior-Based Robotics. The MIT Press, Cambridge (1998)
3. Baudrillard, J.: Videowelt und fraktalen Subjekt. In: Philosophie Der Neuen Technologie, pp. 113–131. Merve Verlag, Berlin (1998)
4. Braitenberg, V.: Vehicles. The MIT Press, Cambridge (1996)
5. Čermák, P., Kelemen, J.: An Attempt to Teaching Programming with Robots. In: Obdržálek, D., Gottscheber, A. (eds.) EUROBOT 2011. CCIS, vol. 161, pp. 78–87. Springer, Heidelberg (2011)

6. Dostál, M.: A Functional Approach to Automatic Programming (PhD Dissertation). Hradec Králové, University of Hradec Králové (2005) (in Czech)
7. Dostál, M.: The KHLI interface for programming Khepera robots in LISP. In: Proc. Kognice a umělý život III, pp. 45–58. Silesian University, Opava (2003) (in Czech)
8. Dudek, G., Jenkis, M.: Computational Principles of Mobile Robotics. Cambridge University Press, Cambridge (1983)
9. Hodges, A.: Alan Turing - The Enigma of Intelligence. Unwin Paperbacks, London (1983)
10. Horáková, J., Kelemen, J.: Robots as in-betweeners. In: Rudas, I.J., et al. (eds.) Computational Intelligence and Engineering, pp. 115–127. Springer, Berlin (2010)
11. Horáková, J., Kelemen, J.: Artificial living beings and robots - one root, variety of influences. Artificial Life and Robotics 13, 555–560 (2009)
12. Horáková, J., Kelemen, J.: The robot story - why robots were born and how they grew up. In: Husbands, P., et al. (eds.) The Mechanical Mind in History, pp. 283–306. The MIT Press, Cambridge (2008)
13. Kelemen, J.: Agents from functional-computational perspective. Acta Polytechnica Hungarica 3, 37–54 (2006)
14. Kelemen, J.: A note on achieving low-level rationality from pure reactivity. Journal of Experimental and Theoretical Artificial Intelligence 8, 121–127 (2006)
15. Kelemen, J.: On the Discreet Charm of Robot Programming. In: Pap, E., Fodor, J. (eds.) Proc. 9th IEEE International Symposium on Intelligent Systems and Informatics, pp. 11–16. IEEE Publication, Piscataway (2011); IEEE Catalog Number CFP1184C-CDR
16. Kelemen, J., Kubík, A.: Robots and agents in basic computer science curricula. In: Kozlowski, K. (ed.) Proc. 3rd International Workshop on Robot Motion and Control, pp. 309–317. Poznan University of Technology, Poznan (2002)
17. Kelemen, J., Kubík, A.: RADIUS - looking for robots' help in computer science research and education. ERCIM News No. 15, 48–49 (2002)
18. Matarić, M.J.: The Robotics Primer. The MIT Press, Cambridge (2007)
19. Murphy, R.R.: Introduction to AI Robotics. The MIT Press, Cambridge (2000)
20. Nemrava, M., Čermák, P.: Solving the box-pushing problem by master-slave robots cooperation. JAMRIS 2(3), 32–37 (2008)
21. Pfeifer, R., Scheier, C.: Understanding Intelligence. The MIT Press, Cambridge (1999)
22. Sieg, W.: Church-Turing Thesis. In: Wilson, R.A., Keil, F.C. (eds.) The MIT Encyclopedia of the Cognitive Sciences, pp. 116–117. The MIT Press, Cabmridge (1999)
23. Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. Proceedings of London Mathematical Society 42, 544–546 (1936); corrections in 43 (1937)
24. Winston, P.W.: Artificial Intelligence, 3rd edn. Addison Wesley, Reading (1992)

Online Generation of Biped Robot Motion in an Unstructured Environment

Borovac Branislav, Mirko Raković, and Milutin Nikolić

Abstract. In this work is demonstrated the possibility of using primitives to generate complex movements that ensure motion of bipedal humanoid robots in unstructured environments. It is pointed out that for the robot's motion in an unstructured environment an on-line generation of motion is required. Generation of motion by using primitives represents superposition of simple movements that are easily performed. Simple movements are either reflex or learned synchronous movements of several joints, and each of these movements represents one primitive. Each primitive has its parameters and constraints that are determined on the basis of the movements capable of performing by a human. A set of all primitives represents the data base from which primitives are selected and combined for the purpose of performing a complex movement.

Keywords: Robot, biped robot motion, humanoid robot, primitive.

1 Introduction

In the recent years we have witnessed an explosive development of Humanoid Robotics. However, in the realization of their movements, many humanoid robots perform motions that are synthesized in advance [4, 18, 5].

Bearing in mind the high sensitivity of humanoid robots to disturbances, it is straightforward that the maintenance of dynamic balance represents the primary control task. In [17, 16], different control algorithms were analyzed for preserving an anthropomorphic motion, whereas a separate control law was defined based on the deviation of the Zero-Moment Point (ZMP) [14] from its reference position, for preserving dynamic balance. Authors, in [17, 16], made a comparison of the fuzzy

Borovac Branislav · Mirko Raković · Milutin Nikolić
University of Novi Sad,
Faculty of Technical Sciences, Serbia
e-mail: {borovac,rakovicm,milutinn}@uns.ac.rs

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 93–114.
DOI: 10.1007/978-3-642-33959-2_6

logic control and Proportional-Integral-Differential (PID) control, and it was shown that these strategies were mainly adequate for compensating small disturbances, whereas the compensation of large disturbances requires a different approach. In [14], Vukobratović et al. considered in detail the notion of dynamic balance, especially from the aspect of biological principles that are used to preserve it, including the case of a non-standard foot-ground contact. The problem of classifying disturbances has been considered in detail in [15].

If robots are to share the living and working space with humans, which assumes their motion in unstructured environments, it is not possible to use a predefined motion since they will have to react to the situations in real time during the motion realization, which will not be possible to plan in advance. A main prerequisite to achieve this is to ensure the robot's efficient motion, that is its ability to compensate for the ever-present disturbance and the ability for online generation in highly unstructured environment. Current techniques applied for online motion synthesis lack fast response and flexibility, and new methods are needed.

From biology we can learn a lot. In [8], it was clearly shown that an electrical micro-stimulation of the same interneuron region of the spinal cord evoked synergistic contractions that generated forces directing the hand limb toward an equilibrium point in space. The collection of the measured forces corresponded to a well-structured spatial pattern (vector field) that was convergent and characterized by a single equilibrium point. Further, there are important findings about modular organization of spinal motor systems in the frog spinal cord. The experiments showed that only a few of distinct types of motor outputs could be evoked by stimulation. However, when the stimulation was applied simultaneously to two different sites in the spinal cord, each of which when stimulated individually produced a different type of motor output, the resulting motor output was a simple combination of the separate motor outputs. Based on these observations it was proposed that complex movements might be produced by the flexible combination of a small number of spinally generated motor patterns.

The use of primitives to generate motion of humanoid robots is not a new approach. In [3, 2, 9, 6, 7], the authors used the whole movement (overall gait, transition from standing to walking, etc.) as a primitive. In [3], authors used a library of motion primitives where each primitive is a single step. The library of motion primitives actually represents a set of different steps. Depending on the requirements and robot state, each time a new primitive is selected from the library. In [2], authors describe an approach to generate walking primitive databases in which each primitive is a cyclic walk with different parameters. They also generate a different primitive for transition from one cyclic walk to another. A general framework for learning motor skills which is based on a thorough analytical understanding of the robot task representation and execution was presented in [9]. Finally, in [7, 6] an approach was presented for online segmentation of the whole-body human motion observation and learning.

An essential feature in the approach we advocate is that the online motion is formed by a combination of simple movements and not of the complex movements recorded in advance. Complex movements are decomposed into simple ones, called primitives (e.g. leg stretching, leg bending, hip turning, etc.). The basic idea is to enable the system to learn to execute online each primitive with different parameters (let say, leg bending to different knee and hip angles) and from different initial positions. A movement may be continued (if needed) with another primitive (also, online selected) to perform a complex movement. For example, the movement of the leg in swing phase consists of leg bending immediately followed by leg stretching. In addition, the primitives can be added on top of each other (superimposed) in case the cur-rent motion is basically properly executed but small corrections are needed. For ex-ample, even when dynamic balance is not directly endangered, human realizes a series of simple movements and changes its motion in real time. Thus, for example, if in the course of the gait a need arises to modify the motion (e.g. there appears an obstacle that is to be bypassed or the need to climb the staircases), the human does not waste time on the calculation of a new and complex motion to be realized and of new kinematic and dynamic parameters of motion, but it selects a most appropriate primitive, by adjusting only some of the basic parameters such as the height of the leg lifting, angle of leg bending or the stride length. By introducing the base of primitives that are realized by taking as a model human's movements, the aim of this work was to demonstrate that the appropriate selection and combination of primitives can yield the realization of a complex motion.

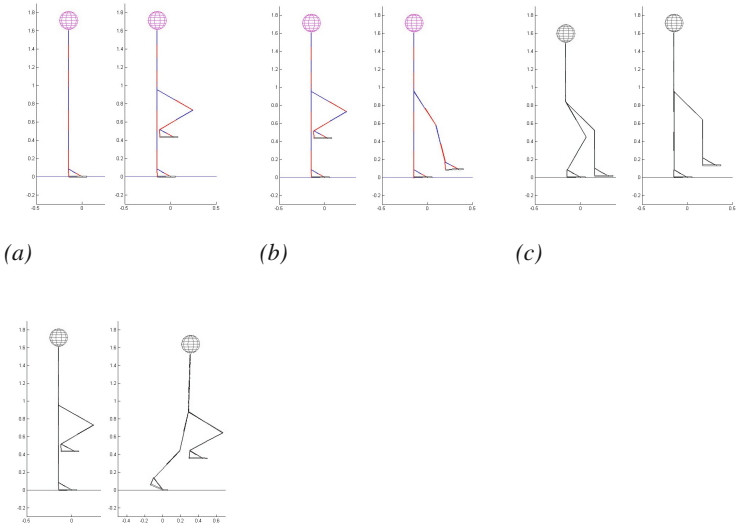
2 Motion Decomposition and Synthesis

Every motion can be considered as being composed of a set of basic movements which can be easily combined. In this chapter we describe how climbing stairs can be decomposed into primitives, and how the primitives can be combined into a more complex motion.

2.1 Primitives

The term primitive stands for a simple movement that a human or robot is capable to realize. A primitive itself should be simple in order it could be easily combined with the other primitives. This implies that a certain primitive can be included in different complex movements. Each primitive is parameterized and has the following parameters: intensity of the movement in the span of 0-1 (which determines the extent to which, for example, a leg is to bend or stretch), time instant from which the primitive execution should start, and duration of the primitive execution.

Figs. 1 (*a*) and 1 (*b*) show stick diagrams representing bending and stretching (at the hip, knee and the ankle) of the leg in swing phase and stretching of the support leg (Fig. 1 (*c*)), as well as the primitive for the mechanism's inclination forward (Fig. 1 (*d*)).



(a) *(b)* *(c)*

(d)

Fig. 1 Stick diagrams of the humanoid robot model in the realization of primitives by the swing leg: a) leg bending, b) leg stretching. And support leg c) stretching, d) inclining, by which the system as a whole moves forward.

Each of the primitives is realized by activating one or more Degrees of Freedom (DOFs). Thus, for example, the primitive for bending the leg in swing phase involves activation of the joints at the hip, knee and ankle of the swing leg. In the realization of the primitive for stretching of the support leg (Fig. 1*(c)*) use is made of the ankle, knee, and hip joints. By the realization of the primitives for stretching of the support leg, these joints move so to ensure that at the end of the movement, when the stretching intensity was preset to 1, the leg is fully stretched, which corresponds to the angles at the joints of 0 rad.

In the realization of the primitive for inclining, the support leg is activated again, and the same joints are used as in the realization of the primitive for stretching the support leg (Fig. 1*(d)*), but now the toes link is also moving since the rear part of the foot (heel) separates from the ground and the system remains supported only on the link of the toes. With this primitive, the angles at the joints of the knee and hip are of the opposite sign to the angles at the links of the toes and ankle, to ensure keeping the trunk in the upright position with respect to the external coordinate frame. The primitives shown here are just examples identified on the basis of human motion for the purpose of climbing stairs, which is in the focus at the moment.

It also should be noted that, considering kinematics, the motion at each joint is of simple ”s” shape. However, the simulation motion was realized by applying appropriate driving torques at the joints or by applying appropriate voltages to the

driving motors[1], assuming that each joint is actuated by its own DC motor. For the same shape of movement (with different starting points), the control variables vary, and their exact values depend of the ”gravity contribution” to the joint load which vary with different position of the leg in space, etc. This means that the shape of the control variable does not always correspond to the shape of the movement.

2.2 *Composing Primitives*

In this section is presented the motion of climbing stairs, entirely synthesized using primitives (Fig. 2). In order to make clearer the process of the synthesis procedure, the motion was split into three phases. First, the robot from a regular walk is brought to the initial upright posture which is adopted to be the initial position for imposing the primitive for climbing stairs. Then, it is to realize climbing the first step and afterwards the second step. Further climbing can be realized by repeating the movements performed to climb up the second step.

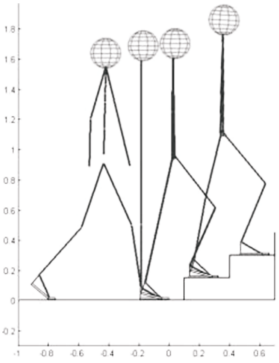


Fig. 2 Stick diagram (only four postures are shown) of humanoid robot climbing stairs

It should be pointed out that there is no reference motion for such type of walk, and the overall motion is to be realized by superimposing the appropriate primitives. Sampling time in this example is 1ms.

Climbing the first step is realized by combining the following primitives. First, to the robot in the upright position the leg-bending primitive is imposed, by which the leg takes off from the ground and passes to the swing phase. This primitive lasts 1650 iterations at an intensity of 0.44. At the same time is imposed the primitive for lifting up the same leg. This primitive is imposed beginning from the 275th iteration and lasts to the 1925th iteration at an intensity of 0.2. In the span from the 1100th to the 2750th iteration the two primitives - for stretching the support leg and bringing

[1] The driving torque of the joints or motor driving voltage for the joint actuation will be called the control variable.

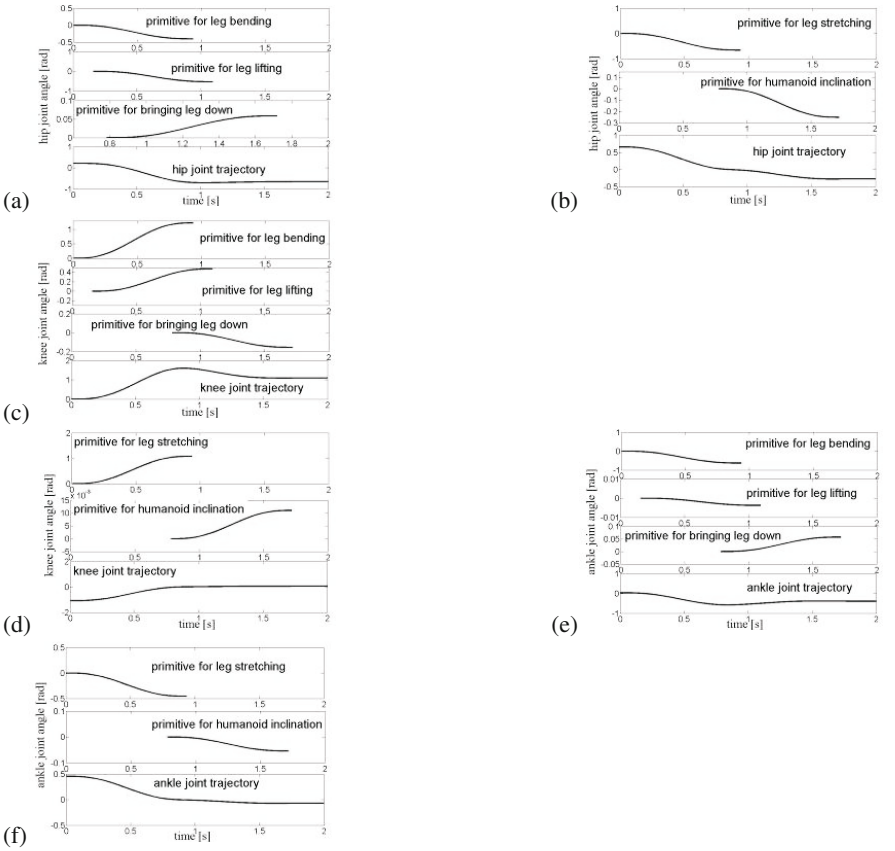


Fig. 3 Diagrams of forming the motion at the joints at the hip, knee and ankle (downwards) of the swing (left) and support (right) leg by superimposing the appropriate primitives: (a, hip of the leg in swing phase), (b, hip of the support leg), (c, knee of the leg in swing phase), (d, knee of the support leg), (e, ankle of the leg in swing phase), (f, ankle of the support leg)

down the swing leg - are imposed simultaneously at the intensities of 0.35 and 0.1, respectively. An additional primitive for inclining forward is imposed to the support leg from the 1350th to the 3000th iteration at an intensity of 0.25.

The third phase is the climbing up the second step, the initial posture being different from the one prior to climbing the first step. In this phase the leg bending is realized, lasting from the first to the 1650th iteration at an intensity of 0.44. The primitive for stretching the support leg is applied simultaneously (lasting also from the first to 1650th iteration at an intensity of 1). After that follows the lifting up of the swing leg, and this primitive lasts from the 275th to the 1925th iteration at an

intensity of 0.23. Then comes the realization of the inclination[2] (in Fig. 3 denoted as the primitive for humanoid inclination) of the supporting leg from the 1350th to the 3000th iteration at the intensity of 0.28, and finally, the landing of the swing leg lasting the same time at an intensity of 0.1. Afterwards, the procedure is repeated from the beginning.

This example is given just to explain and illustrate the basic idea of composing movements. However, primitives are not composed by selecting motion for each joint separately, but by selecting motion "as a whole". For example, the motion of the leg in swing phase is composed by the realization of three primitives: leg bending, leg lifting and leg landing. Each primitive ensures a synchronized motion of all joints involved.

3 Model of the Humanoid Robot

The humanoid is modeled as a set of open kinematic chains connected to the central (basic) link and interconnected by rotational joints with only one DOF [10]. Fig. 4 shows the mechanism structure of anthropomorphic shape consisting of 46 links. The basic link is the pelvis, numbered as 1. The joints with more DOFs (ankle and hip) are modeled as a series of one-DOF joints, connected by massless links of infinitesimal length (fictitious links). For example, the spherical joint at the right hip is modeled by three one-DOF joints (the orth of rotation axes $\mathbf{e_7}$, $\mathbf{e_8}$ and $\mathbf{e_9}$ are mutually orthogonal), connected by the fictitious links 2 and 3. The position and orientation of the pelvis, (the basic link, to which all other kinematic chains are connected), are presented by a vector consisting of three translational and three angular coordinates $X = [x, y, z, \theta, \varphi, \psi]^T$. The other kinematic chains connected to the pelvis are: links 1-9 representing the right leg, the links 1, 10-17 constituting the left leg; links 1, 18-42 representing the 10-link trunk (backbone) and the right arm, whereas the fourth kinematic chain is represented by the 10-link trunk (the links are interconnected by 2-DOF joints) and the left arm (links 1, 18-38, 43-46). The head is included into the link 38. The feet are two-link ones, consisting of the foot body and toes connected by a one-DOF joint (left foot: links 8 and 9; right foot: links 16 and 17).

The motion of each joint is described by one joint coordinate $(q_7, q_8, \ldots, q_{51})$. Taking into account that 6 coordinates are needed for the positioning of the base link in the space we obtain that the overall number of DOFs of the described mechanism is 51, whereas the system's position is described by the following expression:

$$\mathbf{Q} = [\mathbf{X}^T \mathbf{q}^T]^T = [x, y, z, \theta, \varphi, \psi, q_7, q_8, \ldots, q_{51}]^T . \quad (1)$$

[2] Inclination is realized by activating the joint by which the link of toes is connected to the rest of the foot, thus inclining the overall humanoid forward. To preserve the verticality of the trunk, this inclination has to be compensated for, so that the curves of the "primitive for humanoid inclination" are applied at all joints of the support leg, which can be seen in the right-hand diagrams of Fig. 3.

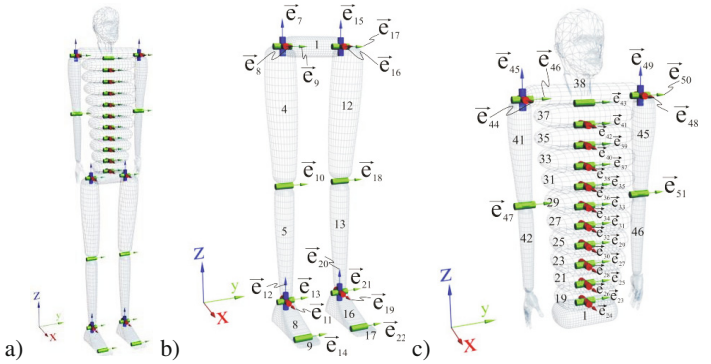


Fig. 4 Kinematic model of the robot: a) the whole robot, b) detailed schematic of the legs, c) detailed schematic of the trunk and arms

Each joint has its own actuator that generates the corresponding driving torque τ_i , whereas the first six DOFs of the vector $\mathbf{Q}$ correspond to the position and orientation of the basic link $(x, y, z, \theta, \varphi, \psi)$, and they are unpowered. The vector of driving torques at the joints is , and the expanded vector of generalized forces of the length 51 is . In the simulation we can select either that the joints are actuated by applying driving torques or the motor voltages. In the case when the joints are powered by DC motors with permanent magnets, the motor is controlled by the rotor coil voltage, and, without the loss of generality, the same motor can be used at each of the joints.

In the course of stepping, the system passes from the single-support to the double-support phase. In the beginning, the leg that performs stepping is not in contact with the ground, the contact is being realized when the stepping is ended. Also, in the course of the contact realization, slipping may occur between the foot and the ground, and during the settling phase the rear leg can separate from the support. Because of that, it is of great importance to model appropriately the foot-ground contact, i.e. to describe in the most reliable way all the effects that may arise between the two bodies in contact.

3.1 Modeling of the Contact

The two-link foot is rectangular, with a flat contact area, so that for an exact identification of the contact type it suffices to observe only six characteristic points shown in Fig. 5. Four contact points (1-4 in Fig. 5) are the foot body corners, whereas the points 5 and 6 are at the top of the toes. By observing only these six points we can describe all the possible configurations of the foot-ground contact. If three or more points (which are not collinear) are in contact with the support, the contact is planar; if two points are involved, it is a linear and, finally, there may exist one-point contact.

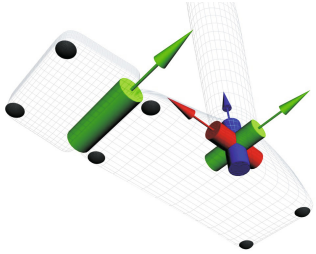


Fig. 5 Positions of the viscoelastic elements through which the foot-ground contact is realized (view from the bottom of the foot)

Since there are two feet, the total number of points that are to be observed is 12. However, not all the points are always in contact with the ground. The moment when a certain point establishes or breaks contact with the ground was calculated using slack variables [12].

3.2 Model of the Overall System

By uniting the model of the robot's mechanical structure and of the viscoelastic foot-ground contacts, the overall system can be described by the following set of differential equations:

$$\mathbf{H} \cdot \ddot{\mathbf{Q}} + \mathbf{h_0} = \mathbf{T} + \sum_{i \in S} \mathbf{J}_i^T \cdot \begin{bmatrix} \mathbf{F}_i \\ \delta_i \times \mathbf{F}_i, \end{bmatrix} \tag{2}$$

where $\ddot{\mathbf{Q}}$ represents the vector of generalized accelerations of the mechanical system; $\mathbf{H}$ is the system inertia matrix; $\mathbf{h_0}$ is the column vector of velocity effects and gravitational moments; $\mathbf{T}$ is the column vector of driving torques; S is the set of points in contact with the ground; δ_i and $\mathbf{F}_i$ are the deformation and contact force at the i-th contact point, respectively; $\mathbf{J}_i$ is the Jacobian matrix calculated for the i-th contact point [1].

In the case when each joint is powered by a separate electric motor, the second-order DC electric motor model is used in the following way:

$$\begin{aligned} u_k &= R_r \cdot i_{rk} + C_e \cdot \dot{q}_k + L_r \cdot \dot{i}_{rk} \\ \tau_k &= C_m \cdot i_{rk} + B \cdot \dot{q}_k + J_r \cdot \ddot{q}_k. \end{aligned} \tag{3}$$

In these equations, τ_k represents the driving torque at the k-th joint (actuator); i_{rk} is the rotor current of the k-th actuator; $\dot{q}_k$ and $\ddot{q}_k$ are the angular velocity and acceleration at the k-th joint. The rotor coil voltage of the k-th actuator is u_k. The actuator parameters are: torque constant (C_m=4.1910 Nm/A), speed constant (C_e= 4.1794 rad/Vs), rotor thermal resistance (R_r=0.3630 Ω), rotor inductance ($L_r = 161 \cdot 10^{-6}$ H), rotor inertia (J_r=0.1459 kgm^2), and coefficient of viscous friction (B=1.268·10$^-$3 Nm/s). The motion of the actuator is controlled by changing the

rotor coil voltage u_k. Therefore, the overall model of the system is presented by equation (2), together with another 45 second-order differential equations (3), describing the behavior of the actuators at each of powered joints.

4 Realization of the Primitives

Simple motion of human limbs requires very complex, simultaneous and well synchronized change of the joints' angles. The task becomes even more complex when the motion has to be synthesized and performed online on the basis of the current state of the robot-environment interaction.

Let us direct our attention to some important issues. The first one is the difference between kinematic and dynamic composition of primitives. In the case of kinematic composition it is just the shape of the movement to be achieved, while dynamics composition supposes the application of a control variable to achieve a desired movement. Let us explain this on the example of leg bending in swing phase. The first characteristic of this movement is that spatial trajectory of the ankle is approximately a straight line during normal walking without obstacles. However, during the motion in an unstructured environment it is not necessary that a certain primitive is always realized in the same way. The bending intensity and bending speed have to be on-line selected on the basis of the current situation in the environment. As it is clear from Fig. 6, illustrating two examples of leg bending, one and the same movement can be realized with different motion parameters and from different initial positions. In both cases, small circles denote the initial and terminal position of the swing leg ankle (terminal position is defined by bending intensity). If the bending intensity is larger, the ankle terminal position will be higher. This issue will be further discussed in the next section.

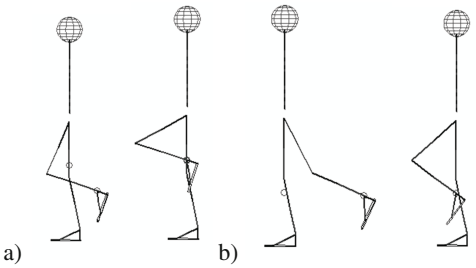


Fig. 6 Examples of leg bending from different starting positions and with different intensities: a) leg bending intensity is 91%, b) leg bending intensity is 50%

The second issue on which we will focus our attention can be explained on an example when the movement has to be changed during the execution or when a primitive is to be executed while the robot is moving. Let us suppose that during the execution of a certain primitive such situation arises in which the system is forced to

stop performing current primitive and starts execution of a new one. The execution of a new primitive should start immediately after the need has been recognized, and it is not necessary to stop the limbs first and then start the new primitive. Hence, such situation can be generalized as the primitive execution in case the velocity at the start-ing point is not zero. This issue will be described in details in Section 4.2.

4.1 Realization of Primitives using Support Vector Machine (SVM)

Having in mind findings reported in [8], we believe that human learns some basic motions and, when needed, they are to be recalled and replayed. Another important point is that the same stimuli drive limb to the same point regardless of its initial position (whole vector field is formed). And finally, those vector fields are additive. We believe that an approach with similar characteristics is needed for robots, too.

4.1.1 Generation of Training Data Set

To enable the application of SVM learning algorithms for the realization of primitives it is necessary to define the sets of appropriate inputs and outputs, which will be explained on the example of the leg bending. The adopted inputs are: the bending in-tensity, movement velocity (specified by movement duration), current velocity of the ankle, current values of angles and current angular velocities at the knee (rotation about the y-axis) and the hip (rotation about the x and y-axes). The outputs are the driving torques at the hip and the knee. At the starting position, the leg is not moving and the driving torques have to be defined so to ensure this (just to compensate for the gravity). Then, the torques should be such to ensure a linear ankle trajectory whose terminal point is defined by bending intensity. When the ankle terminal point is reached, the torques should be such to keep the leg at this position.

If the movement velocity increases, the time needed for motion realization becomes shorter. The angular velocities at the knee and hip joints (constituting the vector $\dot{\mathbf{q}}_L$) can be determined from the relation:

$$\dot{\mathbf{q}}_L = \mathbf{J}_L^{-1} \cdot \mathbf{V}_{skz}. \tag{4}$$

In (4), $\mathbf{v}_{skz}$ denotes the desired ankle velocity, while the values of the Jacobian matrix represent the relationship between the leg's joints angular velocities and ankle linear velocity. The driving torques are calculated from (2).

To use the SVM approximation it is necessary to ensure a proper training set. Let us remind once more that inputs for primitives (and also for SVM) are the bending in-tensity, movement velocity (specified by the movement duration), current ankle velocity, current leg posture, i.e. current values of the angles and of current angular velocities of the performing leg, while the outputs are the driving torques. In view of the fact that bending can be performed from any possible initial position the training

set should span, as much as possible, over all the initial postures that might appear. The training set should also span the whole range of possible bending intensities and velocities.

The procedure of determining the input and output quantities for training data set is as follows:

1. Starting posture of the robot's leg is determined by random selection of the hip and knee joint angles from a predefined range.
2. Intensity (I) of the leg bending and movement duration (S) are also selected randomly.
3. Desired velocity profile of the ankle is calculated.
4. Then, the procedure to compute driving torques is as follows:
   a. The desired ankle velocity profile being known, the corresponding angular velocities at leg's joints are in each iteration calculated from (4);
   b. After that, the angle and angular velocity at each joint are calculated;
   c. Since the whole system state is known (angles, angular velocities and accelerations at all joints) the driving torques are calculated from (2).
5. The procedure is repeated till the ankle is sufficiently close to its terminal position, and the ankle velocity is sufficiently low. Then, the procedure for this movement is stopped, a new ankle starting position is randomly selected (the target point is defined by the intensity of leg bending) and steps 1-4 are repeated.
6. After a sufficient number of performed movements, the procedure is stopped.

In this way the leg bending is simulated from an arbitrary point to the target point defined by the intensity of leg bending. For each time instant, the values of all input and output quantities needed for SVM training are obtained. The input vector $[I\ S\ \mathbf{v}_{skz}\ \mathbf{q}_L\ \dot{\mathbf{q}}_L]^T$ for the training set is of dimension 11, whereas the dimension of the output vector is 3.

4.1.2 SVM Regression

Since the SVM is used to calculate driving torques for primitives it is necessary to briefly describe what the SVM regression stands for. There are a number of algorithms for approximating the function for establishing the unknown interdependence between the input and output data, but an ever-arising question is how good is the approximation of the function $\mathbf{y} = f_a(\mathbf{x})$. In determining the approximation function, it is necessary to minimize some of the error functions. The majority of the algorithms for the function approximation minimize the empirical error. With the function approximation algorithms that minimize only the empirical error, there arises the problem of large generalization errors. The problem appears when the training set is small compared to the number of different data that can appear at the input. Structural Risk Minimization (SRM) [13] is a new technique of the statistical learning theory, which apart from minimizing the empirical errors, also minimizes

the generalization errors (elements of the weight matrix w). Hence, it follows that the structural error will be minimized by minimizing a function of the form:

$$R = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}|y_i - f_a(\mathbf{x}_i, \mathbf{w})|_\varepsilon . \tag{5}$$

In (5), the error function with the $\varepsilon-$ insensitivity zone was used as the norm. The parameter C is the penalty parameter which determines the extent to which the empirical error is penalized relatively to the penalization of the large values in the weighting matrix. The network input is denoted by $\mathbf{x}$, and the desired output by $\mathbf{y}$. The approximating function is denoted by $f_a(\mathbf{x}, \mathbf{w})$, and it has to be chosen in advance. Since the case considered is highly nonlinear, Radial Basis Function (RBF) network with Gaussian kernel was chosen as approximation function, for which the output is calculated from the following relation:

$$f_a(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{N} w_i exp\left(-\gamma\|\mathbf{x} - \mathbf{c}_i\|^2\right) + \rho . \tag{6}$$

The nonlinear SVM regressions (minimization of (5)) determines the elements of the weight matrix $\mathbf{w}$ and bias ρ. During the SVM training, the support vectors $(\mathbf{c}_i)$ are chosen from the set of input training data. Design parameter ρ defines the shape of RBFs, and it is experimentally chosen to minimize the VC-dimension, which provides a good generalization [13].

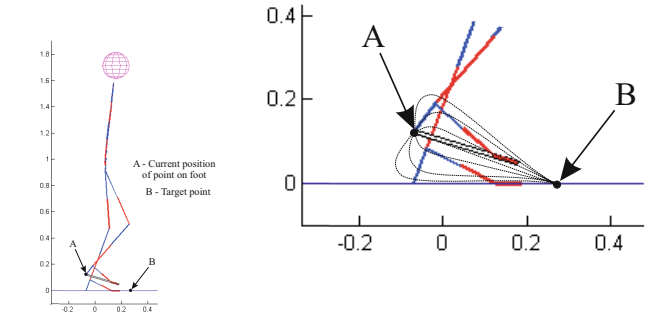
By following the procedure described in the previous section, the training set generation lasted 100s. In this period, 178 different initial postures were generated and the same number of leg bending was performed. The training set was formed on the basis of data from every sampling period (in this case it was 1 ms) and 100 000 input-output pairs were collected. Of all collected data, only 10% were selected for the training data set (for SVM regression) [3]. The penalty parameter C and ε zone of insensitivity were selected as 100 and 0.1, respectively. These parameters are specific for each particular task and their values were obtained by trial-and-error till a satisfactory response was obtained.

4.2 Realization of Primitives by Setting Target Points

Let us consider now the situation when it is necessary to realize a primitive in the case that at the initial moment the system is not at rest. Let us consider this on the example of the system that at the moment of observation is in the single-support phase (as shown in Fig. 7 (*a*)) and the task is to have the swing leg landing by

[3] There are three reasons to select just 10% of available data. A first one is that for the training set a quadratic programming problem has to be solved. This requires large memory resources and training is time consuming. The second reason is that by selecting a training set of such size the problem of over-fitting is avoided. Third reason is that such training set size was proven to be large enough for a successful training dealing with such kind of problems [11].

stepping for-ward. To achieve this, let us observe a characteristic point (in Fig 7 (*a*) it is the heel tip denoted with A), its predefined target position at rest (in Fig. 7 (*a*) denoted with B) and the orientation to be achieved by the realization of the leg stretching. Depending of the velocity of the heel tip at the moment when the leg stretching is to begin, the trajectories of the heel tip to the point A may be different (in Fig. 7 (*b*) shown by thin full lines from A to B).



(*a*) (*b*)

Fig. 7 Humanoid robot prepared for the realization of leg stretching. Current position of the robot's foot is determined with the point A and target is the point B

In other words, at the moment when the realization of the "leg stretching" primitive starts, the velocity of point A may be of arbitrary intensity and direction. Let us denote with $\mathbf{r}_A$ and $\mathbf{r}_B$ the vectors determining positions of the points A and B, so $\mathbf{r}_A$ and $\mathbf{r}_B$ are determined by six components each:

$$\begin{aligned} \mathbf{r}_A &= [r_{Ax}\ r_{Ay}\ r_{Az}\ \phi_{Ax}\ \theta_{Ay}\ \psi_{Az}]^T \\ \mathbf{r}_B &= [r_{Bx}\ r_{By} r_{Bz}\ \phi_{Bx}\ \theta_{By}\ \psi_{Bz}]^T \end{aligned} \tag{7}$$

determining the instantaneous position and orientation and the target (desired) position and orientation of the foot. Vector $\mathbf{r}_A$ is to be calculated while the vector $\mathbf{r}_B$ is predefined with respect to the coordinate system attached to the basic robot link (pelvis).

Let us denote with $\mathbf{v}_A$ the velocity of the point A. The vector $\mathbf{v}_A$ is also determined by six components that represent the linear and angular velocity of the foot link at the point A with respect to the basic robot link:

$$\mathbf{v}_A = [v_{Ax}\ v_{Ay}\ v_{Az}\ \omega_{Ax}\ \omega_{Ay}\ \omega_{Az}]^T . \tag{8}$$

In order to realize a smooth motion of the leg, the desired velocity of point A is determined from the vector field of velocities that lead the point A to the target point B. To form the vector field of velocities it is necessary to predefine the vector $\mathbf{r}_B$ and maximal desired linear and and angular velocity of the foot, v_{max}. The value of the instantaneous velocity is determined in the following way. First, we determine the orth of the vector $\mathbf{r}_e = \mathbf{r}_B - \mathbf{r}_A$. If there would not exist the requirement for

smooth transition from the instantaneous to the desired velocity, then the desired foot velocity would be $v_{max} \cdot \mathbf{r}_e$. The desired foot velocity at the instant t is calculated from the relation:

$$\mathbf{v}_A(t) = (1-b) \cdot \mathbf{v}_A(t_{p0}) + b \cdot v_{int} \cdot \mathbf{r}_e. \tag{9}$$

The coefficient b serves for a gradual transition from the initial foot velocity to the velocity that will bring the foot to the target point. This coefficient is calculated as follows:

$$b = \left(\frac{(t - t_{p0})/dt - 250}{30} \right), \tag{10}$$

where f is the logistic function shown in Fig. 8. It is evident from the figure that 0.3s from the beginning of the realization of the primitive the desired foot velocity is:

$$\mathbf{v}_A(t) \approx v_{int} \cdot \mathbf{r}_e, \tag{11}$$

which means that the vector of linear velocity of point A is directed toward the point B.

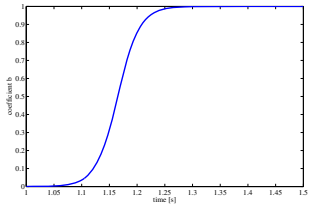


Fig. 8 Values of the coefficient b for the case when $dt = 0.66ms$, $t_{p0} = 1s$, $t = 1.5s$

In order to ensure smooth stopping of the foot at the desired point, the desired velocity is obtained by multiplying the orth of the vector $v\mathbf{r}_e$ with the intensity of the velocity v_{int}. Stopping of the foot begins when the condition $|\mathbf{r}_e| < 0.1 \cdot v_{max}$ is satisfied. Since v_{max} is given in the span $[0-1]$, when, for example, $v_{max} = 0.1$, the foot begins stopping when the point A comes to a distance of 1cm from the equilibrium point B. Hence v_{int} is calculated in the following way:

$$v_{int} = \begin{cases} v_{max} & if \ |\mathbf{r}_e| > 0.1 \cdot v_{max} \\ 10 \cdot |\mathbf{r}_e| & if \ |\mathbf{r}_e| \leq 0.1 \cdot v_{max}. \end{cases} \tag{12}$$

On the basis of the desired velocity $\mathbf{v}_A(t)$, it is possible to calculate the reference velocity at the robot leg joints as follows:

$$\dot{q}_i(t) = J_A^+ \cdot \mathbf{v}_A(t), \tag{13}$$

where J_A^+ is the pseudo-inverse Jacobean calculated for the point A, and the subscript "i" stands for the ordinal numbers of the joints that are included in the realization of the given primitive. Since each leg has 8 DOFs (3 at the hip, 1 at the knee,

3 at the ankle, and 1 at the toes), in equation (13) use is made of the pseudo-inverse Jacobean because the realization of the primitive does not always require exactly 6 DOFs. In order to compensate for the ever-present disturbances, the calculated reference velocities at the joints $\dot{(q)}_i(t)$ are additionally corrected to preserve dynamic balance [4]. The final reference velocity at the robot joint is obtained from the following equation:

$$\dot{q}_i^{ref}(t) = \dot{q}_i(t) + \dot{q}_i^{corr}. \tag{14}$$

In order to follow the reference velocities $\dot{q}_i^{ref}(t)$ PID regulators were implemented at the robot joints, on the basis of which is calculated motor voltage for joint actuation, using the following relation:

$$u_i(t) = u_i(t-dt) - \left(K_P \cdot \dot{q}_i^e(t) + K_I \cdot \sum \dot{q}_i^e(t) + K_D \cdot (\dot{q}_i^e(t) - \dot{q}_i^e(t-dt))\right). \tag{15}$$

5 Simulation Verification

In this section we describe the simulations that were performed to verify the validity of the proposed procedure for generating the robot's motion using only primitives. In the first simulation, the driving torques at the robot's joints were generated by using SVM. In the second simulation use was made of the primitives that were realized by setting the target point and using the model of DC motors for joint actuation.

5.1 Simulation Verification for Primitives Realized Using SVM

After the SVM training has been completed, the performance testing on the humanoid model was performed by simulation. On the basis of specified leg bending intensity, movement duration, current ankle velocity, current leg posture, and angular velocities at the joints (the knee and the hip) the driving torques are obtained from SVM and applied on the robot. The torques drive the system to a new state, new torques are obtained, and the procedure is repeated till the movement is completed. In Fig. 9 are shown two examples of leg bending performed using the trained SVM. The starting posture (angles at the hip and the knee) is selected to correspond approximately to the posture of swing leg after its separation from the ground. Bending intensity and movement duration are defined randomly.

In the example sketched in Fig. 9 (*a*), the initial angles intensities (hip rotation about the x and y axes, and the knee joint rotation about the y axis) were 0^o, 40^o and 10^o, respectively. Bending intensity was 41%, and the motion duration (parameter *S*) 45%. As a consequence, the movement was performed for t_{prim}=0.69s. The distance between the target ankle position (defined by bending intensity) and achieved position was 6.3cm.

[4] The way to calculate the correction $\dot{q}_i^{corr}$ is described in detail in [17].

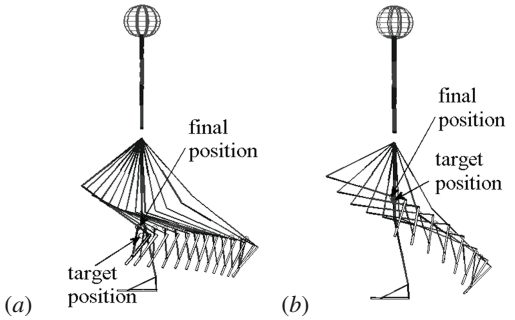


Fig. 9 Examples of leg bending from different starting positions and at different intensities: a) leg bending intensity is 91%, b) leg bending intensity is 50%

In the second example (sketched in Fig. 9 (*b*)), the initial angle values were 0^o, 30^o and 15^o, respectively. Bending intensity was 92%, and motion duration (parameter *S*) was 70%. The motion was performed for t_{prim}=0.46s. The distance between the target ankle position (defined by bending intensity) and achieved position was 2.7cm.

To perform complete walk in swing phase, the leg bending has to be continued by leg stretching. Also, appropriate primitives have to be realized for the supporting leg. In Fig. 10 is shown a stick diagram of the walk composed only from primitives.

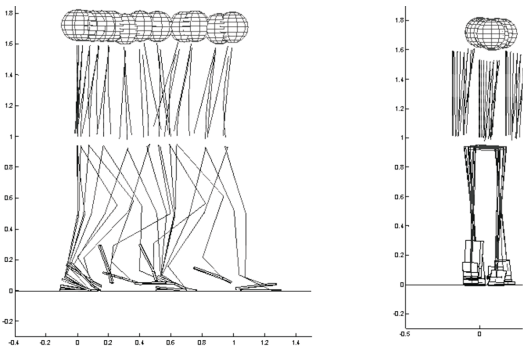


Fig. 10 Humanoid robot walk composed from primitives

At the beginning, the robot stands still with both feet in contact with the ground. When the walk starts, the robot first transfers its weight to left leg in order to be able to step forward and after that right leg lifts up from the ground. Then, the right leg bending is performed first, followed by its stretching. The trunk moves forward and the robot inclines forward too. When the right leg touches the ground, the

double-support phase starts. After that, robot transfers its weight to the right leg and the left leg takes of the ground and system comes again to single-support phase. Then the process continues. The two half-steps lasted 2.3s.

5.2 *Simulation Verification for the Primitives Realized by Setting Target Points*

For the purpose of generating motion by using the primitives that are realized by setting target points, five primitives were implemented, and these are: the primitives for leg bending and stretching, the primitive for robot's forward inclination, the primitive for transferring the robot weight over the support leg, and the primitive for leg landing at the onset of the double-support phase (the first three primitives can be seen in Fig. 1 and the other two in Fig. 11).

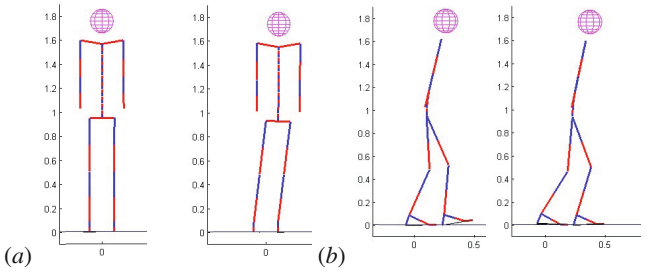


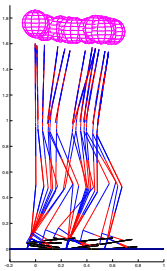
Fig. 11 Stick diagrams of humanoid robot performing the primitives: (*a*) for transferring the robot weight over the support leg, and (*b*) for leg landing

From the aspect of walk realization using primitives, a step can be considered as being composed of four time periods, in accordance with the primitives that are executed during these periods. In the initial moment, the robot is in the double-support phase and standing still with both feet on the ground. To start walking, the robot has to transfer its support to only one foot, so that the other foot could get separated from the ground. Therefore, during the first period, two primitives are successively executed, viz. the primitive for transferring the mass center above the support leg and the leg bending primitive. The execution of the leg bending primitive begins when the projection of the center of mass and ZMP reach the border of the area covered by the foot of the support leg, i.e. when they "enter" the support polygon of the single-support phase, which is just about to begin. A condition for the termination of the first period is that the selected point at the heel tip of the foot of the bending leg is at a 1-cm distance from the target point and that the projection of the mass center of the basic link onto the ground is at the 1-cm distance from the center of the support area of the single-support phase (the area on the ground covered by the foot of the support leg). During this period, in a certain instant, the robot passes from

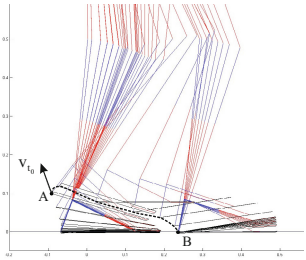
the double-support to the single-support phase. In the second period, the primitives for leg stretching and trunk inclining forward are executed. During this period it is necessary to ensure that in the realization of the leg stretching primitive in swing phase the leg is brought on time to the front of the trunk and thus made ready to form contact with the ground. During this period the humanoid is still in the single-support phase. The second period ends when the leg which was in the swing phase establishes contact with the ground, i.e. when the robot assumes again the double-support phase. The third period is the time of the realization of the primitive for leg landing, while the robot is all the time in the double-support phase. This period terminates when the foot made contact with the ground, at least at three non-collinear contact points. After that comes again the execution of the primitive from the first period (the transfer of the robot's mass center above the support foot). It is necessary to point out once more that the primitives can be realized from the different initial positions and without the need for the robot to stop completely before continuing execution of the next primitive, which is clear from the presented example. At the beginning of the step, the humanoid was standing on both legs, with the parallel feet, and then it realized inclination on the support leg, enabling thus the beginning of the single-support phase. In the last period of the step the system made contact with the ground by its front leg. From that moment on robot was in the double-support phase (the posture of which is different from the double-support phase at the beginning of the step). After that started again the realization of the primitive for transferring robot's mass center above the support leg (of course, the projection of the system's mass center also shifted from the hind to the front leg) in order to enable separation of the hind leg from the ground and re-establishing of the single-support phase. The hind leg separated from the ground just at the moment when the conditions ensuring dynamic balance were fulfilled, i.e. when the projection of the center of mass and ZMP entered the support polygon of the single-support phase which was just about to begin. It should be pointed out once more that the parameters of the primitive for transferring robot's mass center above the support leg was in both cases (i.e. at the beginning and at tend of the half-step) entirely the same. In other words, the parameters of the position of the target point and velocity of the primitive realization were in both cases the same, although the conditions at the beginning of the realization of the primitive were different.

In this way, the cycle of successive execution of the primitives for realization of one half-step is completed. The ending of all three periods described means that the half-step is over, after which the leg that was supporting one, in the next half-step becomes a swing leg and vice versa.

In Fig. 12 (*a*) is presented the result of the simulation of the robot's walk generated by the described procedure. During the execution of the movements it was necessary to activate also the correction for the maintenance of dynamic balance, by which reference velocity of robot's joints was corrected at each instant of the motion. In Fig. 12 (*b*) is shown the position of the point A at the moment of ending of the first and beginning of the second period. At that instant, the velocity of the point A is different from 0, and the linear velocity of A is given by the vector $\mathbf{v}_{t0}$. In the case of an instantaneous change of the direction and velocity of the point A



(*a*)



(*b*)

Fig. 12 Humanoid robot walk composed from the primitives realized by setting target points (*a*); trajectory of point A on foot during realization of leg stretching primitive(*b*)

there would arise large accelerations, which could endanger the dynamic balance of the robot. Due to a gradual change of the desired velocity of the point A, determined from equation 9, as can be seen from Fig. 12 (*b*), the point A in the beginning continues to move in the direction determined by the vector $\mathbf{v}_{t0}$, and then it gradually moves toward the target point B. It should be noted that the positions of the points A and B are determined with respect to the hip of the swing leg. For the sake of a simpler illustration, the position of the point B is shown for the moment when the execution of the primitive for leg stretching is ended.

6 Conclusion

One of the most important characteristics of the environment where humans are living and working is that it is unstructured. In order to be able to share the living and working space with humans, the humanoid robots will have to be capable to perform the online synthesis and execution of a motion that cannot be specified (either defined or programmed) in advance. In this paper, we presented a novel approach to composing complex motions from basic movements, called primitives. A basic characteristic of the proposed method is that the system is able to perform a primitive from different starting points and with different motion parameters. Also, a new primitive can be started when the previous one is completed (or even before the

current movement is ended), without time-consuming calculations. This enables the realization of a human-like motion of the humanoid in an unstructured environment, based on the online synthesis of the motion.

Acknowledgements. This work was funded by the Ministry of Science and Technological Development of the Republic of Serbia in part under contract TR35003 and in part under contract III44008 and by Provincial Secretariat for Science and Technological Development under contract 114-451-2116/2011.

References

1. Borovac, B., Nikolić, M., Raković, M.: How to compensate for the disturbances that jeopardize dynamic balance of a humanoid robot? International Journal of Humanoid Robotics 8, 533–578 (2011)
2. Denk, J., Schmidt, G.: Synthesis of walking primitive databases for biped robots in 3d-environments. In: IEEE International Conference on Robotics and Automation, Proceedings. ICRA 2003, vol. 1, pp. 1343–1349 (2003), doi:10.1109/ROBOT.2003.1241778
3. Hauser, K., Bretl, T., Harada, K., Latombe, J.-C.: Using motion primitives in probabilistic sample-based planning for humanoid robots. In: Akella, S., Amato, N., Huang, W., Mishra, B. (eds.) Algorithmic Foundation of Robotics VII. Springer Tracts in Advanced Robotics, vol. 47, pp. 507–522. Springer, Heidelberg (2008)
4. Hirai, K., Hirose, M., Haikawa, Y., Takenaka, T.: The development of honda humanoid robot. In: Proceedings of IEEE International Conference on Robotics and Automation 1998, vol. 2, pp. 1321–1326 (1998), doi:10.1109/ROBOT.1998.677288
5. Kuffner, J., Nishiwaki, K., Kagami, S., Inaba, M., Inoue, H.: Motion planning for humanoid robots. In: Proc. 11th Int. Symp. of Robotics Research, ISRR 2003 (2003)
6. Kulić, D., Nakamura, Y.: Scaffolding on-line segmentation of full body human motion patterns. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008, pp. 2860–2866 (2008), doi:10.1109/IROS.2008.4650619
7. Kulić, D., Nakamura, Y.: Incremental learning and memory consolidation of whole body human motion primitives. Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems 17(6), 484–507 (2009), http://dx.doi.org/10.1177/1059712309342487, doi:10.1177/1059712309342487
8. Mussa-Ivaldi, F.A., Giszter, S.F., Bizzi, E.: Linear combinations of primitives in vertebrate motor control. Proceedings of the National Academy of Sciences of the United States of America 91(16), 7534–7538 (1994), http://www.jstor.org/stable/2365319
9. Peters, J., Schaal, S.: Policy Learning for Motor Skills. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part II. LNCS, vol. 4985, pp. 233–242. Springer, Heidelberg (2008)
10. Potkonjak, V., Vukobratović, M., Babković, K., Borovac, B.: General model of dynamics of human and humanoid motion: feasibility, potentials and verification. International Journal of Humanoid Robotics 3, 21–48 (2006)
11. Raković, M., Nikolić, M., Borovac, B.: Humanoid Robot Reaching Task Using Support Vector Machine. In: Obdržálek, D., Gottscheber, A. (eds.) EUROBOT 2011. Communications in Computer and Information Science, vol. 161, pp. 263–276. Springer, Heidelberg (2011)

12. Turner, J.D.: On the simulation of discontinuous functions. Journal of Applied Mechanics 68(5), 751–757 (2001), http://link.aip.org/link/AMJ/68/751/1, doi:10.1115/1.1387022
13. Vapnik, V.N.: The nature of statistical learning theory, 2nd edn. Springer-Verlag New York, Inc., New York (2000)
14. Vukobratović, M., Borovac, B.: Zero-moment point - thirty five years of its life. International Journal of Humanoid Robotics 1, 157–173 (2004)
15. Vukobratović, M., Borovac, B., Potkonjak, V.: Towards a unified understanding of basic notions and terms in humanoid robotics. Robotica 25(01), 87–101 (2007)
16. Vukobratović, M., Borovac, B., Raković, M.: Comparison of PID and Fuzzy Logic Controllers in Humanoid Robot Control of Small Disturbances. In: Gottscheber, A., Enderle, S., Obdrzalek, D. (eds.) EUROBOT 2008. Communications in Computer and Information Science, vol. 33, pp. 42–53. Springer, Heidelberg (2009)
17. Vukobratovic, M., Borovac, B., Raković, M., Potkonjak, V., Milinović, M.: On some aspects of humanoid robots gait synthesis and control at small disturbances. International Journal of Humanoid Robotics 5, 119–156 (2008)
18. Yamaguchi, J., Soga, E., Inoue, S., Takanishi, A.: Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking. In: Proceedings of IEEE International Conference on Robotics and Automation 1999, vol. 1, pp. 368–374 (1999), doi:10.1109/ROBOT.1999.770006

Qualitative Evaluation of Flight Controller Performances for Autonomous Quadrotors

Aleksandar Rodić, Gyula Mester, and Ivan Stojković

Abstract. The paper regards to benchmarking and qualitative evaluation of different autonomous quadrotor flight controllers. Three characteristic representatives of frequently used flight control techniques are considered: PID, backstepping and fuzzy. The paper aims to contribute to the objective assessment of quadrotor control performances with respect to the criteria regarding to dynamic performances, trajectory tracking precision, energy efficiency and control robustness upon stochastic internal and/or external perturbation. Qualitative evaluation of the closed-loop system performance should enable the best choice of microcopter control structure. Non-linear modeling, control and numerical simulation of two characteristic flight test-scenarios (indoor as well as outdoor) are described in the paper, too. Obtained simulation results for three representative control algorithms are graphically and table presented, analyzed and discussed.

Keywords: Autonomous quadrotor, flight controller, PID controller, fuzzy controller.

1 Introduction

Over the recent years, many research groups are working in order to exploit the potential advantages of quadrotor rotorcrafts as UMAVs of future. This paper is addressed to problems of controller performances evaluation and analysis. The main benefits of this research concern with achievement of a controller architecture that

Aleksandar Rodić · Ivan Stojković
Mihajlo Pupin Institute, Robotics Laboratory,
Volgina 15, 11060 Belgrade, Serbia
e-mail: {aleksandar.rodic,ivan.stojkovic}@pupin.rs

Gyula Mester
University of Szeged, Faculty of Engineering,
6725 Szeged, Moszkvai krt. 5-7, Hungary
e-mail: drmestergyula@gmail.com

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 115–134.
DOI: 10.1007/978-3-642-33959-2_7

should enable quadrotor high dynamic performances, robustness to external perturbations as well as satisfactory trajectory tracking precision.

The overview of the most important techniques and the respective publications are listed in the text to follow. The Lyapunov Theory of Stability was frequently considered in the literature as for example [1] - [3]. According to this technique, it is possible to ensure, under certain condition, the asymptotical stability of the micro copter.

The linear control techniques based on proportional-integral-derivative (PID) feedback structure [4], [5] are frequently used with micro copters for flight control. The strength of the PID feedback is the exponential convergence property mainly due to the compensation of the Coriolis and gyroscopic terms. On the contrary a PID structure does not require some specific model parameters and the control law is much simpler to implement.

Some researchers used adaptive techniques [6] [7]. These methods provide good performance with parametric uncertainties and unmodeled dynamics.

The control based on a Linear quadratic Regulator (LQR) [4] [8] [9] shows fine results. The main advantage of this technique is that the optimal input signal turns out to be obtainable from full state feedback (by solving the Ricatti equation). On the other hand the analytical solution to the Ricatti equation is difficult to compute.

In the recent time, very popular control technique is done with backstepping control [10]. In the respective publications the convergence of the quadrotor internal states is guaranteed, but a lot of computation is required.

Dynamic control methods are based on use of a dynamic feedback [11] [12]. This technique is implemented in few quadrotor projects to transform the closed loop part of the system into a linear, controllable and decoupled subsystem.

There are also control methods based on utilization of visual feedback information. The camera used for this purpose can be mounted on-board [13] [14] (fixed on the helicopter) or off-board [15] [16] (fixed to the ground).

Other control algorithms belong to the class so called the knowledge-based algorithms. Main characteristics of these methods are that they represent non-linear techniques that do not require knowledge about the model of system. These techniques assume quadrotor plant as a black-box. They use control platform with fuzzy techniques[17], neural networks [18] and reinforcement learning [19].

In [20] a procedure allowing the computation of time-optimal quadrotor maneuvers for arbitrary initial and final states by solving the boundary value problem induced by the minimum principle. The algorithm allows the computation of quadrotor maneuvers that satisfy Pontryagins minimum principle with respect to time-optimality. Such maneuvers provide a useful lower bound on the duration of maneuvers, which can be used to assess performance of controllers and vehicle design parameters. The usage of the computed maneuvers as a benchmark is demonstrated by evaluating quadrotor design parameters, and a linear feedback control law as an example of a control strategy.

The main contribution of the paper regards to design and simulation of the appropriate simulation benchmark procedures (indoor as well outdoor) to be used for the objective assessment and evaluation of different control algorithms applied to microcopoter rotorcrafts.

2 Quadrotor Dynamics Modeling

The quadrotor is satisfactory well modeled [21] - [23] with a four rotors in a cross configuration Fig. 1. This cross structure is quite thin and light, however it shows robustness by linking mechanically the motors (which are heavier than the structure). Each propeller is connected to the motor through the reduction gears. All the propellers axes of rotation are fixed and parallel. Furthermore, they have fixed-pitch blades and their air flows point downwards (to get an upward lift). These considerations point out that the structure is quite rigid and the only things that can vary are the propeller speeds.

As shown in Fig. 1, one pair of opposite propellers of quadrotor rotates clockwise (2 and 4), whereas the other pair rotates anticlockwise (1 and 3). This way it is able to avoid the yaw drift due to reactive torques. This configuration also offers the advantage of lateral motion without changing the pitch of the propeller blades. Fixed pitch simplifies rotor mechanics and reduces the gyroscopic effects. Control of quadrotor is achieved by commanding different speeds to different propellers, which in turn produces differential aerodynamic forces and moments. For hovering, all four propellers rotate at same speed. For vertical motion, the speed of all four propellers is increased or decreased by the same amount, simultaneously. In order to pitch and move laterally in that direction, speed of propellers 3 and 1 is changed conversely. Similarly, for roll and corresponding lateral motion, speed of propellers 2 and 4 is changed conversely. To produce yaw, the speed of one pair of two oppositely placed propellers is increased while the speed of the other pair is decreased by the same amount. This way, overall thrust produced is same, but differential drag moment creates yawing motion. In spite of four actuators, the quadrotor is still an under-actuated system.

The Fig. 1 shows the structure model [22] [23] in hovering condition, where all the propellers have the same speed of rotation $\omega_i = \omega_H,\ i = 1,\ldots,4$. In the Fig. 1 all the propellers rotate at the same (hovering) speed ω_H (rad/s) to counterbalance the acceleration due to gravity. Thus, the quadrotor performs stationary flight and no forces or torques moves it from its position. Even though, the quadrotor has 6 DOFs, it is equipped just with four propellers hence it is not possible to reach a desired setpoint for all the DOFs, but at maximum four. However, thanks to its structure, it is quite easy to choose the four best controllable variables and to decouple them to make control easier. The four quadrotor targets are thus related to the four basic movements which allow the microcopter to reach a certain height and attitude.

To describe the motion of a 6 DOF rigid body it is usual to define two reference frames Fig. 1 : (i) the earth inertial frame (E-frame), and (ii) the body-fixed frame (B-frame).

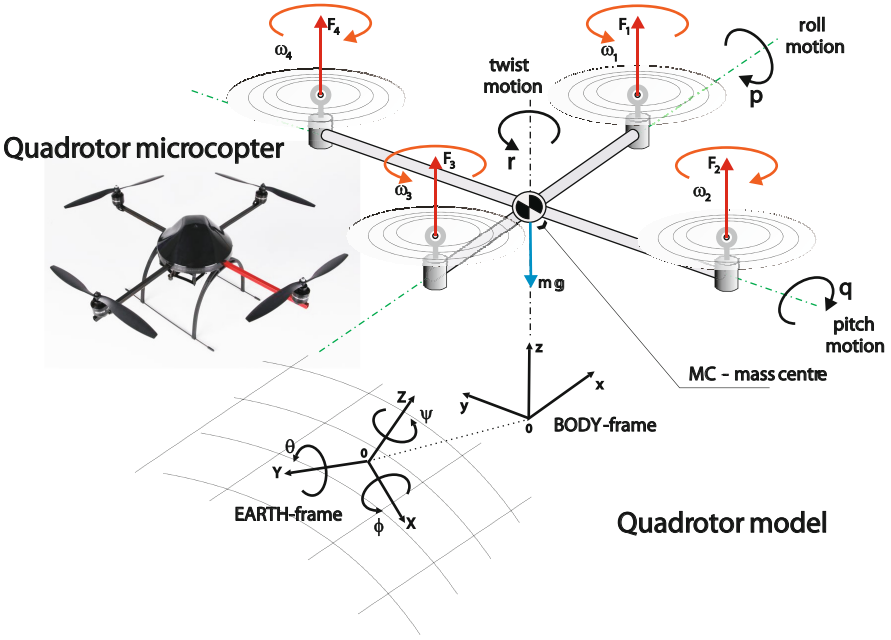


Fig. 1 Quadrotor rotorcraft – a non-linear dynamic system that uses synergy of its rotary-wings to fly. Commonly used model of quadrotor with corresponding degrees of freedom. Coordinate systems assumed to enable model derivation.

The linear position of the helicopter (X,Y,Z) is determined by the coordinates of the vector between the origin of the B-frame and the origin of the E-frame according to equation. The angular position (or attitude) of the helicopter (φ, θ, ψ) is defined by the orientation of the B-frame with respect to the E-frame. The vector that describes quadrotor position and orientation is:

$$s = [X \quad Y \quad Z \quad \varphi \quad \theta \quad \psi]^T. \tag{1}$$

The generalized quadrotor velocity expressed in the B-frame can be written as [1]:

$$v = [u \quad \upsilon \quad w \quad p \quad q \quad r]^T, \tag{2}$$

where, the u, υ, w represent linear velocity components in the B-frame, while p, q, r are corresponding angular velocities of rotation about corresponding roll, pitch and yaw axes. Finally, the kinematical model of the quadrotor correlates the motions in these two coordinate systems [21].

The quadrotor dynamics can be described in the known form [21], extended by adding of the air-resistance member. The equation that describes model in B-frame is:

$$M_B\dot{v} + C_B(v)v + G_B(s) + R_a(v) = O_B(v)\Omega + E_B\Omega^2, \tag{3}$$

where $\dot{v}$ is the generalized acceleration vector with respect to (w.r.t.) the B-frame, M_B is the system inertia matrix, C_B is the Coriolis-centripetal matrix and G_B is the gravitational force vector, all expressed w.r.t. the B-frame. The R_a is the air-resistance vector. The O_B and E_B are the gyroscopic propeller matrix and the movement matrix, successively. The gyroscopic propeller matrix O_B depends on total rotational moment of inertia around the propeller axis and corresponding angular speeds p, q. The matrix E_B depends on the design parameters – thrust and drag coefficients. Air resistance forces appear as an external perturbation to the quadrotor translational movements in longitudinal (x), lateral (y) and normal (z) direction w.r.t B-frame. The drag force R_a depends on a magnitude of body-fluid relative velocity. It takes into account both - air resistance and wind gust.

$$R_a = \frac{1}{2}\rho \cdot \begin{bmatrix} (\dot{x} - v_{w,x})^2 \cdot C_{d,x} \cdot A_x \\ (\dot{y} - v_{w,y})^2 \cdot C_{d,y} \cdot A_y \\ (\dot{z} - v_{w,z})^2 \cdot C_{d,z} \cdot A_z. \end{bmatrix} \tag{4}$$

In equation (4) ρ is the air density, $C_{d,x}$, $C_{d,y}$ and $C_{d,z}$ are drag coefficients, A_x, A_y, A_z are reference areas exposed against the streaming fluid, $v_{w,x}$, $v_{w,y}$, $v_{w,z}$ are wind velocity components in the particular directions w.r.t B-frame.

Equation (3), after certain rearrangement and transformation from the B-frame space to E-frame space, can be written in the scalar form suitable for controller design. Now, the model of quadrotor dynamics can be described by a system of equations [21]:

$$\begin{aligned}
\ddot{X} &= (\sin\psi\sin\varphi + \cos\psi\sin\theta\cos\varphi)\tfrac{U_1}{m} - \tfrac{R_{a,x}}{m}, \\
\ddot{Y} &= (-\cos\psi\sin\varphi + \sin\psi\sin\theta\cos\varphi)\tfrac{U_1}{m} - \tfrac{R_{a,y}}{m}, \\
\ddot{Z} &= -g + \cos\theta\cos\varphi\tfrac{U_1}{m} - \tfrac{R_{a,z}}{m}, \\
\ddot{\varphi} &= \tfrac{I_{YY}-I_{ZZ}}{I_{XX}}\dot{\theta}\dot{\psi} - \tfrac{J_{TP}}{I_{XX}}\dot{\theta}\,\Omega_r + \tfrac{U_2}{I_{XX}}, \\
\ddot{\theta} &= \tfrac{I_{ZZ}-I_{XX}}{I_{YY}}\dot{\varphi}\dot{\psi} + \tfrac{J_{TP}}{I_{YY}}\dot{\varphi}\,\Omega_r + \tfrac{U_3}{I_{YY}}, \\
\ddot{\psi} &= \tfrac{I_{XX}-I_{YY}}{I_{ZZ}}\dot{\varphi}\dot{\theta} + \tfrac{U_4}{I_{ZZ}},
\end{aligned} \tag{5}$$

where the propeller's speed inputs are given through equation (6) [21]:

$$\underline{U} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b\left({\omega_1}^2 + {\omega_2}^2 + {\omega_3}^2 + {\omega_4}^2\right) \\ l\,b\left(-{\omega_2}^2 + {\omega_4}^2\right) \\ l\,b\left(-{\omega_1}^2 + {\omega_3}^2\right) \\ d\left(-{\omega_1}^2 + {\omega_2}^2 - {\omega_3}^2 + {\omega_4}^2\right). \end{bmatrix} \tag{6}$$

The movement vector $\underline{U}$ in (6) relates to the movement force and torques acting in quadrotor body MC. The component U_1 acts along the z-axis of the B-frame while the movement torques U_2, U_3, U_4 act about the particular B-frame axes - roll, pitch and yaw. The overall propeller's speed Ω_r (rad/s) is defined by equation (7):

$$\Omega_r = -\omega_1 + \omega_2 - \omega_3 + \omega_4. \tag{7}$$

Quadrotor is equipped with four fixed-pitch rotors, each one includes a Brush-Less Direct Current (BLDC) motor, a one-stage gearbox and a rotary-wing (propeller). The entire rotor dynamics can be approximated by a first order transfer function, sufficient to reproduce the dynamics between the propeller's speed set-point and its true speed. The model form is given in (8) [21]:

$$G_m(s) = \frac{K_m}{T_m s + 1} \cdot e^{-\tau_m s}, \tag{8}$$

where K_m, T_m and τ_m are corresponding motor parameters related to gain, time constant and time delay.

3 Control System Architecture

Commonly used control system architecture of autonomous quadrotor microcopter is presented in Fig. 2 [21] [24] [25]. The task planning block is in debt to determine referent 3D rotorcraft trajectory as well as to propose the referent flight speed along the trajectory. The task planning block generates referent path based on flight parameters and microcopter task imposed.

Position control block Fig. 2 has to ensure accurate 3D trajectory tracking. It represents so called outside control loop. Based on sensory information (GPS, IR, SONAR) about the referent positions (speeds) and corresponding actual ones defined in the inertial coordinate system (E-frame), the position controller calculates referent attitude position of quadrotor body (pitch θ_{ref} and roll angle φ_{ref}) that have to enable desired motion.

Inner control block represents the core of the control scheme. It is responsible for the attitude control of quadrotor system. Appropriate attitude control ensures in an indirect way required flight performances in the particular directions of motion such as longitudinal, lateral as well as vertical. Inner control block processes the task and sensor data and provides a signal for basic movements which balances the position error. Equation (9) is used in this block to transfer an acceleration command to a basic movement one. The control rules to be used to estimate the acceleration commands are to be considered in the next section.

The essence of building control scheme presented in Fig. 2 is that by controlling a body attitude (within an inner loop) it is enabled controlling of the linear rotorcraft movements. Also, high robustness to parameter and structural uncertainties of system modeling are required in design of attitude control algorithm.

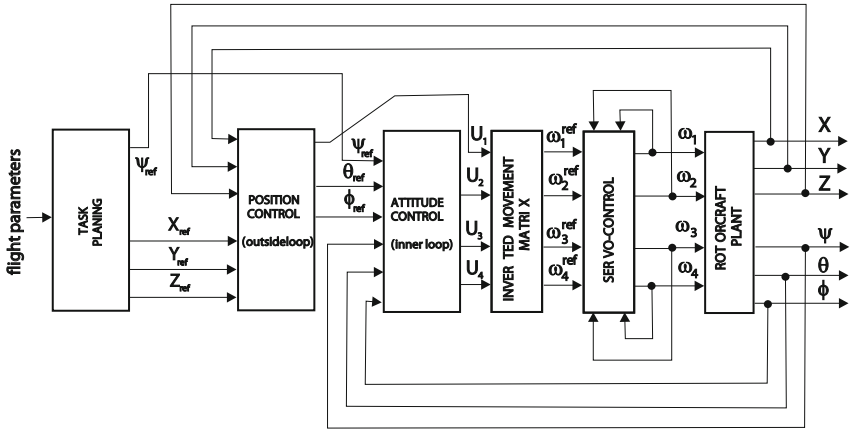


Fig. 2 Block-scheme of the global control system architecture of autonomous UMAVs

Inverted Movements Matrix block Fig. 2 is used to compute the propeller's squared speed from the four basic movement signals. Since the determinant of the movement matrix is different than zero, it can be inverted to find the vector Ω^2. The computation block is shown in equation (9) [21] [24].

$$\Omega^2 = \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}_{ref} = \begin{bmatrix} \frac{1}{4b}U_1 - \frac{1}{2bl}U_3 - \frac{1}{4d}U_4 \\ \frac{1}{4b}U_1 - \frac{1}{2bl}U_2 + \frac{1}{4d}U_4 \\ \frac{1}{4b}U_1 + \frac{1}{2bl}U_3 - \frac{1}{4d}U_4 \\ \frac{1}{4b}U_1 + \frac{1}{2bl}U_2 + \frac{1}{4d}U_4. \end{bmatrix} \tag{9}$$

Variety of control algorithms can be implemented within the flight controller presented in Fig. 2. This paper aims to propose corresponding testing procedure and a qualitative evaluation of three representative flight control techniques. These are: (i) conventional linear PID regulator, (ii) non-linear, model-based backstepping method (BSM), and (iii) non-linear, knowledge-based fuzzy logic control (FLC) based on use of a Fuzzy Inference System. The choice was made due to the reason that the fore mentioned control algorithms are frequently used in the open literature. The PID controller is assumed as the main representative of the linear control techniques. The backstepping method was considered in the paper as a typical representative of the model-based, dynamic, non-linear control method while the fuzzy controller was assumed as the representative of so called knowledge-based control algorithms.

3.1 PID Controller

A proportional-integrative-derivative controller (PID) is most common feedback form in all kinds of control systems, and is also being used for flight control of quadrotor [4] [5]. The ideal PID is represented in continuous time domain as:

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau) d\tau + k_D \frac{de(t)}{dt}, \tag{10}$$

where the terms k_P, k_I and k_D are proportional, integral and derivative gain, respectively, $u(t)$ is the output of the controller, and input $e(t)$, is the reference tracking error. The PID quadrotor flight controller consists of six PID controllers for the particular state coordinates (1), see Fig. 1.

3.2 Backstepping Controller

The backstepping technique is recursive design methodology that makes use of Lyapunov stability theory to force the system to follow a desired trajectory. Backstepping approach to quadrotor flight control was successfully applied in number of researches such as for example [10] [24].

First, the dynamical model from (5) is rewritten in state-space form $\dot{X} = f(X,U)$, by introducing $X = [x_1, \ldots, x_{12}]^T \in \Re^{12}$as space vector of the system (7) [24]:

$$\begin{array}{lll} x_1 = \phi, & x_5 = \psi, & x_9 = Y, \\ x_2 = \dot{x}_1 = \dot{\phi}, & x_6 = \dot{x}_5 = \dot{\psi}, & x_{10} = \dot{x}_9 = \dot{Y}, \\ x_3 = \theta, & x_7 = X, & x_{11} = Z, \\ x_4 = \dot{x}_3 = \dot{\theta}, & x_8 = \dot{x}_7 = \dot{X}, & x_{12} = \dot{x}_{11} = \dot{Z}. \end{array} \tag{11}$$

Next, the x- coordinates are transformed into the new z- coordinates by means of a diffeomorphism [24].

$$\begin{array}{ll} z_1 = x_{1\_ref} - x_1, & z_7 = x_{7\_ref} - x_7, \\ z_2 = x_2 - \dot{x}_{1\_ref} - \alpha_1 z_1, & z_8 = x_8 - \dot{x}_{7\_ref} - \alpha_7 z_7, \\ z_3 = x_{3\_ref} - x_3, & z_9 = x_{9\_ref} - x_9, \\ z_4 = x_4 - \dot{x}_{3\_ref} - \alpha_3 z_3, & z_{10} = x_{10} - \dot{x}_{9\_ref} - \alpha_9 z_9, \\ z_5 = x_{5\_ref} - x_5, & z_{11} = x_{11\_ref} - x_{11}, \\ z_6 = x_6 - \dot{x}_{5\_ref} - \alpha_5 z_5, & z_{12} = x_{12} - \dot{x}_{11\_ref} - \alpha_{11} z_{11}. \end{array} \tag{12}$$

Introducing the partial Lyapunov functions (example for z_1 and z_2) $V_1(z_1) = \frac{1}{2} z_1^2$ and $V_2(z_1, z_2) = \frac{1}{2}(z_1^2 + z_2^2)$, it is possible to determine the control law such that $\dot{V}_2 < 0$ and therefore the global asymptotic stability will be guaranteed, according to Lyapunov stability theorem, and x_1 tends to x_{1_ref}. Applying this procedure to all x- coordinates results in the following backstepping controller (9):

$$
\begin{aligned}
U_X &= \tfrac{m}{U_1}(z_7 - \alpha_7(z_8 + \alpha_7 z_7) - \alpha_8 z_8),\\
U_Y &= \tfrac{m}{U_1}(z_9 - \alpha_9(z_{10} + \alpha_9 z_9) - \alpha_{10} z_{10}),\\
U_1 &= \tfrac{m}{\cos x_1 \cos x_3}(z_{11} + g - \alpha_{11}(z_{12} + \alpha_{11} z_{11}) - \alpha_{12} z_{12}),\\
U_2 &= I_{XX}(z_1 - \tfrac{I_{YY} - I_{ZZ}}{I_{XX}} x_4 x_6 + \tfrac{J_{TP}}{I_{XX}} x_4 \Omega_r - \alpha_1(z_2 + \alpha_1 z_1) - \alpha_2 z_2),\\
U_3 &= I_{YY}(z_3 - \tfrac{I_{ZZ} - I_{XX}}{I_{YY}} x_2 x_6 - \tfrac{J_{TP}}{I_{YY}} x_2 \Omega_r - \alpha_3(z_4 + \alpha_3 z_3) - \alpha_4 z_4),\\
U_4 &= I_{ZZ}(z_5 - \tfrac{I_{XX} - I_{YY}}{I_{ZZ}} x_2 x_4 - \alpha_5(z_6 + \alpha_5 z_5) - \alpha_6 z_6),
\end{aligned}
\tag{13}
$$

where $\alpha_{1,2,\ldots,12} > 0$ are parameters of the backstepping flight controller.

3.3 *Fuzzy Controller*

While some conventional control methods heavily depends on the exact model of controlled system, fuzzy controllers can be designed intuitively in light of the knowledge acquired on the behaviour of the system. This knowledge is often gained through experience and common sense, regardless of the mathematical model of the dynamics governing its behaviour, and it is in the form of set of rules that tries to mimic human-like reasoning. In [25] a Mamdani type of fuzzy inference is used to control quadrotor, and in [26] the comparison of Mamdani and Takagi-Sugeno-Kang (TSK) fuzzy controllers is conducted. The controller that will be implemented here consists of six FLCs, one for each particular state (1), that are in form of zero order TSK fuzzy inference system (14):

$$
u = FLC(e, \dot{e}) = \frac{\sum_{k=1}^{n} \mu_{k,1}(e) \cdot \mu_{k,2}(\dot{e}) \cdot C_k}{\sum_{k=1}^{n} \mu_{k,1}(e) \cdot \mu_{k,2}(\dot{e})}. \tag{14}
$$

FLC described in (14) works with error e and the error rate $\dot{e}$. Actually, inputs in the FLC are first preprocessed, then they are normalized to fit membership function intervals $[-1\ 1]$ and $[-3\ 3]$, and finally feed to FLC. The output of the FLC is control action u. Each input variable possess the corresponding three fuzzy sets NEGATIVE, ZERO and POSITIVE and they are presented in Fig. 3. Output membership functions are fuzzy singletons $C_N = -1$, $C_Z = 0$ and $C_P = 1$.

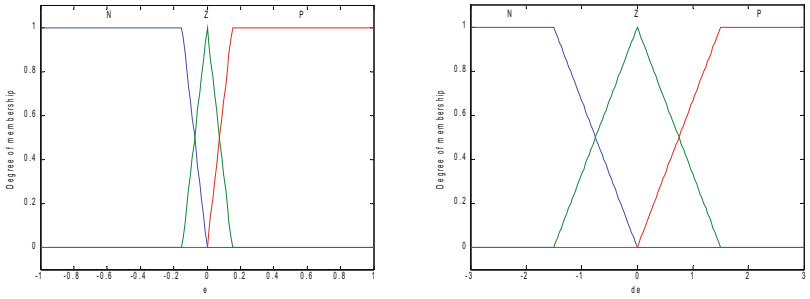


Fig. 3 Membership functions for input variables

Fuzzy rules are in the form: If (e is $\mu_{k,1}$) and ($\dot{e}$ is $\mu_{k,2}$) then (u is C_k), and there are nine rules in the Fuzzy Rule Base that are presented in Table 1.

Table 1 Table of fuzzy rules used for the flight control of quadrotor

Fuzzy rule base			
Rule No	Input e	Input $\dot{e}$	Output u
1	NEGATIVE	NEGATIVE	NEGATIVE
2	NEGATIVE	ZERO	NEGATIVE
3	NEGATIVE	POSITIVE	ZERO
4	ZERO	NEGATIVE	NEGATIVE
5	ZERO	ZERO	ZERO
6	ZERO	POSITIVE	POSITIVE
7	POSITIVE	NEGATIVE	ZERO
8	POSITIVE	ZERO	POSITIVE
9	POSITIVE	POSITIVE	POSITIVE

4 Simulation Experiments and Flight Controller Evaluation

For the purpose of testing flight controller the following quadrotor parameters are assumed Table 2, [22] [23]. The parameters fit the model described in Sect. 2.

For the purpose of analysis and qualitative evaluation of quadrotor flight controller performances, three representative control algorithms (PID, BSM and FLC) are considered. Control parameters of the PID regulator and Backstepping controller are given in Tables 3 and 4. Fuzzy control parameters are given in Sect. 3.3. Control parameters from Tables 3 and 4 are determined by simulation.

Table 2 Quadrotor model parameters used in simulation experiments

Model parameters					
Symbol	Value	Unit	Symbol	Value	Unit
m	1.0	kg	ρ	1.225	kg/m^3
I_{XX}, I_{YY}	$8.1 \cdot 10^{-3}$	Nms^2	A_x, A_y	0.0121	m^2
I_{ZZ}	$14.2 \cdot 10^{-3}$	Nms^2	A_z	0.0143	m^2
b	$54.2 \cdot 10^{-6}$	Ns^2	$C_{d,x}, C_{d,y}$	1.125	–
d	$1.1 \cdot 10^{-6}$	Nms^2	$C_{d,z}$	1.04	–
l	0.24	m	K_m	0.973	–
J_{TP}	$104 \cdot 10^{-6}$	–	T_m	0.113	–
g	9.81	m/s^2	τ_m	0.112	s

Table 3 PID regulator parameters used in simulation

PID gains	State coordinates i=1,...,6					
	1	2	3	4	5	6
proportional	2.22	1.84	1.90	0.30	0.30	0.48
integral	0.01	0.01	0.01	0.20	0.20	0.000001
differential	3.42	2.54	2.28	0.10	0.10	0.04

Table 4 Backstepping controller parameters used in simulation

α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_{91}	α_{10}	α_{11}	α_{12}
10.7	2.0	9.5	3.8	2.2	2.1	2.0	3.0	2.0	3.0	3.0	3.0

Assessment and qualitative evaluation of three representative control techniques frequently used with UMAV are accomplished upon the criteria imposed. The following criteria are introduced: (i) criterion on fine dynamic performances; (ii) criterion on trajectory tracking accuracy; (iii) criterion on control robustness upon the external perturbation; and (iv) criterion on energy efficiency. Control algorithms chosen are evaluated by comparison of the simulation results obtained for the same control object and same flight conditions. Two experimental scenarios are considered as the characteristic benchmarking procedures. These are: (i) dynamic quadrotor flight in the 3D-loop manoever, and (ii) typical cruising flight along the trajectory introduced by setting waypoints with the pre-defined GPS coordinates. Chosen benchmarking tests enable credible assessment of different control techniques under the same conditions.

Dynamic quadrotor flight regards to a microcopter movement in the perpendicular planes in a rather narrow space Fig. 4. It is accomplished by flying in the 3D-loop about a horizontal and a vertical rod set in such a way to be 2 meters far one from another. The curve-linear, smooth loop (trajectory) is pre-defined by introducing 8 key-waypoints Fig. 4. The trajectory defined includes several flight maneuvers: (i) throttle movements in the vertical direction (1-2 and 7-8), (ii) counter-clockwise roll movements (2-3-4 and 5-6-7), (iii) tilt movement about the pitch axis (4-5), and short (iv) hovering with the constant propeller speed (in the point 2 i.e. 7). Quadrotor is required to track the imposed trajectory-loop shown in Fig. 4 moving along at a low speed of maximal value 0.5 (m/s) and to repeat the same path for 33% increased average speed with maximum of 1 (m/s). Flying in the loop, quadrotor is subjected to influence of the inertia and centripetal forces that tend to run a rotorcraft away from the desired path as well as to disturb its dynamic performances (keeping attitude within the allowed range, smooth acceleration profile, no vibration and turbulence, etc.). The obtained simulation results for the three representative control techniques chosen are summarized, analyzed and commented in the text to follow.

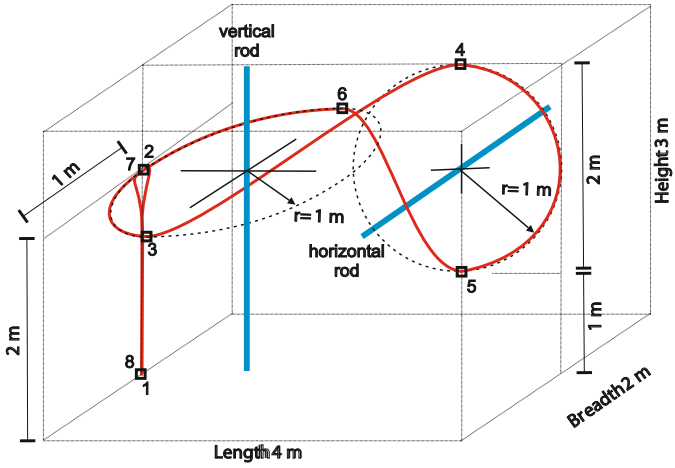


Fig. 4 Trajectory-loop for testing of quadrotor dynamic flight perfromances

Analyzing the simulation results, Backstepping method ensures the best control performances in sense of trajectory tracking precision. Other two concurrent algorithms have slightly better characteristics in sense of energy efficiency (less consumptions). By increasing of flight speed dynamic effects become influential upon the system performances. Consequently, Backstepping method is more sensitive to changing of flight speed than other two controllers PID and FLC. Degradation of control system performances with excitation of dynamic modes in the case of BSM implementation are shown in Figs. 5 and 6.

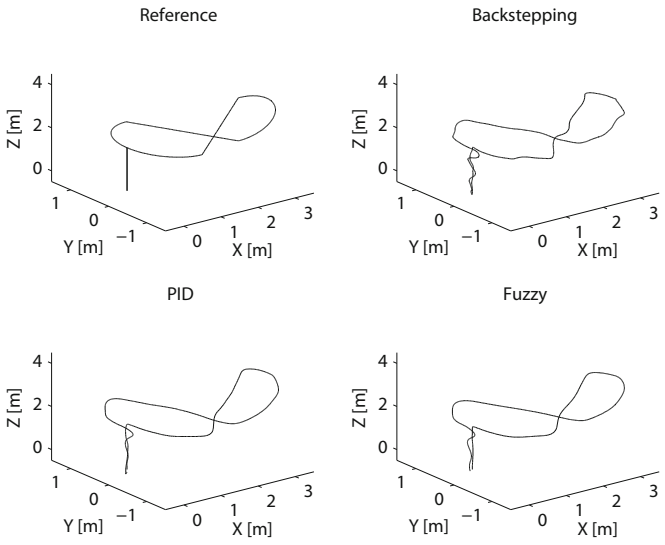


Fig. 5 Trajectory tracking accuracy of the reference path obtained for three examined control techniques and for the case of low speed flight

Corresponding position error in X-, Y-, Z- and Yaw-direction for the "increased" experimental flight speeds is presented in Fig. 7. Corresponding attitude deflections in the roll and pitch directions are shown in Fig. 8.

The second benchmarking procedure (test) considered in the paper regards to assessment of control techniques synthesized to navigate quadrocopter towards predefined waypoints where the microcopter trajectory is imposed by setting of the reference waypoints whose GPS coordinates (longitude, latitude and altitude) are acquired by use of the Google Earth software [27], Fig. 9.

A multi-segment trajectory of 5689 (m) is chosen as test trajectory. It is defined by the 7 waypoints (1-8-3-4-12-10-28-1) whose coordinates are presented in Table 6. Quadrocopter is required to track the trajectory by constant cruising speed of 5 (m/s). When approaching the particular waypoints, quadrocopter varies its flight speed, slows down in order to avoid large deviations of position from the reference path due to inertial effects that become rather expressed especially at high speeds of flight.

Control algorithms PID, BSM and FLC are tested upon the criteria regarding trajectory tracking accuracy, energy efficiency and control robustness upon the internal and external perturbations. In this case, a sudden wind gust is considered as an external perturbation while sensor noise is considered as internal perturbation of the system. Side wind gust is modelled in the paper as additional air-resistance force produced at the quadrotor body due to air streaming (wind blowing). It is assumed the case of south-east wind (143 degrees w.r.t. longitudinal axis) blowing with a constant speed amplitude of 18 (km/h).

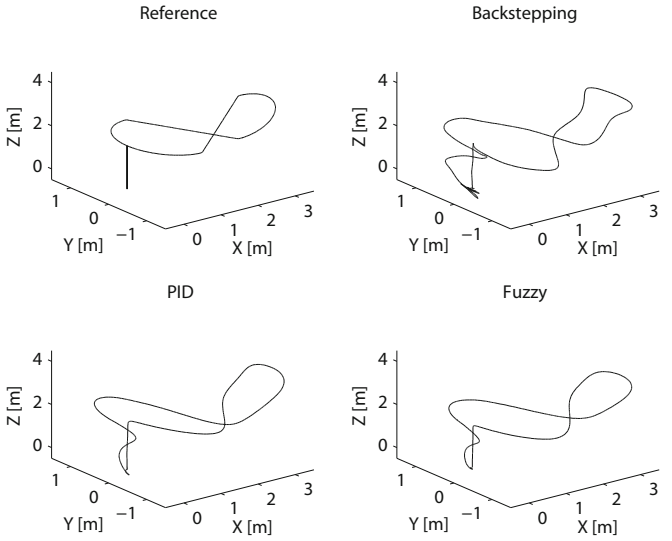


Fig. 6 Trajectory tracking accuracy of the reference path obtained for three considered control techniques and for the case of increased flight speed

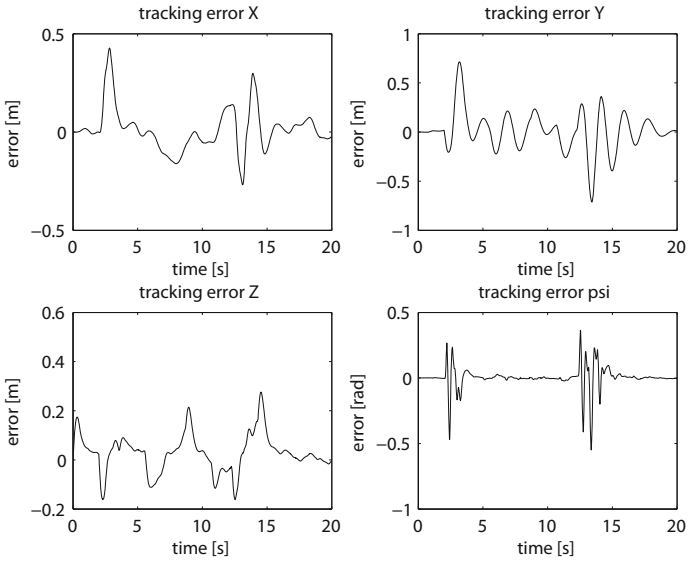


Fig. 7 Position error indicators obtained for the increased flight speed and BSM controller

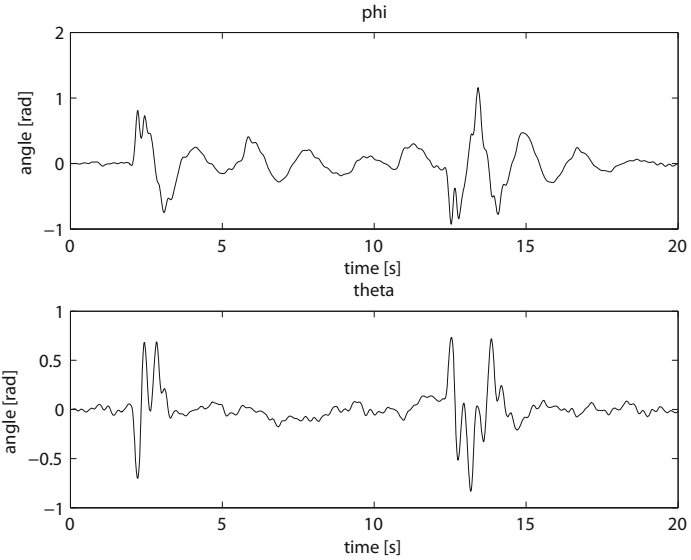


Fig. 8 Attitude deflection (roll and pitch angles) obtained for the increased flight speed and BSM

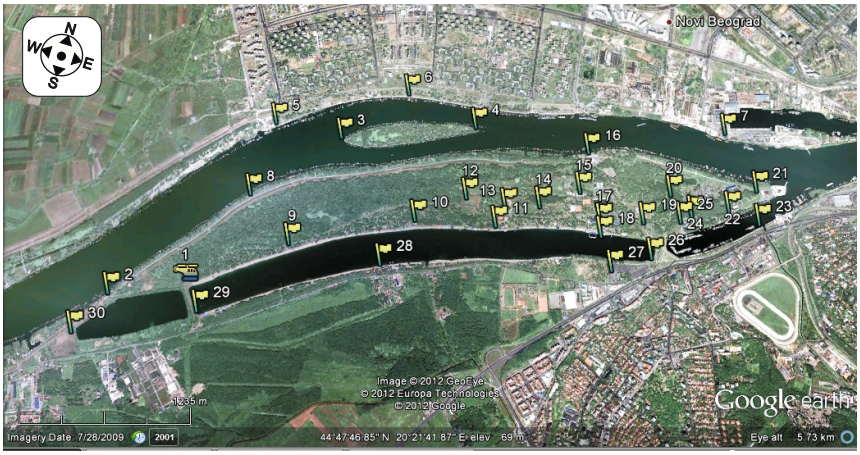


Fig. 9 Fragment of the Google-Earth map [27]. The site "ADA" park of nature (Belgrade, Serbia). Flags in the map are set to mark the particular waypoints used to plot a contour of the desired path model

Table 5 Performance indicators of quadrotor controllers for outdoor flight test scenario

	Perturbation	PID maximal/average	VBSM maximal/average	VFLC maximal/average
Flight duration (s)	noise wind	**1172.00** **1172.20**	1172.30 1172.35	1175.70 1178.30
Energy consumptions (J)	noise wind	**50765** **56930**	52475 56936	50836 57122
Position error w.r.t. reference traject. (m)	noise wind	5.55 / 1.57 7.15 / 1.85	**5.035 / 1.45** **5.46 / 1.558**	8.32 / 2.87 16.03 / 3.99
Deviation from the reference altitude (m)	noise wind	**0.15** / 0.0326 0.035 / 0.0041	0.35 / 0.0698 **0.015** / **0.0001**	0.205 / **0.0305** 0.154 / 0.0022

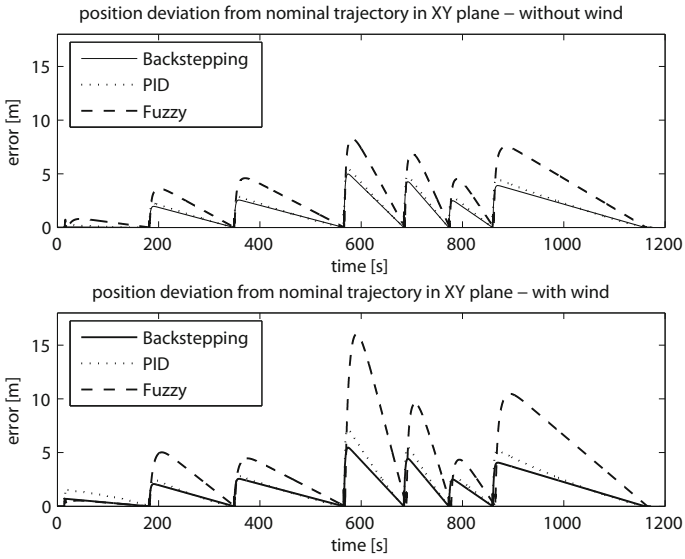


Fig. 10 Trajectory tracking accuracy obtained for implementation of the PID, BSM and FLC controllers in the cases without and with presence external perturbation

Investigating the simulation results reviewed in Table 5, the Backstepping method achieves better control performances than the concurrent PID and FLC controllers bearing in mind the criterion on tracking accuracy. The BSM control shows satisfactory robustness to a side wind gust as external perturbation, too. PID method, as a representative linear technique, achieves satisfactory precision of tracking and better efficiency regarding energy criterion. PID and BSM controllers enable better

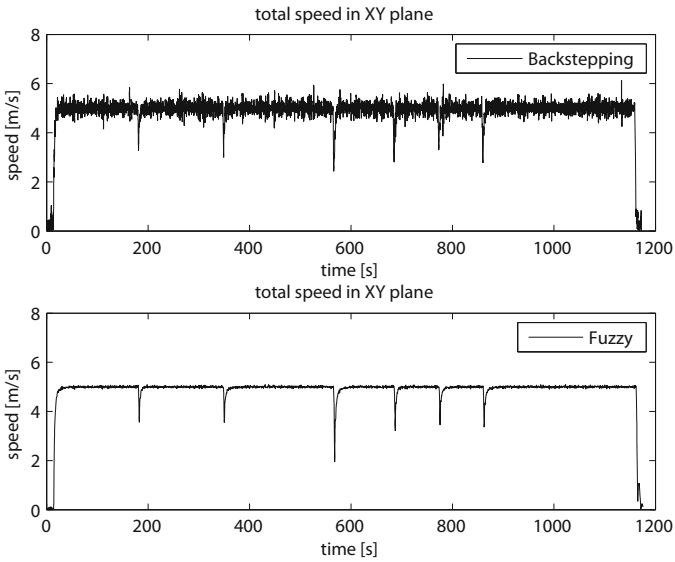


Fig. 11 Flight speed obtained for implementation of the backstepping and fuzzy controller

Table 6 GPS coordinates of the waypoints of the experimental test trajectory 1-8-3-4-12-10-28-1 shown in Fig. 9

Point number	Latitude	Longitude	Altitude (m)
1	44° 46′ 40.82″	20° 22′ 21.21″	72
3	44° 47′ 26.61″	20° 22′ 57.66″	67
4	44° 47′ 38.12″	20° 23′ 14.35″	67
8	44° 47′ 06.30″	20° 22′ 33.93″	74
10	44° 47′ 10.40″	20° 23′ 33.13″	73
12	44° 47′ 19.20″	20° 23′ 48.77″	72
28	44° 46′ 57.54″	20° 23′ 26.45″	70

reference velocity tracking of than fuzzy regulator. Regarding criterion on keeping the reference flight altitude better results are achieved by implementing BSM and PID techniques, too.

Some characteristic simulation results concerning precision of trajectory tracking, keeping reference flight speed and actual motor (rotor) speed as control variable are presented in Figs. 10 and 11. One of the drawbacks of implementation the backstepping controller regards to anisotropic operation of quadrocopter motors Fig. 12 that have rather pronounced oscillation of rotor speeds.

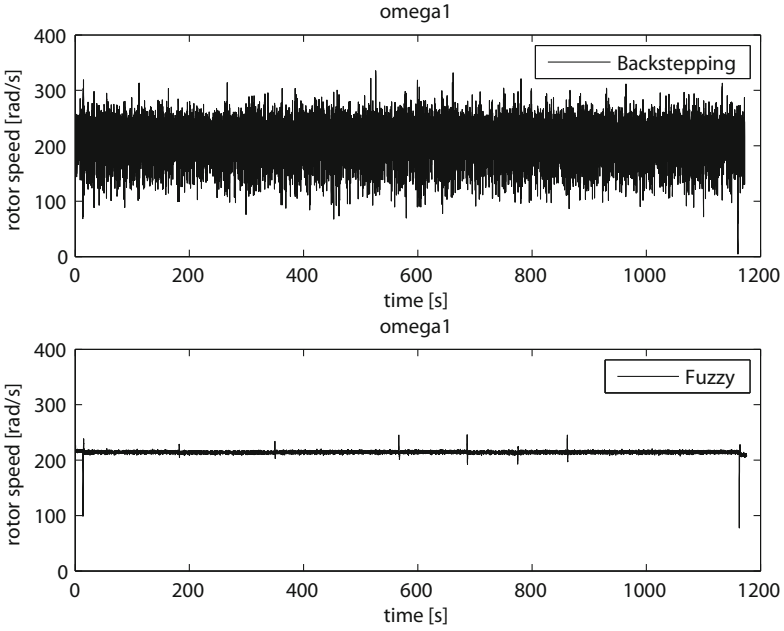


Fig. 12 Rotor speed as control variable obtained at the 1st motor of quadrocopter

5 Conclusion

The paper regards to development of appropriate benchmarking and qualitative evaluation procedures dedicated to exploration, analysis and validation of flight controller performances of quadrocopter UMAVs. The system benchmarking represents inexpensive and no risky procedure for valuable assessment and evaluation of the control quality of the UMAV systems. Because of that it is of great interest to set up appropriate benchmark simulation procedures to be accomplished before implementation of the chosen control algorithm with real microcopters. Two objective benchmark simulation tests are proposed in the paper. One indoor test, capable for exploration of dynamic flight scenarios and another outdoor waypoint navigation and trajectory tracking test are simulated. Three controllers (PID regulator, Backstepping and Fuzzy controller) as typical representatives of linear/non-linear and model-based/knowledge-based control techniques are validated through several closed-loop simulation tests. Based on a qualitative analysis of the obtained simulation results the Backstepping controller was identified as the best flight controller solution in this case. In spite of that, it has a drawback that regards to anisotropic operation of quadrocopter motors that leads towards potential damaging of actuators over longer exploitation and worse energy efficiency. The PID regulator has certain advantages compared to the Backstapping and FLC, regarding to energy efficiency

and precision of reference velocity tracking. Concerning robustness to the internal and external perturbations, the Backstepping controller ensures better characteristics than two concurrent techniques. Generally spoken, existence of perturbations degrades the control system performances in all cases and in that sense need additional compensation. Proposed benchmark and evaluation procedure, described in the paper, can be usefully implemented in evaluation of other control methods in the same way, too. The proposed benchmarking procedure ensures equal conditions for testing and validation of different control architectures of such kind dynamic systems. It enables designer to bring valuable conclusions in the phase of controller development before performing tests with real system and in real exploitation conditions. Further research leads to combining (hybridization) of different types of control algorithms in order to minimize potential drawbacks of the particular techniques and to enhance their existing advantages.

Acknowledgements. This project is supported by Ministry of Science and Education of Republic Serbia under the grants TR-35003 and III-44008 and partially by the TÁMOP-4.2.2/08/1/2008-0008 program of the Hungarian National Development Agency.

References

1. Dzul, A., Castillo, P., Lozano, R.: Real-time stabilization and tracking of a four-rotor mini rotorcraft. IEEE Transaction on Control System Technology 12(4), 510–516 (2004)
2. Lozano, R., Castillo, P., Dzul, A.: Stabilization of a mini rotorcraft having four rotors. In: Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, pp. 2693–2698 (2004)
3. Palomino, A., Salazar-Cruz, S., Lozano, R.: Trajectory tracking for a four rotor mini-aircraft. In: Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, Sevilla, Spain, pp. 2505–2510 (2005)
4. Noth, A., Bouabdallah, A., Siegwart, R.: Pid vs lq control techniques applied to an indoor micro quadrotor. In: Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 2451–2456 (2004)
5. Tayebi, A., McGilvray, S.: Attitude stabilization of a vtol quadrotor aircraft. IEEE Transaction on Control System Technology 14(3), 562–571 (2006)
6. Fradkov, A., Andrievsky, B., Peaucelle, D.: Adaptive control experiments for laas helicopter benchmark, pp. 760–765 (2005)
7. Morel, Y., Leonessa, A.: Direct adaptive tracking control of quadrotor aerial vehicles. In: Florida Conference on Recent Advances in Robotics, pp. 1–6 (2006)
8. Lozano, R., Castillo, P., Dzul, A.: Stabilization of a mini rotorcraft with four rotors. IEEE Control Systems Magazine, 45–55 (2005)
9. Turczi, A.: Flight Control System of an Experimental Unmanned Quad-Rotor Helicopter. In: Proceedings of the 10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics (2009)
10. Madani, T., Benallegue, A.: Backstepping control for a quadrotor helicopter. In: Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3255–3260 (2006)

11. Mokhtari, A., Benallegue, A.: Dynamic feedback controller of euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle. In: Proceedings of the 2004 IEEE International Conference on Robotics and Automation, pp. 2359–2366 (2004)
12. Benallegue, A., Mister, V., M'Sirdi, N.K.: Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback. In: IEEE International Workshop on Robot and Human Interactive Communication, pp. 586–593 (2001)
13. Valenti, M., Tournier, G.P., How, J.P.: Estimation and control of a quadrotor vehicle using monocular vision and moire patterns. In: AIAA Guidance, Navigation, and Control Conference and Exhibit (2006)
14. Hamel, T., Metni, N., Derkx, F.: Visual tracking control of aerial robotic systems with adaptive depth estimation. In: Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, pp. 6078–6084 (2005)
15. Ostrowski, J.P., Altug, E., Taylor, C.J.: Quadrotor control using dual camera visual feedback. In: Proceedings of the 2003 IEEE International Conference on Robotics and Automation, pp. 4294–4299 (2003)
16. Earl, M.G., D'Andrea, R.: Real-time attitude estimation techniques applied to a four rotor helicopter. In: 43rd IEEE Conference on Decision and Control, pp. 3956–3961 (2004)
17. Coza, C., Macnab, C.J.B.: A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization (2006)
18. Tarbouchi, M., Dunfied, J., Labonte, G.: Neural network based control of a four rotor helicopter. In: 2004 IEEE Intrnational Conference on Industrial Technology, pp. 1543–1548 (2004)
19. Jang, J.S., Waslander, S.L., Hoffmann, G.M., Tomlin, C.J.: Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning. In: Proceedings of 2005 (IROS) IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3712–3717 (2005)
20. Hehn, M., Ritz, R., Andrea, R.: Performance benchmarking of quadrotor systems using time-optimal control. Autonomous Robots, 1–20 (2012), doi:10.1007/s10514-012-9282-3
21. Bresciani, T.: Modelling, Identification and Control of a Quadrotor Helicopter. MSc Dissertation, Department of Automatic Control, Lund University (2008)
22. Rodic, A., Mester, G.: The Modeling and imulation of an Autonomous Quad-Rotor Microcopter n a Virtual Outdoor Scenario. Acta Polytechnica Hungarica 8(4), 107–122 (2011)
23. Rodic, A., Mester, G.: Modeling and Simulation of Quad-rotor Dynamics and Spatial Navigation. In: Proceedings of the SISY 2011, 9th IEEE International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, pp. 23–28 (2011), doi:10.1109/SISY.2011.6034325
24. Bouabdallah, S., Siegwart, R.: Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation ICRA 2005, pp. 2247–2252 (2005)
25. Raza, S.A., Gueaieb, W.: Intelligent Flight Control of an Autonomous Quadrotor. In: Casolo, F. (ed.) InTech (2010) ISBN: 978-953-7619-55-8
26. Santos, M., Lopez, V., Morata, F.: Intelligen fuzzy controller of a quadrotor. In: Intrernational Conference on Intelligent Systems and Knowledge Engineering, ISKE 2010, pp. 141–146 (2010)
27. http://www.google.com/earth/index.html

Part III
Applications of Fuzzy Systems

Fuzzy Geometry in Linear Fuzzy Space

Djordje Obradović, Zora Konjović, Endre Pap, and Imre J. Rudas

Abstract. In this paper a new mathematical model of basic planar imprecise geometric objects (fuzzy line, fuzzy triangle and fuzzy circle) are introduced. Also, basic measurement functions (distance between fuzzy point and fuzzy line, fuzzy point and fuzzy triangle, two fuzzy lines and two fuzzy triangles) as well as spatial operation (linear combination of two fuzzy points) and main spatial relations (coincidence, between and collinear)is proposed. Results obtained with our model can be used in various applications such as image analysis (imprecise feature extraction), GIS (imprecise spatial object modeling), robotics (environment models). Imprecise point objects are modeled as a union of linear combinations of fuzzy points in linear fuzzy space. However, it is proved that fuzzy line could be represented only by two and fuzzy triangle with three fuzzy points.

Keywords: Fuzzy point, fuzzy line, fuzzy triangle, fuzzy collinear, linear fuzzy space.

Djordje Obradović · Zora Konjović
Faculty of Technical Sciences/Department of Computing and Automation,
Novi Sad, Serbia
e-mail: {obrad,ftn_zora}@uns.ac.rs

Endre Pap
Faculty of Sciences and Mathematics,
University of Novi Sad,
Trg D. Obradovica 4, Novi Sad, Serbia
Óbuda university, H-1034 Budapest,
Becsi út 96/B, Hungary,
Educons University,
21208 Sremska Kamenica, Serbia
e-mail: pap@dmi.uns.ac.rs,pape@eunet.rs

Imre J. Rudas
Óbuda university,
H-1034 Budapest, Becsi ut 96/B, Hungary
e-mail: rudas@uni-obuda.hu

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 137–153.
DOI: 10.1007/978-3-642-33959-2_8

1 Introduction

Extraction of basic geometric features, such as lines, from digital raster image is one of fundamental processes in image analysis. Common problems in image analysis arise from the fact that a discrete space (digital raster image) is used for real-world elements representation, while spatial relations apply rules of continual space. For example, line is represented as a set of discrete points that usually do not have to be collinear, which contradicts the line definition. Real-world objects are mapped to the digital raster image through a variety of sensors, making the image only an approximation to the real-world object. Due to imperfections in either the image data or the edge detector, there may be missing points or pixels on lines as well as spatial deviations between ideal line and the set of imprecise points obtained from the edge detector. The overall effect is an image that has some distortion in its geometry.

The research topic which is relevant for this paper is modeling of basic planar imprecise geometry objects and their relations, as well as their application to spatial data management systems. In the sequel, the results belonging to this topic will be briefly presented.

An overview of the papers dealing with imprecise point objects modeling is given in [1]. In general, three basic approaches to spatial data uncertainty/imprecision are recognized : exact models [2, 3, 4, 5, 6], probabilistic models [7, 8, 9, 10, 11] and fuzzy models [1], [12, 13, 14, 15, 16, 17, 18].

Löffler [4] has proposed models for imprecise lines as set of possible precise planar lines, while Guibas,Salasin and Stolfiin [2]present basic spatial relations (coincidence, between and collinear) between imprecise points modeled as circles with same diameter. Simplicity in description and efficiency in computation are advantages of such models. However, with this approach, it is not possible to describe the line whose points are known with different levels of accuracy.

In the paper [16] the problem of spatial objects which do not have homogeneous interiors and sharply defined boundaries is considered. For a spatial vagueness called *fuzziness* this paper provides a conceptual model of fuzzy spatial objects that also incorporates fuzzy geometric union, intersection, and difference operations as well as fuzzy topological predicates. In particular, this model is not based on Euclidean space and on infinite-precision arithmetic, but relies on a finite, discrete geometric domain called *grid partition* which takes into account finite-precision number systems available in computers.

In the paper [12] the abstract conceptual model of *fuzzy spatial data types,* such as fuzzy points, fuzzy lines and fuzzy regions, is proposed. The paper focuses on defining their structure and semantics. Theory of fuzzy sets and fuzzy topologies is a formal framework for the conceptual model.

Buckley and Eslami in their paper [15] present fuzzy plane geometry. The fuzzy point and fuzzy line are defined as fuzzy sets with membership functions that are convex and upper semi-continuous.

One of the earliest and the most commonly used technique in feature extraction, known as Generalized Hough transformation, is given by Duda and Hart [19] in

1972. This technique is based on voting procedure which is carried out in parameter space. Line candidates are obtained as local maxima of Hough transformation which highly depends on spatial relation co linearity.

Based on our previous results [1] we introduce a new mathematical model of fuzzy line, fuzzy triangle and fuzzy circle, as well as basic spatial relations: coincidence, between and collinear. Practical applications of the results obtained in this paper are based on simple, yet effective, modeling of imprecise data using fuzzy sets, which enables the gradual estimation of objects spatial relations.

This paper consists of six sections. Following this introductory section several definitions and preliminaries related to imprecise point object model are set out in Section 2. Mathematical models of the basic planar imprecise geometrical objects are set out in Section 3. Section 4 contains basic spatial measurement functions and Section 5 contains basic spatial relations. The final section contains concluding remarks and future research directions.

2 Preliminaries

Definition 1. *Fuzzy point* $P \in \mathbb{R}^2$, denoted by $\tilde{P}$ is defined by its membership function $\mu_{\tilde{P}} \in \mathcal{F}^2$, where the set $\mathcal{F}^2$ contains all membership functions $u : \mathbb{R}^2 \to [0,1]$ satisfying following conditions:

i) $(\forall u \in \mathcal{F}^2)(\exists_1 P \in \mathbb{R}^2)u(P) = 1$,
ii) $(\forall X_1, X_2 \in \mathbb{R}^2)(\lambda \in [0,1])u(\lambda X_1 + (1-\lambda)X_2) \geq min(u(X_1), u(X_2))$,
iii) function u is upper semi continuous,
iv) $[u]^\alpha = X | X \in \mathbb{R}^2, u(X) \geq \alpha$ α -cut of function u is convex.

The point from $\mathbb{R}^2$, with membership function $\mu_{\tilde{P}}(P) = 1$, will be denoted by P (P is the core of the fuzzy point $\tilde{P}$), and the membership function of the point $\tilde{P}$ will be denoted by $\mu_{\tilde{P}}$. By $[P]^\alpha$ we denote the α -cut of the fuzzy point (this is a set from $\mathbb{R}^2$).

Definition 2. $\mathbb{R}^2$ *Linear fuzzy space* is the set $\mathcal{H}^2 \subset \mathcal{F}^2$ of all functions which, in addition to the properties given in Definition 1, are:

i) Symmetric against the core $S \in \mathbb{R}^2$
($\mu(S) = 1$),
$\mu(V) = \mu(M) \wedge \mu(M) \neq 0 \to d(S,V) = d(S,M)$,
where $d(S,M)$ is the distance in $\mathbb{R}^2$.
ii) Inverse-linear decreasing w.r.t. points distance from the core according to:
If $r \neq 0$

$$\mu_{\tilde{S}}(V) = \max(0, 1 - \frac{d(S,V)}{|r_S|}),$$

if $r = 0$

$$\mu_{\tilde{S}}(V) = \begin{cases} 1 & \text{if } S = V \\ 0 & \text{if } S \neq V, \end{cases}$$

where $d(S,V)$ is the distance between the point V and the core $S(V,S \in \mathbb{R}^n)$ and $r \in \mathbb{R}$ is constant.

Elements of that space are represented as ordered pairs $\tilde{S} = (S, r_S)$ where $S \in \mathbb{R}^2$ is the core of $\tilde{S}$, and $r_S \in \mathbb{R}$ is the distance from the core for which the function value becomes 0; in the sequel parameter r_S will be denoted as *fuzzy support radius*.

Definition 3. Let the linear fuzzy space $\mathcal{H}$ be defined on $\mathbb{R}$. Fuzzy relations $\leq^{RF}$ and $\leq^{LF}$ for the set $\mathcal{H}$ are defined by membership functions

$$\mu(\tilde{A} \leq^{RF} \tilde{B}) = \begin{cases} 0 & \text{if } A > B, \\ \frac{B-A}{r_A - r_B} & \text{if } A \leq B \wedge A + r_A > B + r_B \\ 1 & \text{if } A \leq B \wedge A + r_A \leq B + r_B, \end{cases}$$

$$\mu(\tilde{A} \leq^{LF} \tilde{B}) = \begin{cases} 0 & \text{if } A > B \\ \frac{B-A}{r_B - r_A} & \text{if } A \leq B \wedge A - r_A > B - r_B \\ 1 & \text{if } A \leq B \wedge A - r_A \leq B - r_B, \end{cases}$$

respectively, where $\tilde{A} = (A, r_A)$ and $\tilde{B} = (B, r_B)$ are points from $\mathcal{H}$, A is the core of $\tilde{A}$ and r_A is a parameter determining the membership function of point $\tilde{A}$.

3 Basic Fuzzy Plane Geometry Objects in $\mathbb{R}^2$ Linear Fuzzy Space

In this section we shall introduce basic operations over linear fuzzy space $\mathcal{H}^2$ defined on $\mathbb{R}^2$, as well as their properties which will be used in definitions of basic fuzzy plane geometry objects.

Definition 4. Let $\tilde{A}, \tilde{B} \in \mathcal{H}^2$. An operator $+ : \mathcal{H}^2 \times \mathcal{H}^2 \to \mathcal{H}^2$ is called *fuzzy points addition* given by

$$\tilde{A} + \tilde{B} = (A + B, r_A + r_B),$$

where $A + B$ is a vector addition, and $r_A + r_B$ is a scalar addition.

Definition 5. Let $\mathcal{H}^2$ be a linear fuzzy space. Then a function $f : \mathcal{H}^2 \times \mathcal{H}^2 \times (0,1) \to \mathcal{H}^2$ is called *linear combination* of the fuzzy points $\tilde{A}, \tilde{B} \in \mathcal{H}^2$ given by

$$f(\tilde{A}, \tilde{B}, u) = \tilde{A} + u \cdot (\tilde{B} - \tilde{A}),$$

where $u \in [0,1]$ and operator is a scalar multiplication of fuzzy point.

Figure 1 shows alpha-cuts of the fuzzy points $\tilde{A}, \tilde{B} \in \mathcal{H}^2$ and alpha-cut of the fuzzy point $f(\tilde{A}, \tilde{B}, u)$. Cores of the all three fuzzy points are collinear. All straight lines that connects points $X \in (\tilde{A})^\alpha$ and $Y \in (\tilde{B})^\alpha$ always contains at least one point from $(f(\tilde{A}, \tilde{B}, u))^\alpha$.

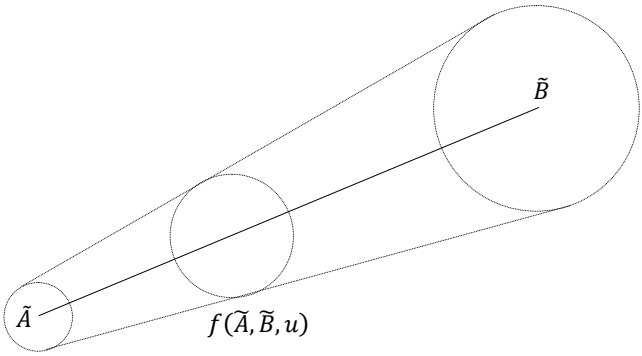


Fig. 1 Geometric illustration of linear combination of the fuzzy points $\tilde{A},\tilde{B} \in \mathcal{H}^2$

Remark. The linear combination of the two fuzzy points could be also expressed as

$$f(\tilde{A},\tilde{B},u) = (1-u)\tilde{A} + u \cdot \tilde{B}.$$

Definition 6. Let $\tilde{A},\tilde{B} \in \mathcal{H}^2$ and $\tilde{A} \neq \tilde{B}$. Then a point $T_{AB} \in \mathbb{R}^2$ is called *internal homothetic center* if the following holds

$$T_{AB} = A + \frac{a_r}{a_r + b_r}(B - A),$$

where $\tilde{A} = (A, a_r)$ and $\tilde{B} = (B, b_r)$.

In the previous definitions we introduced the basic element of a linear fuzzy space and provide an overview of its basic features. Fuzzy points are used to describe the position of a real object when there is some uncertainty to the measured position. Most often this uncertainty in practical applications is ignored. There are applications in which real objects are not only represented by the position but the entire series of uniformly spaced points. These points can be distributed along a curve that has a beginning and an end. Curve that connects two points is called a line or path.

If the points that represent the path are imprecise, then the whole line should be described in way similar to imprecise point's description. In this section we will present mathematical model for such fuzzy line.

Definition 7. Let $\mathcal{H}^2$ be a linear fuzzy space and function f is a linear combination of the fuzzy points $\tilde{A}$ and $\tilde{B}$. Then a fuzzy set $\widetilde{AB}$ is called *fuzzy line* if following holds

$$\widetilde{AB} = \bigcup_{u \in [0,1]} f(\tilde{A},\tilde{B},u).$$

Theorem 1. *Let $\mathcal{H}^2$ be linear fuzzy space, fuzzy line $\widetilde{AB}$ defined by fuzzy points $\tilde{A}$ and $\tilde{B} \in \mathcal{H}^2$. Then following holds*

$$\widetilde{AB} = \widetilde{BA}.$$

Proof. We shall prove that for all $X \in \mathbb{R}^2$

$$\mu_{\widetilde{AB}}(X) = \mu_{\widetilde{BA}}(X)$$

is satisfied. Therefore, we have

$$\mu_{\widetilde{AB}}(X) = \max_{u=[0,1]}(max(0, 1 - \frac{d(A+u(B-A),X)}{|a_r + u(b_r - a_r)|})),$$

which can be expressed as

$$\mu_{\widetilde{AB}}(X) = \max_{u=[0,1]}(max(0, 1 - \frac{d(B+(1-u)(A-B),X)}{|b_r + (1-u)(a_r - b_r)|})). \tag{1}$$

If we take $v = 1 - u$ in equation (1) following is obtained $\mu_{\widetilde{AB}}(X) = \max_{v=[0,1]}(max(0, 1 - \frac{d(B+v(A-B),X)}{|b_r+v(a_r-b_r)|})$, implying

$$\mu_{\widetilde{AB}}(X) = \mu_{\widetilde{BA}}(X).$$

□

Definition 8. Let $\widetilde{AB}$ be *fuzzy line* defined on linear fuzzy space $\mathcal{H}^2$ and $X \in \mathbb{R}^2$. Then a fuzzy point $\widetilde{X'} \subset \widetilde{AB}$ is called *fuzzy image of point X on fuzzy line $\widetilde{AB}$*, and a real number $u \in [0,1]$ is called *eigenvalue of the fuzzy image X* on fuzzy line $\widetilde{AB}$ if following hold

i) $\widetilde{X'} = \tilde{A} + u(\tilde{B} - \tilde{A})$,
ii) $d(X, (\widetilde{X'})^1) = min\{d(X,Y) | \forall Y \in (\widetilde{AB})^1\}$,
iii) $u = min(1, max(0, \frac{(x_1-a_1)(b_1-a_1)+(x_2-a_2)(b_2-a_2)}{(b_1-a_1)^2+(b_2-a_2)^2}))$,

where $X = (x_1, x_2)$, $\tilde{A} = ((a_1, a_2), a_r)$ and $\tilde{B} = ((b_1, b_2), b_r)$.

Remark. If the eigenvalue of the fuzzy image X is equal 0, then fuzzy image is the starting fuzzy point, if eigenvalue is equal 1 it is the final point, otherwise it is the inner point of a fuzzy line.

Figure 2 shows alpha cuts of the fuzzy points $\tilde{A}, \tilde{B} \in \mathcal{H}^2$ and alpha-cut of the fuzzy image of point X on fuzzy line $\widetilde{AB}$ (fuzzy point $\widetilde{X'}$). The nearest point from point X to the straight line $\overline{AB} = [\widetilde{AB}]^0$ is core of its fuzzy image $\widetilde{X'}$.

Theorem 2. *Let $\widetilde{AB} \in L^2$ be fuzzy line, $\widetilde{X'} \in \mathcal{H}^2$ fuzzy image of point $X \in \mathbb{R}^2$ on $\widetilde{AB}$ and $u \in [0,1]$ eigenvalue of the fuzzy image X on $\widetilde{AB}$. Then point X belongs to fuzzy set $\widetilde{AB}$ according to following*

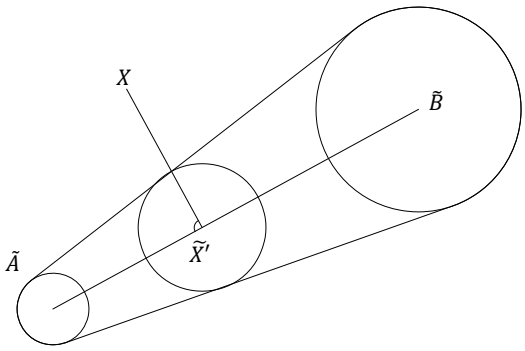


Fig. 2 Geometric illustration of the fuzzy image of point X on fuzzy line $\widetilde{AB}$

$$\mu_{\widetilde{AB}}(X) = \begin{cases} \mu_{\tilde{A}}(X) & if\, u_m = 0, \\ \mu_{\tilde{X}_T}(X) & if\, 0 < u_m < 1, , \\ \mu_{\tilde{B}}(X) & if\, u_m = 1, \end{cases}$$

where fuzzy point $\tilde{X}_T = \tilde{A} + u_m(\tilde{B} - \tilde{A})$ *and* $u_m = u + \frac{(b_r - a_r)}{x_r'} \frac{d(X,X')^2}{d(A,B)^2}$.

Proof. Since, the membership function of the union of two fuzzy sets is defined as maximum of the two individual membership functions, and the fuzzy line $\widetilde{AB}$ is union of all linear combination of fuzzy points $\tilde{A}$ and $\tilde{B}$ it follows that value of the membership function of fuzzy line $\widetilde{AB}$ in point X calculated as $\mu_{\widetilde{AB}}(X) = \max(\mu_{\tilde{Y}}(X))\tilde{Y} \subset \widetilde{AB})$.

Fuzzy point $\tilde{Y} = \tilde{A} + u(\tilde{B} - \tilde{A})$ is represented by corresponding membership function $\mu_{\tilde{Y}}(X) = \max(0, 1 - \frac{d(A+u(B-A),X)}{|a_r + u(b_r - a_r)|})$. It follows that

$$\mu_{\widetilde{AB}}(X) = \max u = [0,1](max(0, 1 - \frac{d(A + u(B-A), X)}{|a_r + u(b_r - a_r)|})) \tag{2}$$

if we take expressions

$$\mu_{\widetilde{AB}}(X) = \max_{u=[0,1]}(f(u)),$$

$$f(u) = \max(0, 1 - g(u)),$$

$$g(u) = \frac{d(A + u(B-A), X)}{|a_r + u(b_r - a_r)|},$$

into Eq. (2), and let $g(u_m) = \min(g(u))$. Since function g is defined for non negative values, it follows that $g(u_1) \leq g(u_2) \rightarrow g^2(u_1) \leq g^2(u_2)$, and $g^2(u_m) = min(g^2(u))$.

Consequence is that we are looking for u_m such that $(g^2(u_m))' = 0$. Solution to this equation is

$$u_m = u + \frac{(b_r - a_r)}{x_r'} \frac{d(X, X')^2}{d(A, B)^2}.$$

□

Definition 9. Let $\tilde{A}, \tilde{B}, \tilde{C} \in \mathcal{H}^2$ be fuzzy points with noncollinear cores $(\tilde{A} \neq \tilde{B} \neq \tilde{C})$ and function f is a linear combination of two fuzzy points. Then the fuzzy set $\widetilde{ABC}$ is called *fuzzy triangle,* if the following holds

$$\widetilde{ABC} = \bigcup_{u=0}^{1} f(\tilde{A}, \bigcup_{v=0}^{1} f(\tilde{B}, \tilde{C}, v), u).$$

It's membership function is denoted by $\mu_{\widetilde{ABC}}(X)$ and determined according to the following formula

$$\mu_{\widetilde{ABC}}(X) = \max_{u \in [0,1], v \in [0,1]} \{\mu_{\tilde{Y}}(X) | \tilde{Y} = f(\tilde{A}, f(\tilde{B}, \tilde{C}, v), u)\}.$$

α -cut of fuzzy triangle $\widetilde{ABC}$ is denoted by $(\widetilde{ABC})^\alpha$.

Figure 3 shows geometric illustration of the fuzzy triangle membership function and corresponding $\alpha - cuts$.

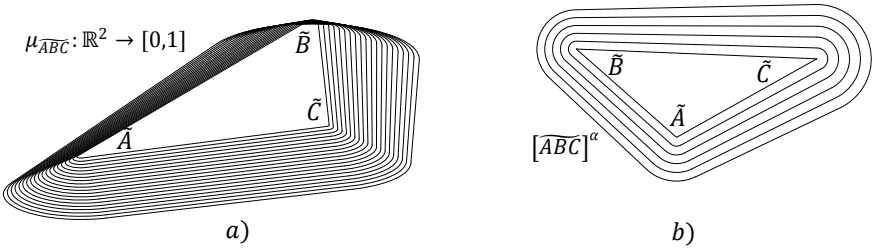


Fig. 3 a) Fuzzy triangle membership function b) Fuzzy triangle $\alpha - cuts$

Definition 10. Let $\widetilde{ABC}$ be a fuzzy triangle defined on fuzzy linear space $\mathcal{H}^2$. Fuzzy point $\tilde{X} \subset \widetilde{ABC}$ is called *edge point of the fuzzy triangle*$\widetilde{ABC}$ if for all $\alpha \in [0,1]$ a point $Y \in (\tilde{X})^\alpha$ exists such that all its neighborhoods contain at least one point from $(\widetilde{ABC})^\alpha$ and at least one point out side of $(\widetilde{ABC})^\alpha$.

Remark. α-cut of all edge points intersect α-cut of fuzzy triangle in at least one point.

Definition 11. Let $\widetilde{ABC}$ be a fuzzy triangle defined on fuzzy linear space $\mathcal{H}^2$. Fuzzy point $\tilde{X} \subset \widetilde{ABC}$ is called *inner point of fuzzy triangle* $\widetilde{ABC}$ if it is not an edge point.

Definition 12. Let $\widetilde{ABC}$ be a fuzzy triangle defined on fuzzy linear space $\mathcal{H}^2$. Union of all edge points of the fuzzy triangle $\widetilde{ABC}$ is called fuzzy edge of fuzzy triangle $\widetilde{ABC}$, denoted by $\partial\widetilde{ABC}$.

Theorem 3. *Let $\widetilde{ABC}$ be a fuzzy triangle defined on $\mathcal{H}^2$. Then, for every fuzzy point $\tilde{X} \in \partial\widetilde{ABC}/\{\tilde{A},\tilde{B},\tilde{C}\}$ and $\alpha \in (0,1)$ the single point $T \in (\tilde{X})^\alpha$ exists such that all its neighborhoods contain at least one point from $(\widetilde{ABC})^\alpha$ and at least one point outside of $(\widetilde{ABC})^\alpha$.*

Proof. This proposition is an immediate consequence of the Theorem 2.1 from [1], where it is proved that the convex hull of fuzzy points consists of lines and arcs. □

Theorem 4. *Let $\widetilde{ABC}$ be a fuzzy triangle defined on linear fuzzy space $\mathcal{H}^2$. Then for all $X \in \mathbb{R}^2$ the following holds*

$$\mu_{\widetilde{ABC}}(X) = \mu_{\widetilde{CAB}}(X) = \mu_{\widetilde{BCA}}(X).$$

Proof. Analogously to the proof of Theorem 3. □

Direct consequence of this proposition is that a fuzzy triangle can be represented by three fuzzy points, i.e., the set $\{\tilde{A},\tilde{B},\tilde{C}\}$.

Fuzzy circle is also one of the basic planar imprecise geometrical objects. Analogously to the definitions of fuzzy line and fuzzy triangle, which is an extension of a precise circle, we define a fuzzy circle as a union of fuzzy points. There by, we also take care that a newly defined geometrical object is appropriate for implementation in GIS applications.

Definition 13. Let be $\mathcal{H}$ a fuzzy space defined on $\mathbb{R}$, fuzzy relation $\leq^{RF}$ be fuzzy ordering in linear fuzzy space, $C \in \mathbb{R}^2$ and $\tilde{R} \in \mathcal{H}$. Then the union of all fuzzy points $\tilde{A} \in \mathcal{H}^2$ such that

$$\mu(\tilde{d}(C,\tilde{A}) \leq^{RF} \tilde{R}) = 1,$$

is called *fuzzy circle* with center C and radius $\tilde{R}$.

Fuzzy circle is represented by the ordered pair $(C,\tilde{R})$.

Theorem 5. *Let $(C,\tilde{R})$ be a fuzzy circle defined on linear fuzzy space $\mathcal{H}^2$. Then the value of the fuzzy circle membership function in point $X \in \mathbb{R}^2$ is determined according to the following formula*

$$\mu_{(C,\tilde{R})}(X) = \max(0, min(1, 1 - \frac{d(X,C) - R}{r_r})),$$

where $\tilde{R} = (R, r_r)$.

Proof. We shall consider three cases: (i) $d(C,X) \leq R$, (ii) $R < d(C,X) \leq R + r_r$ and (iii) $R + r_r < d(C,X)$.

Case (i) is a trivial one. Namely, due to the definition of the fuzzy relation $\leq^{RF}$ the fuzzy point $\tilde{A}$ with core X such that $\mu(\tilde{d}(C,\tilde{A}) \leq^{RF} \tilde{R}) = 1$ exists, which implies $\mu_{(C,\tilde{R})}(X) = 1$. For the cases (ii) and (iii) $d(C,X) = R + x$ where $x \geq 0$ and there exists the fuzzy point $\tilde{A} = (A, r_r)$ such that $d(C,A) = R$, $d(A,X) = x$ and point A is on the line which connects C and X. According to the previous assumptions, $d(X,C) = d(C,A) + d(A,X) = R + x$ and $x = d(X,C) - R$. Therefore, $\mu_{\tilde{A}}(X)$ could be expressed as $\mu_{\tilde{A}}(X) = \max(0, 1 - \frac{d(A,X)}{r_r}) = \max(0, 1 - \frac{d(X,C)-R}{r_r})$ □

4 Spatial Measurement in $\mathbb{R}^2$ Linear Fuzzy Space

Measurement of the space especially the distance between plane geometry objects is defined as a generalization of the concept of physical distance. Distance function or metric is a function that behaves according to specific set of rules. In this section we shell present the basic distance functions between fuzzy plane geometry objects and their main properties according to the set of rules presented in our papers [1], [20].

Definition 14. Let $\mathcal{H}^2$ be a linear fuzzy space, $\tilde{d} : \mathcal{H}^2 \times \mathcal{H}^2 \to H^+$, $L, R : (0,1) \times (0,1) \to [0,1]$ be symmetric, associative and non-decreasing for both arguments, and $L(0,0) = 0$, $R(1,1) = 1$. The ordered quadruple $(\mathcal{H}^2, \tilde{d}, L, R)$ is called fuzzy metric space and the function $\tilde{d}$ is a *fuzzy metric*, if and only if the following conditions hold:

i) $\tilde{d}(\tilde{X},\tilde{Y}) = \tilde{0} \Leftrightarrow (\tilde{X})^1 = (\tilde{Y})^1$.
ii) $\tilde{d}(\tilde{X},\tilde{Y}) = \tilde{d}(\tilde{Y},\tilde{X})$ for each $\tilde{X},\tilde{Y} \in \mathcal{H}^2$.
iii) $\forall \tilde{X},\tilde{Y} \in \mathcal{H}^2$:

a. $\tilde{d}(\tilde{X},\tilde{Y})(s+t) \geq L(d(x,z)(s), d(z,y)(t))$ if $s \leq \lambda_1(x,z) \wedge t \leq \lambda_1(z,y) \wedge s + t \leq \lambda_1(x,y)$;
b. $\tilde{d}(\tilde{X},\tilde{Y})(s+t) \leq R(d(x,z)(s), d(z,y)(t))$ if $s \geq \lambda_1(x,z) \wedge t \geq \lambda_1(z,y) \wedge s + t \geq \lambda_1(x,y)$,

where the α -cut of fuzzy number $\tilde{d}(x,y)$ is given by $[\tilde{d}(\tilde{X},\tilde{Y})]^\alpha = [\lambda_\alpha(x,y), \rho_\alpha(x,y)]$ $(x,y \in \mathbb{R}^+, 0 < \alpha \leq 1)$. The fuzzy zero, $\tilde{0}$ is a *non-negative* fuzzy number with $[\tilde{0}]^1 = 0$.

Remark: Following distance functions are fuzzy metrics

i) $\tilde{d}(\tilde{X},\tilde{Y}) =_{DF} (d(X,Y), (r_X + r_Y))$,
ii) $\tilde{d}(\tilde{X},\tilde{Y}) =_{DF} (d(X,Y), max(r_X, r_Y))$,
iii) $\tilde{d}(\tilde{X},\tilde{Y}) =_{DF} (d(X,Y), |r_X - r_Y|)$.

Distance (iii) also satisfies set of rulles which define clasic metric.

In the following definitions we shall extend distance between fuzzy points to distance between different fuzzy plane geometric objects, such as distance between fuzzy point and fuzzy line, fuzzy point and fuzzy triangle and at last between two fuzzy triangles.

Definition 15. Let $\mathcal{H}^2$ be a linear fuzzy space, L^2 set of all fuzzy lines defined on $\mathcal{H}^2$, $\tilde{d}$ is fuzzy distance between fuzzy points, and μ_L is membership function of the fuzzy relation minimal (Definition 15. in the paper [1]). The function $dist : \mathcal{H}^2 \times L^2 \to H^+$ is called *distance between fuzzy point and fuzzy line* if the following holds:

$$dist(\tilde{T}, \widetilde{AB}) = \tilde{d}(\tilde{T}, \tilde{X}),$$

where $\tilde{X} \in \widetilde{AB}$ such that $\mu_L(\tilde{d}(\tilde{T}, \tilde{X})) = hgt(\{\tilde{d}(\tilde{T}, \tilde{Y}) \forall \tilde{Y} \in \widetilde{AB}\})$.

Definition 16. Let $\mathcal{H}^2$ be linear fuzzy space, $\mathcal{T}^2$ be a set of all fuzzy triangles defined on $\mathcal{H}^2$, $\tilde{d}$ is fuzzy distance between fuzzy points and μ_L is membership function of the fuzzy relation minimal. The function $dist : \mathcal{H}^2 \times \mathcal{T}^2 \to \mathcal{H}^+$ is called distance between fuzzy point and fuzzy triangle if the following holds:

$$dist(\tilde{T}, \widetilde{ABC}) = \tilde{d}(\tilde{T}, \tilde{X}),$$

where $\tilde{X} \in \widetilde{ABC}$ such that $\mu_L(\tilde{d}(\tilde{T}, \tilde{X})) = hgt(\{\tilde{d}(\tilde{T}, \tilde{Y}) \forall \tilde{Y} \in \widetilde{ABC}\})$.

Definition 17. Let $\mathcal{H}^2$ be linear fuzzy space, L^2 set of all fuzzy lines on $\mathcal{H}^2$ $\tilde{d}$ is fuzzy distance between fuzzy points and μ_L is membership function of the fuzzy relation minimal. The function $dist : L^2 \times L^2 \to H^+$ is called distance between two fuzzy lines if the following holds:

$$dist(\widetilde{AB}, \widetilde{CD}) = \tilde{d}(\tilde{X}, \tilde{Y}),$$

where $\tilde{X} \in \widetilde{AB}$ and $\tilde{Y} \in \widetilde{CD}$ such that

$$\mu_L(\tilde{d}(\tilde{X}, \tilde{Y})) = hgt(\{\tilde{d}(\tilde{Q}, \tilde{W}) \forall \tilde{Q} \in \widetilde{AB} \wedge \forall \tilde{W} \in \widetilde{CD}\}).$$

Definition 18. Let $\mathcal{H}^2$ be a linear fuzzy space, L^2 be a set of all fuzzy lines on $\mathcal{H}^2$, T^2 be a set of all fuzzy triangles, $\tilde{d}$ is fuzzy distance between fuzzy points and μ_L is membership function of the fuzzy relation minimal. The function $dist : L^2 \times T^2 \to H^+$ is called distance between fuzzy line and fuzzy triangle if the following holds:

$$dist(\widetilde{AB}, \widetilde{CDE}) = \tilde{d}(\tilde{X}, \tilde{Y}),$$

where $\tilde{X} \in \widetilde{AB}$ and $\tilde{Y} \in \widetilde{CDE}$ satisfies condition

$$\mu_L(\tilde{d}(\tilde{X}, \tilde{Y})) = hgt(\{\tilde{d}(\tilde{Q}, \tilde{W}) \forall \tilde{Q} \in \widetilde{AB} \wedge \forall \tilde{W} \in \widetilde{CDE}\})$$

Definition 19. Let $\mathcal{H}^2$ be linear fuzzy space, T^2 be a set of all fuzzy triangles on $\mathcal{H}^2$, $\tilde{d}$ is fuzzy distance between fuzzy points and μ_L is membership function of the fuzzy relation minimal. The function $dist : T^2 \times T^2 \to H^+$ is called distance between two fuzzy triangles if the following holds:

$$dist(\widetilde{ABC},\widetilde{DEF}) = \tilde{d}(\tilde{X},\tilde{Y}),$$

where fuzzy points $\tilde{X} \in \widetilde{ABC}$ and $\tilde{Y} \in \widetilde{DEF}$ such that

$$\mu_L(\tilde{d}(\tilde{X},\tilde{Y})) = hgt(\{\tilde{d}(\tilde{Q},\tilde{W}) \forall \tilde{Q} \in \widetilde{ABC} \wedge \forall \tilde{W} \in \widetilde{DEF}\}).$$

5 Spatial Relations in $\mathbb{R}^2$ Linear Fuzzy Space

Spatial relations (predicates) are functions that are used to establish mutual relations between the fuzzy geometric objects. The basic spatial relations are *coincide*, *between* and *collinear*. In this section we will give their definitions and basic properties.

Fuzzy relation *coincidence* expresses the degree of truth that two fuzzy points are on the same place.

Definition 20. Let λ be the Lebesgue measure on the set $[0,1]$ and $\mathcal{H}^2$ is a linear fuzzy space. The fuzzy relation $coin : \mathcal{H}^2 \times \mathcal{H}^2 \to [0,1]$ is *fuzzy coincidence* represented by the following membership function

$$\mu_{coin}(\tilde{A},\tilde{B}) = \lambda(\{\alpha | [\tilde{A}]^\alpha \cap [\tilde{B}]^\alpha \neq \emptyset\}).$$

Remark. Since the lowest α is always 0, then a membership function of the *fuzzy coincidence* is given by

$$\mu_{coin}(\tilde{A},\tilde{B}) = \max\{\alpha | [\tilde{A}]^\alpha \cap [\tilde{B}]^\alpha \neq \emptyset\}.$$

Proposition

"*Fuzzy point $\tilde{A}$ is coincident to fuzzy point $\tilde{B}$*"

is partially true with the truth degree $\mu_{coin}(\tilde{A},\tilde{B})$; in the Theorem 6 we present method for its calculation.

Theorem 6. *Let the fuzzy relation coin be a fuzzy coincidence. Then the membership function of the fuzzy relation fuzzy coincidence is determined according to the following formula*

$$\mu_{coin}(\tilde{A},\tilde{B}) = \begin{cases} 0 & if\ |a_r| + |b_r| = 0 \wedge d(A,B) \neq 0, \\ \max(0, 1 - \frac{d(A,B)}{|a_r|+|b_r|}) & if\ |a_r| + |b_r| \neq 0, \\ 1 & if\ |a_r| + |b_r| = 0 \wedge d(A,B) = 0. \end{cases}$$

Proof. It is trivial to prove that $(\tilde{A})^{\alpha_2} \subset (\tilde{A})^{\alpha_1}$ iff $\alpha_2 > \alpha_1$. Also it is obvious that for every set Q_1, Q_2, T_1 and T_2 proposition $Q_1 \subset Q_2 \wedge T_1 \subset T_2 \to (Q_1 \cap T_1) \subset (Q_2 \cap T_2)$ is true. Since, we are looking for the largest α for which $(\tilde{A})^\alpha \cap (\tilde{B})^\alpha \neq \emptyset$ holds, it follows that intersection should be a set with only one element $T \in (\tilde{A})^\alpha \wedge T \in (\tilde{B})^\alpha$.

Since $d(T,A) \leq (1-\alpha_m)|r_A|$, $d(T,B) \leq (1-\alpha_m)|r_B|$ and T is unique, it follows that $d(T,A)+d(T,B) = d(A,B)$. After some arranging we finally obtain three equations:

$$d(T,A) = (1-\alpha_m)|r_A|,$$

$$d(T,B) = (1-\alpha_m)|r_B|,$$

$$d(T,A)+d(T,B) = d(A,B).$$

Solution of this system is

$$\alpha_m = \max\left(0, 1-\frac{d(A,B)}{|a_r|+|b_r|}\right),$$

because α_m could not be less than 0. In the case that $|a_r|+|b_r| = 0$ it follows that $\tilde{A}$ and $\tilde{B}$ are precise points which are coincident only if distance is 0. Finally we obtain:

$$\mu_{coin}(\tilde{A},\tilde{B}) = \begin{cases} 0 & \text{if } |a_r|+|b_r| = 0 \wedge d(A,B) \neq 0, \\ max(0, 1-\frac{d(A,B)}{|a_r|+|b_r|}) & \text{if } |a_r|+|b_r| \neq 0, \\ 1 & \text{if } |a_r|+|b_r| = 0 \wedge d(A,B) = 0 \end{cases} . \qquad \square$$

Fuzzy relation *contains* or *between* is a measure that fuzzy point belongs to fuzzy line or fuzzy line contains fuzzy point.

Definition 21. Let λ be Lebesgue measure on the set $[0,1]$, $\mathcal{H}^2$ linear fuzzy space and L^2 be set of all fuzzy lines defined on $\mathcal{H}^2$. Then fuzzy relation $contain : \mathcal{H}^2 \times L^2 \to [0,1]$ is *fuzzy contain* represented by following membership function

$$\mu_{contain}(\tilde{A},\widetilde{BC}) = \lambda(\{\alpha | [\tilde{A}]^\alpha \cap [\widetilde{BC}]^\alpha \neq \emptyset\}).$$

Remark. Its membership function could be also represented as

$$\mu_{contain}(\tilde{A},\widetilde{BC}) = \lambda(\{\alpha | \exists u \in [0,1] \wedge \exists X \in [\tilde{A}]^\alpha \wedge \exists Y,Z \in [\widetilde{BC}]^\alpha \wedge X = Y + u(Z-Y)\}).$$

Proposition

$$\text{“}\textit{Fuzzy line } \widetilde{BC} \textit{ contain fuzzy point } \tilde{A}\text{”}$$

is partially true with the truth degree $\mu_{contain}(\tilde{A},\widetilde{BC})$; in the Theorem 7 we present method for its efficient calculation.

Theorem 7. *Let $\tilde{A},\tilde{B},\tilde{C} \in \mathcal{H}^2$ be fuzzy points defined on $\mathcal{H}^2$ linear fuzzy space, $u \in [0,1]$ and $\widetilde{A'}$ be fuzzy image of point A on fuzzy line $\widetilde{BC}$. Points T_{AB} and T_{AC} are internal homothetic center fuzzy points for fuzzy points $\tilde{A}$ and $\tilde{B}$ and $\tilde{A}$ and $\tilde{C}$ respectively. Then the membership function of the fuzzy relation fuzzy contain is determined according to the following formula*

$$\mu_{contain}(\tilde{A},\widetilde{BC}) = \begin{cases} \mu_{coin}(\tilde{A},\widetilde{A'}) & if\ u \in \{0,1\} \\ \mu_{\tilde{A}}(A^*) & if\ u \in (0,1), \end{cases}$$

where the point A^* is a projection of the core of $\tilde{A}$ on the line passing through the points T_{AB} and T_{AC}.

Proof. Similarly to previous proof, we are looking for the largest α that satisfies proposition $card([\tilde{A}]^\alpha \cap [\widetilde{BC}]^\alpha) = 1$. It is obvious that such a point should be an intersection of tangents to α cuts of $[\tilde{B}]^\alpha$, $[\tilde{C}]^\alpha$ and $[\tilde{A}]^\alpha$. Direct consequence is that we are looking for α such that there exists a common tangential line for three circles ($[\tilde{A}]^\alpha$, $[\tilde{B}]^\alpha$ and $[\tilde{C}]^\alpha$). It is the largest one because if $\alpha^* > \alpha$ exists then $\emptyset \neq [\tilde{A}]^{\alpha^*} \cap [\widetilde{BC}]^{\alpha^*} \subset [\tilde{A}]^\alpha \cap [\widetilde{BC}]^\alpha$ holds, which is impossible since the set $[\tilde{A}]^\alpha \cap [\widetilde{BC}]^\alpha$ is a minimal non empty set.

According to definition of internal homothetic center it follows that all internal tangents between circles $[\tilde{A}]^\alpha$ and $[\tilde{B}]^\alpha$ pass through T_{AB}, and all internal tangents between circles $[\tilde{A}]^\alpha$ and $[\tilde{C}]^\alpha$ intersects in point T_{AC} ; it also follows that if such $\alpha > 0$ exists, then tangent line common to the circles $[\tilde{A}]^\alpha$, $[\tilde{B}]^\alpha$ and $[\tilde{C}]^\alpha$ pass through the points T_{AB} and T_{AC}. Since the radius of the circle $[\tilde{A}]^\alpha$ is $(1-\alpha)a_r$ it follows that distance between point A and its projection on common tangent line is $d(A,A^*) = (1-\alpha)|a_r|$. From this we can express α as $\alpha = 1 - \frac{d(A,A^*)}{|a_r|}$. Since α cannot be less than it follows that

$$\alpha = \max\left(0, 1 - \frac{d(A,A^*)}{|a_r|}\right)$$

which is, according to Definition 2, same as

$$\mu_{\tilde{A}}(A^*) = \alpha = \max\left(0, 1 - \frac{d(A,A^*)}{(|a_r|}\right).$$

If the eigenvalue of the core A on fuzzy line $\widetilde{BC}$ is $u = 0$ it follows that fuzzy image of $\tilde{A}$ on fuzzy line $\widetilde{BC}$ is $\tilde{B}$, and from $[\tilde{A}]^\alpha \cap [\widetilde{BC}]^\alpha = [\tilde{A}]^\alpha \cap [\tilde{B}]^\alpha$ we finally obtain $\mu_{contain}(\tilde{A},\widetilde{BC}) = \max\{\alpha | [\tilde{A}]^\alpha \cap [\tilde{B}]^\alpha \neq \emptyset\} = \mu_{coin}(\tilde{A},\tilde{B})$. The proof for the case that eigenvalue $u = 1$ is analogous to the proof for the case $u = 0$. □

Collinearity is also one of the fundamental relations between three points in plane geometry. In the following definition we will present our definition of *fuzzy collinearity* in fuzzy linear space, as well as a method for its practical computation.

Definition 22. Let $\tilde{A},\tilde{B},\tilde{C} \in \mathcal{H}^2$ be a fuzzy points defined on $\mathcal{H}^2$ linear fuzzy space and λ be Lebesgue measure on the set $[0,1]$. The fuzzy relation $coli : \mathcal{H}^2 \times \mathcal{H}^2 \times \mathcal{H}^2 \to [0,1]$ is *fuzzy collinearity* between three fuzzy points and it is represented by following membership function

$$\mu_{coli}(\tilde{A},\tilde{B},\tilde{C}) = \lambda\left(\{\alpha \mid \exists u \in R \wedge \exists X \in (\tilde{A})^\alpha \wedge \exists Y \in (\tilde{B})^\alpha \wedge \exists Z \in [\tilde{C}]^\alpha \wedge A = B + u(C-B)\}\right).$$

Proposition

$$\textit{"Fuzzy points } \tilde{A},\tilde{B} \textit{ and } \tilde{C} \textit{ are collinear"}$$

is partially true with the truth degree $\mu_{coli}(\tilde{A},\tilde{B},\tilde{C})$; in the Theorem 8 we present method for its calculation.

Theorem 8. *Let $\tilde{A},\tilde{B},\tilde{C} \in \mathcal{H}^2$, fuzzy relation contain be fuzzy contain. Then a membership function of the fuzzy relation fuzzy colinearity is determined according to the following formula*

$$\mu_{coli}(\tilde{A},\tilde{B},\tilde{C}) = \max(\mu_{contain}(\tilde{A},\widetilde{BC}),\mu_{contain}(\tilde{B},\widetilde{AC}),\mu_{contain}(\tilde{C},\widetilde{AB})).$$

Proof. Analogously to the Theorem 7 , we ought to prove that the largest α corresponds to case that all three circles $([\tilde{A}]^\alpha$, $[\tilde{B}]^\alpha$ and $[\tilde{C}]^\alpha)$ have a common tangent line. In general there are three possible solutions: (i) Fuzzy line $\widetilde{BC}$ contains fuzzy point $\tilde{A}$. (ii) Fuzzy line $\widetilde{AC}$ contains fuzzy point $\tilde{B}$ and (iii) Fuzzy line $\widetilde{AB}$ contains fuzzy point $\tilde{C}$. □

6 Conclusion

In this paper we have proposed a new mathematical model of the imprecise basic plane geometric object (fuzzy line, fuzzy triangle, fuzzy circle), proven, theirs main properties, introduced basic distance functions and introduced the basic imprecise spatial relations (coincidence, contain and collinear).

Our model of the imprecise line object is based on the model of fuzzy imprecise point presented in [1] as the union of linear combination of two fuzzy points. Consequence is that fuzzy line can be represented only by two fuzzy points, which is simple, yet descriptive extension of precise ideal line. Imprecise spatial relations proposed in this paper are based on fuzzy relations between fuzzy points and fuzzy lines.

The proposed models of imprecise line objects could be used in various applications, such as image analysis (imprecise feature extraction), GIS (imprecise spatial object modeling) and robotics (environment models). In the our paper [21] we use fuzzy line as model of the road lane. The algorithm for lane detection is primarily based on fuzzy spatial relations introduced by this work, and it is characterized by reduced computational complexity versus the standard Hough transformation.

In order to reach our final goal, which is development of an effective applicable framework for dealing with imprecise spatial data, it is necessary to develop new specialized fuzzy indexing structures analogous to R tree, Quad tree and GRID. In fact, this is one of the main research directions related to the development of fuzzy linear space-based algorithms.

Another future research direction would be in the extension of the introduced two dimensional concepts to the three dimensional linear fuzzy space.

Acknowledgements. Results presented in this paper are part of the research conducted within the Project Technology Enhanced Learning Infrastructure in Serbia, Grant No II 47003 of the Ministry of Education and Science of the Republic of Serbia and Provincial Secretariat for Science and Technological Development of Vojvodina. The third author was supported by the project MPNRS 174009.

References

1. Obradović, D., Konjovic, Z., Pap, E., Ralević, N.M.: The maximal distance between imprecise point objects. Fuzzy Sets and Systems 170(1), 76–94 (2011)
2. Guibas, L., Salesin, D., Stolfi, J.: Epsilon geometry: building robust algorithms from imprecise computations. In: Proceedings of the Fifth Annual Symposium on Computational Geometry, New York, NY, USA, pp. 208–217 (1989)
3. Löffler, M., van Kreveld, M.: Geometry with Imprecise Lines. In: 24th European Workshop on Computational Geometry (2008)
4. Löffler, M.: Data Imprecision in Computational Geometry. PhD Thesis. Utrecht University, Utrecht (2009)
5. Löffler, M., van Kreveld, M.: Largest and Smallest Convex Hulls for Imprecise Points. Algorithmica 56(2), 235–269 (2008)
6. Yang, K., SamGe, S., He, H.: Robust line detection using two-orthogonal direction image scanning. Computer Vision and Image Understanding 115(8), 1207–1222 (2011)
7. Kiryati, N., Eldar, Y., Bruckstein, A.M.: A probabilistic Hough transform. Pattern Recognition 24(4), 303–316 (1991)
8. Stephens, R.S.: Probabilistic approach to the Hough transform. Image and Vision Computing 9(1), 66–71 (1991)
9. Galambos, C., Matas, J., Kittler, J.: Progressive probabilistic Hough transform for line detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1 (1999)
10. Matas, J., Galambos, C., Kittler, J.: Robust Detection of Lines Using the Progressive Probabilistic Hough Transform. Computer Vision and Image Understanding 78(1), 119–137 (2000)
11. Galambos, C., Kittler, J., Matas, J.: Using gradient information to enhance the progressive probabilistic Hough transform. In: Proceedings 15th International Conference on Pattern Recognition, ICPR 2000, Barcelona, Spain, pp. 560–563 (2000)
12. Schneider, M.: Fuzzy topological predicates, their properties, and their integration into query languages. In: Proceedings of the 9th ACM International Symposium on Advances in Geographic Information Systems, pp. 9–14 (2001)
13. Rosenfeld, A.: The fuzzy geometry of image subsets. Pattern Recognition Letters 2(5), 311–317 (1984)
14. Schneider, M.: Design and implementation of finite resolution crisp and fuzzy spatial objects. Data Knowl. Eng. 44, 81–108 (2003)
15. Buckley, J.J., Eslami, E.: Fuzzy plane geometry I: Points and lines. Fuzzy Sets and Systems 86(2), 179–187 (1997)
16. Schneider, M.: Spatial Data Types for Database Systems: Finite Resolution Geometry for Geographic Information Systems. Springer (1997)
17. Obradović, D., Konjović, Z., Pap, E.: Extending PostGIS by imprecise point objects. In: IEEE 8th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, pp. 23–28 (2010)

18. Verstraete, J., Tré, G., Caluwe, R., Hallez, A.: Field based methods for the modeling of fuzzy spatial data. Fuzzy Modeling with Spatial Information for Geographic Problems, 41–69 (2005)
19. Dudaand, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Commun. ACM 15(1), 11–15 (1972)
20. Hadžić, O., Pap, E.: Fixed point theory in probabilistic metric spaces. KluwerAcademic Publishers, Dordecht (2001)
21. Obradović, D., Konjović, Z., Pap, E., Rudas, I.J.: Linear Fuzzy Space Based Road Lane Model and Detection. Knowledge-Based Systems (to appear)

An Object Oriented Realization of Perceptual Computer

Dragan Šaletić and Mihajlo Anđelković

Abstract. In the chapter basic concepts of type-2 fuzzy logic, computing with words and perceptual computing are presented. Architectural details of an object-oriented realization of a software library for developing perceptual computers are exposed and explained. Thereby, following topics are covered: mathematical models for data types, class hierarchies for types and inference operators, operation in multithreaded environment, comparison with an existing MATLAB realization, and a short code sample. Introducing a class hierarchy for inference operators is motivated by prior work of the authors that implied novel weighted averages can be replaced with other operators. Development of a perceptual computer and its usage for hierarchical decision making in solving known problem of missile system selection is described. Input values and produced results are presented in details in the chapter. The results are consistent with results from the literature. Conclusions are given, as well as possible directions for further research and application work.

Keywords: Fuzzy logic, type-2 fuzzy logic, perceptual computer, hierarchical decision making, MATLAB, missile system selection.

1 Introduction

When modeling phenomena from the real world with objective to make basis for software implementations of intelligent systems (agents) [1], incomplete or uncertain information may be encountered. As such information hinders a more structured approach to modeling, it is natural to deal with the uncertainty through fuzzy sets theory, [2]. This theory provides a methodology for dealing with phenomena too complex or ill-defined to be analyzed by conventional means. In general, uncertainty

Dragan Šaletić · Mihajlo Anđelković
School of Computing, University "Union",
Belgrade, Knez Mihailova 6/V, Serbia
e-mail: dsaletic@raf.edu.rs,
michael.angelkovich@gmail.com

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 155–173.
DOI: 10.1007/978-3-642-33959-2_9

may arise from imprecise, inaccurate, or incomplete information, from verbal ambiguity, and from disagreement between different sources of information. In many real world phenomena, both numerical and linguistic imprecision may appear in the available data. When it can not be clearly determined whether the membership of an element in a set is 0 or 1, this necessitates the application of fuzzy models, founded on the theory of fuzzy sets [3], [4]. A fuzzy set has the characteristic that a membership of an element in a set is not only 0 or 1, as is the case with classical (crisp, Cantor) sets. Membership degree (grade) to a fuzzy set is a crisp number from $[0, 1]$.

Fuzzy sets theory has been around for nearly 50 years, and it generated widely accepted applications, among which fuzzy-logic controllers are the most noticeable one. Very soon after introducing fuzzy sets (FS), L. Zadeh, the father of fuzzy logic, in the paper [5] in which he introduced many important notions of fuzzy sets, introduced also a generalization of his fuzzy sets now called type-2 fuzzy sets (T2 FSs): fuzzy sets with fuzzy membership functions. Now, fuzzy sets with membership degrees that are crisp numbers from $[0, 1]$ are called type-1 fuzzy sets (T1 FSs).

However, T2 FS, that is, Zadeh's generalization of T1 FS have not been actively studied until Zadeh's paper from 1996. with the title "Fuzzy Logic = Computing With Words", [6]. Research activities dealing with computing with words based on T2 FSs have been intensified, [7], [8], after that Zadeh's paper. In last ten years, mainly Mendel and his collaborators have developed one class of applications for computing with words, [9], [10], for aiding people in making subjective judgments. He called it perceptual computing.

After this introductory section, in Section 2 a brief overview of employed theory terms concerning fuzzy logic, computing with words and perceptual computing is given. In Section 3 our work, reported in [11], is extended, covering all important architectural details of our perceptual computer (Per-C) realization. Section 4 concerns reproduction of results from [10] for the problem of missile system selection, using our Per-C. Section 5 deals with conclusions and directions for further research work in the field.

2 An Overview

Perceptual computing is a methodology of building an interactive device that could aid people in making subjective judgments, a device that would propagate random and linguistic uncertainty into subjective judgments, but in a way that could be modeled and observed by judgment maker, [10]. A Per-C, Fig. 1 is such a device, designed for each specific problem using the methodology of perceptual computing. This chapter describes building of a Per-C with object oriented techniques and its application in solving a problem from literature, the problem of missile system selection. In the considered problem, each missile system is evaluated by several experts, each of which has his subjective judgment regarding the given score. Decision-makers set their priorities in order to choose the system that optimally fits their requirements.

The block from Fig. 1 denoted by "encoder" consists of a model of sensor information. Elements of that block are described in the sequel. In modeling the phenomena from the problem considered, T2 FSs have to be used, and their special case, interval type-2 fuzzy sets (IT2 FSs) are the most popular at the time. IT2 FSs can model first-order linguistic uncertainty, which becomes necessary when there are several decision-makers willing to use data given by numbers, number intervals and words, whereas the T1 FSs can not. T1 FS can model words, but only by expressing point of view for one person, as *words mean different things to different people*.

So, when real world phenomena are so fuzzy that it is difficult to determine the membership degree even as a crisp number in $[0,1]$, fuzzy sets of type-2 are used. Symbol A is used for a T1 FS, whereas symbol $\widetilde{A}$ is used for a T2 FS. Type-1 FS A defined on a domain (universe of discourse) X, is given by its membership function (MF) shown in Fig. 2(a), and denoted by $\mu_A(x)$. Blurring the type-1 membership function (T1MF) depicted in Fig. 2(a) (blurring corresponds to different meaning of a word to different people), an entity given by Fig. 2(b) can be obtained.

Having the blurred MF from Fig. 2(b), at a specific value of x, say x', the membership function takes on values wherever the vertical line intersects the blur. Ordinary

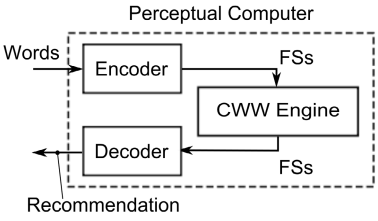


Fig. 1 The architecture of a Per-C. The entry point is encoder, that receives words. Those words are being prepared and encoded into FSs, and then sent to the CWW engine. After a result has been acquired, it is still a FS. As such, it is sent to decoder in order to be decoded into output consisting of words.

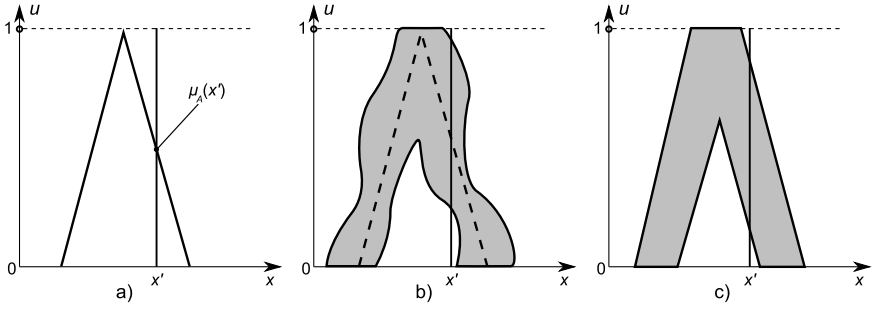


Fig. 2 a) A type-1 fuzzy set, b) A fuzzy set with blurred membership, and c) FOU of an interval type-2 fuzzy set

type-2 fuzzy sets add weights to these primary membership grades, as depicted in Fig. 3. When all of those values are weighted as 1 for all x', an interval type-2 fuzzy set is obtained. Such a FS, $\widetilde{A}$, is completely described by its so-called footprint of uncertainty (FOU), FOU($\widetilde{A}$), an example of which is depicted in Fig. 2(c). The FOU($\widetilde{A}$), in turn, is completely described by its lower and upper membership functions, LMF($\widetilde{A}$) and UMF($\widetilde{A}$). These functions can have many shapes, for example, triangles, trapezoids, Gaussians, and others.

The union and intersection of IT2 FSs, Fig. 4, and complement of those sets can be computed in terms of simple T1 FS operations [3], [4], that are performed only on LMFs or UMFs of IT2 FSs. This makes such FSs useful for practical applications.

The centroid of $\widetilde{A}$, $C_{\widetilde{A}}$, provides a measure of the uncertainty about IT2 FS. The centroid of $\widetilde{A}$ is an interval of numbers that has both a smallest and a largest value, that is, $C_{\widetilde{A}} = [c_l(\widetilde{A}), c_r(\widetilde{A})]$, and $c_r(\widetilde{A}) - c_l(\widetilde{A})$ is small for thin FOUs and is large for broad FOUs. The quantities $c_l(\widetilde{A})$ and $c_r(\widetilde{A})$ must be computed, but there are no closed-form formulas for doing that. Karnik and Mendel [12], [10], have developed

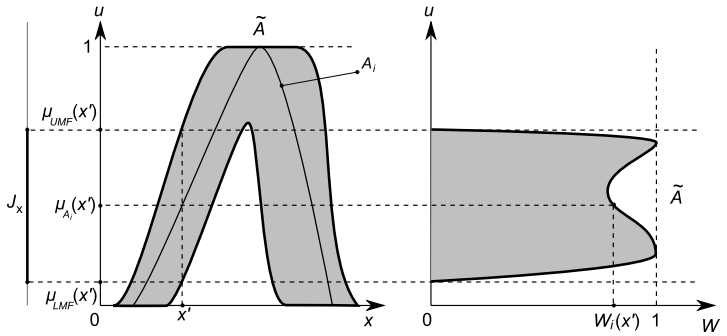


Fig. 3 A type-2 fuzzy set. For a given variable x' are depicted the membership grades to UMF and LMF, and also membership grade $\mu_{A_i}(x')$ to an embedded T1 FS A_i. On the right half of the image is depicted a continuous cut of weights, and the weight for the given variable x' and embedded T1 FS A_i has been marked as $W_i(x')$.

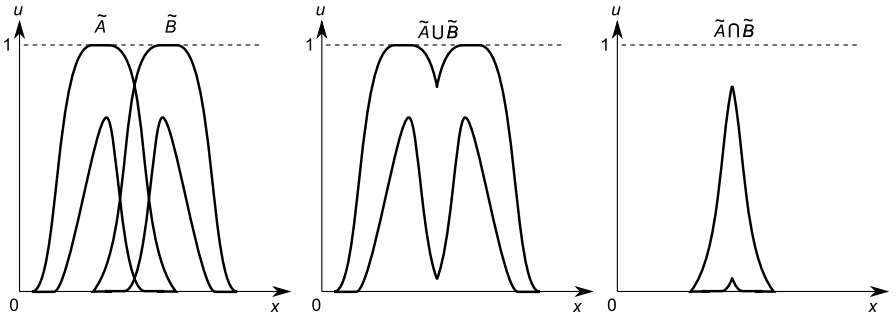


Fig. 4 An union and an intersection of two IT2 FSs

iterative algorithms, now known as Karnik-Mendel (KM) algorithms, for computing $c_l(\widetilde{A})$ and $c_r(\widetilde{A})$.

Cardinality of a crisp set is a count on the number of elements in that set. The cardinality of a T1 FS A defined on a finite domain X, is, [13], a sum of membership degrees of elements of the FS

$$\text{card}(A) = |A| = \sum_{x \in X} A(x). \tag{1}$$

For the continuous case, the definition of the cardinality of a T1 FS is, [14],

$$\text{card}(A) = |A| = \int_X A(X)\, dx. \tag{2}$$

The cardinality of an IT2 FS $\widetilde{A}$ is an interval of numbers, the smallest number being the cardinality of LMF($\widetilde{A}$) and the largest number being the cardinality of UMF($\widetilde{A}$). The average cardinality of $\widetilde{A}$ is the average of these two numbers.

An IT2 FS is used as a FS model of a word because it is characterized by its FOU and, therefore, has the potential to capture word uncertainties implied by different meanings of a word to different decision-makers.

How an IT2 FS model for a word is obtained? First, a continuous scale must be established for each variable of interest. After that, a vocabulary of words must be created so that their FS models (MFs) cover the entire scale. For perceptual computing, one begins by establishing a vocabulary of application-related words. The collection of words, $\widetilde{W}_i$, in the vocabulary and their IT2 FS models, FOU($\widetilde{W}_i$), constitutes a codebook for an application (Ap), that is, Codebook = $\{(\widetilde{W}_i, FOU(\widetilde{W}_i)),\ i = 1, \ldots, N_{\text{Ap}}\}$. After a scale is established and a vocabulary of words is created, interval end-point data are collected from a group of subjects. This is done in two steps: in the first one the words are randomized, and, in the second step a group of subjects is measured to provide end-point data for the words on the scale. A practical type-2 method for modeling a word [10] is called the interval approach (IA). It consists of two parts, a data part and a fuzzy set (FS) part. In the data part, data intervals that have been collected from a group of subjects are preprocessed, after which data statistics are computed for the surviving intervals. In the FS part, FS uncertainty measures are established for a prespecified triangle T1MF. Here we are always beginning with the assumption that the FOU is an interior FOU (as in Fig. 5), and, if need be, later switching to a shoulder FOU (as in Fig. 5).

Then the parameters of the triangle T1MF are determined using the data statistics, and the derived T1MFs are aggregated using union leading to an FOU for a word, and finally to a mathematical model for the FOU. The three FOU shapes that can be obtained for a word using the IA are the ones depicted in Fig. 5, and so these FOUs are referred to herein as canonical FOUs for a word.

The block from Fig. 1 denoted by CWW represents approximate reasoning in an IT2 FS environment; in that block a CWW engine is implemented. Elements of that block are described in the sequel. That block can be built in two forms, [3], [4]: as a fuzzy decision-making system, and as a fuzzy expert system. As the

problem this chapter is dealing with, the problem of missile system selection, is considered as a fuzzy decision-making system, only that form of CWW engine is here shortly described. The second form, a fuzzy expert system variant of CWW engine is described in literature, for example, in [10]. The decision-making form of the CWW engines, as any other form of decision-making, involves aggregation of numerical subcriteria (data, features, decisions, recommendations, judgments, scores, etc.). Aggregation obtained by using a weighted average of numbers connected with criteria is quite common and widely used. In many situations, however, providing a single number for either the subcriteria or weights is problematic (there could be uncertainties about them), and it is more meaningful to provide intervals, T1 FSs or IT2 FSs, or a mixture of all these, in such cases. A novel weighted average (NWA) is a weighted average in which at least one subcriterion or weight is not a single real number, but is instead an interval, a T1 FS, or an IT2 FS. NWAs are umbrella term for arithmetic weighted average (AWA), interval weighted average (IWA), fuzzy weighted average (FWA), and linguistic weighted average (LWA). A decision-making form of the CWW engine is built using NWA as aggregation operator.

When at least one subcriterion or weight is modeled as an interval, and all other subcriteria or weights are modeled by numbers, the resulting WA is called an IWA, denoted Y_{IWA}. On the other hand, when at least one subcriterion or weight is modeled as a T1 FS, and all other subcriteria or weights are modeled by lesser types[1], the resulting WA is called a FWA, denoted Y_{FWA}. And, finally, when at least one subcriterion or weight is modeled as an IT2 FS, the resulting WA is called a LWA. The AWA, IWA and FWA are special cases of the LWA; hence, here our focus is only on the latter. The following equation presents LWA in form that denotes it as a generalization of AWA:

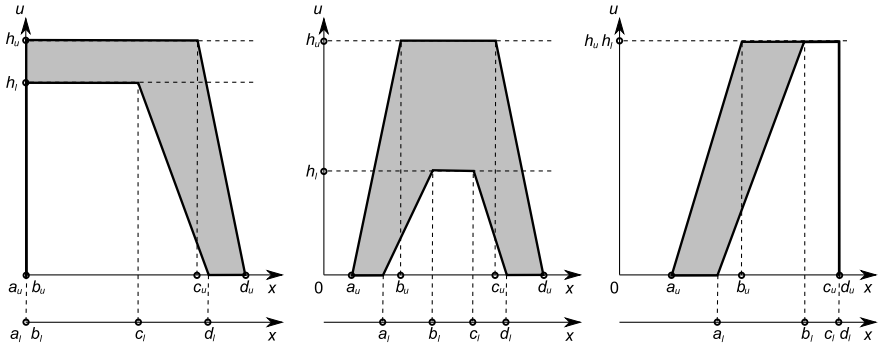


Fig. 5 Three types of FOU (from left to the right): A left shoulder FOU, an interior FOU, and a right shoulder FOU. On each, domain values of UMF and LMF are labelled on separated scales. A UMF is defined by $(a_u, b_u, c_u, d_u, h_u)$ and a LMF by $(a_l, b_l, c_l, d_l, h_l)$.

[1] Here we establish a convention that a Per-C type is called lesser than another if it is less capable of expressing uncertainty. We also use to say that a lesser type has a lower rank. For example, IT2 FS have the highest rank, and numbers are lesser type than intervals.

$$\widetilde{Y}_{LWA} = \frac{\sum_{i=1}^{n} \widetilde{X}_i \widetilde{W}_i}{\sum_{i=1}^{n} \widetilde{W}_i}, \tag{3}$$

where subcriteria $\widetilde{X}_i$ and weights $\widetilde{W}_i$ are characterized by their FOUs, and $\widetilde{Y}_{LWA}$ is also an IT2 FS. This is called an expressive way to summarize the LWA rather than a computational way to summarize the LWA, because the LWA is not computed by multiplying, adding, and dividing IT2 FSs. It is more complicated than that. The KM algorithms can be used as tools for the computations. So, given FOUs for $\widetilde{X}_i$ and $\widetilde{W}_i$, it is possible to compute FOU($\widetilde{Y}_{LWA}$).

The decoder maps a resulting FOU into an output that consists of words. This approach is supposedly correct, in accordance to the fact that "not only do words mean different things to different people," but they must also mean similar things to different people, or else people would not be able to communicate with each other. Known output types for a Per-C are:

1. *Word*. This is the most typical case. The FOU at the output of the CWW engine is mapped into a word (or a group of similar words) in the codebook so that it can be understood. Similarity measures that compare the similarity between two FOUs are needed to do this.
2. *Rank*. In some decision-making situations, several strategies/candidates are compared at the same time to find the best one(s). For example, three missile systems are compared to find the one with the best overall performance. Ranking methods should be used to do this.
3. *Class*. In some decision making applications, the output of the CWW engine has to be mapped into a class. Classifiers are needed to do this.

Obviously, if two FOUs have the same shape and are located very close to each other, they should be linguistically similar; or, if they have different shapes and are located close to each other, they should not be linguistically similar; or, if they have the same or different shapes but are not located close to each other they should also not be linguistically similar.

Different similarity measures exist, as for T1 FSs, so for IT2 FSs. Among them, the Jaccard similarity measure, which utilizes both shape and proximity information about an FOU simultaneously, provides a crisp numerical similarity measure that agrees with all three of the previous similarity cases.

Simply stated, the Jaccard similarity measure is the ratio of the average cardinality of the intersection of two IT2 FSs to the average cardinality of the union of the two IT2 FSs.

One of the ranking methods that is very simple is based on the centroid of an IT2 FS. First, the centroid is computed for each FOU, and then the center of each centroid is computed, after which the average centroids for all FOUs are sorted in increasing order to obtain the rank of the FOUs.

Classifiers can be based on the notion of subsethood, which defines the degree of containment of one set in another. Subsethood is conceptually more appropriate for a classifier than similarity because $\widetilde{A}$ and class-FOUs belong to different domains. The subsethood between two IT2 FSs, $\widetilde{A}$ and $\widetilde{B}$, $\mathrm{ss}(\widetilde{A},\widetilde{B})$, may either be an interval of numbers, $\mathrm{ss}(\widetilde{A},\widetilde{B}) = [\mathrm{ss}_l(\widetilde{A},\widetilde{B}), \mathrm{ss}_r(\widetilde{A},\widetilde{B})]$ or a single number.

3 Our Per-C Realization

In this section details of our object-oriented realization of Per-C are presented. Basically, it is explained how the data is internally structured and how the class hierarchy is chosen. What follows after that is a short code sample before a demonstration on the well known problem and data from the literature, in Section 4.

3.1 Data Model

Following the model that is proposed in [10], trapezoidal shape is used to represent T1 FSs. In further text, it is refereed to T1 FS form as (a,b,c,d,h). Fig. 6 illustrates how to project this five-tuple to actual T1 FS values. As it is suggested on the same figure, both lower types can also be expressed with this five-tuple, as follows:

1. A number a: $(a,a,a,a,h=1)$.
2. An interval $[a,b]$: $(a,a,b,b,h=1)$.

An IT2 FS can be expressed with two T1 FSs that represent corresponding UMF and LMF. Technically, this architecture allows both T1 FSs and IT2 FSs to have highest membership degree that is lesser than 1, although this feature is not required to reproduce results from [10]. This chapter will not address practical implications of such design.

In addition to the four types featured in [10], **fuzzy singleton** and **fuzzy interval** are added as separate types. They internally resemble to a number and an interval, and the difference is that they have an additional parameter for membership degree. However, during computations, these data types are treated as T1 FSs. Hence, the further text is not referring to them as to special types. It can be assumed that everything that applies for T1 FSs also applies to fuzzy singletons and fuzzy intervals, unless specified otherwise.

3.2 Considerations for Object-Oriented Design

When designing an OOP based software library, the primary concern should be what the end-users[2] see and how user-friendly the produced library is. Also, one should take advantage of OO programming paradigm and make the actual objects correspond to the real world objects in sense of their functionality and appearance.

[2] By term *user* is meant fellow programmers.

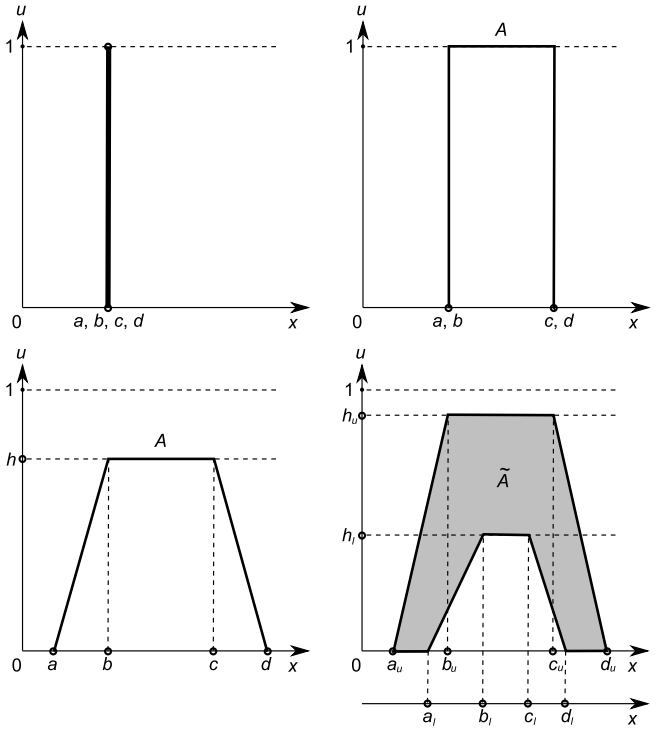


Fig. 6 All four Per-C types, with corresponding labels to map them to T1 FS model

There are several approaches to OO realization of Per-C types. What the types should have in common is a base class or interface, that inserts a crossing point between Per-C type class hierarchy and the general superclass or pointer type, such as `Object` class or `void*` type pointer. Advantage of this is being able to define a common set of methods and variables as well as common state and behavior that all other classes in a class hierarchy should have. Regarding Per-C, one may feel tempted to implement each lower type as a subtype of its closest higher type. We do not recommend such subtyping, because it would require that even a number uses as much memory as an IT2 FS. However, we do aim to achieve the same coding style as if exactly that was done.

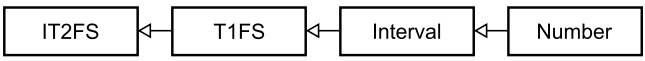


Fig. 7 An undesirable class hierarchy for Per-C types. Here `IT2FS` is the base class, and further types gradually reduce its expressional capabilities down to a number, while using the same amount of space

Regarding programming language, the focus of the realization is on Java. Java has been chosen with the purpose to target the widest auditorium possible. Silently, a C++ realization is being developed with a lower priority. The basic package for classes is `rs.edu.raf.*`, to which will be referred to as `*.` in further text.

3.2.1 Strict or Weak Typization

The major differences between considered realizations may concern common set of methods of the base type. Regarding what set of base methods is selected for the base type, it can be said that the class hierarchy is oriented toward strict or toward weak typization.

Let's say that all four basic types have their own sets of methods: $S = \{S_N, S_I, S_{T1FS}, S_{IT2FS}\}$. Should it be chosen that the base class `PerCType` contains only the methods that are found in intersection of all those subsets, direct knowledge and use of types will be enforced. That is, this base type will limit available method set even if its reference or pointer are employed to use an IT2 FS as an IT2 FS. Its use would be limited only to transportation of the values under a common type.

It is observable that KM and enhanced KM (EKM) algorithms, that are basis of present Per-C computations, require interpreting all aggregated values as if they were of the highest type met among them. That is, if there is an array of numbers and T1 FSs to be aggregated, all values should be treated as T1 FSs. To achieve this with strict typization, one needs either type conversion, delegates, or multiple adapters. Authors would make remark of inconveniences concerning these approaches:

1. The type conversion solution would require extra memory allocation and initialization for every object that is type converted. This is not desirable at all, if optimal time and space performance are sought. It should also take extra coding time to do this.
2. Delegates provide a clean way to code while working with different types, but each lower type needs one delegate type for each higher type. This approach may also include a factory pattern to decide which delegate is to be used. This would in total mean having six delegate types, which (regardless of a not large number), would not be a desirable option, for it would at lest require extra coding time.
3. Finally, using adapters would mean ability to interpret an `PerCType` as another, using an all-knowing adapter class, or set of adapters. This would be equally as much complicated as using delegates, although it seems to be even less clean coding solution.

In contrast to that, weak typization can be used. This means the base type `PerCType` will have union of all sets of methods from S, or most of them. Hence, the class hierarchy itself will be the carrier of the meta-information needed to implicitly interpret any of the lower types as a higher one. No explicit conversions would be needed. Authors consider this an advantage that surpasses other disadvantages of the approach, because it minimizes the amount of required coding while keeping up with the optimal performance.

The major disadvantage of this approach is ambiguous meaning of some methods in context of certain data types. For example, calling `getLeft()` from an interval type clearly asks for its left bound. But calling it from an IT2 FS would rise a question: left bound of what? It may refer to the left bound of UMF's or LMF's core, to the left bound of its IT2 FS's centroid, or to the left bound of the domain that either UMF or LMF cover.

Such effects can be mitigated by adding interfaces that would help users access only methods that are supposed to be available from a certain type. Runtime exceptions should be thrown on undesirable program states, and prevent users from getting results that may not be what they actually wanted. Also, provided that documentation is a common way of communication, it may be relied on the facts (1) that programmers who would want to use the library would also have interest in knowing at least type-2 fuzzy logic basics, (2) and that they would read the documentation.

The bottom line is that extra methods do not do actual harm to the overall performance. In turn, they optimize amount of written code, and there are several mechanisms to mitigate their side effects. The authors believe it to be a fair trade for ability to create a variable of type number, and use it straight as an IT2 FS without spending extra computer resources and coding time to achieve the same result.

A lesser disadvantage of all mentioned approaches would be that there is no easy way to do transformations between types. That is, these approaches will not let a number internally become an interval without creating or reusing another object. This could be overcome by allocating new parameters dynamically within an object, as it shifts the type. However, such operations are not common in the studied literature. Thus, that is neither implemented nor considered to be an actual disadvantage in the scope of this chapter.

3.2.2 The Class Hierarchy

The final set of major developed Per-C type classes is listed below. Some classes have their interfaces that can be used to limit their methods only to those that are expected within their actual type. The interfaces are named with prefix *In*.

1. `PerCType` - Superclass for all Per-C types
2. `Number` - Holds a crisp number. Its interface is `InNumber`.
3. `Interval` - Holds a crisp interval. Its interface is `InInterval`.
4. `T1FS` - Holds a T1 FS. It offers constructors for trapezoidal, triangular, and symmetrical triangular FS with ± 1 domain around the core. Its interface is `InT1FS`. It is superclass of the following classes:
   a. `LST1FS` - Uses `T1FS` base to construct a left shoulder T1 FS.
   b. `RST1FS` - Uses `T1FS` base to construct a right shoulder T1 FS.
5. `FSingleton` - Holds a fuzzy singleton. Its interface is `InFSingleton`.
6. `FInterval` - Holds a fuzzy interval. Its interface is `InFInterval`.
7. `IT2FS` - Holds an IT2 FS. It has constructors that take either ten real values for UMF and LMF, or T1 FSs that represent UMF and LMF. The class also has a static method that can take an arbitrarily long array of `T1FS`s, and construct

its FOU based on them, using the algorithm presented in [11], that implements behavior described in [10]. Its interface is `InIT2FS`. It is superclass of the following classes:

a. `RSIT2FS` - Has constructors that either take six values or two `RST1FS`s. It also has a static method that can initialize the FOU from an array of `RST1FS`s.
b. `LSIT2FS` - Has constructors that either take six values or two `LST1FS`s. It also has a static method that can initialize the FOU from an array of `LST1FS`s.

In our realization, the types are found in the package `*.perc.type`, and the interfaces are in `*.perc.type.in`.

3.2.3 Other Classes

As it can be observed in MATLAB examples from [10], Per-C computations rely on knowing bounds of a global domain. There it is uniquely set to $[0, 10]$. We decided not to pass this domain through methods calls, but to extract it as a static singleton of type `Interval`, that is globally available from class `*.perc.Standards`. This same class should be used for holding any further global settings. A question, that can be asked, is what happens if different applications want to use different domains. We made it overridable on the process level, and so it can be customized. Next question may be: what happens if multiple threads want to use different global domains. That use case has also been taken in consideration. The class `Standards` stores the custom global intervals in a map that maps thread identifiers to the interval objects, thus making the architecture scalable and stable in multithreaded environment, in addition to not requiring from users to worry about switching the contexts on their own. All they need to do is to set the desired global domain interval from a thread, and it will be available in the thread. The rest of the library is also thread-safe.

Class `*.perc.KMAlgorithms` implements statically the KM Algorithms, that expect to get two arrays or `Lists` of `Interval` type variables. As a result, they give an `Interval` type value.

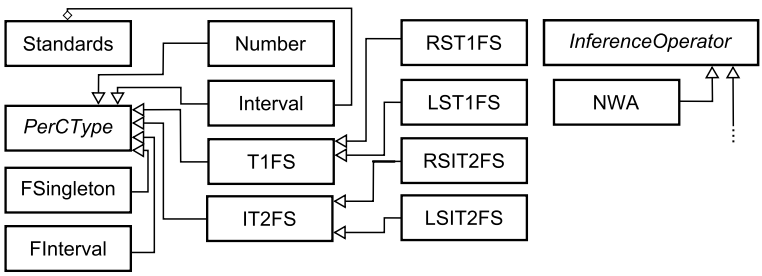


Fig. 8 The major classes of desirable class hierarchy for Per-C types

Package `*.perc.agg.*` is intended for aggregation operators. We have taken the liberty of creating a base abstract class `InferenceOperator` that is used as a base for implementation of NWAs, and may be used for realization of other operators. The only reason that an interface has not been used, is allowing user to call either `process(List<PerCType>, List<PerCType>)` or `process (PerCType[], PerCType[])` without having to implement both. In the realization, the former is the default abstract method, and the latter is implemented method that re-packs two arrays as `Lists` and passes them to the former method. It can be overridden by the users. The class that implements NWAs is found at `*.perc.agg.NWA`.

There are also various petty classes, like `PerCException` or problem solving classes like `HDM`, for hierarchical decision making. The former group of classes is common for any Java software and thus it does not call for special attention. The latter group of classes does not impact the presented architecture. The realizations from this group are stand-alone problems that are not addressed in details within this chapter.

3.2.4 Type Control

Although the class hierarchy purposely blends the types, they still play a significant role in present Per-C computations and algorithms. For example, even though the NWAs do not require to know exact type of each aggregated value, they still need to know the highest type in order to apply an optimal version of the algorithm and to produce the result of the right type. It may also be required to use a type in a switch-statement or to sort values by type rank.

Therefore, in addition to the values that need to be held in each type, an additional variable is introduced to the `PerCType` class. It is intended to keep track of the real type of a variable, and it is advisable for it to be a natural number. Advisable Java type representatives are `byte`, `char` and at most `int`. More complex subtyping may require dividing this variable with lower and higher bit mask, but that is not being done in our realization. The convention we use is that type=1 stands for a number, type=2 signifies an interval, type=3 means a T1 FS, and type=4 is taken for IT2 FS. All other subtypes take their type from an adequate representative. That is, fuzzy singletons and fuzzy intervals are of type=3.

3.3 Code Sample

Lets code the example from [10], p. 148, using the developed classes. The example is an application of KM algorithms to aggregate an array of five intervals that is weighted with five interval-weights. Result is an interval.

```
// Within the class Test
  public static void main(String[] args) {
    // Create the input data
    PerCType[] values = { // Aggregated values
      new Interval(8.2, 9.8),
```

```
        new Interval(5.8, 8.2),
        new Interval(2, 8),
        new Interval(3, 5),
        new Interval(0.5, 1.5)
      };

      PerCType[] weights = { // Corresponding weights
        new Interval(1, 3),
        new Interval(0.6, 1.4),
        new Interval(1.7, 8.9),
        new Interval(2.4, 5.6),
        new Interval(5, 7)
      };

      // Create operator for novel weighted averages
      InferenceOperator operator = new NWA();

      // Aggregate and print the result
      PerCType result = operator.process(values, weights);
      System.out.println(result);
    }
```

Produced output describes interval $[2.0190954773869345, 6.356521739130436]$. This result agrees with result from the literature.

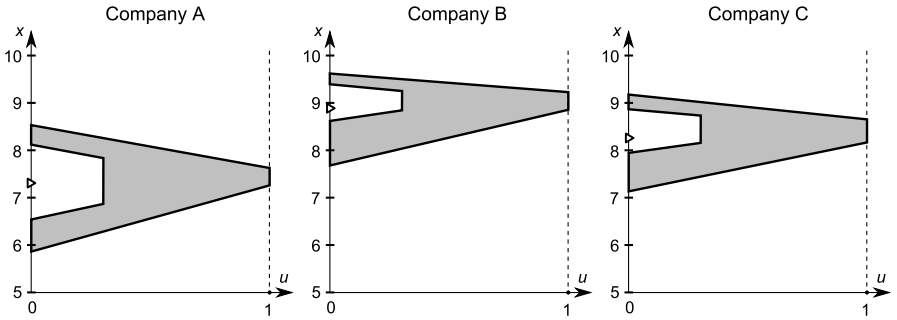


Fig. 9 Results of data aggregation for companies A, B and C

4 Hierarchical Decision Making: Missile System Selection

On purpose of demonstrating the developed architecture, a hierarchical decision making mechanism has been implemented. This implementation mimics behavior presented in [10]. Missile system selection problem appeared a number of times. For example, in [15], [16], [17]. In this chapter is aimed to present the example with no tolerances applied to the source values.

4.1 The Problem

In this problem, three alternatives — companies A, B and C — are evaluated. There are main five criteria, that are divided into further 23 criteria. If this is presented as a tree data structure, only leaves are supposed to have actual values. In this case, the 23 subcriteria are evaluated using numbers and words. However, both criteria and subcriteria are weighted with T1 FSs, in order to express favoring some criteria and subcriteria over others at the same level. Aggregation is being done from the leaves to the root. That is, first a score of each criterion is acquired by aggregating its subcriteria. Then all criteria are aggregated to acquire the final score of an alternative. The input data are given in Tables 4 and 5.

A remark should be made that each weight is a triangular T1 FS with tolerance ± 1 around its core. Exceptions to this are smallest and largest weight. That is, $\widetilde{1}$ is $(1,1,1,2,1)$, and $\widetilde{9}$ is $(8,9,9,9,1)$.

4.2 Per-C Architecture for Hierarchical Decision Making

In [10], a procurement judgment advisor (PJA) has been presented as assistance in hierarchical decision making (HDM). Follows an overview of coder, CWW engine and decoder of this Per-C.

The coder's role is to normalize input data by its connotation and by its value. The scores that are T1 FS or IT2 FS are supposedly already normalized to the domain $[\underline{X},\overline{X}]$, which is $[0,10]$ in this example. Hence, they will be only normalized by their connotation, which is the first step. Connotation can be positive (more=better) or negative (less=better). Only the scores with negative connotation will be affected by this normalization. Having a numerical score x_i, in [10] is proposed to normalize its negative connotation by $x_i \to x_i^* = 1/x_i$. T1 FSs are normalized by being replaced with their antonyms, defined as $\mu_A(x) \to \mu_{A^*}(\overline{X} - x)$. This is extendible to IT2 FSs, by applying it to their LMF and UMF. The second step is normalization by domain. Having three scores of the same criterion $X_i = \{x_i(A), x_i(B), x_i(C)\}$ for three alternatives, the normalization is done by division with $\max(X_i)$ and multiplication by $\overline{X} = 10$.

NWAs are the **CWW engine**, and **the decoder** does ranking by defuzzified centroids of the resulting IT2 FSs.

4.3 The Data and Results

The input data are given in Table 4. The score connotations are labeled with $\uparrow$ (positive) and $\downarrow$ (negative) in the beginning of each row. Triangular T1 FS weights have been labeled with a tilde above them. The used words can be divided in two groups: (1) Low and High, and (2) Poor, Average, Good and Very Good. Since this chapter aims to reproduce the results from [10], word definitions are provided as UMF/LMF pairs in Table 5. Whenever a height is not specified as the fifth parameter, it may be assumed that it equals 1.

In further text, numerical results of aggregation for each criterion are provided in Tables 1, 2 and 3. Each final result is presented both numerically and graphically, see Fig. 9.

The recommendation given by the system is the company B. The second place is taken by the company C, and the third place remains for the company A. Similarities between the resulting IT2 FSs by companies are, as follows: $\mathrm{sm}_J(A,B) = 0.05226$, $\mathrm{sm}_J(A,C) = 0.17721$ and $\mathrm{sm}_J(B,C) = 0.31190$. All results are consistent with results provided in [10].

Table 1 Results for company A

Aggregation result	UMF	LMF
Tactics	T1FS(9.05573, 9.17447, 9.17447, 9.24949)	
Technology	UMF(8.24879, 8.86759, 9.00374, 9.11501)	LMF(8.57315, 8.98297, 9.00374, 9.11501)
Maintenance	UMF(4.24313, 6.56368, 7.43684, 8.78182)	LMF(5.53062, 6.59171, 7.4279, 8.16864, 0.53)
Economy	UMF(6.86012, 7.70238, 8.0, 8.7917)	LMF(7.04013, 7.58224, 8.02686, 8.35352, 0.53)
Andvancement	UMF(1.93143, 3.83333, 4.95333, 6.70375)	LMF(3.55857, 3.95616, 4.89323, 5.29286, 0.27)
Final	UMF(5.84673, 7.24328, 7.66767, 8.52748)	LMF(6.55344, 6.88888, 7.86374, 8.07115, 0.27)
Centroid	[6.92295, 7.66822] defuzzified to 7.295589	

Table 2 Results for company B

Aggregation result	UMF	LMF
Tactics	T1FS(8.87148, 9.06815, 9.06815, 9.21781)	
Technology	UMF(8.17754, 8.68956, 8.88571, 9.18593)	LMF(8.68683, 8.80396, 8.8738, 9.10516, 0.57)
Maintenance	UMF(5.84188, 7.85895, 8.78684, 9.73)	LMF(7.99591, 8.22408, 8.76919, 9.13045, 0.3)
Economy	UMF(8.614, 9.40476, 9.70238, 9.98364)	LMF(9.337, 9.46089, 9.61954, 9.69636, 0.3)
Andvancement	UMF(6.50125, 8.52467, 9.25333, 9.79429)	LMF(8.28875, 8.8021, 9.19632, 9.64429, 0.57)
Final	UMF(7.72819, 8.8895, 9.2137, 9.60886)	LMF(8.66804, 8.83498, 9.31372, 9.45663, 0.3)
Centroid	[8.588739, 9.191289] defuzzified to 8.890014	

Table 3 Results for company C

Aggregation result	UMF	LMF
Tactics	T1FS(9.25819, 9.30729, 9.30729, 9.35168)	
Technology	UMF(8.33372, 8.86595, 9.0621, 9.33513)	LMF(8.83236, 8.96747, 9.03916, 9.25383, 0.57)
Maintenance	UMF(4.3575, 6.69737, 7.75526, 9.18091)	LMF(6.41, 6.84849, 7.87427, 8.366, 0.3)
Economy	UMF(7.44957, 8.17074, 8.46836, 8.91002)	LMF(8.10684, 8.22162, 8.40207, 8.49403, 0.3)
Andvancement	UMF(5.15429, 7.082, 7.84667, 8.9875)	LMF(6.77, 7.05493, 8.05444, 8.37, 0.27)
Final	UMF(7.10336, 8.29954, 8.63903, 9.1773)	LMF(7.96407, 8.17024, 8.75713, 8.90594, 0.27)
Centroid	[7.986224, 8.644687] defuzzified to 8.315456	

Table 4 Scores, criteria and weights for the three alternatives

Item	Weighting	Company A	Company B	Company C
Criterion I: Tactics	$\widetilde{9}$			
↑ 1. Effective range (km)	$\widetilde{7}$	43	36	38
↓ 2. Flight height (m)	$\widetilde{1}$	25	20	23
↑ 3. Flight velocity (M. No)	$\widetilde{9}$	0.72	0.80	0.75
↑ 4. Reliability (%)	$\widetilde{9}$	80	83	76
↑ 5. Firing accuracy (%)	$\widetilde{9}$	67	70	63
↑ 6. Destruction rate (%)	$\widetilde{7}$	84	88	86
↑ 7. Kill radius (m)	$\widetilde{6}$	15	12	18
Criterion II: Technology	$\widetilde{3}$			
↓ 8. Missile scale (cm) (l × d-span)	$\widetilde{4}$	521×35-135	381×34-105	445×35-120
↓ 9. Reaction time (min)	$\widetilde{9}$	1.2	1.5	1.3
↑ 10. Fire rate (round/min)	$\widetilde{9}$	0.6	0.6	0.7
↑ 11. Anti-jam (%)	$\widetilde{8}$	68	75	70
↑ 12. Combat capability	$\widetilde{9}$	Very Good	Good	Good
Criterion III: Maintenance	$\widetilde{1}$			
↓ 13. Operation condition requirement	$\widetilde{5}$	High	Low	Low
↑ 14. Safety	$\widetilde{6}$	Very Good	Good	Good
↑ 15. Defilade[a]	$\widetilde{2}$	Good	Very Good	Good
↑ 16. Simplicity	$\widetilde{3}$	Good	Good	Good
↑ 17. Assembly	$\widetilde{3}$	Good	Good	Poor
Criterion IV: Economy	$\widetilde{5}$			
↓ 18. System cost (10,000)	$\widetilde{8}$	800	755	785
↑ 19. System life (years)	$\widetilde{8}$	7	7	5
↓ 20. Material limitation	$\widetilde{5}$	High	Low	Low
Criterion V: Advancement	$\widetilde{7}$			
↑ 21. Modularization	$\widetilde{5}$	Average	Good	Average
↑ 22. Mobility	$\widetilde{7}$	Poor	Very Good	Good
↑ 23. Standardization	$\widetilde{3}$	Good	Good	Very Good

[a] *Defilade* means to surround by defensive works so as to protect the interior when in danger of being commanded by enemy's guns

Table 5 The codebook with six words used in missile system selection problem

Word	UMF	LMF
Low	UMF(0.09, 1.25, 2.5, 4.62)	LMF(1.67, 1.92, 1.92, 2.21, 0.3)
High	UMF(5.38, 7.5, 8.75, 9.81)	LMF(7.79, 8.3, 8.3, 9.21, 0.53)
Poor	UMF(0.09, 1.5, 3, 4.62)	LMF(1.79, 2.28, 2.28, 2.81, 0.4)
Average	UMF(3.59, 4.75, 5.5, 6.91)	LMF(4.86, 5.03, 5.03, 5.14, 0.27)
Good	UMF(5.98, 7.75, 8.6, 9.52)	LMF(8.03, 8.36, 8.36, 9.17, 0.57)
Very Good	UMF(7.37, 9.41, 10, 10)	LMF(8.72, 9.91, 10, 10, 1)

5 Conclusions

In the chapter the basic concepts of computing with words and perceptual computing are presented. An original OO realization of a library for developing Per-Cs is presented. Its architecture is observed from standpoint of important decisions that affected it. Advantages and disadvantages of the decisions are discussed in details. An implementation of Per-C for hierarchical decision making has been developed and used to demonstrate the library. A well known problem of selecting missile system is employed on this purpose. The acquired results correspond to the results from [17] and [10]. It is pointed out that weighted mean based aggregation techniques other than NWAs may exist, as presented in [18].

A perceptual computer is an artificial intelligent device, and is validated according to classical artificial intelligence approach, using the Turing test, [1], [10]. However, perceptual computer is an intelligent agent, [1], an entity with sensors accepting information modeled as IT2 FS, and acting through outputs in the form of recommendations and data. It might be of research interest to explore the possibility of introducing a validation criterion in form of a measure of performance in developing such an agent, instead of the Turing test. Further, perceptual computers are developed having in mind only Zadeh's operators max and min for FSs union and intersections, [10]. It might be of research interest also to consider other forms of perceptual computers, with union and intersection operators other than max and min. Both forms of CWW engine, the first one with aggregation using NWAs, and the second one with a fuzzy expert system in the form of if-then rules heavily use aggregation operators. Interesting field of research might be considering extensions of perceptual computer using different aggregation functions, [19], [20], [21]. It seems that possibility of using some form of the Turing test remains, in choosing adequate aggregation operators.

There are many possibilities, not only for further research work, but also work in developing applications of perceptual computers.

Acknowledgements. This paper is partly supported by Ministry of Education and Science – Serbia, project #36002, "Traffic and communication, planning and management, using computational intelligence techniques".

References

1. Russell, S., Norvig, P.: Artificial Intelligence, A Modern Approach, 3rd edn. Pearson, Upper Saddle River (2010)
2. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)
3. Ross, T.: Fuzzy Logic with Engineering Applications, 2nd edn. John Wiley and Sons, Chichester (2004)
4. Šaletić, D.Z.: Fuzzy Sets and Fuzzy-Logic Systems. Eduka, Belgrade (2012) (in Serbian)
5. Zadeh, L.A.: The concept of linguistic variable and its application to Approximate Reasoning. Information Sciences 8, part I, 199–249 (1975)
6. Zadeh, L.A.: Fuzzy logic = Computing With Words. IEEE Trans. on Fuzzy Systems 4, 103–111 (1996)

7. Zadeh, L. A., Kacprzyk, J.: Computing with Words in Information/Intelligent Systems. Book 1, Foundations, Book 2 Applications. Physica-Verlag, Heidelberg (1999)
8. Castillo, O., Melin, P.: Type-2 Fuzzy Logic: Theory and Applications. Springer, Berlin (2008)
9. Mendel, J.M.: The perceptual computer: An architecture for computing with words. In: Proc. IEEE Int. Conf. Fuzzy Syst (FUZZ-IEEE 2001), Melbourne, Australia, pp. 35–38 (December 2001)
10. Mendel, J.M., Wu, D.: Perceptual Computing: Aiding people in Making Subjective Judgments. John Wiley & Sons Inc., Hoboken (2010)
11. Šaletić, D.Z., Anđelković, M.: A Perceptual Computer Software Model Applied to Hierarchical Decision Making. In: Pap, E., Fodor, J. (eds.) Proceedings of the IEEE 9th International Symposium on Intelligent Systems and Informatics, SISY 2011, Subotica, Serbia, September 8-10, pp. 145–150 (2011); 27-sisy2011.pdf, IEEE Catalog Number: CFP1084C-CDR, Digital Object Identifier: 10.1109/SISY.2011.6034311, ISBN: 978-4244-7395-3
12. Karnik, N.N., Mendel, J.M.: Centroid of a type-2 fuzzy set. Information Sciences 132, 195–220 (2001)
13. Zadeh, L.A.: The role of fuzzy logic in the management of uncertainty in expert systems. Fuzzy Sets and Systems 11, 199–227 (1983)
14. De Luca, A., Termini, S.: A definition of non-probabilistic entropy in the setting of fuzzy sets theory. Information and Computation 20, 301–312 (1972)
15. Šaletić, D., Velašević, D.: Missile System Selection Based on the Fuzzy Sets Theory. Yugoslav Journal of Operations Research 10(1), 47–61 (2000)
16. Gardašević-Filipović, M., Šaletić, D.: Multicriteria Optimization in Fuzzy Environment: System Selection by the Analytic Hierarchy Process. Yugoslav Journal on Operation Research 20(1), 71–85 (2010)
17. Wu, D., Mendel, J.M.: Computing With Words for Hierarchical Decision Making Applied to Evaluating a Weapon System. IEEE Transactions on Fuzzy Systems 18(3) 2010, 441–460 (2010)
18. Anđelković, M.: Object-Oriented Realization of Perceptual Computer and a New Inference Operator. e-RAF Journal on Computing, Electronic Scientific Journal 3, 17–33 (2011)
19. Grabish, M., Marishal, J.L., Mesiar, R., Pap, E.: Aggregation Functions. Cambridge University Press (2009)
20. Dujmović, J.J.: Weighted conjunctive and disjunctive means and their application in system evaluation. Journal of the University of Belgrade, EE Department, Series Mathematics and Physics, No. 461-497, 147–158 (1974)
21. Dujmović, J.J.: Continuous Preference Logic for System Evaluation. IEEE Trans. on Fuzzy Systems 15(6), 1082–1099 (2007)

Classical and Fuzzy Approaches to 2–DOF Control Solutions for BLDC–m Drives

Alexandra-Iulia Stinean, Stefan Preitl, Radu-Emil Precup,
Claudia-Adina Dragos, and Mircea-Bogdan Radac

Abstract. This chapter gives two–degree–of–freedom (2–DOF) speed control solutions for brushless Direct Current motor (BLDC–m) drives with focus on design methodologies. A classical 2–DOF structure, 2–DOF proportional-integral (PI) and proportional–integral–derivative (PID) structures and 2–DOF fuzzy control solutions are presented and approaches regarding the methods are highlighted. A case study concerning a BLDC–m drive with variable moment of inertia is presented. Comparative studies based on digital simulation results are included to exemplify the design methods.

Keywords: Speed control, 2–DOF control, brushless direct current motor, PID control.

1 Introduction

The research results obtained in mechatronics systems during the last decade have been focused on setting theoretical foundations and, based on it, on enlarging the application domains. The studies are oriented to assessing the quality of motion control systems and on disclosing the insurmountable performance limitations inherent in the mechanical structures. Both accuracy and robustness are essential characteristics in high performance motion applications with variable moment of inertia (VMI), [1]. As result, robust control solutions for servomechanisms have been

Alexandra-Iulia Stinean · Stefan Preitl · Radu-Emil Precup ·
Claudia-Adina Dragos · Mircea-Bogdan Radac
"Politehnica" University of Timisoara,
Department of Automation and Applied Informatics,
Bd. V. Parvan 2, RO–300223 Timisoara, Romania
e-mail: kassandra3107@yahoo.com,
{stefan.preitl,radu.precup}@aut.upt.ro,
{claudia.dragos,mircea.radac}@aut.upt.ro

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 175–193.
DOI: 10.1007/978-3-642-33959-2_10

proposed to realize effective disturbance suppression in the presence of stability degree constraints [2], [3].

The main advantages of using two–degree-of-freedom (2–DOF) control solutions concern simultaneous good feedback properties, reference tracking and disturbance rejection. One drawback of 2–DOF controllers is that the overshoot reduction is paid by slower set–point responses [3], [4], [5]. So, the design of 2–DOF controllers represents a multi-objective problem. With this regard, Miklosovic and Gao offer in [3] a robust 2–DOF control design technique that extends the concepts of active disturbance rejection control.

Several 2–DOF control structures (CSs) have been proposed during the last decades. They are characterized by different combinations of the inclusion in the feed-back loop the reference part and of the disturbance part requirements. Even the classical design has different approaches [6], [7], [8], [9], [10]. For the development of classical 2–DOF PI(D) CSs for low order plants, Araki and Taguchi present in [6] similarities between 2–DOF control structures and one–degree–of–freedom (1–DOF) controllers (mainly PI(D) ones) extended with input filters (reference and feedback).

Alternative approaches to the design of 2–DOF proportional–integral (PI) and proportional–integral–derivative (PID) control solutions are the ESO–m [11] and the 2p–ESO–m design methods [12]. These approaches are recommended mainly for applications with VMI. The computer–aided design of various types of 2–DOF controllers based on algebraic methods is analyzed in [12], [14], [13].

The fuzzy logic technique can be also inserted in the 2–DOF CSs, and several approaches are suggested for 2–DOF fuzzy control (FC) structures. A 2–DOF controller consisting of a one-step-ahead fuzzy pre-filtering in the feed–forward loop and a PI–fuzzy controller in the feedback loop dedicated to the foot trajectory tracking control is discussed in [15] and [16], where self–tuning and model reference adaptive 2–DOF PID–fuzzy controllers are presented. A new framework for the design of generic 2–DOF linear and fuzzy controllers dedicated to plants with integral components and nonlinearities is proposed and applied in [17] and [18]. In addition, the variability of the plant parameters needs sometime also the permanent adaptation of the control algorithm or of its parameters as shown in [13], [15] and [16].

The presented research results are based mainly on easy accessible references. They are focused on two classes of 2–DOF controllers, the classical 2–DOF (and its PI(D) representation) and 2–DOF fuzzy controllers. The applications involve a class of mechatronics systems.

The chapter is structured as follows. Section 2 presents a brief overview on the classical 2–DOF controller approach based on [14] and [19], on 2–DOF–PI(D) equivalent structures providing the foundation for discussion and comparison for the design methodology, and on the basic structures for 2–DOF fuzzy controllers derived from a PI(D) approach. Section 3 gives a short description of the plant, i.e., a servo system application built around a brushless Direct Current Motor

(BLDC–m) with VMI, with inner control (current) and usable models for design. Section 4 discusses some aspects regarding speed control solutions with 2–DOF controllers, experimental scenarios and simulation results. Section 5 is dedicated to the concluding remarks.

2 Classical Structures of 2–DOF and 2–DOF PI(D) Controllers

2.1 Basic Structure and Polynomial Design of 2–DOF Controllers

The 2–DOF CS in its classical (discrete or continuous) form [4], [19], uses two distinct controllers (Fig. 1): the reference controller $T(z)/R(z)$, where $T(z)$ is the reference filter, and the feedback controller $S(z)/R(z)$. The polynomial $R(z)$ is the common part which include mainly the integral components. The classical design of the unknown polynomials $T(z)$, $S(z)$ and $R(z)$ is known as polynomial design problem based on solving the polynomial Diophantine equation [5], [14] under different particularities in treating the constrains and causality (degree) conditions for the polynomials; such conditions are exemplified in [4] and [5]. Unlike the 1–DOF CS, in case of the 2–DOF controller the enlisted attributes can be separately adjusted without influencing one another.

In the usual discrete form the plant is characterized by the pulse transfer function (t.f.) calculated from the continuous model

$$P(z) = (1 - z^{-1})Z\{\frac{P(s)}{s}\} = \frac{B(z)}{A(z)}, \tag{1}$$

where the plant parameters can be time variable. The servo performances are given by a reference model in the form $\frac{B_m(z)}{A_m(z)}$ with an additional condition for the zero control error. Finally, the t.f. of the CS obtains the form

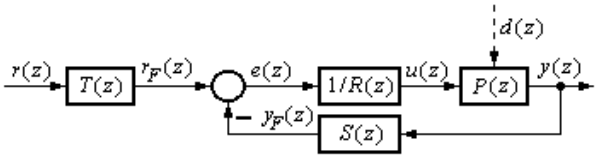


Fig. 1 Structure of classical 2–DOF controller and control structure with the $R(z)$ component placed in the loop

$$\begin{aligned} &\frac{T(z)}{A(z)R'(z)+B^{-}(z)S(z)} = \frac{B'_m(z)}{A_m(z)} \cdot \frac{A_o(z)}{A_o(z)}, \\ &T(z) = B'_m(z)A_o(z), \\ &\mathrm{A}(z)\mathrm{R}'(z) + \mathrm{B}^{-}(z)\mathrm{S}(z) = \mathrm{A_m}(z)\mathrm{A_o}(z). \end{aligned} \tag{2}$$

The last equation in (2) is a Diophantine equation over the ring of polynomials, and its solutions are the coefficients of the polynomials $T(z)$, $R(z)$ and $S(z)$ [12].

2.2 PID Controllers. 2–DOF Controller Interpretation

For 1–DOF controllers, the CS performance can be improved using several particular controller structures with non-homogenous dynamics with respect to the two inputs [5] as shown in Fig. 2. Each controller block can be characterized by its own t.f.s. The presented approach permits also an easy 2–DOF interpretation of the design as discussed in [7] for classical PI(D) controllers and in [8] for fuzzy controllers.

Three such 2–DOF CSs are presented in Fig. 3 and referred to as [17] the reference input filter structure, Fig. 3 (a), the feed-forward structure, Fig. 3 (b), and the feedback structure, Fig. 3 (c). The connections between 2–DOF and extended with input filters of 1-DOF controller structures are synthesized in Table 1, where: P – proportional, D – derivative, I – integral, L1(2) – first (second) order lag filter. The choice of a certain representation of the controller depends on the structure of the available controller and on the adopted algorithmic design method and the result of this design.

Using the approach proposed in [6], the main PI(D) controller component – or/and $C^*(s)$ - defined in Fig. 3 (a) to (c) are characterized by the following t.f.s:

$$C^*(s) = \begin{cases} \frac{k_c(1+T_F s)}{s} & \text{PIcontroller} \\ \frac{k_c(1+T_F s)(1+T'_c s)}{s[1+(T_d/N)s]} & \text{PIDcontroller} \end{cases}, \tag{3}$$

with a possible additional block $C_F(s)$ which accelerates the effect of the reference input in the control signal:

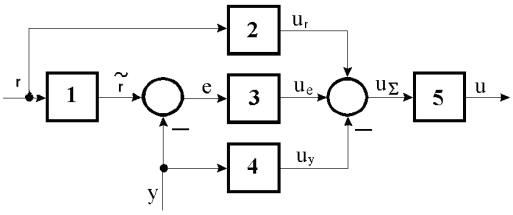


Fig. 2 Typical controller structures and particular forms of the modules

Table 1 Connections between 2–DOF controllers and extended 1–DOF controller structures

Fig. 3 (a)		$F(s)$	–	$F(s)C(s)$	$C(s)$	Remarks
Fig. 3 (b)		–	$C_F(s)$	$C(s)$–$C_F(s)$	$C(s)$	-
Fig. 3 (c)		–	$C_F(s)$	$C^*(s)$	$C^*(s)$+$C_F(s)$	-
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)				

$$C_F(s) = \begin{cases} k_c(T_c - T_F) & \text{PIcontroller} \\ \frac{k_c(T_c - T_F)(1+T_c's)}{1+(T_d/N)s} & \text{PIDcontroller} \end{cases}$$

The digital implementation can be supported by the classical informational diagram presented in detail for example in [5]. The bump-less switching between two control algorithms (c.a.s) – connected to linearized plant models and referred as c.a. (1) and c.a. (2) – needs a permanent modification of the tuning parameters and the reconsideration of the past values in the control algorithms.

2.3 2–DOF Takagi–Sugeno Fuzzy Control Structures for PID Controllers

The main advantage of the classical Takagi–Sugeno (TS) fuzzy control (FC) structure concerns an easy modeling of the nonlinearities. They also do not need the special bump-less circuit. Extended 2–DOF FC structures can be defined on the ba-

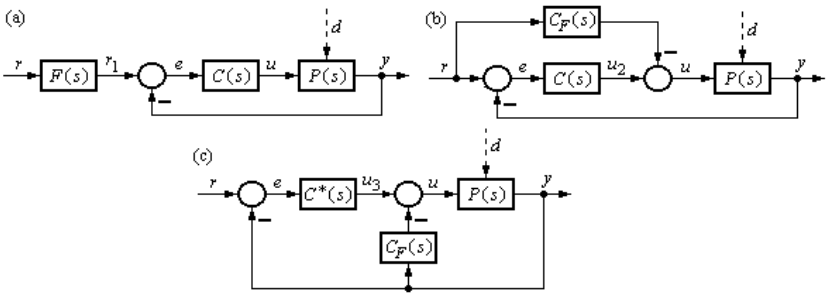


Fig. 3 Structures for 2–DOF controllers as extensions of 1–DOF controllers

sis of the structures given in Figs. 2 and 3; for example, [5] and [17] offer 2–DOF FCs defined around TS fuzzy blocks FB–T_c implemented in terms of Figs. 4 to 7.

The development of the extended 2–DOF PI–FC starts with the definition and development of the classical PI block with the t.f.

$$G^{\tau}(s) = \frac{k_c}{s}(1 + s\tau),$$

with $\tau \geq 0$.

The main PI(D) block is fuzzified in the set-point filter structure or in the feed-forward structure, and the transfer function $C(s)$ (or $C^*(s)$) is expressed in (3). The fuzzification of the generic PI block with the t.f. $G^{\tau}(s)$ leads to the fuzzy block FB–τ; it is accepted that the continuous-time linear block with the t.f. $G^{\tau}(s)$ has the control error e as input and the control signal u as output (other variants are also possible). The structure of the block FB–τ is presented in Fig. 8, where FB is the TS fuzzy block without dynamics, with:

$$\Delta e(k) = e(k) - e(k-1)$$

is the increment of control error,

$$\Delta u(k) = u(k) - u(k-1)$$

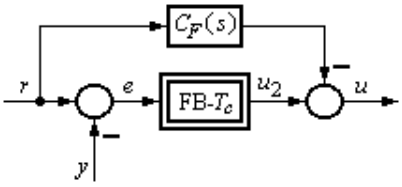


Fig. 4 Structure of feed-forward 2–DOF PI-fuzzy controller

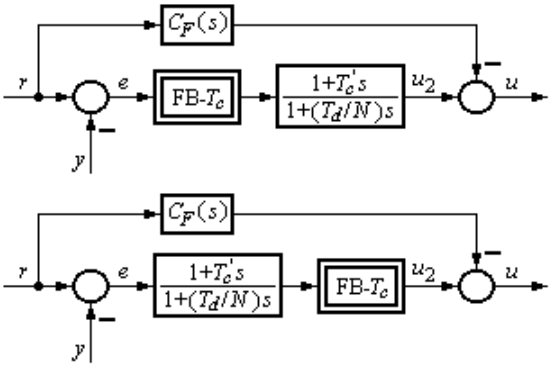


Fig. 5 Structures of feed-forward 2–DOF PID–fuzzy controllers

is the increment of control signal, and k is the index of the current sampling interval because the block FB–τ is implemented as a digital controller.

The fuzzification in the block FC is based on the input membership functions illustrated in Fig. 9, which can be applied for the TS fuzzy block FB–τ. For a low-cost implementation of the 2–DOF fuzzy controllers initially three input membership functions are defined. Fig. 9 points out the tuning parameters, B_e and $B_{\Delta e}$.

More membership functions can be defined for nonlinear plants and high performance specifications. The analysis of and design of the fuzzy controllers should account for the necessary nonlinear scaling factors of the input and output variables of the block FB which must be inserted in the plant. Accepting the sampling period T_s, Tustin's method can be applied to discretize the continuous-time linear generic PI block with the t.f. $G^{\tau}(s)$. This results in the following recurrent equation of the incremental digital generic PI block and its parameters:

$$\Delta u(k) = K_P\,[\Delta\, e(k) + \mu\, e(k)],\;\; K_P = k_c\,(\tau - T_s/2),\;\; \mu = 2T_s/(2\tau - T_s).$$

The complete rule base of the TS fuzzy block FB–τ is given by

Rule 1: IF $e(k)$ IS N AND $\Delta e(k)$ IS P THEN $\Delta u(k) = K_P^1[\Delta e(k) + \mu^1 e(k)]$,
Rule 2: IF $e(k)$ IS ZE AND $\Delta e(k)$ IS P THEN $\Delta u(k) = K_P^2[\Delta e(k) + \mu^2 e(k)]$,
Rule 3: IF $e(k)$ IS P AND $\Delta e(k)$ IS P THEN $\Delta u(k) = K_P^3[\Delta e(k) + \mu^3 e(k)]$,
Rule 4: IF $e(k)$ IS N AND $\Delta e(k)$ IS ZE THEN $\Delta u(k) = K_P^4[\Delta e(k) + \mu^4 e(k)]$,
Rule 5: IF $e(k)$ IS ZE AND $\Delta e(k)$ IS ZE THEN $\Delta u(k) = K_P^5[\Delta e(k) + \mu^5 e(k)]$,
Rule 6: IF $e(k)$ IS P AND $\Delta e(k)$ IS ZE THEN $\Delta u(k) = K_P^6[\Delta e(k) + \mu^6 e(k)]$,
Rule 7: IF $e(k)$ IS N AND $\Delta e(k)$ IS N THEN $\Delta u(k) = K_P^7[\Delta e(k) + \mu^7 e(k)]$,
Rule 8: IF $e(k)$ IS ZE AND $\Delta e(k)$ IS N THEN $\Delta u(k) = K_P^8[\Delta e(k) + \mu^8 e(k)]$,
Rule 9: IF $e(k)$ IS P AND $\Delta e(k)$ IS N THEN $\Delta u(k) = K_P^9[\Delta e(k) + \mu^9 e(k)]$.

This rule base shows, by the additional upper indices in the rule consequents, that the TS fuzzy block FB–τ can be obtained from the separate tuning of nine linear blocks FB–τ. Therefore the TS fuzzy block FB–τ exhibits like a bump-less interpolator of nine separately tuned linear PI blocks defined in accordance with (9). The SUM and PROD operators are used in the inference engine of the TS fuzzy block FB–τ, and the weighted average method is used in the defuzzification. The modal equivalence

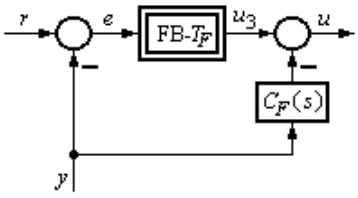


Fig. 6 Structure of feedback 2–DOF PI–fuzzy controller

principle is applied to guarantee the quasi–PI behavior of the fuzzy block FB–τ. This results in the useful tuning conditions

$$B_{\Delta e} = \mu\, B_e,\ \ B_{\Delta u} = K_P\, \mu\, B_e. \tag{4}$$

The tuning conditions are applied in the TS fuzzy block FB–τ. The restructuring of the controller structure allows: accounting for the experience design with PI and PID controllers; the easy introduction of additional specific facilities specific to PI(D) controllers (output or inner limitations, anti windup reset referred to as AWR, smoothing the transition from one algorithm to another one), and the conversion of a PI, PID controller into a 2–DOF controller and vice versa. Such conversion relations are given in [17] and [18].

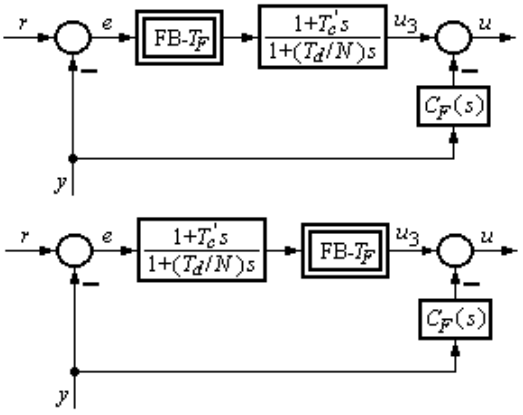


Fig. 7 Structures of feedback 2–DOF PID–fuzzy controllers

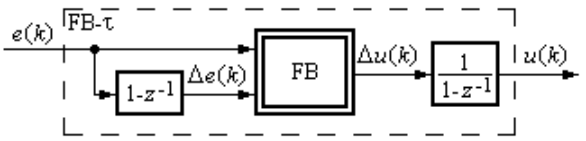


Fig. 8 Structure of FB–τ

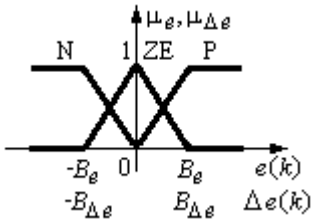


Fig. 9 Input membership functions of FB–τ

The improvement of CS performance can be ensured by the proper choice of the parameter $B_e > 0$ in (4). This can be assisted by the stability analysis accompanied by useful tools specific to the analysis and modeling of fuzzy control systems [20], [21], [22], [23], [24], [25], [26], [27], [28], [29].

3 Mathematical Modeling of Plant as BLDC–m Drive

The controlled plant is represented by a BLDC–m drive with internal current control loop. In the symmetrical operating mode the mathematical models (MMs) of classical DC motors and of BLDC–ms are very close. This fact leads to some similarities in the development of control solutions, relative simple control structure and cheap implementation of the control algorithms [30], [31]. In case of BLDC–ms the current switch is obtained by specialized converters whose commutation time is determined by the position of the rotor, determined either by position sensors or by sensor-less techniques. The major advantages of BLDC–ms are lifespan, high efficiency, very good torque-speed characteristics, and quiet operation. It is accepted that in case of vector control value of the current trends to be zero and the speed control is achieved through the current. Moreover, it is assumed the excitation flux is constant; and the nonlinear effects due to different constructive elements are neglected.

The matrix form of the main equations of the MM of a BLDC–m is presented in [30]. The electromagnetic torque m_e is used in the movement equation

$$m_e = J_e \frac{d}{dt}\omega_r + k_f \omega_r + m_{Load},$$

where m_{Load} is the load torque (i.e., a time variable load disturbance input); the moment of inertia of the driven mechanism J_{mech} can be constant or time-variable:

$$J_e(t) = J_{BLDC} + J_{mech}(t).$$

Two case studies will be considered in the next section. The first case study, with time-variable reference input (including regions characterized by $r(t)$ = const), is representative for defining the adopted controller design methods (the classical 2–DOF and the case with 2–DOF PI controller and feed-forward filter). The main CS performance indices and some simulation results are synthesized in [31].

The second case, to be treated in this chapter, considers the application (the plant) as a simulated plant for a winding process with VMI and constant linear speed, $v_t(t)$ = const according to Fig. 10, where the reference input $r(t)$ for the angular speed $\omega(t)$ must be correlated with the modification of the working roll radius $r_r(t)$. In this context, the CS should ensure the modification of reference input $r(t)$ and the tuning and retuning of the controller parameters as well.

To treat the first aspect, the following condition must be fulfilled, and the measurement of $r_r(t)$ enables the continuous modification of the reference input $r(t)$:

$$v_t(t) = \text{const} \Rightarrow r(t) = k/r_r(t) \;\; (a), \;\; J_e(t) = \frac{1}{2}\rho\pi r_r^4 t \;\; (b). \tag{5}$$

In the design step the inner loop can be characterized by linearized equivalent second order benchmark-type t.f. connected to representative operating points [7]. Such MMs, expressed as benchmark-type t.f.s, are:

- * in the speed control applications:

$$H_P(s) = \frac{k_P}{(1+sT_\Sigma)(1+sT_1)}, \;\; T_1 >> T_\Sigma, \tag{6}$$

$$H_P(s) = \frac{k_P}{(1+sT_\Sigma)(1+sT_1)(1+sT_2)}, \;\; T_1 > T_2 >> T_\Sigma, \tag{7}$$

- * in the position control applications:

$$H_P(s) = \frac{k_P}{s(1+sT_\Sigma)}, \tag{8}$$

$$H_P(s) = \frac{k_P}{s(1+sT_\Sigma)(1+sT_1)}, \;\; T_1 >> T_\Sigma, \tag{9}$$

with time-variable $T_1 = f(J_e(t))$.

The main PI(D) controllers can be tuned by the Extended Symmetrical Optimum method (ESO–m) [11] which can improve the CS performance. The main advantage consist ins fact that only one design parameter (referred to as β) must be adopted. Useful design diagrams concerning the choice of the parameter β and the tuning relations are given in [11]:

$$k_c = \frac{1}{k_P \beta \sqrt{\beta} T_\Sigma^2}, \;\; T_c = \beta \, T_\Sigma, \;\; T_c' = T_1. \tag{10}$$

The CS performance indices can be improved further by introducing the reference filter of first-order (a) or of second-order (b), with the t.f.s:

$$F(s) = \frac{1}{1+\beta \, T_\Sigma s} \;\; (a) \;\; or \;\; F(s) = \frac{1+(\beta - \sqrt{\beta})T_\Sigma s + \beta \, T_\Sigma s^2}{(1+\beta \, T_\Sigma s)(1+T_\Sigma s)} \;\; (b). \tag{11}$$

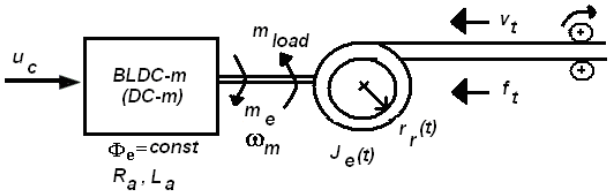


Fig. 10 Block diagram of controlled plant for winding process with VMI

Since $r(t)$ is time-variable and the inner loop can be characterized by a simplified, benchmark-type model, the main (speed) controller is a PID one, extended or not with an integral (I) component. The open–loop CS has $k_0 = k_P k_c$, $T_1 >> T_\Sigma$, k_P and T_1 – time–variable, T_1 must be compensated (applying the pole–zero compensation technique, $T_c' = T_1$), and the gain k_0 must be maintained constant using the permanent recalculation of k_c according to (10).

The considered BLDC–m–based servo system with VMI is characterized by the following parameters: $p = 2$, $R_a = 1\ \Omega$, $L_a = 0.02$ H, $V_{DC} = 220$ V, $J_{e0} = 0.005$ $kg \cdot m^2$. The inner loop, which contains an on–off controller, ensures a second-order (with lag) behavior of the plant. The controlled parameters of the BLDC–m, θ and lpm, were set to ensure that the motor can operate at any desired speed within the range $0 \leq \omega \leq 314\ \mathrm{s}^{-1}$. The functional diagram of a speed controlled BLDC–m drive is presented in Fig. 11.

The internal loop (block diagram) contains the PWM inverter, the current controllers (on–off–type controller with hysteresis and also the current sensors). The main loop contains the actual speed controller and the incremental speed (position) sensor [32], [33], [34], [35]. The phase selection block ensures the proper switching of the phases and the initialization as well.

In the design step of the speed controller the inner control loop characterized by the equivalent second–order benchmark–type t.f. (6) with $k_P = 40$, $T_1 = 0.03$ s and $T_\Sigma = 0.015$. Accepting the variation range of the equivalent moment of inertia $_e(t)$ as relatively small, the controller is tuned using the linearized models (6)–(9). In the first step the classical PI(D) speed controller is designed and tuned using the ESO–m for $\beta = 12$ (or in its extended form of double parameterized (2p–SO–m) form [12]). Finally, due to the fact that the reference input is permanently variable, the reference filter (11) (b) is applied. So the design becomes as a classic case of a

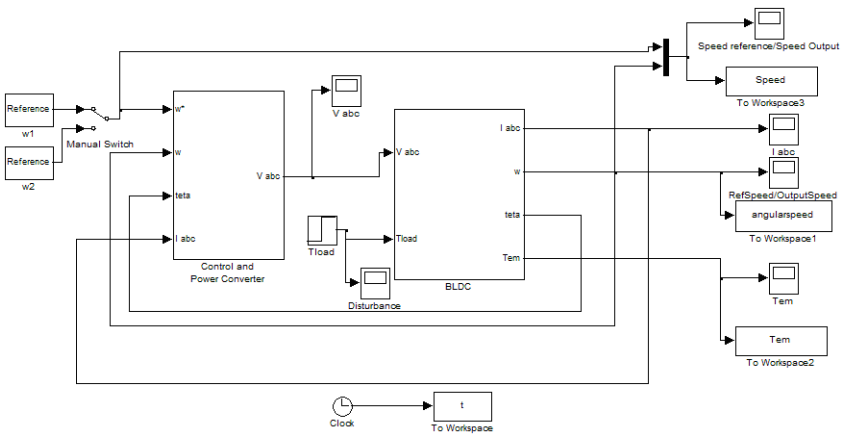


Fig. 11 Functional diagram of a speed controlled BLDC–m drive

2–DOF PI(D) controller, Fig. 3 (a). Accepting a high value for β, which ensures a great value for the nominal phase margin, the controller's parameters are calculated for the average value of $J_e(t)$, without parameter adaptation and with AWR.

4 Experimental Scenarios and Simulation Results

This chapter investigates analytical structures of two-input single-output (TISO) Takagi–Sugeno fuzzy PI(D) controllers versus conventional PI(D) control and variable gain control. Generally, to design and tune PI(D) fuzzy controllers, the continuous–time PI controllers are discretized resulting in the incremental versions of the quasi–continuous digital PI controllers with input/output integration.

The higher operating speed can be accounted for the components of the trapezoidal speed curve, resulting in an average speed equal to the movement speed. The selected structures of 2–DOF PI(D) CSs and of 2–DOF fuzzy PI(D) CSs are presented in detail in Fig. 12 (a), (b) and (c).

In the first experimental scenario the reference input contains (in chronological order) an acceleration part (0 – 1.0 s), a part with constant velocity (1.0 – 1.5 s), a deceleration part (1.5 – 3.0 s), a part with constant velocity (3.0 – 3.5 s), a part characterized by a step disturbance input applied at 3.1 s, and finally a part of deceleration until a stop is reached (3.5 – 4.5 s). This chapter presents the results of the tests conducted with the presented CSs in a first step through simulation for a driving system with BLDC–m with constant moment of inertia.

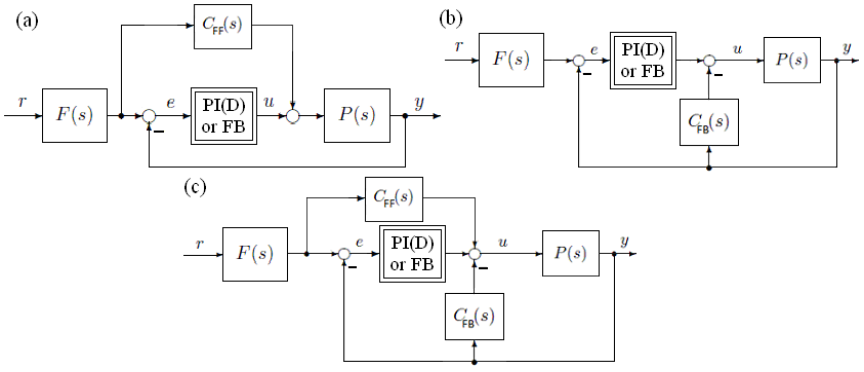


Fig. 12 (a): structures of feed-forward-set-point filter (FF–SP) 2–DOF PI(D)/fuzzy PI(D) CSs, (b): structures of feedback–set–point filter (FB–SP) 2–DOF PI(D)/fuzzy PI(D) CSs, (c): structures of feed–forward–feedback–set–point filter (FF–FB–SP) 2–DOF PI(D)/fuzzy PI(D) CSs

4.1 Feed–Forward–Set–Point Filter Structures

The main simulation results are synthesised in Fig. 13. Fig. 13 (a) illustrates for the simulation scenario the system's output, the angular speed $\omega(t)$ of BLDC–m drives in case of the FF–SP structure. The output is almost the same for both cases, the FF–SP–2–DOF PI(D) controller and the FF–SP–2–DOF fuzzy PI(D) controllers. Fig. 13 (b) details the output around the portion between 0.95 and 1.25 seconds sustaining that the FF–SP structure with 2–DOF fuzzy PI(D) controller ensures better behavior. The control error versus time is presented in Fig. 13 (c); it can be seen that in the tracking phase the control error for FF–SP structure with 2–DOF fuzzy controller is less than the control error for the FF–SP structure with 2–DOF PI(D) controller. For both cases if the angular speed is constant, the control error reaches zero (due to the presence of the I component in controller's structure).

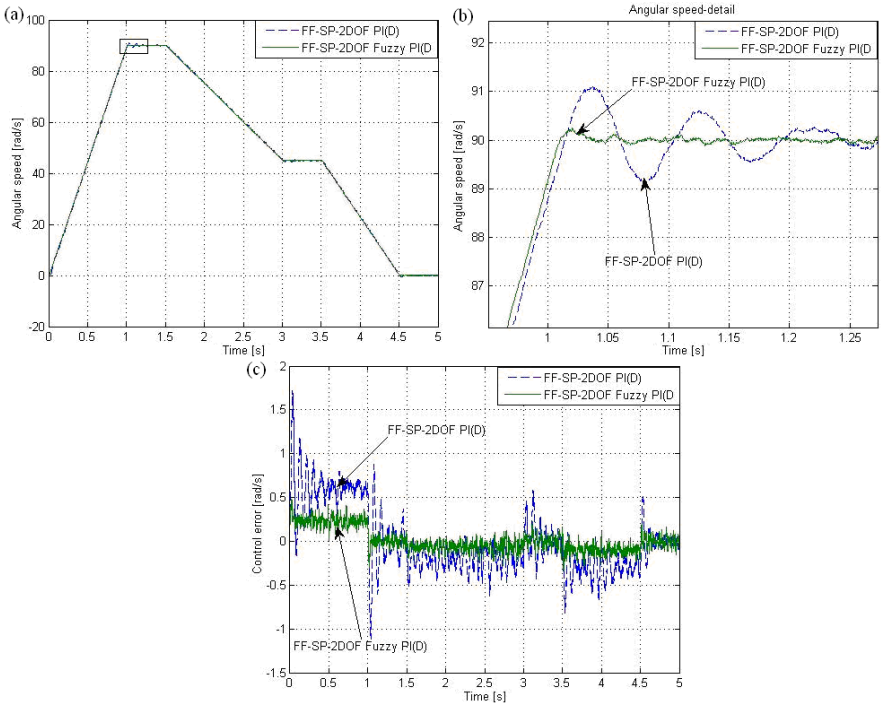


Fig. 13 Simulation results for the FF–SP structures

4.2 *Feedback–Set–Point Filter Structures*

For the same simulation scenario applied to the FB–SP structure, the system performance indices can be computed using the system responses given in Fig. 14 (a), (b) and (c). For the FB–SP structure with 2–DOF fuzzy PI(D) controller and for the FF–SP structure with 2–DOF PI(D) controller the angular speed are almost the same as suggestively illustrated in Fig. 14 (a).

The results presented in Fig. 13 and 14 show that no discontinuities in the variation of $v(t)$ are observed. During the winding regime the output of the controller remains within the limitations.

According to Fig. 14 (b), the overshoot for the FB–SP structure with 2–DOF PI(D) controller is less than the overshoot for the FF–SP structure with 2–DOF PI(D) controller. In comparison with the FB–SP structure, the FF–SP structure has a more oscillatory character. Finally, Fig. 14 (c) presents the evolution of the control error for the FB–SP structure with 2–DOF PI(D) controller and for the CS with 2–DOF fuzzy PI(D) controller.

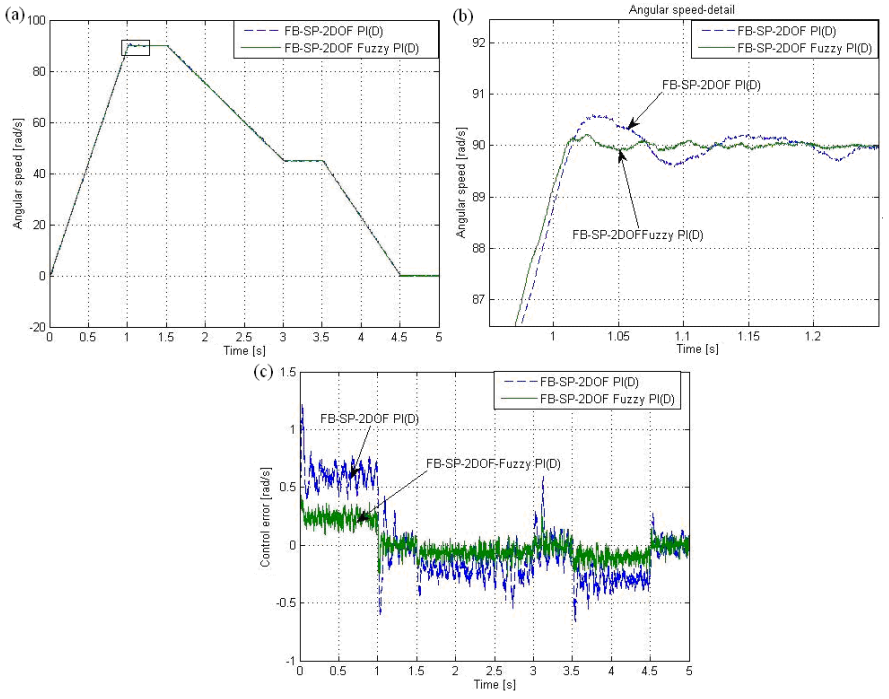


Fig. 14 Simulation results for the FB–SP structures

4.3 Feed–Forward–Feedback–Set–Point Filter Structures

The same simulation scenario is considered for both FF–FB–SP structures. Fig. 15 (a) offers the evolution of the angular speed. Fig. 15 (b) offers some details concerning the angular speed around 1.0 - 1.5 seconds, and it points out that the FF–FB–SP structure with 2–DOF fuzzy controller performs better in comparison with the FF–FB–SP structure with 2–DOF PI(D) controller. Fig. 15 (c) illustrates the control error versus time.

The analysis of these simulations results shows the main conclusion which states that the FF–FB–SP structure with 2–DOF fuzzy PI(D) controller has the best transient behaviors. Therefore, in this case the FF–FB–SP structure with 2–DOF fuzzy PI(D) controller will be adopted for the next experiments (simulations) focused on the case of BLDC–m drive with VMI.

Fig. 16 (a) to (d) synthesizes the main simulation results for fixed controller's parameters according to (10). Since the reference input is permanently variable, the reference filter (11) (b) is applied. The drum radius is calculated according to (5) (a), and the moment of inertia $J_e(t)$ is calculated according to (5) (b). The simulation scenario illustrated in Fig. 16 consists of the start regime, and the angular velocity modification to ensure the desired linear speed v_r, which needs a proper modification of reference input, corresponding to increasing the radius $r_r(t)$ and to the variation of the moment of inertia.

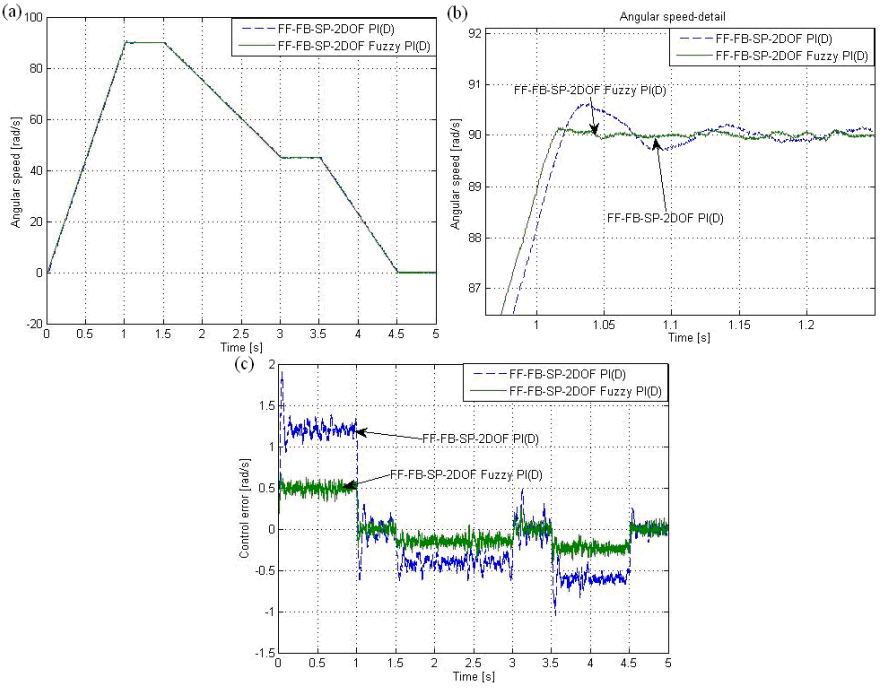


Fig. 15 Simulation results for the FF–FB–SP structures

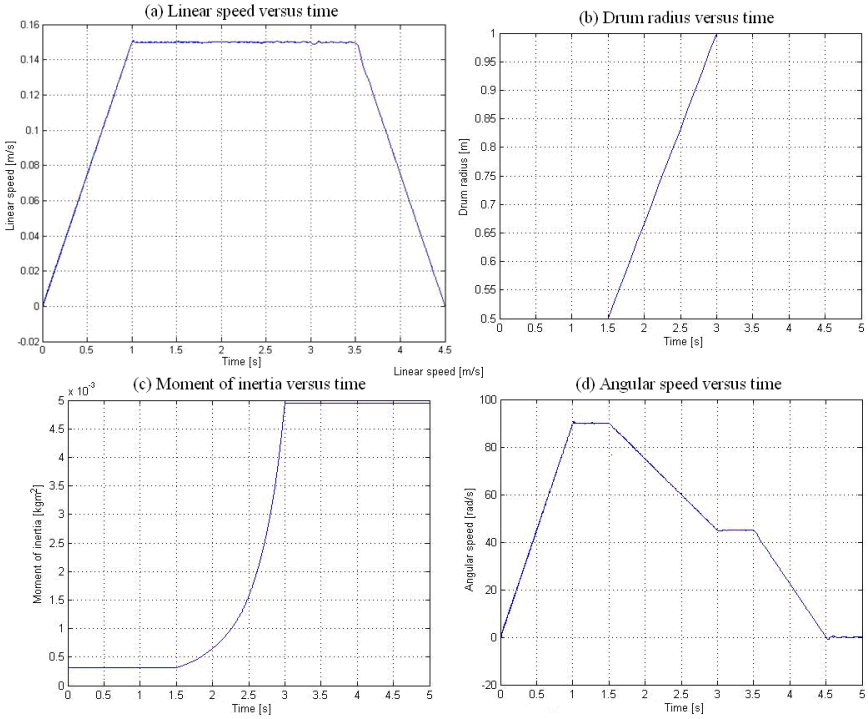


Fig. 16 Simulation results for the FF–FB–SP structure with VMI.

5 Conclusions

This chapter has presented control solutions and methods based on extensions of 2–DOF PI(D) control structures and derived Takagi–Sugeno 2–DOF fuzzy controllers, focused on three basic structures: the FF–SP–, the FB–SP– and FF–FB–SP– structures. The application is a BLDC–m based servo system driving system. The integral element specific to the 2–DOF controllers is included in the forward channel of the control loop.

The advantage of fuzzy logic is the ability to tune certain variables easily by varying the linguistic rules or input variables. The main feature of TS fuzzy models is the expression of the local dynamics of each fuzzy rule by linear system models, and this has been employed in our control solutions.

The proposed controller structures are tuned by the straightforward adaptation of the tuning relations given in the literature. The choice of different representations depends on the structure of the controller, the methods used in controller design and tuning and the final form of the t.f.

Due to the nonlinearities in the plant, the fuzzy control solutions are more advantageous in comparison with other BLDC–m control solutions reported by the

state–of–the–art. The proposed controller structures can be implemented relatively easily in quasi continuous digital version by using well–known approaches [4], [5], [7].

The application related to a BLDC–m drive system with VMI confirms the applicability of the methods. Other aspects of interest for future research include sliding mode control and state observers with disturbance observation.

Acknowledgements. 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 partially supported by the strategic grant POSDRU ID 77265 (2010) of the Ministry of Labor, Family and Social Protection, Romania, co–financed by the European Social Fund – Investing in People.

References

1. Isermann, R.: Mechatronic systems: Fundamentals. Springer, Heidelberg (2005)
2. 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. Electron 47, 315–324 (2000)
3. Miklosovic, R., Gao, Z.: A robust two–degree–of–freedom control design technique and its practical application. In: Proceedings of 39th IAS Annual Meeting Conference, Seattle, WA, USA, vol. 3, pp. 1495–1502 (2004)
4. Landau, I.D., Zito, G.: Digital control systems: Design, identification and implementation. Springer, London (2006)
5. Preitl, S., Precup, R.E., Preitl, Z.: Control structures and algorithms. Editura Orizonturi Universitare, Timisoara (2009) (in Romanian)
6. Araki, M., Taguchi, H.: Two–degree–of–freedom PID controllers. Int. J. Control Automat. Syst. 1, 401–411 (2003)
7. Astrom, K.J., Hagglund, T.: PID controllers theory: Design and tuning. Instrument Society of America, Research Triangle Park (1995)
8. Leva, A., Bascetta, L.: On the design of the feed-forward compensator in two-degree-of-freedom controllers. Mechatronics 16, 533–546 (2006)
9. Alfaro, V.M., Vilanova, R., Arrieta, O.: Robust tuning of Two-Degree-of-Freedom (2–DoF) PI/PID based cascade control system. J. Process Control 19, 1658–1670 (2009)
10. Cheng, Z., Yamada, K., Sakanushi, T., Murakami, I., Ando, Y., Nguyen, L.T., Yamamoto, S.: A design method for two–degree–of–freedom multi–period repetitive controllers for multiple–input/multiple–output systems. In: Preprints of 18th IFAC World Congress, Milano, Italy, pp. 5753–5758 (2011)
11. Preitl, S., Precup, R.E.: An extension of tuning relations after symmetrical optimum method for PI and PID controllers. Automatica 35, 1731–1736 (1999)
12. Preitl, Z.: Model-based design methods for speed control applications. Editura Politehnica, Timisoara (2008)
13. Peng, Y.Q., Luo, J., Zhuang, J.F., Wu, C.Q.: Model reference fuzzy adaptive PID control and its applications in typical industrial processes. In: Proceedings of IEEE International Conference on Automation and Logistics (ICAL 2008), Qingdao, China, pp. 896–901 (2008)
14. Preitl, Z., Levendovszky, T.: Computer aided design of two–degree–of–freedom (2DF) controllers. Scientific Bulletin of "Politehnica" University of Timisoara Romania. Transactions on Automatic Control and Computer Science 48(62), 70–75 (2003)

15. Visioli, A.: Fuzzy logic based set–point weight tuning of PID controllers. IEEE Trans. Syst. Man. Cybern. A Syst. Humans 29, 587–592 (1999)
16. Shu, S.Q., Ding, X.Y., Wu, W., Ren, H.Y.: Application of a self–tuning two degree of freedom PID controller based on fuzzy inference for PMSM. In: Proceedings of International Conference on Electrical Machines and Systems (ICEMS 2008), Wuhan, China, pp. 1629–1632 (2008)
17. 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)
18. 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)
19. Horowitz, I.M.: Synthesis of feedback systems. Academic Press, New York (1963)
20. 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)
21. Baranyi, P., Yam, Y., Varkonyi–Koczy, 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)
22. Skrjanc, I., Blazic, S., Matko, D.: Direct fuzzy model–reference adaptive control. Int. J. Intell. Syst. 17, 943–963 (2002)
23. Johanyak, Z.C.: A brief survey and comparison on various interpolation based fuzzy reasoning methods. Acta Polytechnica Hungarica 3, 91–105 (2006)
24. Fodor, J., Rudas, I.J.: On continuous triangular norms that are migrative. Fuzzy Sets Systems 158, 1692–1697 (2007)
25. Blazic, S., Skrjanc, I., Matko, D.: Globally stable direct fuzzy model reference adaptive control. Fuzzy Sets Systems 139, 3–33 (2003)
26. Mihailovic, B., Pap, E.: Asymmetric general Choquet integrals. Acta Polytechnica Hungarica 6, 161–173 (2009)
27. Vascak, J., Madarasz, L.: Adaptation of fuzzy cognitive maps – a comparison study. Acta Polytechnica Hungarica 7, 109–122 (2010)
28. Johanyak, Z.C.: Student evaluation based on fuzzy rule interpolation. Int. J. Artif. Intell. 5, 37–55 (2010)
29. Linda, O., Manic, M.: Uncertainty-robust design of interval type–2 fuzzy logic controller for delta parallel robot. IEEE Trans. Ind. Informat. 7, 661–670 (2011)
30. Stinean, A.I., Preitl, S., Precup, R.E., Pozna, C., Dragos, C.A., Radac, M.B.: Speed and position control of BLDC servo systems with low inertia. In: Proceedings of 2nd International Conference on Cognitive Infocomunications (CogInfoCom 2011), Budapest, Hungary, p. 8 (2011)
31. Stinean, A.I., Preitl, S., Precup, R.E., Dragos, C.A., Radac, M.B.: 2–DOF control solutions for BLDC–m drives. In: Proceedings of IEEE 9th International Symposium on Intelligent Systems and Informatics (SISY 2011), Subotica, Serbia, pp. 29–34 (2011)
32. Baldursson, S.: BLDC motor modelling and control – A Matlab/Simulink implementation. M.Sc. Thesis, Institutionen for Energi och Miljo, Goteborg, Sweden (2005)
33. Nasar, S.A., Boldea, I.: Electric drives, 2nd edn. CRC Press, Taylor and Francis, New York (2005)
34. Boldea, I.: Advanced electric drives. PhD courses (2010-2011), "Politehnica" University of Timisoara, Timisoara, Romania (2011)

35. 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)
36. ECP: Industrial emulator/servo trainer model 220 system, testbed for practical control training, Bell Canyon, CA, USA. Educational Control Products (2010)
37. Preitl, S., Precup, R.E., Dragos, C.A., Radac, M.B.: Tuning of 2–DOF fuzzy PI (D) controllers laboratory applications. In: Proceedings of 11th International Conference on Computational Intelligence and Informatics (CINTI 2010), Budapest, Hungary, pp. 237–242 (2010)
38. Horvath, L., Rudas, I.J.: Modelling and solving methods for engineers. Academic Press, Burlington (2004)
39. Vascak, J.: Navigation of mobile robots using potential fields and computational intelligence means. Acta Polytechnica Hungarica 4, 63–74 (2007)
40. Dankovic, B., Nikolic, S., Milojkovic, M., Jovanovic, Z.: A class of almost orthogonal filters. J. Circ. Syst. Comp. 18, 923–931 (2009)
41. Iglesias, J.A., Angelov, P., Ledezma, A., Sanchis, A.: Evolving classification of agents' behaviors: a general approach. Evolving Syst. 1, 161–171 (2010)
42. 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)
43. Linda, O., Manic, M.: Self-organizing fuzzy haptic teleoperation of mobile robot using sparse sonar data. IEEE Trans. Ind. Electron. 58, 3187–3195 (2011)
44. 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)
45. 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)
46. Obradovic, D., Konjovic, Z., Pap, E., Rudas, I.J.: Linear fuzzy space based road lane model and detection. Know. Based Syst. (2012), doi:10.1016/j.knosys.2012.01.002

Part IV
Applications in Medicine

Virtual Doctor System (VDS) and Ontology Based Reasoning for Medical Diagnosis

Hamido Fujita, Masaki Kurematsu, and Jun Hakura

Abstract. VDS is a system built as intelligent thinking support for assisting medical doctor in a hospital to do medical diagnosis based on the avatar of that doctor. The medical knowledge is also collected from the doctor based on his/her experience in diagnosis. The avatar construction is mimicking real doctor. The avatar interacts with patients through their voices, and other sensors to read patient mental state and physical state that are used in aligned manner to assess the patient sickness states through Bayesian network. The physical view is represented as physical ontology. The mental view is represented as mental ontology. These two ontologies aligned on medical knowledge for diagnosis and reasoning based on similarities computation. These two types of ontologies have been mapped and aligned for reasoning using a simple Bayesian Network for causal reasoning to find related query decision case based diagnosis collected from expert doctors. The system is implemented and tested. We have constructed an integrated computerized model which reflects a human diagnostician and through it; an integrated interaction between that model and the real human user (patient) is utilized for 1^{st} stage diagnosis purposes recalled as simple cases.

Keywords: Virtual doctor system, medical diagnoses, Bayesian network, avatar, expert doctor. decision making.

1 Introduction

Virtual Doctor System (VDS) is reported in [2][9][10] and is related to design avatar that resemble a real human doctor and act to interact with patient user to establish a diagnosis scenarios. The system outline is shown on Fig. 1.

Hamido Fujita · Masaki Kurematsu · Jun Hakura
Iwate Prefectuarl University,
Intelligent Software Systems Laboratory
152-52 Sugo, Takizawa, Iwate-gun, Iwate, 020-0173, Japan
e-mail: {issam,kure,hakura}@iwate-pu.ac.jp

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 197–214.
DOI: 10.1007/978-3-642-33959-2_11

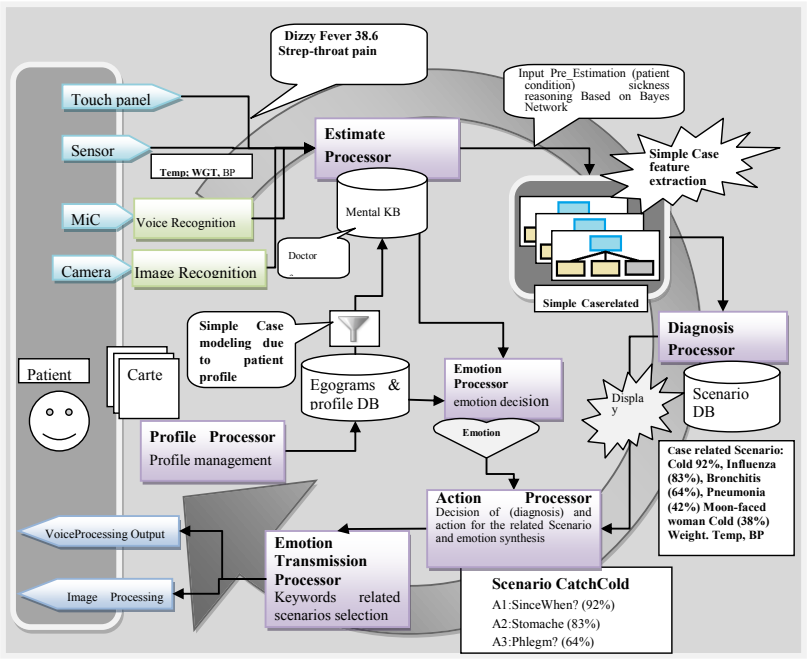


Fig. 1 The VDS system outline

We have created a related technology, reflecting the state of art on creating a program that resembles the user mental psychological behavior through a face, this concept we called it mental cloning [1] [2] [3] [10] [11]. The mental cloning is used to collect on the built avatar reflecting a real person, the animated real-time images created in real-time on this avatar resembling the emotional behavior of that person articulated through this avatar in the same manner the real person interact with certain world in similar invocation. This is represented by using that person ego state [3] [11].

In this paper the system is expanded to reflect on reasoning issue that the VDS uses to interact with human patient for medical diagnosis. The inter operability is represented by utilizing the medical diagnosis cases of medical doctor, and represented in machine executable fashion based on human patient interaction with virtual avatar resembling a real doctor. The Virtual Doctor System (VDS) is installed in a local hospital in Iwate-Japan where that doctor is regularly, practicing her medical diagnosis in real situation and environment. The avatar or VDS is working as a 1^{st} glance diagnosis to classify patients based on the criticality, emergence related to examination parameters and diagnosis scenarios outcome.

2 VDS System Outline

The medical scenarios used in medical diagnosis are defined on general guide lines formalization, and customization according to the subject doctor experience and related specialization in the course of medical practices. This system is to help the real doctor by filtering the outpatient (when they come to the hospital) who waiting to see the real doctor. The virtual doctor sees patients by interacting with them and issues a decision making for simple cases and non-simple cases categorical based analysis. The *simple medical case* in our context; is defined as the case that usually medical doctor reaches through medical diagnosis, and is considered by medical Doctor namely A, as a state that the outpatient can be recovered by taking a rest, or simple medical supplement, a case resulted from stress, heavy work or tiredness or else.

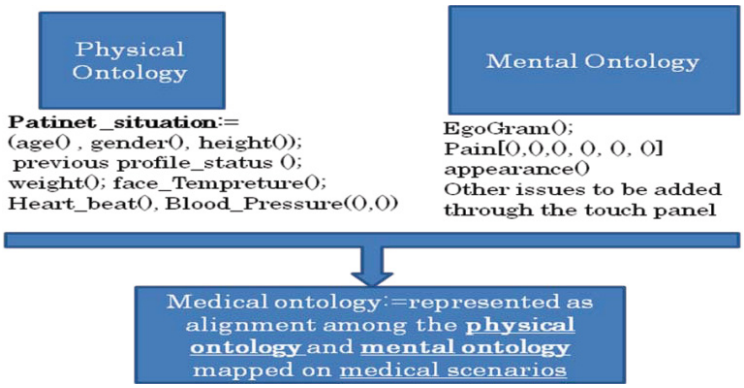


Fig. 2 Outline of VDS conceptual reasoning

The simple case treatment is in most cases is to ask the patient to rest and come back after few days if the recovery is not achieved or correlated physical (body) phenomenal sign is emerged, or stayed (not relieved). We have selected the*simple case* approach due to the following: We can test our system and its reasoning design framework. We use the system for helping the medical doctor to classify medical cases for outpatient based on criticality issues. Criticality issues are estimation of the outpatient sickness state. This is based on his/her mental and physical reasoning that is achieved (i.e., reached) collectively (inferred) by the VDS. We have constructed two concepts that the system uses to do reasoning. One is called *Physical Ontology* , and the other is called *Mental Ontology* . We define the concept of "pain" as part of mental ontology; we implement the system using Owl (Ontology Web Language). The two represented ontologies are merged to yield a third ontology called *Medical oncology*, representing the diagnosis processor in Fig. 1. The action processor is the decision making that use Bayesian Network, shown in Fig. 2, is utilized to calculate the Conditional Probability Table (CPT) that assist the action processor to select

the best scenarios that would readout the selected diagnosis fitting to the patient instance, in the same manner as the real doctor does. On [11] we appended several video that demonstrate how the VDS interact with the patient and how the system's camera eyeing on patient face to extract values to reason through on patient state. This link below has demo files and news available for download, you can reference to them.

http://www.somet.soft.iwate-pu.ac.jp/system_news/

3 Reasoning Framework

There are several parts related to the VDS interaction parts that have already reported in published work and are related to voice and face emotional recognition [4][6][11] and voice emotional recognition [7][14]. We have built a concept we call it as mental cloning [1] we could collect user emotion and mentality reflected through face and voice to understand the mental state of the user. These are used for creating output for the avatar and input data due to user emotional change (engagement) with the avatar [2]. The reasoning issues are not reported yet, and this paper is discussing the reasoning framework related to VDS application for diagnosis. We have restricted our diagnosis reported in this paper to *simple cases*. The context of the engagement is defined in advance (In this paper the context is medical diagnosis based on Doctor A). So collecting the user mental state is to have the system adapt to changes that would have the user be engaged with the system in a positive manner (forwarding interactive style of communication).

The avatar in our system that we built as experiential is virtually constructed to resemble real Doctor objectified for subjective human patient to interact with. We have a mask face model for real doctor practicing his/her medical provision in hospital in our town [2][1]. Such face (system) will interact with the user in emotional based manner [11]. The face would smile or else and act in emotional manner according to the context and engagement style of the user. The face mental background resembling an ego state reflected through the egogram resembled person (medical doctor) and represented in the system as a program [11]. We have studied this aspect and we created a program that can interact with the user using transactional analysis [3]. The face states are the primitive states that the system would select interactively according to the user engagement as cognitive state (shown in Fig. 3). The user ego state is also collected from the best match from the database based on what we called universal template. A set of egograms is stored in the system and indexed according to universal templates [3].

We evaluated these universal template based on experimenting them with Miyazawa Kenji avatar that experimented in A museum [11].

These stored classified egogram are to work as templates to test user ego states (emotion). User observed ego state is measured through a set of universal templates. The measurement of face parts movements are referenced (computed) to a indexed templates collected from many Japanese subject (people) contributed in our

experiment [3][6][7][11]. On [11] you can view (download to view) the movies showing how the system works and also for public news on the project, (in Japanese media).

The system would test the mental states of the user based on these egograms, and interact with the user based on instantiation of observed changes on the face parts collected due to emotional reasoning based engagement. Also, the same is done on the voice as well. The voice emotional features are examined to reflect the patient voice sound features like soaring throat related sound feature, or related expression to pain or else. The same also, is for expressing the dialogue with patient by doctors with emotional voice to patient synchronized with their mental situation.

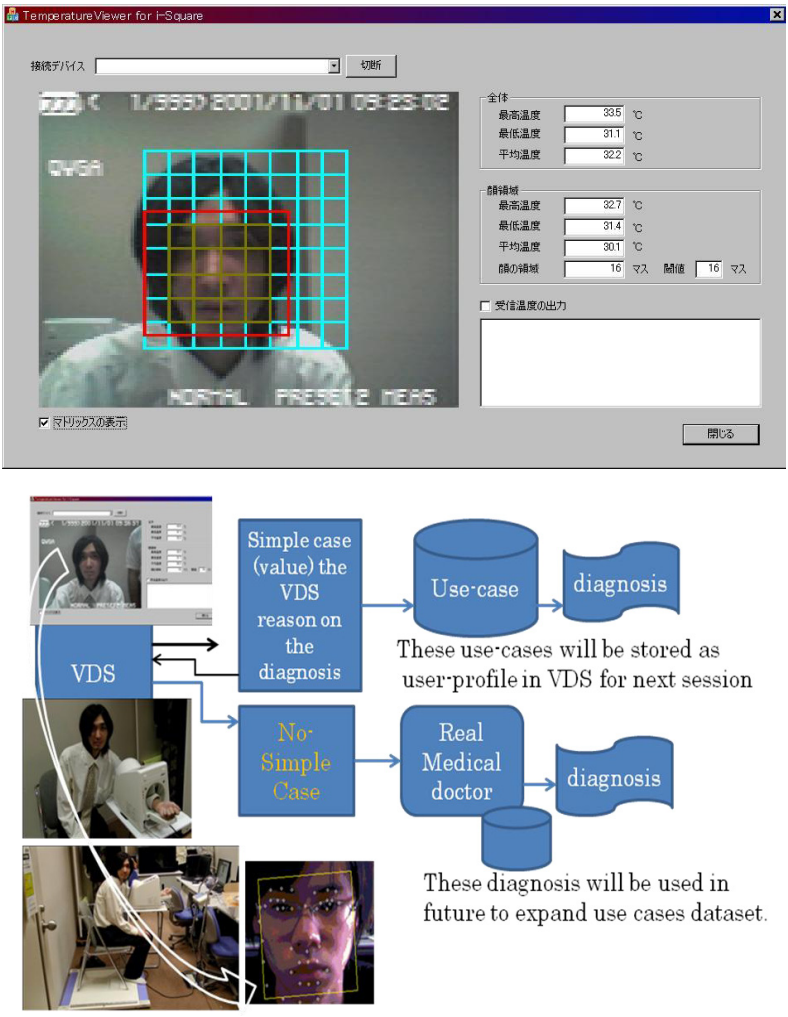


Fig. 3 The physical ontology and VDS related simple case

We have not presented patient voice emotional recognition or VDS generated emotional readout diagnosis in this paper [7], [14]. Taken all these issues into account we are reporting in this paper, using this established technology, we have examined and experimented reasoning related matter reflected to medical use-case provided by two medical doctors in Morioka city. We have built the system that ensemble a medical doctor interacting with the patient based on the framework mentioned above, to do diagnosis on patient at clinic or hospital in Morioka city, Japan. The system is working as a filter (sorter) to do the 1^{st} diagnosis based on provided medical guidelines specialized by these two nominated medical doctors working in that clinic. This is especially useful in Japanese local hospitals when patient usually wait for one to two hours to see the doctor (human physician). The system is assisting the hospital to set patient into simple cases (with category, and without physical observation) and no-simple cases, (need doctor observation). The main issue that reported in this paper are related to new findings that we have collected in relation to VDS is using the mental cloning concept and the avatar technology together to construct a copy of Medical Doctor (MD) avatar [9][10]. We have experimented two MDs'. These two hospitals' doctors' avatars are constructed [11]. The avatar face and voice with related diagnosis scenarios on "simple cases" have constructed. We have selected these two doctors based on the style of their diagnosis. As, Dr. A uses patient appearance in reasoning and diagnosis (with certain physical touch), while Dr. B uses egogram based certification to analyze the patient mental states and do diagnosis on patients' condition by navigating in these states, through specific scenarios and networked style of decision making. She integrates all these decision based on her experience, represented as decision network style. These two instances of MD style of reasoning (diagnosis) are examined and represented in the VDS system presented; based on provided instances of simple cases medical practices data.

We define what is the simple case, and what are the formal guidelines defining the simple case (Fig. 3). The relative customization of such *simple cases* due to the doctor experiences in diagnosis. It is those cases that the MD concludes as non critical symptoms, or symptoms that may need later on further observation, or situation that is not necessarily be recovered by medication or surgery or else. A relative medical advice, or supplementary medication supporting the medical case in hand, or/and appointment to come again to confirm the sustainability of such case, and are to be provided by MDs' for these called *simple case*.

Ontology can provide means to allow human to understand meanings of the elements and concepts or things in defining the problem, and also a means to reason on these classified items, based on such semantically based representation. Semantic technologies contribute to provide machine-executable metadata for reasoning purposes.

4 Conceptual Reasoning Based on Ontology

The conceptual reasoning framework is based on representing two types of ontologies, reflecting patient (user) physical conceptual status as seen by the VDS, as shown in Fig. 3. We have used a combination of probabilistic techniques, ontology representation and inference to determine the *simple case,* we defined in our context. The target of the system is identifying the patient case as simple, with weight. (Simple_?), where? is: High, medium and low, or not-simple. The decision taken as "not simple" means go to the next room to see the real doctor. The related technical reasoning is also sent to the real doctor.

We have defined two types of ontologies (as shown in Fig. 2).

Physical Ontology definition: PhO.

Mental Ontology Definition: MeO.

Each ontology represents causal relation articulated from physical view analysis, and mental view analysis.

Probabilistic model has been used to reference and infer to doctor diagnosis. This part is taking use of alignment of the two defined ontologies, and to do diagnosis based on probabilistic type calculation to compute values that would be used to make the decision related to special cases, this is modeled on Bayesian network.

Probabilistic model has been used to reference and infer to doctor diagnosis. This part is taking use of alignment of the two defined ontologies, and to do diagnosis based on probabilistic type calculation to compute values that would be used to make the decision related to special cases, modeled on Bayesian network.

Values represented as Meta model by the PhO and MeO, as types reflecting collected data from patient observation.

Ontology alignment is very challengeable problem and active research in ontology engineering [8].

5 Bayesian Network for Decision Making

Inference uses the Bayesian network (as belief network) aligned and reflected on these two ontologies. On Fig. 5 we can see the big arrow at the left side. This arrow is reflecting to collecting data through the user interface to articulate in collective manner on the PhO issue, aligned with the MeO for reasoning on medical ontology, to select the best scenarios through the diagnosis processor. User situation classify the meaning of metadata based on (gender, egogram, age, history) [2]. These are variables, acting as values that in a collective manner classify the medical scenarios articulated on the ontologies representation inference reflected on the two types of nominated MDs' Simple Cases. These values articulate to diagnosis in probabilistic manner to reflect on the aligned mental view and physical view. The both are aligned to articulate to the status of the outpatient in probabilistic manner; and to reason on probabilistic combination of symptoms modeled as belief networks; that is used to find the best value related remedy or the cause of anomaly.

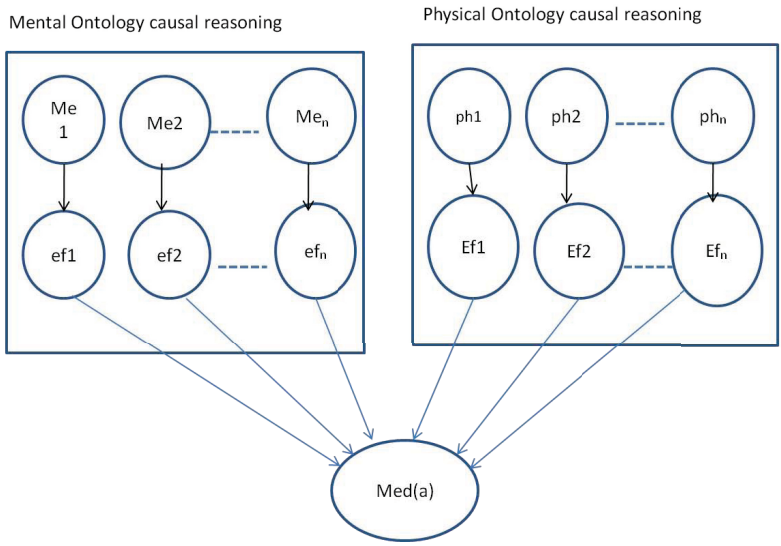


Fig. 4 The Bayesian network concept in our model

The history is related to last state patient state. If the last state is ”simple,” then we carried on. If not ”simple” then continue with extracting data and send these to real doctor, for real-diagnosis.

The variables are those values defined by the PhO and MeO views. These variables are any values between 0 and 1.

These values are representing the temperature in relation to threshold values (representing normal situation articulated on user situation). The total weight total of these values should be one.

Symptoms type Physical: Are those symptoms observed on the patient by devices or previous documented data (Fig. 5). In our system, we have the patient be seated on a chair with three types of devices that read: The body weight, temperature distribution on the face, and blood pressure. There are also other data that can be collected from previous history or document, referencing to previous physical state and articulates on the new state.

Symptoms type Mental: These are the observed behavioral patterns on the patient face, articulated through templates to reflect to the mental state of the patient, if she/he is in a pain or a sort of situation.

These above two situations each are reflected and represented on ontology reflecting the medical ontology specified by the two medical doctors and specialized by the difference in their ontology in patient diagnosis. The Symptoms reflected on Physical ontology are those reflected on mental ontology are mutually independent. The medical ontology represents the conceptual (abstract) view of medical diagnosis. The view is specialized by the doctor type, reflected as a model spe cialization,

on the usage of the PhO and MeO in diagnosis. The simple case is defined in conceptual view and generalized form, the specialization due to the type definition of simple cases according to the doctor experiences, and is represented on medical ontology. The style of reasoning diagnosis is also relative to the doctor diagnosis ontology (as a specialization to medical ontology).

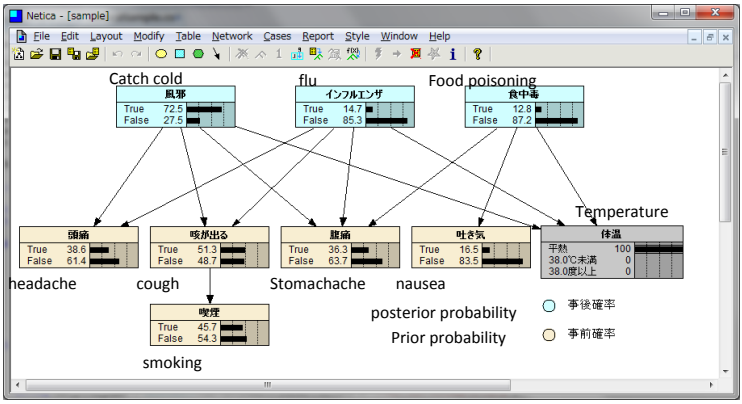


Fig. 5 Applying netica in MD provided simple case

The variables outcome would infer to the medical cases and invoke certain scenarios. These scenarios are explained to the outpatient as question or comments, expressed by the VDS in the same manner the real doctor does.

Ontology is a set of concepts related to set of attributes on to each concept, and set of relationship among these concepts. All variables values (relationship or attributes) are computed based on mental and physical observation in the model. Variables are computed and collected by initiating scenarios with outpatient to collect these causes' values (probabilistic) for decision making.

With these medical doctors help we have established several cases resembling different types of configuration of variables. These configurations can resemble different types of simple cases. We could create more complicated configuration in the same manner, however this would make the Bayesian network more complicated. At the moment our interest is to build the system and test it in these two hospitals. There are several issues that need to be examined from real experiment. Not only the inference complexity, but the practicality on having such system in medical practices for simple case scenarios needs also, to be investigated.

The probability weight of each symptom is calculated with weights representing the importance of that variable. All variables in the PhO, MeO are classified weight reflecting the structure of medical knowledge. The knowledge base is categorized and classified due to importance weight of these variables as part of the two defined ontologies. The evaluation is not numerical but subjectively qualitative and oracle. The PhO, is mapped with related aspect collected from the input to related aspect

from the MeO, The alignment is articulated on MedO through causal relation based on Bayesian Network which is proved to be useful in reasoning on imperfect knowledge like these related to medical diagnosis on patient, with imperfect knowledge. The result would produce a keyword (or a set of data) or a statement as query that reason through the reasoner for the related scenarios that can reflect a nest of related constraints. The medical alignment ontology is a conceptual view of simple medical case that is defined in terms of PhO and MeO aligned on medical diagnosis. The specialization is due to the type definition of simple cases implied to the doctor experiences. The style of reasoning diagnosis is also, relative to the doctor diagnosis ontology.

A Bayesian network is a graphical model for probabilistic relationships among a set of variables, represented using a directed acyclic graph, that arrows represent causal influence among its nodes. The Noisy-OR network [12] has the assumption that the made model is of causal independence among the modeled causes and their common effect. The word noisy reflects that the interaction among the causes and the effect is not deterministic, so it is not possible to capture all the possible causes of an effect.

We have used Netica[15] to construct the BN. Netica allows network construction and parameter learning from data. We derived the parameters for conditional probability values from medical studies related to simple cases observations, to model causal reasoning in medical ontology.

The causal independence needs to be facilitated through the conditional probability theory $P(A_i \backslash f(A_i))$.

As mentioned above a diagnosis is instantiated from physical variables, (ph1, ph2,$\cdots$phn) and mental variables (me1, me2,$\cdots$men). These variables are mutual independent. There are also effect that articulated from each variables reflected from corresponding ontology, and all the summation of these effect would lead to med ical scenario conclusion. The medical scenario also, leads to other set of physical and mental variables that again in collective manner, may lead to other medical decision it is an integrative causal reasoning based on Bayesian network.

$$P[(med(a) \backslash me_1, \ldots, me_n, ph_1, \ldots, ph_n)$$
$$= \sum_i^n P(med(a) \backslash me_i) \sum_i^n P(med(a) \backslash ph_i) \prod_i^n P(ef_i \backslash me_i) \prod_i^n P(Ef_i \backslash ph_i),$$

where Ef or EF are Boolean value as either true or False mapped to the Med(a). Also these ef(i) and Ef(i) are conditionally independence between the both, as the signs (symptoms) resulted from the physical effect and mental effect together are due to certain disorder. These are not necessarily correlated by their relation on the medical diagnosis reflected in the above formula implied to establish decision making for causal reasoning on diagnosis. For example,*mild* blood pressure, with high_mild disgust, could lead to mild stress. It is a type of Simple case as reflected from MDr. B. We have noticed that; this is a sort of belief network for a noisy-OR type causal reasoning in context-specific independence [12]. The noisy-OR (ef(i)) is like a regular OR function, all its parents are binary values in [0,1]. As we have

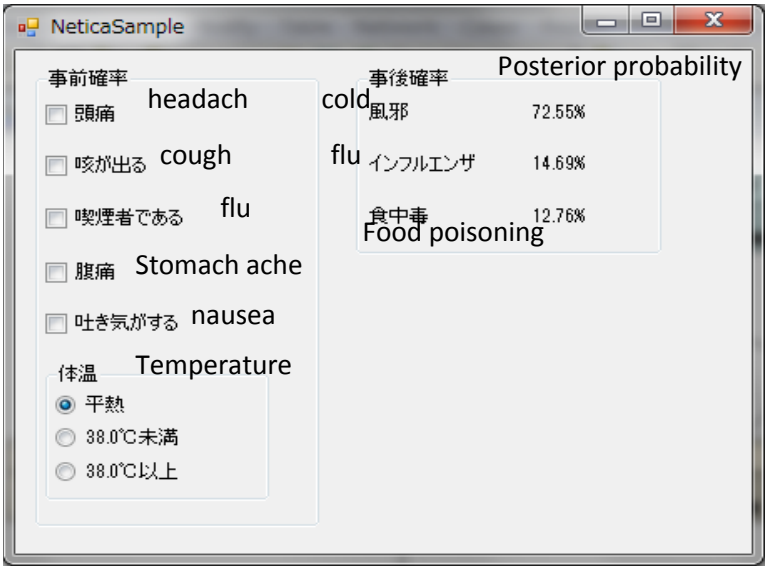


Fig. 6 Netica based API application for the simple case in Fig. 5

the PhO and MeO symptoms related nodes are causal independence related to ef(i), therefore, $P[(med(a)\backslash me_1,\ldots,me_n,ph_1,\ldots,ph_n)$ is represented by terms of noisy hidden variables $ef(i)$.

Therefore we calculate the conditional probability of all states. The nodes in Fig. 5, (believe networks concept based on the aligned ontology) are of two types: discrete nodes representing either symptoms observable or not. It is a binary value. There is continuous node representing values like temperature or weight (i.e., PhO), and these values are represented as conditional probability distribution. In other words, the temperature, for example is converted to high in relation to sym p toms or low. This is possible by storing conditional mean and variance in each decision node namely the MeO(i) and phO(i).

Recall again to Fig. 2, as resembles the type of reasoning to the simple case causal based reasoning. The causal network calculates to reach the appropriate de cision through the different dataset provided by these two doctors.

The analysis using the above formula is to calculate the probability of medical diagnosis based on collecting the effect (ef(i) collected from MeO) and (Ef(i) collected from PhO), respectively. The collected computed probability will be according to threshold values specifying the patient as either simple case or not. For simple case, the system would advise the patient on his/her condition using a set of Simple cases categorized scenarios organized due to the physical and mental data, For example; temperature combination values (PhO) and disgust values (MeO). The conclusion will be presented to the patient as readout scenarios, toned based on user situation, in the same manner as the real doctor does. For any other case, concluded

as not simple the system may advise the patient to move to the next door to see the real doctor. The diagnosis scenarios and copy of it will be transferred as a file to the real doctor's computer. We have used Netica application [15] to build the Bayesian network reasoning engine. In addition, we use Netica Java application (API) to do the application that interact and reason on values for diagnosis cascaded with the OWL reasoning engine. The implementation has done using JAVA with Netica connected with Jess reasoner. The program resembling the two ontologies based inference build the Bayesian Network in the memory for decision making and reflect this in the reasoner such that to have the diagnosis processor (Fig. 1) to select the appropriate scenarios according to Medical doctor position participating in the analysis.

6 Case Based Evaluations

We have trained Bayesian network on sample mapping related to "simple case: denoted as SC" such that we produce merged Bayesian network resembling the medical ontology with appropriate CPT that reflecting the MD estimated diagnosis.. We have trained the network with 53 patients data based on classification of MeO and PhO using measures and mapping rules extracted from real case data to produce the medical ontology with high probability CPT. The simple case with probabilistic decision is as in Fig. 6, showing the patient select symptoms on the left, and the reasoner responses on the right. These data and related reasoning are needed for diagnosis processor (Fig. 1).

The system is tested by MD_A who provided diagnosis scenarios used to interact with the system, based on the ontology related to SC provided by the medical 6 doctors' VDS committee (MD_A is among them). Tuning has been carried on to fit into the system that the MD_A usually does in her diagnosis practices. Table 3 indicates the outcome of this test.

Patients come to the hospital and fill a forms of questionnaire that specify their egogram which is done by touch panel monitor. These egograms are stored in patient profile database associated with properties and restrictions according to the patient state as part of patient concept. Physical data on patient states are automatically collected,. The reasoner creates the network corresponding to the two ontologies related to the simple case, and accordingly, creates the classification as specialization for the simple case for such patient. The system computes through the simple case the CPT at the 1^{st} pass then another question that articulates the special case be given to construct the mapping classification and the network to compute the 2^{nd} pass using again the CPT for the 2^{nd} pass simple case scenario and so on. The tested patients are volunteers selected with the help of D_A.

Tested persons (acting as patient) are given a short introductory on the system purpose and what is the SC we are trying to diagnosis through them, acting as patients. We asked them to behave normal as if they are being examined by the MD_A.

The confidence in selecting the related set of inter_concepts is related to associating rules mining in the related ontology, as either the MeO, or the PheO represented

as Bayesian Network. Related concepts can produce cluster that is reflected as alignment and pairing of cluster reflected from these two type ontologies. Feature can be computed by information gain reflected through patient profile feature and collected from the physical and mental attributes aggregation.

$$Entropy(f) = -\sum_{i=0}^{n} p(i\backslash feature) log_2 p(i\backslash feature)$$

Feature reflected on either type of ontology, $p(i|feature)$ is the probability for the *i-type* user related feature *feature*. O_c is the ontology as either MeO or PheO related set of coherent set concepts related to the feature *feature* and patient type *i*.. Concepts gathered in relation to feature are weighted according to degree of entropy and the similarity of concepts calculated. The higher the weight is the better concepts reflecting the patient related diagnosis as collection of concepts. Concepts are aggregated to two types, mental concepts and physical related concepts. Each concept is an aggregation to patient feature, patient patterns, and patient p related details can be referenced to [16]. The similarity is computed as weighted nearest neighbor [17]. Concepts are class with related feature that is reflected as query vector. The set of concepts C1 is aligned with query keywords on other concepts class (C2).

$$Similarity_{feature}(C_1, C_2) / \sum_{i=1}^{n} sim_{feature}(C_{1i}, C_{2i})$$

$Similarity_{feature}(C_1, C_2)$ is normalized value. The semantic similarity measurement is utilized to produce weighted content information by taking similarity among concepts related to ontology (PhO, or MeO), reflected by features collected from input as shown in Fig. 3. a_{ji} is the *j*-the medical case with*i* feature.

MeO related concepts and Physical related concepts are aligned (Pair_wised) to using Bayesian Network. The alignment paired set of concepts is reflected as asset of key words inquiry to search for best fit for medical diagnosis case using Ecludian measure.

$$similarity(a_1, a_2) = \sum_{i=1}^{n} w_i * sim(a_{n1}, a_{n2}) / \sum_{i=1}^{n} w_i$$

a_{n1} is medical case, 1 for feature i of weight w_i , and a_{n2} medical case 2, for feature i of weight w_i. The similarity is computed on related medical case that is pairwise collection of merged weighted feature reflected from medical knowledge retrieving (deriving) related similar cases using k-nearest neighbor. It is a vector that can be treated as support vector machine (SVM) to derive similar concepts medical cases. We have used weighted average aggregation operation tool called as PA introduced by Yager [18].

$$PA(a_1, a_2, \ldots, a_n) = \sum_{i=1}^{n} (1 + T(a_i)) a_i / \sum_{i=1}^{n} (1 + T(a_i))$$

Where $T(a_i) = \sum_{j=1, j\neq i}^{n} Sup(a_i, a_j)$, $Sup(a,b)$ is the support for a from b satisfying: $Sup(a,b) \in [0,1]$, The more similarity, the closer the two values are, the more support each other. Yager [18] developed order weight operator (OWA) $OWA(a_1, a_2, \ldots, a_n) = \sum_{i=1}^{n} w_i a_{in(i)}$ where in is an indexing function such that in(i) is the index of the i-th largest attribute, w_i is weight of attribute. Xu[19] introduced power order weighted geometric operator (POW G) such that $POWG(a_1, a_2, \ldots, a_n) = \prod_{i=1}^{n} a_{in(i)}$. Using these operators we computed the similarity index to aggregate attributes to the medical argument in the knowledge base. The attributes related to decision making on patient having temperature is related to the POWG of attributes to physical ontology aggregation and mental ontology aggregation. The weight is representing the importance of the attributes in diagnosis as specified by the medial doctor. The values of the attributes are either crisp value of fuzzy values represented in linguistic terms, e.g., High, Low etc. The crisp values are related to arguments that have values either higher or lower threshold values. The knowledge representation is a combination of argument with attributes that are combination of crisp fuzzy set or fuzzy values related to symptoms defined as membership function. The membership functions (Symptoms) are categorized to two groups: Mental group and physical group membership functions. The threshold values are used to set argument crisp fuzzy values.

The decision making is carried out based on normalizing the attributes related to either groups, then all attributes will be transformed into fuzzy numbers, then all attributes related to either ontology using weighted geometric sum. Then at last optimize the sum to have the collective values of all attributes.

The system is installed at a Hospital in a room that approved by the MD_A [12]. Patients arriving to the hospital are requested by the MD_A voluntary to be also, examined by our system. The request is granted after they are examined by the MD_A, according to the case based SC scenarios that she provided us with. Therefore, the patients who are selected by the MD_A to be tested by VDS are those who belong to simple case category as examined and concluded by MD_A.

Before we have them entering the VDS's room (space) we give them a short introductory on the system and its usage to increase their awareness.

We want to measure the system performance and validity. What is the accuracy that the system concludes SC for an actual SC? What is the VDS fitness to the scale of a medical problem as it is carried in a hospital? Is the technical aspect reflected in the system is appropriate from processing time and practicality views. We have asked patients who participated in the experiment to fill a sheet of paper answering questioners on the system performance that they have tested to diag nose them. The questioners are related to how much the system is practical for their case, are they satisfied with it as medical service provider, and other types of questioners.

The patients participated in our experiment during August December 2010 carried out at the hospital where MD_A is practicing the medical diagnosis. The ex periments started on August which is a season of hot temperature and people who

have a change in their life (normal life routines) style due to holidays. Such change may cause a sick or stress lead to side affect or disorder. This year especially was special for our region as well as other region in Japan, as it was hotter than usual years [21]. So we could examine our system based on provided simple case related to such flu or anomaly.

Table 1, represents the data collected. In this table the 6_MDs are related to experiment based on cases provided by 6 medical doctors' evaluation committee. The data 15/4 for example, represents 15 patients of Class_1_F (20s' years old Female class patients) who accepted to use the experiments, for diagnosis at the time on the period August and September (two months), and among them there 4 patients who are diagnosed as simple case. Note that in Table:3 we have four classes of patients related to gender and age. F: represent female patient and M: represents Male patients. Class_1: is for 20's years old patients , Class_2 is for 30's old patients, Class_3 is for 40's years old patients and Class_4 is for 50's and above years old patients. The test is done for cases according to two months period. The diagnosis compared to actual diagnosis is done by the medical doctors MD_A, for comparison. The number in brackets ( ) indicates those patients who are diagnosed by the system and also by the MD_A as having simple cases (for validation purpose) . This (number) is the counting of patients overlapped in the both diagnosis. This is reflecting the validity of the system. We think this is a suitable and sufficient for the objective that we have set; however, still the precision in regard to the total performance is not complete yet.

The experiment and discussion with MD_A, reveals on issues reflected from old female patients (Class_4_F) who have mental views (e.g., compulsion related issues) that are no covered yet in the CBR in complete manner. We need to enhance the mental views reflecting observations collected from such instances in differentiation manner among other cases to increase the reasoning prediction.

The outcome is based on refining the simple case diagnosis scenarios. Some of the errors are related to touch panel problems as some patients pushing on several times and the system cannot recognize such style data re-entering. Among these testing cases there are technical problem related cases, like those established due to patient's non-predictable behaviors or responses, (e.g., standing in the middle of the engagement session, or looking left and right). The patients, who are above 65 years old, have responded to the system scenarios with overstated responses not like those usually given to human doctor. The patients participated are of age between 20s' to 50s', most of them have background in using computers. They have interest in positive manner to the experiment on using VDS. All patients have simple introductory guidance on the appropriate way to interact with the system. This guidance helped us to have better interaction between patient and VDS and increase the patient awareness.

Due to the experiment statistical analysis the system could evaluate the simple cases for patient with hit ratio of average 70%. This contributes efficiently to utilize Medical Doctors working time for diagnosis. MD_A expresses a good sign and optimism on saving time that can be achieved by filtering patients' medical cases. The performance can predicated through Table 1, (1) The system carry out diagnosis on

behalf of the doctor for Simple case (2) All diagnosis done by the VDS, (tested as non-simple) and its related data is transferred to the MD monitor desktop for further patient diagnosis. These two merits are for the sake of Doctor and hospital resources time saving.

Moreover, Patient acceptance is essential part in the evaluation procedure. What we have discovered as we tested our system on the period between August and December 2010; that the patients gradually has an increasing trust value (i.e., safeness) in the VDS when he/she is diagnosed again by the doctor and similar compiling results are reached. These validation procedures create a sort of reputation state in the neighborhood on having this hospital using VDS. System diagnosis reputation has increased (i.e., patient acceptance) by the time we have the system running in the hospital for constructive correct diagnosis practices that are having sound evaluation by the MDs.

Table 1 Case study related evaluation

Patient Class	August/Sept (2010)		Oct/Nov (2010)		Dec/Jan (2010)	
	6_MDs	MD_A	6_MDs	MD_A	6_MDs	MD_A
Class_1_F	15/4	15/5 (4)	11/4	11/6 (3)	21/8	21/9 (7)
Class_1_M	11/3	11/4 (2)	8/3	8/5 (2)	15/5	15/7 (4)
Class_2_F	21/11	21/10 (9)	17/10	17/12 (8)	27/8	27/10 (7)
Class_2_M	15/2	15/4 (3)	12/5	12/7 (6)	22/7	22/11 (7)
Class_3_F	31/8	31/11 (8)	25/8	25/12 (5)	38/13	38/12 (7)
Class_3_M	22/3	22/5 (1)	19/7	19/10 (7)	29/9	29/11 (8)
Class_4_F	58/28	58/31 (15)	43/14	43/17 (13)	55/23	55/25 (14)
Class_4_M	44/13	44/18 (11)	36/7	36/14 (8)	48/15	48/16 (11)

Medical applications have a particular requirement that justify the development of certain CBR , for example, [22] has presented a CBR for cancer diagnosis for ex pert users.

The CBR should have a sufficient utilization articulated to the user needs more than the expert interpretation to the medical cases [22]. Having a harmonized engagement between the patients; reflecting a typical instantiation of medical cases, and the VDS. This participates in searching the suitable fitness on related scenarios. This is essential for a successful reasoning situating cases. The extract ion is best on collected invariants from patient in interactive manner mimicking a real doctor negotiation style for best medical interface usage. Related contextual knowledge can be helpful in efficient dynamic fitness for diagnosis based on medical cases related embedment by deploying semantic (context) in reasoning.

7 Conclusion

The project reported here is one step to design high definition intelligent application for medical diagnosis based on patient interaction to instantiate his/her physical and emotional attributes (symptoms) to reason in medical knowledge base to compute the decision making using medical knowledge. The system is constructed and tested in a hospital.

The technology developed through VDS is to implement the system into mobile services with join collaboration with mobile company. This is to have all services be integrated as mobile service provider.

The reasoning is based on interpreting the knowledge of doctor based on similarity concept using Bayesian network that is implemented for decision making. The system also has used fuzzy reasoning techniques for decision making. Idea on using combination of fuzzy system is presented in [20]. The results on using fuzzy decision are to be reported in another contribution.

Acknowledgements. This research is supported by the Ministry of Internal Affairs and Communications of Japan under the Strategic Information and Communications R&D Promotion Programme (SCOPE). We would like to give our gratitude to the Medical Doctors Committee board of VDS of SCOPE project as the medical application board advisory of this research, who are providing experience and advises on their medical analysis on patients diagnosis, and simple cases outline scenarios.

References

1. Fujita, H., Hakura, J., Kurematsu, M.: Intelligent human interface based on mentalcloning-based software. International Journal on Knowledge-Based Systems 22(3), 216–234 (2009)
2. Fujita, H., Hakura, J., Kurematsu, M.: Virtual Doctor System (VDS): Medical Decision Reasoning Based on Physical and Mental Ontologies. In: García-Pedrajas, N., Herrera, F., Fyfe, C., Benítez, J.M., Ali, M. (eds.) IEA/AIE 2010, Part II. LNCS, vol. 6097, Springer, Heidelberg (2010)
3. Fujita, H., Hakura, J., Kurematsu, M., Chida, S., Arakawa, Y.: Empirical based Techniques for Human Cognitive Interaction Analysis: Universal Template Design. In: The 7th New Trends in Software Methodologies, Tools and Techniques (Proceedings of SoMeT 2008), pp. 257–277. IOS press (2008) ISBN: 978-1-158603-916-5
4. Fujita, H., Hakura, J., Kurematsu, M.: Virtual Doctor System (VDS): Framework on Reasoning issues. In: Proceedings of the 9th New Trends in Software Methodologies, Tools and Techniques (Proceedings of SoMeT 2010). Frontiers in Artificial Intelligence and Applications, vol. 217, pp. 481–489. IOS Press (2010) ISBN 978-1-60750-628-7
5. Hakura, J., Kurematsu, M., Fujita, H.: An Exploration toward Emotion Estimation from Facial Expressions for Systems with Quasi-Personality. International Journal of Circuits, Systems and Signal Processing 1(2), 137–144 (2008)
6. Hakura, J., Kurematsu, M., Fujita, H.: Facial Expression Invariants for Estimating Mental States of Person. In: New Trends in Software Methodologies, Tools and Techniques (SoMeT 2009). Frontiers in Artificial Intelligence and application series, vol. 199, pp. 978–971. IOS press (2009) ISBN: 978-1-60750-049-0

7. Kurematsu, M., Ohashi, M., Kinoshita, O., Hakura, J., And Fujita, H.: An Approach to implement Listeners Estimate Emotion in Speech. In: New Trends in Software Methodologies, Tools and Techniques (SoMeT 2009). Frontiers in Artificial Intelligence and application series, vol. 199, pp. 978–971. IOS press (2009) ISBN: 978-1-60750-049-0
8. Kalfoglou, Y.: Scholermmer: Ontology Mapping: the state of the art. The Knowledge Engineering Review (2003)
9. Fujita, H., Hakura, J., Kurematsu, M.: Virtual Medical Doctor Systems: Status progress report on Virtual Medical Doctor (VMD). In: Proceedings of Third International Conference on Health Informatics, Valencia, Spain, pp. 38–45 (2010)
10. Fujita, H., Hakura, J., Kurematsu, M.: Multiviews Ontologies Alignment for Medical basedReasoning. In: 11th IEEE International Symposium on Computational Intelligence and Informatics (CINTI 2010), pp. 15–23 (November 2010)
11. (1) http://www.somet.soft.iwate-pu.ac.jp/system_news/News_2009.MPG
(2) http://www.somet.soft.iwate-pu.ac.jp/system_news/System_flow_operation.wmv
(3) http://www.somet.soft.iwate-pu.ac.jp/system_news/VDS_Sample2.mpg
(4) http://www.somet.soft.iwate-pu.ac.jp/system_news/VDS_sample.wmv
12. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988)
13. Zhonil, D.: Bayse OWL: Probabilistic framework for Semantic Web. Ph.D thesis, University of Maryland, Baltimore (December 2005)
14. Kurematsu, K., Chiba, H., Hakura, J., Fujita, H.: A Framework of Emotional Speech Synthesize Using a Chord and a Scale. In: New Trends in Software Methodologies, Tools and Techniques (SoMeT 2010), vol. 217. IOS press (2010) ISBN: 978-1-60750-628-7
15. Netica software packaga, NORSYS Software Corporation http://www.norsys.com/
16. Fujita, H., Kurematsuand, M., Hakura, J.: Virtual Doctor System (VDS): Aspects on Reasoning Issues. In: New Trends in Software Methodologies, Tools and Techniques (SoMeT 2011), pp. 293–304. IOS Press, Saint Petersburg (2011)
17. Wang, Y., Xu, X., Zhao, H., Hua, Z.: Semi-supervised learning based on nearest neighbor rule and cut edges. Knowledge-Based Systems 23(6), 547–554 (2010)
18. Yager, R.R.: The power average operator. IEEE Trans. Syst. Man Cybern. A. Syst. Humans 31, 724–731 (2001)
19. Xu, Z.S., Yager, Z.S., Power-geometric, R.R.: operators and their use in group decision making. IEEE Trans. Fuzzy System 18, 94–105 (2010)
20. Fujita, H., Rudas, I., Fodor, J., Kurematsu, M., Hakura, J.: Fuzzy Reasoning Decision Making on Multiviews Fuzzy Ontologies Alignment. In: IEEE 10th Jubilee International Symposium on Applied Machine Intelligence and Informatics, SAMI (2012)
21. http://search.japantimes.co.jp/cgi-bin/nn20100831a6.html (August 31, 2010) or a copy of it can be accessed at, http://www.somet.soft.iwate-pu.ac.jp/system_news/nn20100831a6.html
22. López, B., et al.: eXiT*CBR: A framework for case-based medical diagnosis development and experimentation. Artificial Intelligence in Medicine (October 2010) (in press), doi:10.1016/j.artmed.2010.09.002

Attention and Emotion Based Adaption of Dialog Systems

Sebastian Hommel, Ahmad Rabie, and Uwe Handmann

Abstract. In this work methods are described, which are used for an individual adaption of a dialog system. Anyway, an automatic real-time capable visual user attention estimation for a face to face human machine interaction is described. Furthermore, an emotion estimation is presented, which combines a visual and an acoustic method. Both, the attention estimation and the visual emotion estimation based on *Active Appearance Models* (AAMs). Certainly, for the attention estimation *Multilayer Perceptrons* (MLPs) are used to map the *Active Appearance Parameters* (AAM-Parameters) onto the current head pose. Afterwards, the chronology of the head poses is classified as attention or inattention. In the visual emotion estimation the AAM-Parameter will be classified by a *Support-Vector-Machine* (SVM). The acoustic emotion estimation also use a SVM to classifies emotion related audio signal features into the 5 basis emotions (neutral, happy, sad, anger, surprise). Afterward, a *Bayes network* is used to combine the results of the visual and the acoustic estimation in the decision level. The visual attention estimation as well as the emotion estimation will be used in service robotic to allow a more natural and human like dialog. Furthermore, the human head pose is very efficient interpreted as head nodding or shaking by the use of adaptive statistical moments. Especially, the head movement of many demented people are restricted, so they often only use their eyes to look around. For that reason, this work examine a simple gaze estimation with the help of an ordinary webcam. Moreover, a full body user re-identification method is described, which allows an individual state estimation of several people for hight dynamic situations. In this work an appearance based method is described, which allows a fast people re-identification over a short time span to allow the usage of individual parameter.

Keywords: Multilayer Perceptron, Bayes network, active appearance model, support-vector-machine.

Sebastian Hommel · Ahmad Rabie · Uwe Handmann
University of Applied Sciences Ruhr West,
Computer Science Institute, Tannenstr. 43, 46240 Bottrop, Germany
email: {sebastian.hommel,ahmad.rabie,
uwe.handmann}@hs-ruhrwest.de

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 215–235.
DOI: 10.1007/978-3-642-33959-2_12

1 Introduction

Due to the growing occurrence of service robots more and more unexperienced and non-instructed users are getting in touch with service robots. Therefore a lot of effort has been spent on enabling a natural human-robot dialog in service robotics during the last years. Nowadays, service systems like shopping robots or ticket machines are established in our society. Furthermore, service systems are getting more and more important in home environment. This ranges from robotic animals for amusement to service robots which help with housework, scheduling, home health care etc.. Especially for the acceptance of these systems they must be easy to use, since non-instructed users should be able to operate these systems. The most intuitive kind to interact with a technical system is the human like communication. Essential parts in human like communication is to know the emotional state of the dialog partner. For example, in tutoring systems or computer games, knowing about the user's feeling of boredom, frustration or happiness can increase learning success or fun in the game. Especially in human-robot interaction, affective reactions of the robot, following the recognition of the user's emotional state, can make the interaction more natural and human-like.

A further helpful information, for a user interaction is to know whether the dialog partner is attentive to the service system or to any other [9]. There are many methods to estimate the user attention, like full body movements or the used attention estimation which based on the direction of view. To estimate the emotion of the dialog partner, it is possible to interpret the facial shape and texture, the voice and some times full body movements. In this work an emotion estimation method is used, which combines the cues of facial expressions and audio signal information.

To estimate the user attention and the facial expression in a 2D image, a head description is needed which model the head of the current user. Two *Active Appearance Models* (AAMs) are used to realize this head models in real time. *Active Appearance Models* have been established to characterize non-rigid objects, like human heads, and can be used to analyze the user's state based on visual features. Therefore, the parameters of the AAM are adapted, so that the model fits to the current face in shape and appearance.

Especially in home health care for demented people, often it is not possible to use the head pose for attention estimation, because many demented people are restricted in their head movements. These people mostly look around by moving merely their eyes. To allow a visual attention estimation in this case, a simple eye tracker is considered which operate with an ordinary webcam. This eye tracker based on the eye position determined by an AAM, too. Further essential informations for a gentler and more natural dialog, which can be estimated with the help of the head pose are head nodding and shaking. Afterwards, the head nodding and shaking can be interpreted as Yes or No to allow a simple nonverbal gestural answering.

To classify the chronology of the head poses to attention or inattention, an adaptive variance is used in this work. To classify the chronology of the head poses to head nodding, shaking or others, an adaptive excess kurtosis is used.

For the estimation of users state it is often helpful to use an individual reference, like the *Individual Mean Face* which is described in [10]. Especially in home environment the number of interaction partners is limited and the most interaction partners come again. So it is helpful to use a former estimated reference. For this reason, a user identification is needed. A full body method for identification is described in this work. This method interprets the full body appearance which is used for a re-identification in a brief span.

2 Related Work

The work comprises the tasks of extracting the visual focus, head gesture classification as well as emotion estimation. All of these tasks require information about the user's head. A wide variety of methods with different kinds of feature extraction and classification approaches can be found in the literature. In the following, we give a brief overview of different methods, which have been applied for the specific tasks.

2.1 Visual Focus of Attention

Basically, a person's visual focus is determined by eye gaze. However, the proposed systems in the literature require high resolution of the eyes [23]. Nevertheless, the head pose can be regarded as a low pass filtered eye gaze and therefore the head pose can be utilized to get information about the visual focus [25]. Hidden Markov Models are a very common way to extract the focus of attention from a sequence of head poses [25, 22, 1]. However, the mentioned methods try to extract a focus of attention in terms of certain objects or persons, which lies not in the scope of this paper.

2.2 Head Gesture Recognition

Known approaches for head gesture recognition are quite similar to visual attention estimation. Again, the head pose is extracted and evaluated over time. A common way to enable the time based evaluation is to apply Neural Networks as shown in [12]. Furthermore, SVM Classification as utilized in [14] or Hidden Markov Models [13] can be applied for head gesture classification. Nevertheless, the proposed methods are also not able to learn head gestures online and therefore are not appropriate to learn new head gesture semantics during the human-robot dialog.

2.3 Audio-Visual Emotion Recognition

Joining several modalities in a single multimodal emotion recognition system could be achieved in several fusion levels, which will be detailed discussed in Sec 4.3. Paleari and Lisetti proposed a general framework for multimodal information fusion towards multimodal emotion recognition. They discussed that the fusion of the

information takes place at signal, feature and decision levels. However, the work did not report any practical implementation and experimental results [17]. De Silva and Chi exploited a rule based method for decision level fusion of speech and vision based systems. The multimodal results showed an improvement over both of the individual systems [21]. Zeng et al. used a voting method to combine output of audiobased and vision-based recognition systems for person-dependent emotion recognition [32].

However, the above listed works of audio-visual-based emotion recognition did not consider the influence of the speech-related configuration of the face on the emotion-related facial changes and consequently on the accuracy of recognizing emotions in natural and unconstrained conversational human robot interaction, which is challenged in this work.

3 Head Pose Interpretation

In this work the user's head poses are extracted and used for an attention and head gesture estimation. The attention estimation in this paper is based on the parameters of a fitted AAM. Trefflich [27] showed that the head movements have a strong correlation to eye movements, because the head movements are the low-pass filtered eye movements. So the 3D head pose is used for the attention estimation. The described system interpret the AAM-Parameters as head pose and analyzes them for attention and gesture estimation [11]. For some applications like the emotion estimation a very detailed but only frontal face model is needed. Due to their complexity the model does not fit enough to different head poses. However, for attention estimation and detection of head shaking or nodding a model is needed that fits good to a rotated face. This is possible by using a model that is not detailed in shape. An overview of the used architecture is shown in Fig. 1.

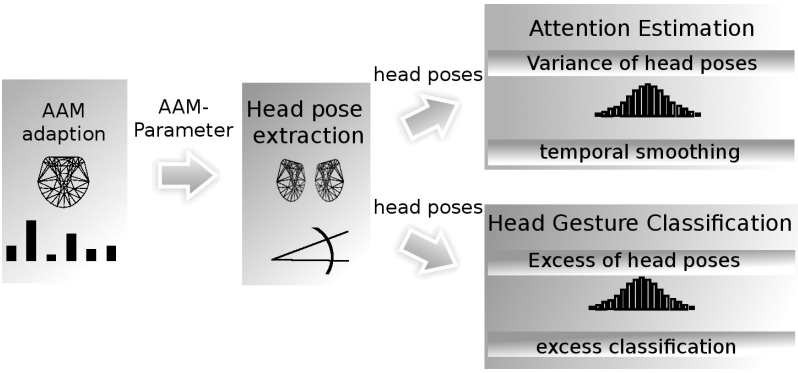


Fig. 1 Head Pose Interpretation. The attention estimation and head gesture classification is done by extracting the head pose from AAM-parameters. The poses are aggregated over time to compute the variance and excess kurtosis. The variance is used to compute the user's attention while the excess kurtosis is applied for head gesture recognition. Temporal smoothing is applied for both subsystems to reduce input parameters noise.

3.1 Head Pose Estimation

Both the attention and the gestures of Yes and No can be determined by head poses. Trefflich shows that the visual attention correlates with the head pose [27]. The first part in the estimator is to extract the head pose. Afterwards, the attention and head gestures are estimated by using statistics. Experiments have shown that some AAM-parameters correlate with the head pose once the training dataset contain head poses. For head pose estimation an own dataset is used which contains mixed facial image sequences of male and female people who rotate their heads around. Each image of this dataset is labeled by the current head pose which is determined by the so called *Flock of Birds*. The *Flock of Birds* is a two parted system which determined the head pose by using magnetic fields. The one part is fixed and must be positioned near by the camera, the other part must be mounted on the top of the head. This system must be calibrated for each user to get correct values. Few samples of this dataset is shown in Fig. 2 .

Fig. 2 Head Pose Dataset. This is an example of the used dataset for the head pose estimation.

The correlation between head poses and AAM-parameters can not be generalized, thereby it is necessary to test each AAM. In the used AAM-approximation the first four shape parameters are generated by a face detector. These so called *global shape parameters* only describe the position of the face in the image. The other shape parameters which describe the individual form are so called *local shape parameters*. Fig. 3 shows the correlation between the first ten possible local shape parameters and the head rotation. By reason of this correlation it is possible to use an AAM with only the first two local shape parameters.

Through this, the used model includes 50 texture parameters but only six shape parameters. Whereof two shape parameters describe the head rotation, one for horizontal and one for vertical head rotation. To have a similar model a method to generate an Active Appearance Model with two separate example sets is favorable [26]. In the used method the texture is learned first. After that, the shape could be learned by separate examples. For texture training the FG-Net dataset [30] is used. An own dataset of one person who move the head horizontally and vertically in separated time frames is used for learning the shape parameters. To map the two shape parameters which describe the head pose onto the correlated real world head pose, two simple *Multilayer Perceptrons* (MLPs) are used. One MLP maps the AAM-Parameter which describes the horizontal head pose onto the correct linear head pose. Equivalent to this, the other MLP maps the AAM-Parameter which describes

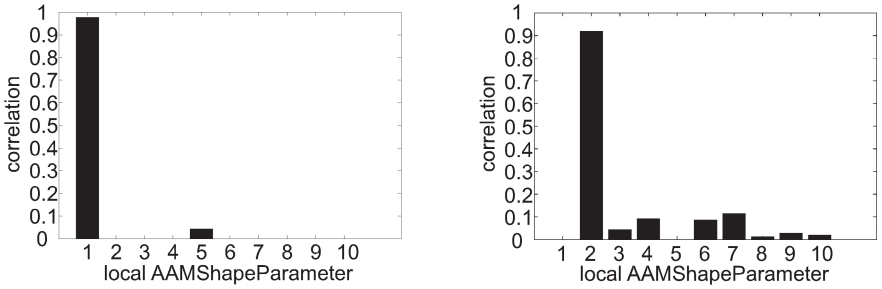


Fig. 3 Correlation between the local AAM-Parameters and the Head Poses. The left diagram shows the correlation with the horizontal head rotation and right diagram shows the correlation with the vertical head rotation.

the vertical head pose onto the correct linear head pose. In this way the horizontal and vertical head poses are described in degree. Each MLP consists in one input neuron, two hidden neurons and one output neuron.

3.2 Attention Estimation

To analyze the head poses, the statistics is used, since it allows a direct attention and gesture estimation without training data. For efficient calculation of these statistics, an *adaptive recursive method* is used, which was developed by Grießbach [6]. All used *adaptive recursive methods* employ a constant weight c which range from 0 to 1. It is possible to use different weights for each adaptive recursive method. For that reason we use the indexed weights c_M, c_{Z^2}, c_A and c_I. By the use of a tall c, the method is more sensitive to input chances. The variance Z^2 (Fig. 4) of the head poses are used to get a classification onto attentive or inattentive. To calculate the variance at the time $t+1$, the temporal mean M_{t+1} and the input value (current head pose) X_{t+1} is needed. In this work, the temporal mean is also adaptive recursive calculated. For the use of adaptive recursive methods, well initializations are favourable. In this work, we postulate that the user is initially attention. Therefore, we use 0 for m_0 and z_0:

$$M_0 = m_0,$$
$$M_{t+1} = M_t + c_M \cdot (X_{t+1} - M_t),$$
$$Z_0^2 = z_0,$$
$$Z_{t+1}^2 = Z_t^2 + c_{Z^2} \cdot \left((X_{t+1} - M_{t+1})^2 - Z_t^2 \right).$$

A sequence of head poses are classified as attentive once the variance is lower than a threshold with a value of 40 degree. The aim in this work is to get a continuous value for the attention A_t. For this an adaptive recursive mean is updated by 100 once the person is attentive and with 0 otherwise:

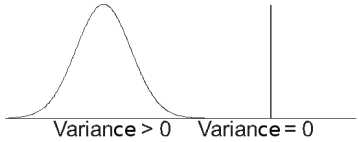


Fig. 4 Variance. When the variance is greater than 40°, the current part of the sequence is classified as inattention and otherwise as attention

$$A_0 = 100,$$
$$A_t = A_{t-1} + c_A \cdot (100 - A_{t-1}) \quad \text{, if attention,}$$
$$A_t = A_{t-1} + c_A \cdot (0 - A_{t-1}) \quad \text{, if inattention.}$$

When a person is attentive mostly it is in interest where the people look too. This point of interest I can be estimated by the adaptive mean of the head poses p:

$$I_0 = 0,$$
$$I_t = I_{t-1} + c_I \cdot (p_t - I_{t-1}).$$

3.3 Head Gesture Estimation

Furthermore, the developed estimator is able to detect head shake and nodding from a face image sequence. Afterwards, a context-sensitive interpretation as Yes or No is possible. When a person shakes their head, only few changes of the head pose in the vertical are generated, but lots of changes in the horizontal are expected. Thereby, the excess kurtosis of the horizontal head poses is platykurtic and the excess kurtosis of the vertical head poses is leptokurtic. The effect is reverse by nodding. Since this condition is unequivocal it can be used for detection. (Fig. 5) The adaptive excess kurtosis ε can also calculated by a method of Grießbach:

$$Z_0^4 = z_0,$$

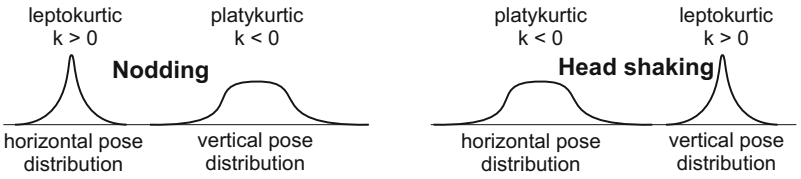


Fig. 5 Head Gesture. Performing a nodding gesture involves changes in the vertical head pose. Therefore, the vertical head pose distribution becomes platykurtic while the horizontal distribution is leptokurtic. Head shaking involves changes of the horizontal head pose. Therefore, the distribution characteristics are inverted.

$$Z_{t+1}^4 = Z_t^4 + c \cdot \left((X_{t+1} - M_{n+1})^4 - Z_t^4 \right),$$

$$\varepsilon = \frac{Z_n^4}{(Z_n^2)^2} - 3.$$

3.4 Eye Tracker

To estimate the attention of demented people the eye gaze is needed, since they are mostly unable to move the head full freely. So these people look around only with their eyes and without head movements. To estimate the eye gaze an ordinary webcam is used. First, the AAM affords the eye position, so it is possible to focus only to this image parts. A gray level eye consist of a white plane, a darker iris and a black pupil, so the pupil is detected as the darkest point in the eye. To establish the eye gaze it is necessary to know the possible eye movements. Speckmann and Hescheler reported that the healthy eye is able to move 20° to the left and 20° to the right [24]. This is an interest area of 40° onto the curvature of the eye. Up to this value it is possible to assume a linear correlation between the pupil position into the 2D image and the pupil position onto real world eye.

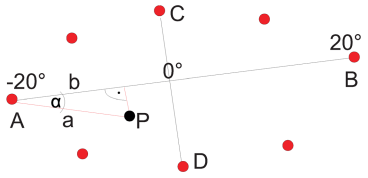


Fig. 6 Eye Tracker. Triangulation to calculate the horizontal and vertical pupil position for the eye gaze estimation

The same assumption is used for the vertical eye movement. Thereby, it is possible to estimate the line of gaze for each pupil position by the use of triangulation. This is exemplary shown in Fig. 6. To estimate the global eye gaze it is necessary to add the head pose, too.

4 Emotion Recognition

Possible modalities to exploit for automatic recognition are language (acoustic-tic and linguistic information), facial expressions, body gestures, bio signals (e. g. heart rate, skin conductance), or behavioral patterns (such as mouse clicks). Though one modality alone can already give information on the affective state of a user, humans always exploit all available modalities, and if an automatic systems attempts to reach human performance, the need for multi-modality is obvious. Thereby not only consent results of different modalities lead to more confident decisions, but also conflicting results can be helpful [17], e. g. to detect pretended or masked emotions, or

to find out more reliable modalities for certain emotions. The most obvious modalities in human-human conversation, and also in human-robot conversation which we aim to enhance, are speech and facial expressions.

In our work we challenge this approach by analyzing the auditory and visual stimuli with respect to their general discriminative power in recognizing emotions. Note that in our work we focus on interactive scenarios and are thus targeting at systems that are able to work online. The approaches we present in this paper are, therefore, not only being tested offline on existing databases but have proven their applicability in robotic applications in real world settings [8, 19, 18]. This is in contrast to other work (e.g. [3]), which has focused on offline emotion recognition only. The following three sections will provide a brief introduction on the respective unimodal analysis techniques as well on the proposed probabilistic decision level fusion.

4.1 Visual Facial Expression Recognition

In order to recognize basic emotion visually, we take a closer look into the interlocutor's face. The basic technique applied here are Active Appearance models (AAMs) first introduced by Cootes et al [5]. The generative AAM approach uses statistical models of shape and texture to describe and synthesize face images. An AAM, that is built from training set, can describe and generate both shape and texture using a single appearance parameter vector, which is used as feature vector for the classification. The active component of an AAM is a search algorithm that computes the appearance parameter vector for a yet unseen face iteratively, starting from an initial estimation of its shape. The AAM fitting algorithm is part of the integrated vision system [19] that consists of three basic components. Face pose and basic facial features (BFFs), such as nose, mouth and eyes, are recognized by the face detection module [4]. The coordinates representing these features are conveyed to the facial feature extraction module. Here, the BFFs are used to initialize the iterative AAM fitting algorithm. After the features are extracted the resulting parameter vector for every image frame is passed to a classifier which categorizes it in one of the six basic emotions in addition to the neutral one.

Besides the feature vector, AAM fitting also returns a reconstruction error that is applied as a confidence measure to reason about the quality of the fitting and

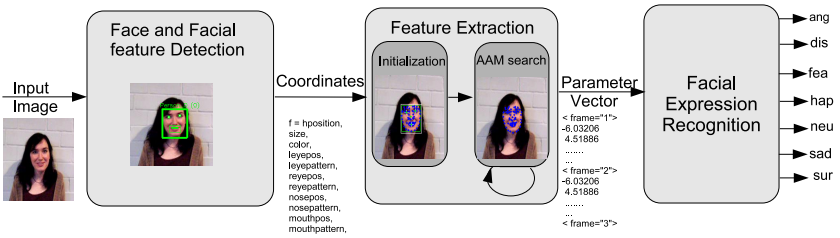


Fig. 7 Emotion Recognition. Architecture of the facial analysis sub-system

also to reject prior false positives resulting from face detection. As classifier a one-against-all Support Vector Machine is applied. The whole system is applicable in soft real-time, running at a rate of approximate 5 Hz on recent PC hardware.

4.2 Emotion Recognition from Speech

For the recognition of emotions from speech, EmoVoice, a framework that features offline analysis of available emotional speech databases, as well as online analysis of emotional speech for applications, is used [29]. This approach is purely based on acoustic features, that is no word information is used. The first step is the extraction of emotion-related features from audio signal. A large vector of statistical features based on prosodic and acoustic properties of the speech signal was calculated for each utterance in the DaFEx database. To reduce the size of feature vectors (over 1400 attributes) correlation-based feature subset selection is used. This selection is necessary to increase performance as well as speed of classification. By this way, 71 features related to pitch, energy, MFCCs, linear regression, range of the frequency spectrum of short-term signal segments, speech proportion, length of voiced and unvoiced parts in an utterance, and the number of glottal pulses remained. For classification support vector machines are used, but with a linear kernel. The feature selection is typically done offline, but the feature extraction and classification can be done either offline or in real-time.

4.3 Probabilistic Decision Level Fusion

Multimodal information fusion is the task of combining some interrelated information from multiple modalities. In an emotion analysis system, while a unimodal system incorporates features of a single modality (visual, audio, tactile, or body information) the multimodal systems use information from multiple different modalities simultaneously.

As affective states in interaction are usually conveyed on different cues at the same time, we agree with other works summarized in [31] that a fusion of visual and acoustic recognition yields significant performance gains. Hence, we followed the idea of an online integration scheme based on the prior offline analysis of recognition results on a database.

In current fusion research, three types of multi-modal fusion strategies are usually applied, namely data-/signal-level fusion, feature-level fusion, and decision-level fusion [17]. Signal-level fusion is applicable solely to sources of the same nature and tightly synchronous. Generally it is achieved by mixing two or more physical signals of the same nature (two auditive signals, two visual signals of two cams, etc). This type of mixing is not feasible for multimodal fusion due to the fact that different modalities always have different captors and different signal characteristics (auditive and visual).

Feature-level fusion means concatenation of the features outputted from different signal processors together to construct a joint feature vector, which is then

conveyed to the affect analyzer. It is used when there is evidence of class-dependent correlation between the features of multiple sources. For example, features can be extracted from a video processor (facial expression) and speech signal (emotion-related prosodic features). Feature-level fusion benefits of interdependence and correlation of the affective features in both modalities but is criticized for ignoring the differences in temporal structure, scale and metrics. Although, feature-level fusion demands synchronization of some extent between modalities. Yet another drawback of such a fusion strategy is that it is more difficult and computationally more intense than combining at the decision level.

The third fusion strategy combines the semantic information captured from the individual unimodal systems, rather than mixing together features or signals. Due to the advantages of (I) being free of synchronization issues between modalities, (II) using relative simple fusion algorithms, and (III) their low computational requirement in contrast to the feature-based methods, decision-level fusion methods are adopted from the vast majority of researchers in the field of multimodality emotion recognition. Following this conclusion we decided a probabilistic-based decision level fusion method to join the facial expression-based, and the acoustic information-based emotion recognizers into bimodal one.

The proposed decision-level fusion method is probabilistic approach based on a top-down-reasoning Bayesian network with a rather simple structure depicted in Fig 8. Based on the classification results of the individual visual and acoustic classifiers, we feed these into the Bayesian network as evidence of the observable nodes (Acoustic and Visual, respectively). By Bayesian inference the posteriori probabilities of the unobservable affective fusion (Fusion) node are computed as:

$$\mathbf{P}(\text{Fusion} = e_f | \text{Visual} = e_v, \text{Acoustic} = e_a), \tag{1}$$

where, e_f, e_v, e_a can belong to any one of seven emotion classes mentioned above, and taken as a final result.

The required probability tables of the Bayesian network are obtained from a performance evaluation of each individual classifiers in an offline training phase based

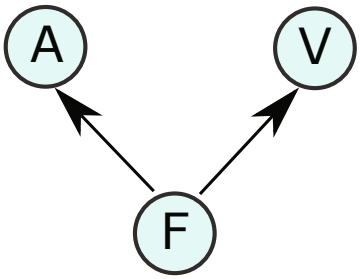


Fig. 8 The Bayesian network. The structure of the Baysian network used to fuse cues of both uni-modals. Evidences of observable nodes acoustic (A) and visual (V) are fed as input into the corresponding nodes. The posteriori probabilities of the unobservable node are computed, wich give fusion (F) as the final result.

on ground-truth-annotated databases [12]. Therefore, confusion matrices of each classifier are turned into conditional probability tables modeling the dependent observation probabilities of the model according to the arrows in Fig. 8.

5 User Re-identification

To use individual settings of a dialog adaption system for different users, a user recognition is essential. To allow a contactless interaction, the system could use visual or acoustic information. Whereas acoustic informations of the user are not evaluable all the time, visual informations are used in this work. Especially for the described facial methods a facial user re-identification could be used [7]. Certainly, in the case of searching or monitoring the dialog partner, facial information are often not available. In this case and to boost the facial re-identification, a whole-body user recognition is used. This work shows an appearance based user re-identification. Before an re-identification is possible, the person must be detected. In this work Histograms of Oriented Gradients are classified by a Support Vector Machine. To be real time capable, a GPU (Graphics Processing Unit) based algorithm is used [16].

The appearance based whole-body user re-identification is useful for monitoring multiple dialog partner in one scene. This kind of re-identification is not helpful for an application over a long period of time since humans whole-body appearance varies by changing clothes. Certainly, the appearance based features are divided into color and texture features. Whereas the texture is naturally independent of the illumination. Typically, color image pixels are described with the help of three color channels, which represents the red, green and blue parts of the current color (RGB). This results in different pixel representations given different illuminations. For this reason the RGB color representation is transformed into a color representation which allows to separate the illumination influence (Formula 2). The used color space represents the pixel color also in three channels, which describes separately the hue, saturation and the value (HSV). The hue range from 0° to 360°, the saturation range from 0% to 100% as well as the value. In this work only the illumination independent hue and saturation of the color are used:

$$
\begin{aligned}
&R,G,B \in [0,1]; MAX = max(R,G,B); MIN = min(R,G,B)\\
&H = \begin{cases} 0^\circ, & \text{if } MAX = MIN \Leftrightarrow R = G = B,\\ 60^\circ \cdot (0 + \frac{G-B}{MAX-MIN}), & \text{if } MAX = R,\\ 60^\circ \cdot (2 + \frac{B-R}{MAX-MIN}), & \text{if } MAX = G,\\ 60^\circ \cdot (4 + \frac{R-G}{MAX-MIN}), & \text{if } MAX = B, \end{cases}\\
&H = H + 360^\circ, \text{ if } H < 0^\circ,\\
&S = \begin{cases} 0, & \text{if } MAX = 0 \Leftrightarrow R = G = B = 0,\\ \frac{MAX-MIN}{MAX}, & \text{other}, \end{cases}\\
&V = MAX.
\end{aligned}
\tag{2}
$$

Three fixed parts of the detected people are selected to eliminate color and texture informations Fig. 9. One rectangle part of the lower body is separated to determine the mean hue and saturation. Furthermore, the mean horizontal and vertical texture rates as well as the mean hue and saturation is calculated from a rectangle part of the upper body. One histogram of the hues and one histogram of the saturations are calculated from an oval area of the upper body.

Fig. 9 Feature extraction. The used features for the full body people re-identification will be extracted from three areas. The location of this areas is relative to the people detection.

Whereas, the upper body appearance is typically more complex the texture rates and the histograms are calculated only from this part. The mean horizontal and the mean vertical texture rates describe the strength of the texture at the selected area. To calculate the horizontal and vertical texture, the Scharr filter [20] is used. The mean hue and saturation describe the ground color of the user's lower and upper body, while the histograms describe more detailed the upper body colors. By using the hue and saturation histograms, even detailed color prints etc. at the clothes are represented in a very compact form. In this work normalized histograms are used due to their scale independency. To handle minor changes of the color, the hue and saturation are both divided into only 16 parts for the histograms. To allow more robust identification, all the features will be tracked over the time. This tracks are used to build a user feature space. This kind of people description is very compact and easy to save. To compare the saved feature spaces with the current features, only the normalized differences of all the features are calculated and summarized. Certainly, during the calculation of the difference D between the hues of the searched person feature space (H_i) and the hue of the current hypotheses (H_c), it must heeded that the hue is represented as a circle. To calculate the difference of the hue (Formula 3) is used:

$$\begin{aligned} D &= |H_i - H_c|, \\ D &= 360^\circ - D, \text{ if } 360^\circ - D < D. \end{aligned} \tag{3}$$

In this way, the summarized differences are used as a score for user identification. A small score means a hight probability to find the correct person.

6 Experimental Results

This section presents experimental results achieved by using the described approaches. Furthermore, a number of test sequences are recorded to evaluate the proposed attention measuring system, the head gesture recognition, the eye tracker and the emotion recognition.

6.1 Head Pose Estimation

To evaluate the presented AAM and MLP based head pose estimation, 8 test sequences are recorded whereby the people can look around. By recording this sequences the head pose which is determined by the *Flock of Birds* is simultaneous recorded. Then the head poses which are estimated with the help of the presented system is compared to the head poses of the *Flock of Birds*. Thereby, the RMS for vertical pose is 0.1387° and the RMS for horizontal is 0.1546°. This result is shown in Fig. 10.

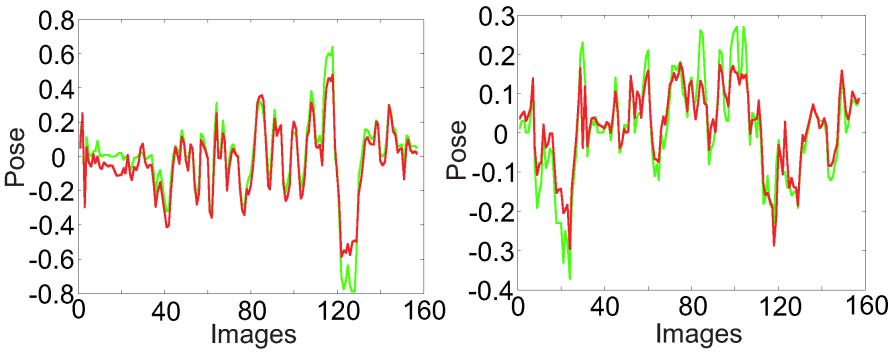


Fig. 10 MLP-Output vs. ”Flock of Birds”. The green curve represent the head poses which are determined with the help of the ”Flock of Birds” and the red curve represent the head poses which are estimated by the presented system. *left:* horizontal head poses; *right:* vertical head poses.

6.2 Attention Estimation

To evaluate the proposed attention measure under real world conditions, 23 test sequences from 8 different persons are recorded. The people were asked to watch several video clips in front of a computer monitor, while they were monitored with the help of a frontal camera. During the first stage, the person were watching an exciting movie. In the second stage, the same persons were watching a boring video,

while another person enters the room and tries to distract the test persons by talking to them or letting things falling on the ground. Afterwards, the recorded image sequences were labeled manually by several people in terms of visual attention. For evaluation purposes the same sequences were presented to the proposed attention estimation system. The average divergence between the presented system and the labels is only 4 percentage points which is shown in Table 1.

Table 1 Attention Estimation. This table shows the rate of attention for each evaluation sequence which is determined by the presented system and by human interpretation

Sequence	1	2	3	4	5	6	7	8	9	10	11	...
System	23.6%	16.8%	3.1%	1.1%	8.5%	4.9%	0%	0%	25.5%	0%	0.2%	...
Human	0%	15.5%	0%	2.3%	12.9%	6.7%	0%	0%	7.5%	0.7%	0%	...
Difference	23.6%	1.3%	3.1%	1.2%	4.4%	1.8%	0%	0%	18%	0.7%	0.2%	...

Sequence	12	13	14	15	16	17	18	19	20	21	22	23
System	4.3%	1.1%	0%	8.6%	14.8%	31.6%	0%	0%	27.1%	13.4%	28.4%	69.6%
Human	0%	0%	0%	0%	7.8%	26.9%	0%	0%	20.6%	15.6%	27.1%	68.3%
Difference	4.3%	1.1%	0%	8.6%	7%	4.7%	0%	0%	6.5%	2.2%	1.3%	1.3%

The described system only fails massively at three image sequences. By visual analyzing of these sequences, it is prominent that the high divergence of these sequences is generally caused by a bad model fitting leading the head direction estimation to fail.

Before the variance for the attention estimation is used, the excess kurtosis was also tried to use, since the excess kurtosis have no scale unit. However, the use of the excess kurtosis failed because it is zero when the head pose histogram is normally distributed and this is able during attention and during inattention (Fig. 11).

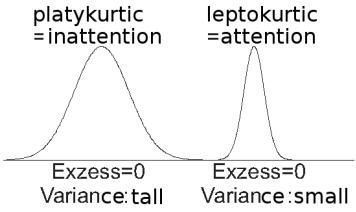
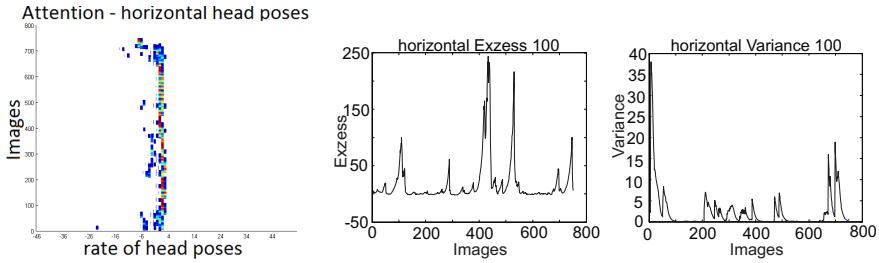


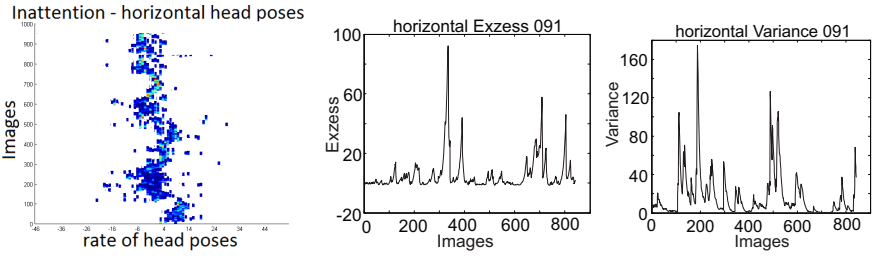
Fig. 11 Excess Kurtosis vs. Variance. A wide and a small histogram can be Gaussian distribution, so the excess kurtosis can be 0, once the variance become tall when the user is inattentive

Fig. 12 and Fig. 13 shows exemplary the head poses, the excess kurtosis and the variance of an attention and an inattention sequence.



(a) **Horizontal Head Poses** increasing rate from blue to red *(b)* **Excess Kurtosis** *(c)* **Variance**

Fig. 12 Excess Kurtosis vs. Variance of an attention sequence. (a) shows the temporal histogram for each time. A small histogram with many red areas leads to a small variance (c) but to a fluctuating excess kurtosis (b).



(a) **Horizontal Head Poses** increasing rate from blue to red *(b)* **Excess Kurtosis** *(c)* **Variance**

Fig. 13 Excess Kurtosis vs. Variance of an inattention sequence. (a) shows the temporal histogram for each time. A wide histogram with less red areas leads to a large variance (c) but to a fluctuating excess kurtosis (b).

6.3 Head Gesture Estimation

To evaluate the Head Gesture Estimation, 4 sequences from several people, which nod and shake their heads in addition to further head movements, are recorded. On this experiment each head nodding and shaking could be detected by no false positive detection. Already a single fully Yes (head nodding) or No (head shaking) head gesture could be detected, where the system reaction is shortly delayed, once the used adaptive calculation.

6.4 *Eye Tracker*

The simple eye tracker is only evaluated by three sequences. In these sequences the people look straight ahead onto the camera and move only their heads, so the eye gaze should roughly 0° where the head pose differ. In Fig. 14 and Fig. 15 the vertical and horizontal eye gazes and head poses is exemplary visualized for one sequence.

(a) **horizontal head pose** *(b)* **horizontal eye gaze**

Fig. 14 Histogram of horizontal head pose vs. eye gaze

(a) **vertical head pose** *(b)* **vertical eye gaze**

Fig. 15 Histogram of vertical head pose vs. eye gaze

Furthermore, the Table 2 shows the variance of the head poses and the eye gazes of the three sequences. This results shows that it is principal possible to estimate the eye gaze by this simple method with a small error. This estimation should be exact enough to use it in an attention estimation.

Table 2 Variance Head Pose vs. Eye Gaze. This table shows the variance of the vertical and horizontal head poses and eye gazes of 3 test sequences from several person.

Person	hor. head pose	hor. eye gaze	ver. head pose	ver. eye gaze
1	71.4	4.3	15.6	1.8
2	39.5	6.7	2.0	0.8
3	176.8	3.0	2.8	0.6

6.5 *Bimodal Emotion Recognition*

As we are striving in this work to give the robot a bimodal emotion recognition ability that is based on analyzing facial expressions and speech information, the systems are afresh evaluated on data set with subjects in a real-life conditions. Four subjects have participated in this test (one female and three males). The whole procedure is divided into training and test phases. For one subject both phases were conducted in the same day; for two others the test was is conducted in the following day, while for the fourth subject the time interval was two days.

In the training phase the subjects are asked to display facial expressions of live emotion classes: anger, happiness, neutral, sadness, and surprise with and without speaking. The average amount of data captured from each subject for each facial expression class was 246 images. To create conditions of real-life human-robot interaction as much as possible, the subjects are allowed to move arbitrarily in front of the camera. During this phase a person-independent AAM, which is built from a subset of the DaFEx database of talking and non-talking subjects, is used to extract the emotion-related facial features. These features are then conveyed to train a person-dependent SVM.

In the test phase the subjects are asked to display facial expressions and utter a few sentences (in general five) expressing as much an emotions as possible [1]. The above-mentioned AAM is used to extract facial features, which are labeled with the proper emotional class by the above-trained SVM. In this session a person-independent speech-based emotion recognizer is utilized to categorize each utterance into the proper emotional class. An average of 145.25 images from each subject for each emotion are used as test data. The validation matrix for the fusion scheme of each subject was an averaged confusion matrix (CPT), which is obtained from the performance of both individual systems on the three remaining subjects.

Table 3 Emotion Recognition. The performance of each stand-alone unimodal systems against the one of the bimodal system. All results are obtained from a test in a real-life condition.

	Anger	Happiness	Neutral	Sadness	Surprise	Total
Vis	75.00	43.75	50.00	60.28	47.92	55.39
Aco	33.04	15.42	36.25	23.06	10.42	23.63
Audio-Visual	75.00	50.00	68.75	49.03	47.92	58.14

Table 3 illustrates the results obtained from both the stand-alone and bimodal systems. The low rates delivered by the speech-based emotion analysis system - the first raw - could be because a person-independent classifier is used, which is trained on a speech-based emotion database that does not include the subjects participating in the evaluation procedure. Nevertheless, it can be seen that the whole performance of the bimodal system has an advantage over both facial-expression- and speech-information-based systems, which satisfy the goal of the fusion scheme proposed

[1] The sentences were emotional words free.

previously. However, when the performance of each channel on each emotion is considered it is notable that the recognition rate of happiness and neutral is enhanced when the bimodal system is employed, which indicates that the cues of both modalities comprise complementary information for these two emotions. In contrast, from the first and fifth rows, it is noticeable that both unimodal cues comprise only redundant information so that combining both modalities yields no improvement with regard to discrimination ability for the recognition of anger and surprise. Furthermore, the fourth column indicates that both modalities deliver conflicting information, which causes sadness to be recognized even less than the stand-alone facial-expression-based modality.

7 Conclusion

In this paper, approaches for extracting the user attention, the head gestures and the emotions are presented. These approaches utilize the shape and texture parameters from a fitted Active Appearance Model. This paper focus on improving the human-robot interaction. An attention and head gesture estimation, which use the AAM shape parameters to estimate the user's head pose is applied. For the measurement of attention the distribution of the head pose over time are used. By comparison the hand labeled attention values with the system output, the presented system seems to be able to estimate the attention value quite well. In addition, a head gesture recognition based on the temporal event mapping approach is proposed.

In order to enhance the naturality of the interaction an approach, which provide the ability for the dialog system to analyze the emotional state of the interaction partner, is implemented. The proposed approach fused a facial-expression-based emotion analysis system with a speech-based analysis system in a bimodal emotion analysis system. A probabilistic decision level fusion approach is used. That makes benefits of its simplicity, no mandatory of synchronization, and the general discriminative power of each unimodal compared to the other fusion methods. Five of the seven basic emotions of Ekman are considered in a life-like scenario of human robot interaction. The bimodal system outperforms both uni-modal system in a natural and life-like conversational human-robot interaction.

Continuing this work, could be possible by integrating the proposed systems into a dialog system [15]. This will be very helpful to examine how the proposed attention values as well as the emotion can be utilized to enable a more natural human-robot interaction. Furthermore, the possibility to track the human eyes with the help of an ordinary webcam with a small estimation error is shown. To enable the use of individual parameter, a re-identification method is presented.

Non-basic emotions such as frustration, sleepiness and satisfaction could be considered in future works by adopting the dimensional approach of emotion categorization. A further work on the topic of attention and emotion estimation could be the usage of full body motion. Furthermore, it is possible to use full body motion as a further feature for the full body re-identification.

Acknowledgements. This work was partly funded by the German Federal Ministry of Education and Research (BMBF) in the framework of the APFel project.

References

1. Ba, S., Odobez, J.: Recognizing Visual Focus of Attention from Head Pose in Natural Meetings (2009)
2. Busso, C., Deng, Z., Yildirim, S., Bulut, M., Lee, C.M., Kazemzadeh, A., Lee, S., Neumann, U., Narayanan, S.: Analysis of Emotion Recognition using Facial Expressions, Speech and Multimodal Information. In: Proc. Int. Conf. Multimodal Interfaces (2004)
3. Caridakis, G., Malatesta, L., Kessous, L., Amir, N., Raouzaiou, A., Karpouzis, K.: Modeling naturalistic affective states via facial and vocal expressions recognition. In: Proc. Int. Conf. Multimodal Interfaces, pp. 146–154. ACM, New York (2006)
4. Castrillón, M., Déniz, O., Guerra, C., Hernández, M.: ENCARA2: Real-time detection of multiple faces at different resolutions in video streams. Journal of Visual Communication and Image Representation 18(2), 130–140 (2007)
5. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active Appearance Models. PAMI 23(6), 681–685 (2001)
6. Grießbach, G.: Weiterentwicklung und Anwendung komplexer adaptiver Schätzalgorithmen in der Biosignalanalyse, der Bildverarbeitung und der Klassifikation zur EBG-Analyse kognitiver Prozesse. DFG-Antrag Gr1 55511-2 (1998)
7. Handmann, U., Hommel, S., Brauckmann, M., Dose, M.: Face Detection and Person Identification on Mobile Platforms. Springer Tracts in Advanced Robotics (STAR). Springer, Germany (2012)
8. Hegel, F., Spexard, T., Vogt, T., Horstmann, G., Wrede, B.: Playing a different imitation game: Interaction with an empathicandroid robot. In: Proc. Int. Conf. Humanoid Robots, pp. 56–61 (2006)
9. Hommel, S.: Zeitliche Analyse von Emotionen auf Basis von Active Appearance Modellen. GRIN Verlag GmbH (2010)
10. Hommel, S., Handmann, U.: AAM based Continuous Facial Expression Recognition for Face Image Sequences. In: 2011 12th IEEE International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, pp. 189–194 (2011)
11. Hommel, S., Handmann, U.: Realtime AAM based User Attention Estimation. In: 2011 IEEE 9th International Symposium Intelligent Systems and Informatics (SISY), Subotica, Serbia, pp. 201–206 (2011)
12. King, L.M., Taylor, P.B.: Hands-free Head-movement Gesture Recognition using Artificial Neural Networks and the Magnified Gradient Function. In: IEEE Conf. of the Engineering in Medicine and Biology Society, pp. 2063–2066 (2005)
13. Lu, P., Zhang, M., Zhu, X., Wang, Y.: Head nod and shake recognition based on multiview model and hidden Markov model. In: Computer Graphics, Imaging and Vision: New Trends, pp. 61–64 (2005)
14. Morency, L.-P., Trevor, D.: Head gesture recognition in intelligent interfaces: the role of context in improving recognition. In: Proceedings of the 11th International Conference on Intelligent User Interfaces, IUI 2006, pp. 32–38. ACM, New York (2006)
15. Müller, S., Schröter, C., Gross, H.-M.: Aspects of user specific dialog adaptation for an autonomous robot. In: IWK (2010)
16. Prisacariu, V., Reid, I.: fastHOG - a real-time GPU implementation of HOG. Department of Engineering Science, Oxford University, Tech. Rep. 2310/09 (2009)

17. Paleari, M., Lisetti, C.L.: Toward multimodal fusion of affective cues. In: Proc. ACM int. Workshop on Human-Centered Multimedia, pp. 99–10. ACM, New York (2006)
18. Rabie, A., Handmann, U.: Fusion of Audio- and Visual Cues for Real-Life Emotional Human Robot Interaction. In: DAGM 2011 (2011)
19. Rabie, A., Lang, C., Hanheide, M., Castrillon-Santana, M., Sagerer, G.: Lang, Ch., Hanheide, M., Castrillon-Santana, M., Sagerer, G.: Automatic Initialization for Facial Analysis in Interactive Robotics. In: Proc. Int. Conf. Computer Vision Systems, Santorini, Greece (2008)
20. Scharr, H.: Optimal operators in digital image processing. Ph.D. thesis, Interdisciplinary Center for Scientific Computer, Ruprecht-Karls-Universität, Heidelberg (2000)
21. Silva, L.D., Chi, P.: Bimodal Emotion Recognition. In: Fourth IEEE Int. Conf. on Automatic Face and Gesture Recognition (2000)
22. Smith, K., Ba, S.O., Odobez, J.-M., Gatica-Perez, D.: Tracking the Visual Focus of Attention for a Varying Number of Wandering People. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 1212–1229 (2008)
23. Smith, P., Member, S., Shah, M., Lobo, N.D.V.: Determining Driver Visual Attention with One Camera. IEEE Trans. on Intelligent Transportation Systems 4 (2003)
24. Speckmann, E.J., Hescheler, J., Köhling, R.: Repetitorium Physiologie, 2nd edn. Urban & Fischer Verlag (2008)
25. Stiefelhagen, R., Finke, M., Yang, J., Waibel, A.: From Gaze to Focus of Attention (1998)
26. Stricker, R., Martin, C., Gross, H.-M.: Increasing the Robustness of 2D Active Appearance Models for Real-World Applications. In: Fritz, M., Schiele, B., Piater, J.H. (eds.) ICVS 2009. LNCS, vol. 5815, pp. 364–373. Springer, Heidelberg (2009)
27. Trefflich, B.: Videogestützte Überwachung der Fahreraufmerksamkeit und Adaption von Faherassistenzsystemen. Technische Universität Ilmenau (2009)
28. Viola, P., Jones, M.: Robust Real-time Object Detection. In: Second Int. Workshop on Statistical and Computational Theories of Vision-Modeling, Learning, Computing, and Sympling (2001)
29. Vogt, T., André, E., Bee, N.: EmoVoice — A Framework for Online Recognition of Emotions from Voice. In: André, E., Dybkjær, L., Minker, W., Neumann, H., Pieraccini, R., Weber, M. (eds.) PIT 2008. LNCS (LNAI), vol. 5078, pp. 188–199. Springer, Heidelberg (2008)
30. Wallhoff, F.: Facial Expressions and Emotion Database. Technische Universität München (2006), http://www.mmk.ei.tum.de/~waf/fgnet/feedtum.html
31. Zeng, Z., Pantic, M., Roisman, M.I., Huang, T.S.: A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. IEEE Transaction on Pattern Analysis and Macine Intellegence 31, 39–58 (2009)
32. Zeng, Z., Tu, J., Liu, M., Zhang, T., Rizzolo, N., Zhang, Z., Huang, T.S., Roth, D., Levinson, S.: Bimodal HCI-related Affect Recognition. In: ICMI (2004)

Topological Modelling as a Tool for Analysis of Functioning Systems

Ivars Karpics, Zigurds Markovics, and Ieva Markovica

Abstract. This article presents a mathematical model construction approach for functioning systems. To perform functioning system analyses with the goal to establish correct work conditions, heuristical problem solving way and mathematical modeling can be used. Topological modeling is an effective tool to develop mathematical models for heterogeneous systems when the available information is insufficient. Within this article, the authors provide a theoretical background and introduce topological model elements, functions, features, and construction phases. A practical model construction process is adapted to be used for medicine tasks. A topological model for multiple diseases is developed. It is used as a mechanism to model the course of a disease and the effect of the applied therapy. Using the proposed criteria for evaluating the chosen therapy and multi-objective optimization techniques make it possible to prescribe the optimal therapy complex for an individual patient.

Keywords: Heterogeneous system, therapy, individual patient, multi-objective optimization.

1 Introduction

Functioning processes are observed in nature, society, technology and other real world domains. Each process in itself involves characteristics, features and other structural elements which all are mutually connected by cause- effect relations. Various approaches and methods exist to describe and analyse the functioning of a system. Part of these methods is based on statistical methods and mathematical modelling approaches. While statistical computation methods are inflexible, use voluminous processing resources and do not always provide a solution, mathematical modelling and heuristic approaches are flexible with regard to changes in the input

Ivars Karpics · Zigurds Markovics · Ieva Markovica
Riga Technical University, Meza str. 1/3-3
e-mail: {Ivars.Karpics,Zigurds.Markovics,Ieva.Markovica}@rtu.lv

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 237–253.
DOI: 10.1007/978-3-642-33959-2_13

of information and can be adjusted to an acceptable solution. The main advantage of heuristic approaches, however, is that they guarantee an acceptable solution regarding the stated preciseness, constraints and evaluation of the final solution. [1]

During the functioning process, all systems lose their resources and working capacity which leads to a general malfunction. The state of a normal functioning needs to be re-established by adding external influences. This is mostly a recovery strategy. A correctly selected strategy resumes the state of normal functioning. When using mathematical modelling and a heuristic approach a recovery strategy is selected in three steps (Fig. 1), [24]:

1. Full investigation of a functioning system by using expert inquiry methods [13]: summary, formalisation and arrangement of all available knowledge in a common knowledge base;
2. Development of the mathematical model[1];
3. Recovery strategy selection by using the created mathematical model and decision-making techniques.

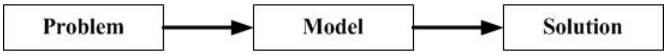


Fig. 1 Selection of a recovery strategy for a functioning system

This article focuses on the first two steps of an appropriate resumption strategy selection and introduces a new method for an appropriate model construction. This method is based on the topological modelling (TM) approach. At the beginning of the article, the theoretical background of TM is provided. In the second part, a practical example is given which includes the development of a model for the analysis of multiple diseases. The model is implemented in a computer system for an optimal therapy selection. This implementation is introduced to present one possible way of using the TM approach. The article concludes with an outlook on potential advancements of the method.

2 Mathematical Model of a Functioning System

A mathematical model represents an abstract, simplified mathematical construction that reflects reality and is created for defined research purposes. Such a model is described by a group of characteristics and logical regularities and defines an idealised performance of the analysed system. A model connects input/ output information, external influences and disturbances in one united mathematical abstraction [1].

[1] In the following when referring to mathematical models it will be used the term 'model'.

The main requirements for a model are:

- The model includes all available knowledge about the analysed functioning process;
- It is developed by knowledge engineers and system developers of expert systems;
- The model needs to be capable to include various types of knowledge (quantitative, linguistic, rule based and other types);
- The construction of the model has to be unsophisticated and foreseeable/predictable, but at the same time it needs to be enough detailed to manage projective research tasks;
- Structural changes of the system can be easily added and implemented in the model;
- The mathematical calculus of the model is unsophisticated and can easily be implemented in the expert system.

2.1 The Use of Topological Modelling for the Construction of a Mathematical Model

One of the mathematical methods for analysing a functioning system is the topological modelling. This method is broadly used in the analysis of various systems. Applications for technical and electrical systems [17], information systems [3, 19], biochemical and biological processes [23], robot control [6] and other systems have been created. Lately, a fast progress of TM for software development and Model Driven Architecture (MDA) applications can be observed [18].

The fundamentals of the topological modelling [17] are based on the assumption that a complex functioning system can be described by abstract concepts like the topological space. The principle of the system's functioning, i.e. of the functional mathematical model, is represented by a topological space in the form $T = (X, \theta)$. X is a functional characteristic space $X = x_1, x_2, \ldots, x_l$ and θ is a topology that describes the relationships between these functioning characteristics in a binary form [17]. The topology in the set X is each system θ that consists of open subsets A, and which satisfy two Kolmogorov axioms (1, 2) [10, 11]:

A set X belongs to the defined topology $\theta(X \in \theta)$ and also an empty set belongs to the topology:

$$\emptyset \in \theta \tag{1}$$

Each split or union of the subsets belongs to the topology:

$$\forall_\eta \left(\bigcup_{\eta=1}^{m} A_\eta \in \theta \right); \forall_\varphi \left(\bigcap_{\varphi=1}^{k} A_\varphi \in \theta \right) \tag{2}$$

The topological model is created in the space of all systems' characteristics, but it also includes information about the structure of the system (defines the topology). One of the solutions how to represent the topology is the description in X open subsets. It is assumed that two points (characteristics) a and b which belong to the

set $X(a \in X, b \in X)$, but are not equal $(a \neq b)$, can be located in the following relations (R: a binary relation exists, $\bar{R}$: a binary relation does not exist):

- aRb, bRa: direct and inverse binary relations between a and b (Fig. 2a);
- $aRb, b\bar{R}a$: direct binary relation from a to b (Fig. 2b);
- $a\bar{R}b, bRa$: inverse binary relation from a to b (Fig. 2c);
- $a\bar{R}b, b\bar{R}a$: no direct and no inverse binary relation from a to b (Fig. 2d).

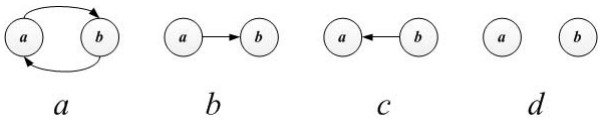


Fig. 2 Binary relations between structural elements

After the collection of all available information, all characteristics are defined and the structure of the topology is known. As the number of characteristics is finite number, it is possible to represent the topology in an oriented graph (i.e. arcs have directions) of the form $G(X;U)$, where X is the set of the functioning characteristics and U is the arc (link) set (Fig. 3). Mostly, in calculations and computer applications an incidence matrix representation is used (Table 1).

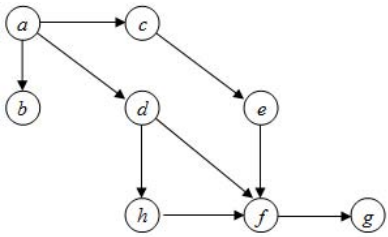


Fig. 3 Representation of a topological model in a graph

2.2 *The Construction Stages of a Model*

A topological model is developed in four steps:

1. Investigation and analysis of a characteristic set $Z = N \cup M$ of the functioning system. The set Z is distributed in two subsets:

   (i) N: internal features subset,
   (ii)M: external features subset.

2. Formation of the nodes set X that describes all important features of the functioning process;

Table 1 Representation of a topological model in the incidence matrix form

	a	b	c	d	e	f	g	h
a	0	1	1	1	0	0	0	0
b	0	0	0	0	0	0	0	0
c	0	0	0	0	1	0	0	0
d	0	0	0	0	0	1	0	1
e	0	0	0	0	0	1	0	0
f	0	0	0	0	0	0	1	0
g	0	0	0	0	0	0	0	0
h	0	0	0	0	0	1	0	0

3. Definition of the topology θ. The topology is defined by summarising the information about the cause-effect relations between X nodes. The cause-effect relations are marked as graph links. A link describes a situation when the change of one feature initializes or provokes the change of another feature without any other external influence. Definite links combined together form causal chains which can be used for logical reasoning and decision-making;
4. Topological model extraction from the topological space.

In the process of the model construction it is assumed that the system is closed and independent from the environment. Accordingly the graph is bounded which defines logical chains within the model [17]. The model allows analyzing the performance of the functioning system in the normal and in the incorrect state, considering a cyclic or non-cyclic performance. For a large scale and hierarchical system, a structural distributed form can be used. In the construction of a large scale model it is possible to use various methods for an extraction and combination of separated models.

It is possible to make the following assumptions:

- A system with an internal feature set L can be a subsystem of P only if $[L] \cap P \neq \emptyset$;
- The systems L and P with their internal features set are independent if $[L] \cap P = L \cap [P] = \emptyset$.

2.3 Model Homomorphism

In the process of developing a model it is often necessary to specify structural parts of the graph, e.g. analyze one of the graph nodes more deeply. In this case, the node is extended to several new nodes and together they form a new additional subgraph.

Graph homomorphism provides a mutual tieback and mapping between basic and detailed graphs. In graph theory it describes the mutual depiction of the separated graphs, maintaining the structural features of each graph [4, 5].

Graph homomorphism f or graph homomorphic depiction from a direct graph $G(X,U)$ to a direct graph $G'(X',U')$, written as $f : G \to \acute{G}$ is a mapping $f : X \to$

$\acute{X}$ such that $\{f(x_i), f(x_j)\} \in U'$. The two graphs G and $\acute{G}$ are homomorphically equivalent only if $G \to \acute{G}$ and $\acute{G} \to G$.

If in the graph G between two nodes $x_i, x_j \in X$ exists a direct link $q = \{z_i, x_j\} \in U$ then in the graph $G' = f(G)$ the link $f(q) \in U$ is orientated from portrait $x'_i = f(x_i)$ to the portrait $x'_j = f(x_j)(x'_i, x'_j \in X')$. The nodes of the graph G can be portrayed as a single G' node. An example of homomorphism is given in Fig. 4.

The main properties of homomorphism are:

- A homomorphical portray frequently allows to specify the initial topological model;
- A complex system can be created as a separated homomorphical graph structure by creating substructures (prototypes) of the system with definite detail level;
- The use of detail levels allows visually and logically separate functioning levels of the system;
- By using homomorphism in the structure analysis it is possible to perform a transition from the set of model features to the models parameters set.

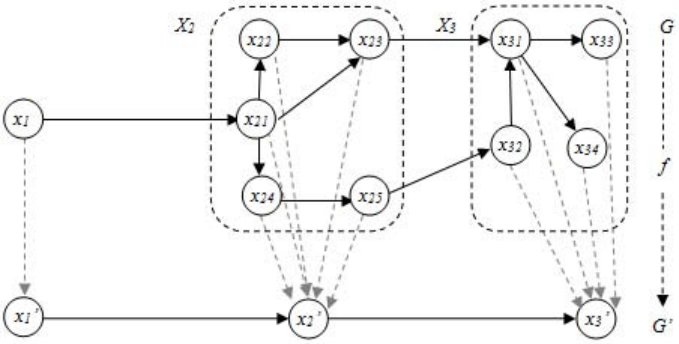


Fig. 4 Graph homomorphism

2.4 Decomposition

Structure of the system can be created and observed in various detail level. These degrees define feature sub sets. A possibility to create detail levels of the system is provided by one of the most important functions of topological model- decomposition. Decomposition allows to observe functioning features in a more detailed view. It is used also to cut out the graph parts. For a functioning feature a subset with nodes is created and then by making further subsets it is possible to create hierarchical levels of the topological model. Decomposition example is given in Fig. 5, where nodes $\acute{x}_2$ and $\acute{x}_3$ of the graph $\acute{G}$ are expanded to the set $X_2 = \{x_{21}, x_{22}, x_{23}, x_{24}, x_{25}\}$ and $X_3 = \{x_{31}, x_{32}, x_{33}, x_{34}\}$, respectively. After the creation of the new subset, mutual links between graph elements are re-established.

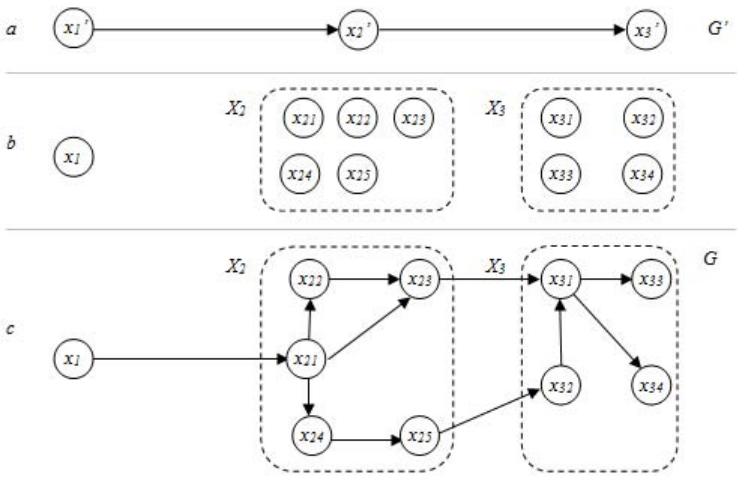


Fig. 5 Graph decomposition

3 Topological Model of Multiple Diseases

One of the main advantages of topological modeling is the possibility to use and in the model include various types of knowledge that can be expressed in numerical, functional or non-analytic form. From this feature follows that the model can be used to mathematically describe a heterogeneous system in a situation when the available information is insufficient. Physiological, biological and chemical processes often have complex structures. They are complex because their features are not structurally even and it is not possible to use differential or other strict type mathematical equations. In the field of medicine, pathogenesis[2] has been defined as a functioning process, which consists of complex physiological processes that are interrelated among each other and arise when the organism is affected by pathological exogenous or endogenic factors. These processes emerge in the form of clinical syndromes and symptoms. The topological model of a pathogenesis is created by summarizing the knowledge of medics and the overall used treatment keynotes and strategies of a disease.

3.1 Topological Model Elements

To analyze a process of the pathogenesis physical type, models of the graph structure can be used [12, 21, 22]. Miscellaneous topological models for differential diagnostic, diagnostic parameter selection, prediction of a diseases' state and therapy selection has been created [14, 15]. To perform all these tasks one common type

[2] Pathogenesis- the science of the causes and effects of diseases, especially the branch of medicine that deals with the laboratory examination of samples of body tissue for diagnostic or forensic purposes.

of a topological model is used and slightly adjusted to the each subtask. It is possible to adjust the topological model for modeling and analyzing a state of disease by defining all structural elements of the model:

- Graph nodes describe the pathogenesis occurrences, processes, parameters and other features:

  1. The circle nodes x are used to describe the organism subsystems, also called as pathogenesis basic mechanisms (Fig. 6a). A node describes each organism subsystem and its functioning level, which is involved in the pathogenesis process (examples are given in Table 2). All x nodes form a set $X = \{x_1, x_2, ..., x_{xl}\}$;
  2. The triangle nodes t defines external treatments, which are used as recovery strategies. In the pathogenesis example, recovery strategies are medicament groups (Fig. 6b). Medicament group examples are given in Table 3. All medical treatments form a set $T = \{t_1, t_2, ..., t_{tl}\}$;
  3. The square nodes y describes side effects of a therapy (Fig. 6c). and they are collected in a set $Y = \{y_1, y_2, ..., y_{yl}\}$. Frequently therapy side effects overlap and create a united set of them. Examples of the side effects are given in the Table 4.

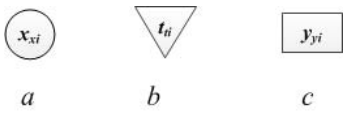


Fig. 6 Nodes of the pathogenesis topological model

- Links show cause-effect relations between graph nodes, where one node by changing its functioning level initiates a performance change of the other node. The link between two nodes can express a linear or non-linear correlation between these functioning levels. Also analytic expressions and the production law `(If..., Then...)` rules can be used to describe the essence of the link. In the process of the topological model development, the link definition is a time consuming procedure and it has the highest importance because links defines the topology of the model, i.e. the structural behavior of the system. Within the process of link definition a large amount of physiological, chemical and biological regularities are taken into account. Usually, interrogatory methods of an expert group are used [13]. The model can include four different link types:

  1. A link between two organism subsystem nodes x_{xi} and x_{xj} (Fig. 7a). This link is used to describe a basic mechanism of a pathogenesis by using a functional expression. Each link is unique and has an individual description of the relationship between these two nodes;
  2. A link connecting the therapy t_{ti} and x_{xj} (Fig. 7b) defines the therapy influence on the basic mechanism of the pathogenesis. The link is used to connect the overall process of the pathogenesis with the external influence (therapy);

Table 2 Organism subsystems' nodes examples

Abbr.	Title
x_1	Hypothalamus zone hyperactivity
x_2	Sympathetic nerve system hyperactivity
x_3	Arteriol hyper alfa energy
x_4	Heart hyper beta energy
x_5	Heart frequency rise
x_6	Systolic volume rise
x_7	Heart minute volume rise
x_8	Systolic blood pressure rise

Table 3 Therapies nodes examples

Abbr.	Title
t_1	ACE inhibitors
t_2	Betablockers
t_3	Calcium antagonists
t_4	Diuretics
t_5	Central simpatolitics

Table 4 Therapy side effects examples

Abbr.	Title
y_1	Bronchial spasm
y_2	Diarrhea
y_3	Anxiety, angst
y_4	Restless sleep
y_5	Dryness in mouth
y_6	Overall weakness
y_7	Negative inotropism
y_8	Hyperlipidemia

3. A link between the therapy t_{ti} and the side effect y_{yi} (Fig. 7c) is used to describe an association between the dose of a therapy and the corresponding side effect;
4. A link from x_{xi} to y_{yi} (Fig. 7d) is used to describe the connection between the organism subsystem and the therapy side effect. The probability of the existence of this link is low, but in the real-life practice of medical doctors this occurrence is observed;

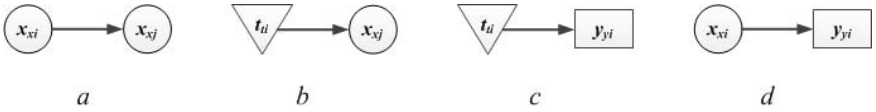


Fig. 7 Link types of the pathogenesis topological model

Each node of the model has an initiate functioning level, which for every individual human is different. Altogether, these functioning levels define the clinical conditions of the patient. Model nodes use the following functioning indicators:

- αx_{xi}: functioning level of the organism subsystem node x_{xi}. The value is given in the range from 0 to 1, where 1 indicates a fully normal functioning and 0 describes a malfunction. The initial value of αx_i is evaluated by a doctor by using the value definition table. In the Table 5, an example describing the functioning level of the organism subsystem 'Sympathetic nerve system hyperactivity' is given;
- βy_{yi}: functioning level of the side effect y_{yi}. Also this value is in the scale [0, 1] and given by a doctor. The value definition is similar to the functioning level definition of αx_{xi};
- τ_{ti}: therapy dose. Quarter (1/4), half (1/2) and full (1) doses (values) are used in equations. The therapy dose level does not change during modeling and analyzing a pathogenesis.

3.2 Development of the Pathogenesis Model

Within the framework of the authors' research work a model for united arterial hypertension (AH), atherosclerosis (AS) and diabetes mellitus (DM) have been developed. These three diseases have been consciously selected because they show common manifestations, has a similar progress mechanism and they partly involve equal organism subsystems. They are mutually connected and frequently one disease provokes other one and they cannot be divided.

The AH model is used from the results of previous research and developed medicine computer system [8, 14, 15]. The model is edited, simplified and used for the construction of the united pathogenesis model. In contradiction to the previous AH model, in which diagnoses of the pathogenesis is the main task to solve, the contemporary model is used to provide information for the selection of the optimal therapies complex[3] in the situation when multiple diseases appear. By adding new AS and DM models, the united pathogenesis model has been created to solve new tasks:

[3] Therapies complex is a multiple medicament combination which is used as a treatment for the patient. In the following when referring to 'therapies complex' it will be used the term 'therapy'

Table 5 Definition scale of the functioning level of the organism subsystem

ID	x_2	
Name	Sympathetic nerve system hyperactivity	
Value	Evaluation by clinical symptoms or by:	
	Noradrenaline d/n excretion mg	Adrenaline d/n excretion mg
	$\leq 100 - 0.1$	$\geq 30 - 0.1$
	$90 - 0.2$	$25 - 0.2$
	$80 - 0.4$	$20 - 0.4$
	$70 - 0.5$	$17 - 0.5$
	$50 - 0.7$	$13 - 0.7$
	$40 - 0.8$	$11 - 0.8$
	$\leq 30 - 1.0$	$\leq 8.0 - 1.0$
	$n = 28.32 \pm 7.52$	$n = 7.81 \pm 1.53$

- The united pathogenesis model is created to investigate the mutual relations between three diseases;
- The model can be used to predict the influence of a therapy;
- The efficiency of each therapy is evaluated by performance criteria. These criteria are the basic information for the optimal selection of a recovery strategy.

At first, a primary model is created on pathogenesis level (Fig. 8). In the second step of the model construction, external influences are added by using the graph composition method. In the composition procedure, all therapies are defined by a label, name and influenced nodes set (Table 6). Each therapy is represented as a graph mini model (Fig. 9) and consists of the therapy node t_{ti}, corresponding side effects and organism subsystem functions which are influenced by the therapy. The final step in the composition process is the adding all mini models to the primary model. At first, therapies are linked through the influenced nodes (Table 3 column 4) to the primary topological model. Then the side effects of the therapies are merged. In Fig. 10 the final united pathogenesis topological model is given. In the model, five different markings (Fig. 11) of nodes are used to portray the membership of a node to the particular pathogenesis. Based on this marking and by decomposing the graph, separated pathogenesis topological models are created (Fig. 12a arterial hypertension TM, Fig. 12b atherosclerosis TM and Fig. 12c diabetes mellitus TM). From the common topological model and separated sub-models it is visible that there do exist nodes which belong to several pathogeneses. These are pathogenesis transition nodes, which are the key elements analyzing the mutual relations between diseases. The decomposition of the united model is used to analyze separate pathogenesis.

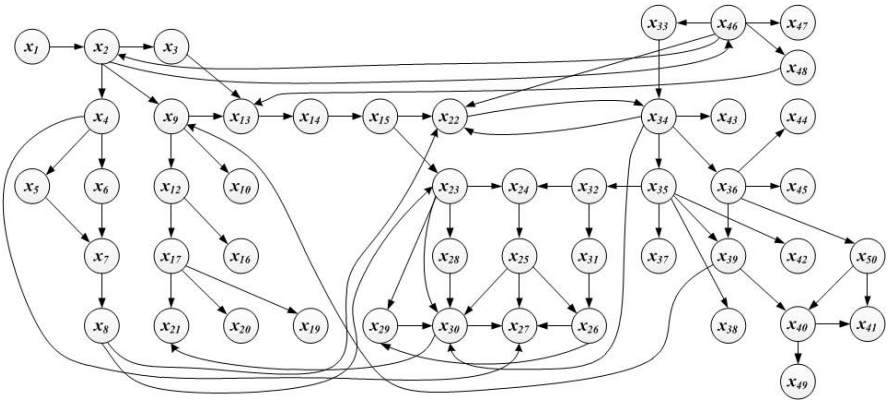


Fig. 8 Primary topological model of multiple pathogenesis

Table 6 Therapy side effects' nodes examples

Abbr.	Name	Side effects	Influenced nodes
t_1	ACE inhibitors	$Y_{t1} = (y_{14}, y_{15}, y_{16}, y_{17}, y_{18}, y_{19})$	$X_{t1} = (x_9)$
t_2	Beta-blockers	$Y_{t2} = (y_1, y_2, y_3, y_4, y_6, y_8, y_9)$	$X_{t2} = (x_2)$
t_3	Calcium antagonists	$Y_{t3} = (y_7, y_{11}, y_{20})$	$X_{t3} = (x_{13})$
t_4	Diuretics	$Y_{t4} = (y_6, y_{12})$	$X_{t4} = (x_{17})$
t_5	Central sympatholytic	$Y_{t5} = (y_5, y_{13})$	$X_{t5} = (x_1)$
t_6	Selective alfa- blockers	$Y_{t6} = (y_5, y_{10}, y_{13})$	$X_{t6} = (x_3)$
t_7	Statins	$Y_{t7} = (y_{13}, y_{21}, y_{22}, y_{23}, y_{24})$	$X_{t7} = (x_{22}, x_{23})$
t_8	Anti diabetes therapy	$Y_{t8} = (y_2, y_6, y_{12}, y_{21})$	$X_{t8} = (x_{46})$

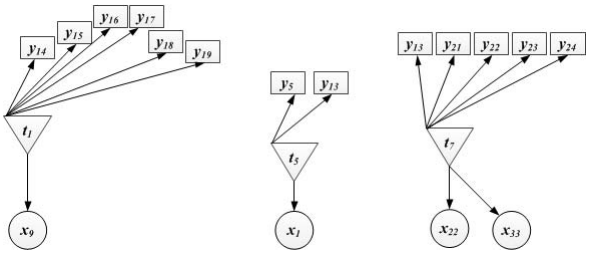


Fig. 9 Examples of therapy mini models

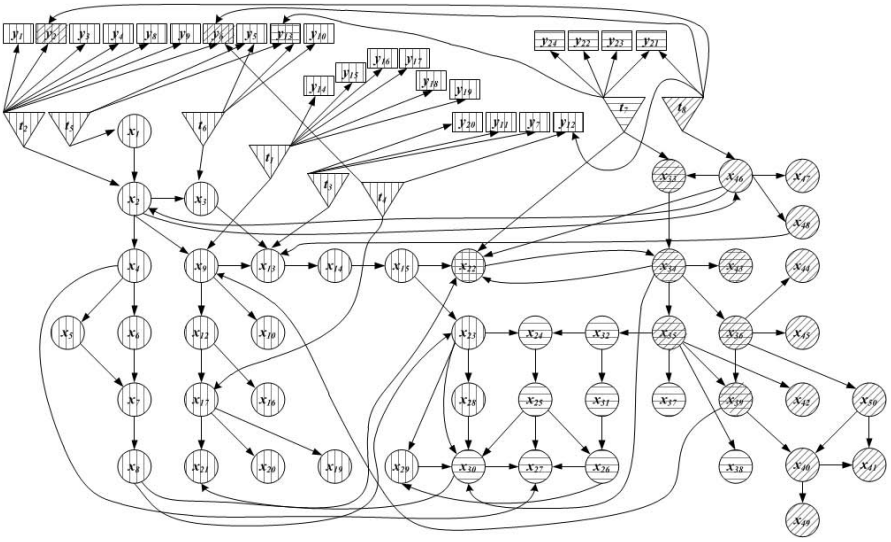


Fig. 10 Final united topological model of multiple pathogenesis

Fig. 11 Examples of topological model marking

4 Decision Support System for Optimal Solution Selection

The third stage of solving the problem (see introduction) is the use of decision-making techniques for the recovery strategy selection for a functioning system. Strict mathematical equations and reasoning methods for the recovery selection in the pathogenesis case do not exist, but the developed model allows analyzing the process of the pathogenesis. The developed model can be used to model the performance of a recovery strategy (several therapies complex) and predict the ending state of the pathogenesis. The results of each therapy are evaluated by the following performance criteria [7]:

- The efficiency rate of the therapy influence shows how each therapy changes the health state of the patient. The rate is found by modeling the performing therapy, which mathematically is a result of recalculating all functional levels of the topological model nodes and summarizing the changes in one value. The recalculation is based on the best-first search method [20], performed from each therapy node;

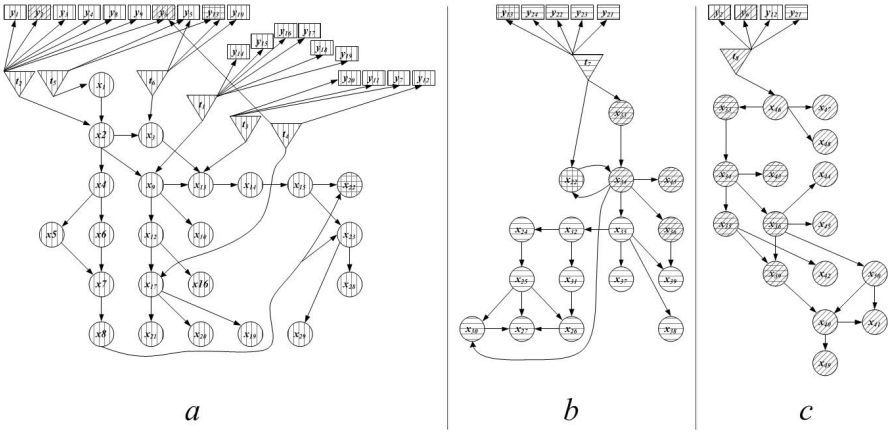


Fig. 12 Topological models for separated pathogenesis

- A complex influence rate to an essential part of the process. As the pathogenesis process is heterogeneous, diverse process parts have a diverse influence (importance) rate. By using expert evaluation methods an importance level of the each node is assigned. The final importance level of the complex is calculated by summing up the importance levels of all covered nodes;
- The recovery coverage level is used to describe an influence spectrum of the therapy. The criterion indicates how the therapy covers a periphery and hardly reachable nodes of the model;
- Provoked side effects criteria is the first negative parameter to be minimized. The parameter is calculated by summing up the functioning levels of the provoked side effects of the therapy;
- A recovery complex cost is the second criteria with negative manner. The criterion is calculated by using expert group inquiry methods.

The above mentioned performance criteria are normalized in the common scale and the first three parameters are with positive manner (a higher value means better), but the last two have negative characteristics (a higher value indicates a worse solution). The common scale provides the possibility to compare therapies by these criteria.

As decision-making method, a multi-objective (multi criteria, vectorial) optimization is used. Optimization is an appropriate approach for a compromise solution selection in a situation when more than two objectives are conflicting [2, 9, 16].

All previously stated functionalities, including the topological model and therapy efficiency criteria, are implemented in the intellectual medicine system, which could be used as an assistant by the doctor in his medicine praxis. In the computer system, all input data (the initiate functioning levels of the organism subsystems and the therapy side effects nodes) have to be entered. For decision making five multi-objective optimization methods are used:

- Weighted sum method;
- Weighted goal method;
- Two objectives weighted sum method;
- E- constrain method;
- Lexicographic approach.

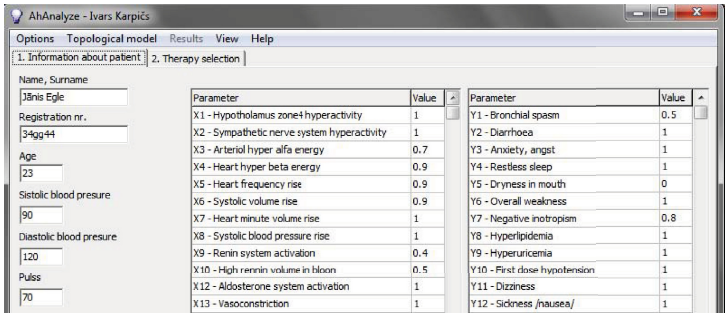


Fig. 13 Fragment of the data input window

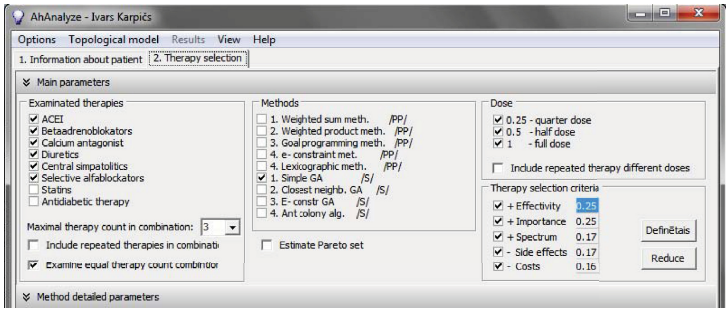


Fig. 14 Therapy selection parameters

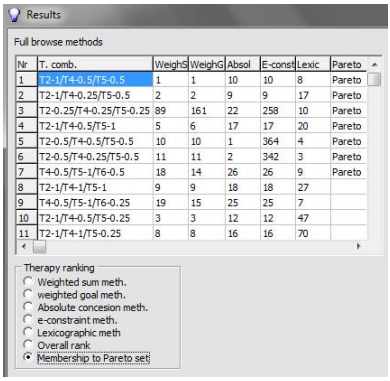


Fig. 15 Suggested therapies by five multi- objective optimization methods

Software includes multi language interface and all patients' data input (Fig. 13), (Fig. 14). As a result, the computer software provides all suggested therapies in a ranked manner, obtained by each of the optimization techniques (Fig. 15). This medicine computer system is still in the development phase and needs to be tested and improved.

5 Conclusion

The analysis of functioning systems is an actual task domain in computer science, mathematics and other engineering fields. An appropriately selected recovery strategy allows re-establishing the correct work of a system when the structural parameters and the behavior of the system is in dysfunction. One of the methods how to model the state of a system and predict its performance after the influence of a recovery strategy is the use of mathematical modeling and heuristic approaches. These methods model the state of a system before and after the recovery strategy. They provide calculus information for the task of selecting a recovery strategy.

In this paper, an approach of mathematical model development based on topological modeling is introduced. A formal description, structure elements and the features of the topological model are provided. Within the article, a practical example for the use of topological modeling is introduced. A united topological model for three diseases is created to analyst the process of each pathogenesis, predict the ending state of a patients health and evaluate the efficiency of the selected therapy. The united topological model is implemented in a medicine computer system, which consists of multi-objective optimization techniques for the optimal recovery strategy selection. The developed computer system is used as a testing system for the topological model and decision-making techniques. By further improvements of the decision-making techniques, the computer system could be used as an advice or guidance system for the praxis of personalized medicine.

Acknowledgements. This work has been supported by the European Social Fund within the project 'Support for the implementation of doctoral studies at Riga Technical University'.

References

1. Bender, E.A.: An Introduction to Mathematical Modelling, p. 272. Dover, New York (2000)
2. Collette, Y., Siarr, P.: Multiobjective optimization: principles and case studies, 1st edn. Corr 2nd printing, p. 293. Springer (2003)
3. Grundspenkis, J., Isajeva, L.: Qualitative Analysis of Complex Systems within the Framework of Structural Modelling. In: Proceedings of the International Conference Modelling and Simulation of Business Systems, pp. 180–184. Kaunas University of Technology Press, Technologia (2003)
4. Hahn, G., Tardif, C.: Graph homomorphisms: structure and symmetry. In: Graph Symmetry. ASI ser. C, pp. 107–166. Kluwer (1997)

5. Hell, P., Nesetril, J.: Graphs and homomorphisms. Oxford lecture series in mathematics and its applications, p. 244. Oxford University Press (2004)
6. Kamat, R.K.: Modeling and analysis of biologically inspired robot. International Journal of Advanced Computer and Mathematical Sciences 1(1), 7–11 (2010)
7. Karpics, I., Markovics, Z.: Development and evaluation of normal performance recovery method of a functional system. In: Scientific Proceedings of 9th IEEE International Symposium on Applied Machine Intelligence and Informatics (SAMI 2011), pp. 171–175 (2011)
8. Karpics, I., Markovics, Z., Markovica, I.: Composition of united multiple diseases evolution topological model. In: IEEE 9th International Symposium on Intelligent Systems and Informatics (SISY 2011), pp. 65–69 (2011)
9. Keeney, R.L., Raiffa, H., Decision, H.: with Multiple Objctives: Preferences and value tradeoff., p. 569. Cambridge University Press (1993)
10. Kolmogorov, A.N., Fomin, S.V.: Elements of the theory of functions and functional analysis. Metric and Normed Spaces, vol. 1, p. 129. Dover, New York (1957)
11. Kolmogorov, A.N., Fomin, S.V.: Elements of the theory of functions and functional analysis. Measure. The Lebesgue integral. Hilbert space, vol. 2, p. 128. Dover, New York (1961)
12. Laue, R.: Elemente der Graphentheorie und ihre anwendung in den biologischen Wissenschaften, p. 137. Akad. Verlagsgesellschaft, Leipzig (1970)
13. Markovics, Z.: Expert evaluation methods (RTU, Latvia), p. 110 (2009)
14. Markovics, Z., Markovica, I.: Variants of Computer Control for Systems Recovering. Scientific Journal of RTU. 5. series Computer science - 6. Computer control technologies (RTU, Latvia), 6–15 (2001)
15. Markovics, Z., Markovica, I., Makarovs, J.: The Alternative Conception on Therapy Selec-tion. Scientific Journal of RTU. 5. series., Computer science - 11. Computer control technologies (RTU, Latvia), pp. 19–27 (2002)
16. Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Struct. Multidisc. Optim. 26, 369–395 (2004)
17. Osis, J.: Topological Model of System Functioning. Automatics and Computer Science. J. of Acad. of Sc, Riga, Latvia, 44–50 (1969)
18. Osis, J., Asnina, E.: Model-Driven Domain Analysis and Software Development: Architectures and Functions, p. 489. IGI Global, New York (2011)
19. Osis, J., Sukovskis, U., Teilans, A.: Business Process Modeling and Simulation Based on Topological Approach. In: Proceedings of the 9th European Simulation Symposium and Exhibition, Passau, Germany, pp. 496–501 (1997)
20. Pearl, J.: Heuristics: Intelligent Search Strategies for Computer Problem Solving, p. 48. Addison-Wesley (1984)
21. Rashevsky, N.: A remark on the possible use of nonoriented graphs in biology. Bulletin of Mathematical Biology 30(2), 30 (1968)
22. Rosen, R.: A relational theory of the structural changes induced in biological system. Bull. Math. Biophysics 23(2), 165–171 (1961)
23. Stalidzans, E., Krauze, A., Berzonis, A.: Modelling of energetical balance of honeybee wintering generation. In: Proceedings of 1st European Scientific Apicultural Conference, pp. 55–62. Psczelnicze zestyty naukowe, Poland (2000)
24. Zbigniew, M., Fogel, D.B.: How to Solve It: Modern Heuristics, 2nd edn. Revised and Extended, p. 554. Springer, Heidelberg (2004)

Some Aspects of Knowledge Approximation and Similarity

Aleksandar Jovanović, Aleksandar Perović, and Zoran Djordjević

Abstract. In the representation and processing of knowledge, the need for knowledge approximation and similarity is frequent. Concepts needed to deal with these issues are emerging; however, the unified treatment is still missing. With this in mind, we discern two sorts of knowledge, continuous represented, which we usually meet in the sensory based knowledge systems, and the other of discrete information structures, involved in automatic reasoning and processing of symbolic sequences in various contexts. The issues related to knowledge similarity and approximation are discussed in some special cases (separately following our initial knowledge division), which are spanning more general field and is used for exposition of our ideas and solution proposals. We have presented the concepts of knowledge similarity and approximation, together with appropriate scaling - degrees of similarity/approximation, in metric spaces for continuous case and the solution for discrete information spaces- DIS spaces for the later case.

Keywords: Knowledge approximation, similarity, molecular biology, chromosome, gene.

1 Introduction

Overture

Axiom of Comprehension is generally acceptable and accepted, except for the treatment of some most abstract and fundamental issues. In practice everyone applies it.

Aleksandar Jovanović · Zoran Djordjević
University of Belgrade, Faculty of Mathematics,
Group for intelligent systems, Studentski trg 16, 11000 Belgrade, Serbia
e-mail: {aljosha.jovanovich,dzoran1}@gmail.com

Aleksandar Perović
University of Belgrade, Faculty of Transportation and Traffic Engineering,
Vojvode Stepe 305, 11000 Belgrade, Serbia
e-mail: pera@sf.bg.ac.rs

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 255–282.
DOI: 10.1007/978-3-642-33959-2_14

Extensionality determines concepts as sets; thence, operations on concepts become set operations. When we deal with operations on concepts, we have the intuition of operations similar to the operations on other things. The knowledge systems operate with concepts in different ways. Here, the concepts of conceptual similarity and approximation of concepts are in general use, less or more openly. We expect that the concept approximation in such operations behaves in a manner somewhat similar or approximate to the way the continuous functions work: the slight change of argument results in a slight or controlled change of the value of the function. We turn our attention to the contexts where such analogy is extensively exploited, extracting the involved mathematical representations of two sorts, continuous and discrete information structurality, as key concepts in conceptual modeling and discuss the nature of similarity and approximation on such representations. The concepts of fuzzy sets enriched the need for mathematical representations of concepts to operational degrees needed in large variety of contexts, involving general life, scientific research and technological development. In fact, practically all concepts in practice are fuzzy concepts and fuzzy sets, hence the operation on concepts needs to be tuned respecting this fact, and hence all operations on concepts are fuzzy operations. This does not pass by the concepts of similarity and approximation.

Referring to knowledge, knowledge manipulating systems and the tasks involved in their processing and management, it is common to think of knowledge within systems processing sensory information or those involving formal contexts. The former, systems integrated over the sensory inputs and sensory knowledge, are in accelerated broad development and already include all inventory of Artificial Intelligence - AI. Now days, we have highly efficient systems of rather diverse sorts and high complexity, where the intelligence is a key ingredient. Analysis and recognition tasks involving ultrasound, radar, visual, infrared, X-ray, MR, other sensory inputs in science, technology, military, all share a common corpus of modeling, simulation and solution methods. Pattern representation, classification and similarity are of key importance and are indispensable. The later, materialized in the processing of formalized contexts, traditional information systems, intelligent information systems, engage higher or lower intelligence/reasoning capacity, depending on the complexity of mathematical models in their cores. The fundamental issues are well established with respect to the involved methods of representation and modeling of phenomena in their scope, utility and way of implementation. The interesting and important examples would include the natural or formalized languages, with the common structurality, inference and processing of logic structures in automated reasoning, correctness and formal issues in programming, sequence processes in molecular biology and genomics, music structural modeling and processing. There are rather simple demands posed by practice which need treatment and solutions.

Evolution of the methods in these largely separated problem areas was divergent, and our past practice kept them disjoint, mainly due to the general separation of Mathematics of continuum and discrete Mathematics, or more precisely, Mathematics of continuum and Mathematics of discrete information structures (structurality), DIS, as two distinctive ways of observing and understanding nature and its expansions. Finally, it is worth mentioning that the two areas are not so far apart as it

might seem, for, on one side we have the continuum type of modeling with classic distance/metrics as prevailing, while, on the other, the information-structural kind of modeling, with a need for similar concepts of similarity and distance-metrics on discrete information bearing structures. Note that the fusion of syntactically structured intelligence into complex sensory based systems is generally present either at the top layer - intelligent hat, or embedded and distributed within its hierarchy.

Here we discuss models and representations, presenting the material on the important examples and treating the central concepts, similarity and approximation of knowledge in both areas, which is of basic importance for all sorts of knowledge extraction and knowledge management. In the selection of illustration examples we preferred to offer some which are inner counterpointed. We chose interesting examples of transversal relationship from our practice, which are partly available on our web, as presentations/software

`http://poincare.matf.bg.ac.rs/~aljosha//GISS/sbgis.htm`

In this way our semantically independent examples are in function of spanning broad space of methods, as e.g. the linearly independent vectors are used in spanning of subspaces.

2 Method

We will present some specific cases of knowledge representation, similarity and approximation, which belong to both problem areas, possessing characteristics and invariants which could be well generalized and applied elsewhere. Besides, we discuss more general issues. Among the selected examples, there are some with inherent duality in their nature, thus having both continuous characterizations with similarity modeled in metric spaces, and discrete nature at some other point of view with dominant discrete information structurality and similarity related to such structures. In some cases, this far distance of mathematics applicable, depends on certain operational aspects used in problem perception and definition, or even on some subjective preferences.

The concept of similarity originates in geometry where it is properly defined and extensively used. The geometric similarity is an equivalence relation, preserving the relationships of fundamental invariants of objects and shapes. The measurement corresponding to this relation is based mainly (but necessarily) on Euclidian distance, which induces continuous notion of similarity approximation for the objects derived from originals by some geometric deformation/mutation, in a minor or larger degree. The need to equalize the measured objects by their volume measure arises already here. Take e.g. two similar simple polygons, one with the 1mm diameter, the other with 100mm diameter. In spite of the fact that their invariants are all in identical relationships, often in practice, we would not consider one as an even rough approximation of the other (such considerations are not applicable to self similarity, where similarity is the part of recursive pattern characterization). Thus, similarity is of highest importance in similarity comparison in practice; however, the congruence is really the starting point in a variety of modeling of similarity and approximation,

maintaining the global weight - e.g. geometric volume rather close. These concepts simply generalize to metric spaces with similarity relation, $(X,d,\sim)$, where X is any set (dots, vectors or functions), d is a distance in X, $\sim$ a similarity relation; similarity measurement and approximation are based on a chosen metrics, appropriate for the invariants of concern. Arguably, among the simplest examples of this kind is the concept of similarity of two curves in the metric space $(C^{\infty}[0,1],d)$, where d is the uniform metrics on $C^{\infty}[0,1]$ defined by $d(x,y) = \sup\{|x(t) - y(t)| : t \in [0,1]\}$. Two curves x and y are said to be similar iff $d(x,y) < \varepsilon$ (here ε is some fixed positive threshold). Note that the mentioned similarity relation is not transitive.

When we deal with similarity and approximation of discrete information structures, something remains similar to the context of continuous similarity and approximation, but some other properties demand attention as well. Here, similarity is simple in the special case of identical structures, including some simplest alterations. The same applies to the need to maintain mathematical invariants to a satisfactory degree. Here, geometric and metric invariants in metric spaces are replaced by general structural invariants. The geometric distance and distances in metric spaces capturing in continuous fashion computational necessities in similarity assessment in continuous cases, do not have proper counterparts in the case of discrete information structural similarity and approximation assessment. One of the important reasons is that we do not have proper metrics in discrete information structure (DIS) spaces, which would be the vectors of the sort $(X,d/\tau,\sim)$, where, now, X consists of informational structures, e.g. logic formulae, recursive functionals or relationals, type structures, programs, music pieces or linguistic contents, automats /machines, structures in molecular biology; τ is some sort of tree topology appropriate for nearness/separation definition, d a distance like function in X compatible with τ, but not necessarily a metric, and $\sim$ some corresponding similarity.

However, we immediately know that we need structural similarity and approximation properties similar to those in metric spaces - thus securing that the nearer is better approximating than the further away, that similarity relationship and measurement methods comprise identity as a special case, and that these methods should be somehow smooth (continuous) in the degrees of similarity, starting from identity. That means that non smooth switching of the similarity/approximation methods between the different structural levels is strongly undesirable. Generally, we cannot exaggerate in insisting in the harmony of perceived context. E.g. in the genetic examples presented below, DNA at the level of chromosome packs, expressing many important genetic properties as chromosome features and at the level of molecular - nucleotide sequences, expressing partly different kind of genetic issues, we deal with two representation spaces, one metric continuous and smooth, usually observed in the magnifications of few thousand (limits of optic microscopy) or at a couple of higher magnitudes (electron microscopy), while the other, discrete, with nucleotides visible, just one magnitude still higher (atomic force - AF or scanning electron, the SE microscopy). Thus, continuous and discrete phenomena are distant approximately one order of magnitude, since metric spaces representing the chromosome level of phenomena and the space of genetic discrete informational structures, in these fundamental genetic examples are distant one order of observational

magnitude. Within a broad interval of magnifications we observe the continuous nature and related mathematical invariants of phenomena, while one magnitude higher resolving molecular structure, immediately switches context to the symbolic sequence structures.

Never the less, the subject has been growing in importance continuously over last few decades and there are numerous solutions appropriate to the concepts involved and computational demands, with special algorithms realized and they represent the accumulated richness of this area development.

2.1 Selected Examples

As examples of sensory recordings which are used as direct or transformed pattern inputs into knowledge systems fed by sensors, we show a few from our own experience. They do not present a representative collection; we use them to illustrate our arguments. Any other sort of knowledge system feeding using sensory information, should have a sort of problems involved, feature representation and processing, finally, pattern matching and other resolutions based on some criteria, which often need to be fully automatized, which are common to some extent to our examples. It is important to notice that all sensory information, within the appropriate knowledge systems, are mapped to a predicative - formalized form, suitable for the formal treatment and this is done by some semantic mappings, which are generally fuzzy functions, e.g. in the simple case we have a semantic mapping of Real World Sensory Features into the entities of a Fuzzy Relational Data Base, the metamorphosis aggregation, which could be a part of a more abstract DIS space.

$$f : RWSF \longrightarrow FRDB \tag{1}$$

(examples: classifiers [1, 2], thought mapper [3]). Obviously, before such a mapping can take place, the sensory features need to be identified, localized and extracted, and, often, transformed within some normalization procedures. The shown material is briefly commented, stressing only the points needed for our main purpose. More elaborated treatment, covering all related problems and methods is presented in the largely growing literature and listed references. We have developed systems for processing of signals and images of various origin, including electro cardio, blood pressure, intraocular pressure, polygraphs, electro encephalography - brain, speech, acoustic, music, sonar, radar, magnetic resonance, radio telescope, telescope, micro wave, microscopy, other, providing structured insights into sensory data (see [4, 5]).

In the analysis of signals, Fourier spectroscopy, using classic or other orthonormal systems of functions provided superior and unique insights into the vast reality of investigated phenomena observable by very reach collection of sensors and represents a major method for determination of the observed reality inherent mathematical invariants. The orthonormality of base system of functions is a necessary property in Fourier developments and systems of functions used for similar deconvolutions lacking it are of rather reduced semantic value, see e.g. [6].

We will briefly comment the offered experimental examples, skipping operational details. In the Fig. 1 left, we present recorded signals with speech, arterial pressure, electroencephalogram, heart rate in the left column, then, in the right column, the beginning of the famous music performance, intraocular pressure, submarine echogram, and encephalogram of acoustic stimulus - the tune c1 played on organ. As immediately noticed, the shown signals exhibit certain specific properties which are directly observable and measurable. Some of them are apparently rather similar, in spite of their distant semantics, e.g. right column first two signals (beginning of piano performance and intraocular pressure), or the signals in the last raw (heart rate and an EEG signal). Besides apparent and fluctuating importance, directly observable properties generally lack characteristics which could be considered as information of dominant significance and, hopefully, closely related to the major intrinsic nature of recorded phenomena. Importance of individual combinatorial local or global properties (local extremes, saddle points and other locally focused or global and statistical properties), generally considered as significant, in the situations of higher complexity could be rather limited, as the knowledge of those will not explain what is really going on in the nature of the observed phenomena. However, Fourier spectroscopy and its modifications and generalizations (here shown classic Fourier time -spectrograms) might provide the desired insights to a desirable degree. Often, when we face complex situations, mathematical theorems are not tightly related to the problem solving. Rather, we are at the entrance of the new, large space of mathematical experimentation, where certainty and safety we used to have during the history of Mathematics become looser, opening variety of deep gaps, not all well visible. When applying Fourier spectroscopy, in order to reach meaningful semantics some demands need be met. The first is the spectral stability. If we have no information on how dominant spectral structures change in time, we can easily make misleading or wrong conclusions.

Secondly, using different spectral resolutions one needs certain estimation of dynamics of investigated properties. For example, the periodic components of signal which are short in time, relative the length of the time window analyzed, will be presented with reduced or even much reduced amplitudes, which is affecting also the apparent amplitudes of components with relatively fast frequency changes. Finally, in a variety of applications, individual spectra do not exhibit the relevant information, rather spectral time traces - spectrograms posses the information which correspond well to the phenomena investigated. Again, in spectrograms there are subtle issues related to the semantic stability, which we shall omit here (more details are available in [7]). In a large number of issues involving spectrograms, there are certain patterns that could be important, the spectrogram features. They can vary in size and topological properties. Roughly, we distinguish small size features - i.e. those corresponding to the short frequency pulses in signals, then, aggregations of small size or dot like objects, with certain statistical properties characterizing them. Opposite to the former, there are topologically contingent aggregations which have distinguished contours and variety of shapes. All sorts of features correspond to the invariants in investigated signals.

In Fig. 1, right and Fig. 2, are shown spectrograms of recorded speech, intraocular pressure and submarine echo, from signals in Fig. 1- left, exhibiting quite well features which correspond to the important characteristics - invariants of processes present in the recorded signals. Those distinguished features can be used to detect and determine the contents in the signals which are of major importance to us.

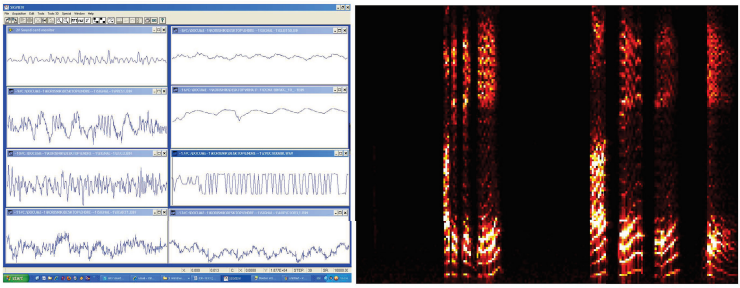


Fig. 1 left image: top raw left, speech capturing, while pronouncing repeatedly: "A, C , T, G"; top raw right, recording of G. Gould -Aria from Goldberg variations; second raw: recordings of arterial and intraocular pressure; third raw: recordings of electro encephalogram - EEG and submarine sonar; forth raw: heart rate and EEG recording of stimulation with the organ c1 tune. Right image: real time spectrogram of the speech signal in the first raw.

In Fig. 3, spectrograms of electro encephalographic signals in the experiments with imagined activity (imagined tone on the left and imagined motor activity on the right) are shown. The characteristic features corresponding to the imagined activity are revealed, which can be used to determine the distinguished controllable states, for the brain computer interfaces, BCI ([8]-[14]). The electro and magneto encephalography (EEG and MEG), reached high development acceleration recently, expecting more from new methodologies based on brain connectivity modeling and both method and technology successes in high resolution EEG and MEG tomography (see [15, 16]).

In Fig. 4 on the left, we have a spectrogram of the beginning of the performance of J. S. Bach's BWV 988, with the spectral features directly corresponding to the music text on the right, though the proper recognition would demand some preprocessing (e.g. spectral recalibration (e.g. [7], or some advances involving separation of individual tunes). Music structure is determined by frequency, duration, intensity and color of individual tunes and their distribution in time. Often, the first two tonal attributes are treated individually, while the remaining two are sometimes defined more generally - e.g. for a part of a musical composition, or, sometimes left unspecified. The problem of recognition of music structure from the recorded/played music has a few approaches, from simplified to those comprising more complex needs, which results in a sequence of a few recognition degrees (the more complex methods refining the simpler).

In the case of simpler melodious patterns generated on piano, organ or other instruments with fixed tonal values, the fundamental recognition would consist of a

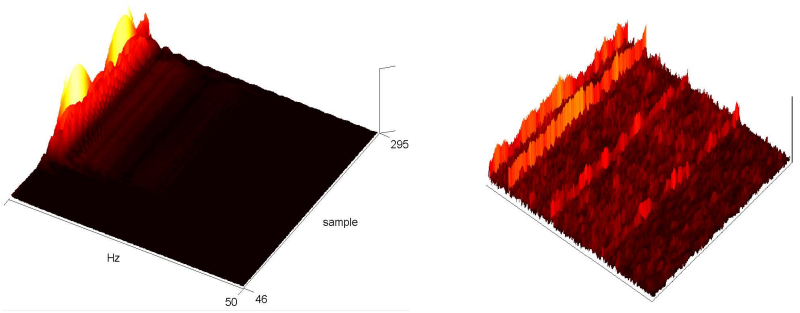


Fig. 2 left: the spectrogram of signal of intraocular pressure, exhibiting characteristics of normal eye pressure; right: the spectrogram of submarine echogram, showing the presence of other vessels with frequencies characteristic to their engines, which can be used for object classification and estimation of location and speed

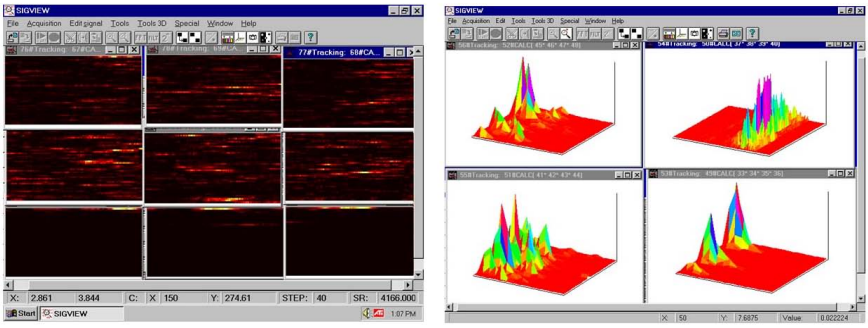


Fig. 3 spectrograms of EEG recordings; left: features showing traces of imagined tones; right: features corresponding to the imagined actions of the kind which are used as different controllable states, as command states in brain computer interfaces - BCI

generation of music structure in some of the well known formal denotation systems, like one developed by d'Arezzo. That process has to start with the separation of basic tonal frequencies, their location in (time, frequency) space, continuing with comparison with the set of precalibrated tonal frequencies, in order to generate the formal structure, like in example in Fig. 4. That process results in generation of semantic mapping which would generate DIS (discrete information structure) output for a given music signal. In such simplified recognition, the individual tune intensity (and frequency) dynamics, as well as recognition of individual aliquots, their separation and their correspondence to specific tunes is omitted. However, the need of recognition of these attributes is not denied in recognition of more complex or music produced by string or wind instruments or in any serious performance analysis.

All other shown examples, share some problems and solutions with this specific example of music recognition. This include, localization of time-frequency

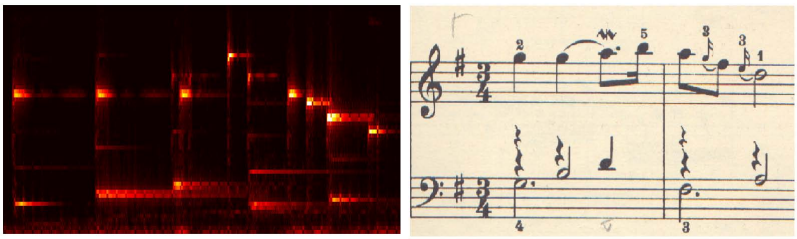


Fig. 4 On the left is the spectrogram of the recording of the beginning of Bach's Goldberg variations played by Glen Gould, showing some specific performance quality; with some simplifications, using certain fuzzy filters, it is possible to map the features in the spectrogram to their corresponding formal denotations in the standard (formal) system of music notation, as shown on the right side, or some other purely syntax form

features, after noise and artifact reduction, then matching (estimation of similarity/approximation) of localized features with some set of etalon objects, which would provide the desired semantic mapping of input signals into a discrete or formal structure. In this way, the features in submarine echo spectrogram, would provide information on certain characteristics of vessels in the area, like size, with classification of objects, behavioral parameter estimation and position localization, which could be the entries to the intelligent command system, which could provide commands for intelligent actions. In the case of arterial or intraocular pressure spectrogram features, they characterize the regular/normal physiological states, altered states and their causes, thus providing analytic knowledge of the investigated systems and physiology. The features in the spectrograms of electro encephalograms, characterize mental states, here, those corresponding to some imagined actions, which used as different controllable states are mapped to the (formal) commands of the BCI after recognition process. In the discussed examples, similarity degree and approximation estimation are used in the recognition processes, as instances in appropriate continuous metrics. Note that all shown features in spectrograms are with fuzzy contours. The relevant feature shapes could be analyzed using the methods for analysis of shapes in images as well, which are discussed further below. In simpler cases, the recognition - similarity estimation procedure can be performed using simple fuzzy filters, resulting in the metamorphosis semantical mappings of type (1).

Individual spectra are vectors. After noise and artifacts are removed, Euclidian metrics or modifications are applicable for the estimation of similarity of spectra, and their approximation. Thus, as broadly used in our examples, for spectra (more often power spectra) u and v, we can define the degree of similarity by

$$d_1(u,v) = \left(\sum_{i=1}^{n} |u_i - v_i| \right)^{\frac{1}{2}}, \tag{2}$$

or, (if normalized with respect to the first harmonic), by

$$d_2(u,v) = \left(\sum_{i=1}^{n} \left| \frac{u_i}{u_0} - \frac{v_i}{v_0} \right| \right)^{\frac{1}{2}}, \tag{3}$$

or by (using normalized sequences)

$$d_3(u,v) = \sum_{i=1}^{n} |\mathrm{sgn}(u_i) - \mathrm{sgn}(v_i)|, \tag{4}$$

or simply using indicators like dot vector product or the same on normalized sequences

$$I_4(u,v) = (|u_1 \cdot v_1|, \ldots, |u_n \cdot v_n|), \quad I_5(u,v) = (\mathrm{sgn}|u_1 \cdot v_1|, \ldots, \mathrm{sgn}|u_n \cdot v_n|). \tag{5}$$

Above metrics can be used similarly on boxed spectra (obtained by simply reducing the resolution by a boxing factor, then determining maximum in each box), as a way of simple discretisation of initial continuous structures. Rather than using standard metrics in similarity estimation, the use of their fuzzification is more appropriate. The prime examples of fuzzy metric spaces are structures of the form $(X, M, \cdot)$, where (X,d) is a metric space, $\cdot$ is the usual multiplication of the reals and $M : X^2 \times (0,+\infty) \longrightarrow (0,1]$ is defined by

$$M(u,v,t) = \frac{\alpha t^{\beta}}{\alpha t^{\beta} + \gamma d(u,v)}, \ \alpha, \beta, \gamma > 0. \tag{6}$$

In the case when $\alpha = \beta = \gamma = 1$ we get the standard fuzzification of d. Fuzzy metric spaces provide a natural environment for the study of similarity and approximation. Besides, fuzzification is consistently applied in recognition process, in a number of related techniques, as those using fuzzy filters and fuzzy tools (for general method see [17, 18]). The above metrics, other related recognition tools, and their fuzzifications easily generalize to spectrograms dominated by simple features, those which are constant in time (in the examples, spectrograms in Fig. 2 to 4), while, on more complex features other extended methods, e.g. like one developed for chromosome analysis could perform rather well ([7, 19, 20]). After noise cancelation, certain preprocessing, feature localization, extraction and normalization, similarity including the approximation quality/degree estimation could be properly represented by fuzzy metrics in many different contexts, enabling appropriate pattern recognition and thus providing semantic mappings of spectral features, as mathematical representations of relevant contents in the sensory inputs, into the set of referent objects, the entries in fuzzy relational data bases or other organized discrete structures, for classification and approximation determination, where the original feature is corresponded to the formal/discrete information in appropriate space of discrete information structures. In the sequel, we proceed with some examples of image analysis.

The first visible genetic structures are DNA packs, chromosomes, which form during cell division. Their purpose is to transfer the genetic material to the daughter cells. There are many genetic properties which are expressed at the level of chromosomes, e.g. individuals of the nearest species with differing chromosome numbers cannot have fertile descendants, and the difference in chromosome number is the first indicator of evolutionary branching; large number of genetic deficiencies are characterized by certain chromosomal changes; presence/absence of individual genes on chromosomes is responsible for an organism embryonic development, later life normal or abnormal development; variety of malignant syndromes have chromosome characteristic changes, etc. Chromosomes are often prepared to exhibit the enhanced light absorption areas, G bands, for which it was found long ago to appear in certain patterns which are characteristic for individual chromosomes, which are therefore used to create libraries of referent, etalon chromosomes comprising all cataloged diversity and for the analysis of detailed structure of this level of genetic expression, including the inventory of genes and their addressing and dynamics in the chromosomal spaces. This area of investigation is of crucial importance in evolutionary research, in cytology, cytogenetics and molecular cytogenetics and related areas of biological research and medical diagnostics and advances. With a variety of problems and demands, it attracted respective attention in recent decades, which resulted in a large accumulation of method and technology developments. The problems we meet in the analysis of chromosomes are of rather general nature in the image analysis, which is the reason to show briefly some achievements, which are presented here as good illustrations for our purposes.

In the left side of Fig. 5 and 6 we have two chromosomal distributions, typical for cell divisions. Chromosomes are G-banded, showing the characteristic absorption distributions. However, both examples contain irregular chromosomes, which we use to demonstrate rather simple concept of geometric/combinatorial equations application to the solution of genetic origin of irregular chromosomes. In Fig. 5 left, at 1 o'clock we marked a couple of irregular chromosomes. The little on top is shortened; the lower, longer chromosome is prolonged. Rather natural hypothesis would be that the genetic material from the shortened chromosome is somehow transferred to the chromosome which is prolonged by about the volume which is missing in the former. Both chromosomes should appear in duplicate pairs. The match for the small chromosome is the green rounded somewhat larger chromosome, positioned near central line, close to the top. The match for the longer chromosome is the green rounded chromosome at 7 o'clock, which is somewhat shorter than it. After extraction, separation and normalization of the 7 o'clock chromosome, we use the 3D absorption distribution representation manifolds to represent all those objects, shown in Fig. 5, center. Left justifying the large chromosomes immediately reveals, their very good matching in distribution of absorption extremes, while the lower obtained one little pack - one extra light band; in the right column, on top is the normal match of the small chromosome which is shortened by volume equivalent to one light band, noticeable as the difference of the top and lower chromosomes in the right column. The difference corresponds well to the part which is added in

the formation of the lower left chromosome. This anomaly is called Philadelphia syndrome; it indicates very serious health problem. In the example in Fig. 6, at 9 o'clock, we have green rounded marker chromosome, a big pack of genetic material that does not exist in regular cell division. In the same cell division, one chromosome is missing, the Y chromosome, responsible for sex determination. We borrowed the missing chromosome from another image, and after photometric analysis of potential chromosome candidates as genetic material origins, we managed to compose the congruence of manifolds: on the right image, on the top is marker chromosome photo morphology manifold, below are the photo morphology manifolds corresponding to (missing/borrowed) Y chromosome, and longer arm of chromosome one; the concatenation of the last two is congruent to the top, preserving all invariants in the photometric manifold. In both examples, before the photometric manifold matching, some normalization was necessary: namely, some of the involved chromosomes are bent in these microscopic preparations and they needed slight modifications - they were transformed in such a way that the obtained images looked the same as if originally they were straight. Otherwise, it would be difficult to justify the achieved congruence. Note also, that the object contours in these images are fuzzy, which is well seen in the representing photometric manifolds. Setting threshold to the level of background, the contours can be detected. We used common geometric similarity to discern the transfer of genetic material and to reconstruct the origin of a regularly not existing chromosome.

In Fig. 7 we exhibit the detailed treatment of chromosome similarity, which is implemented as semiautomatic or fully automatic procedure. In the mitosis on the left side in Fig. 7, the two large central chromosomes are selected (encircled). They are extracted and normalized (for method details see e.g.[7, 19]), shown with their representative photometric longitudinal sections one on top of the other, left image, right to the mitosis, which exhibit perfect matching, with high degree of similarity, they are almost identical. This is confirmed by similarity measurement using appropriate metrics in the space of chromosomal representations, showing very small relative displacements of distributions of coordinates of local maxima. In the center image, we present the normalized representing photometric manifold of the top chromosome from the left image, with two different longitudinal sections in order to illustrate the variation of local extremes distribution in chromosomal longitudinal photometric sections. The two sections local extreme distribution patterns are quite similar, but not identical. Every metrics measuring similarity of chromosomes should be first tuned to compensate these individual variations. Of course, when these variations are the aim of analysis, then the slightest difference has to be maintained. In the right image of Fig. 7, on the lower left we have a mitosis with a couple of corresponding chromosomes, the two shown magnified in the right lower corner, extracted on top. Their photometric representations exhibit: globally, quite good matching of positions of local extremes; locally, otherwise not easily identifiable difference: one of the chromosomes has one local maximum in excess (the second from the right side in the right chromosome), as verified with the similarity measurement with the same metrics as the one used in previous example. The noticed difference corresponds to a severe genetic syndrome - the threesomy. Note that

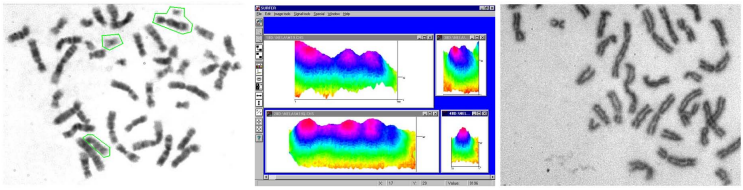


Fig. 5 left, center: geometric manifold equation demonstrating transfer of genetic material from the small chromosome at 1 o'clock to the tail of its nearest chromosome (encircled). Their twins are marked as well. Right: chromosomes in a malignant syndrome; the whole topological structure of chromosomes is getting looser and disaggregated, with some visible disconnections, the contours and inner details becoming very fuzzy. Such properties are well describable and can be used in automatized analysis.

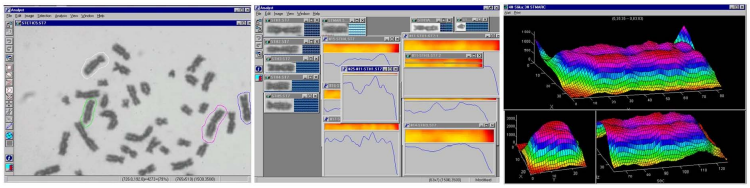


Fig. 6 Geometric similarity used to determine the origin of normally non existing chromosome, the marker chromosome, marked with green at 9 o'clock, left; center: the search for photometric similarity with photometric sections of suspects exposes similarity of photometric sections of bottom curve, the photometry of marker, and the first above it, the photometry of the long part of chromosome one; sections justified from the deep local minimum towards the right, exhibit very good matching of distributions of local maxima; on the right side we have nice congruence of the marker and concatenation of Y (missing in this mitosis) and long arm of chromosome one. This equation work confirmed a rare hematology syndrome (the eighth detected case).

chromosomes have fuzzy contours. Before matching of chromosomes, the contours need be defuzzified in order to determine the beginning of chromosomal photometric representation, which is performed as in the previous example.

We will briefly discuss similarity and approximation on the example of chromosomes. Before approaching this issue, the following has to be done: noise reduction, object contour recognition (all objects have fuzzy contours), object extraction and normalization ("rectification", explained in detail in [19]). After these steps are accomplished, the chromosome representations should be determined. This step depends on the application and needs to be performed with higher or reduced precision, e.g. the analysis of chromosome subtle changes and fine details and the addressing of the genes in chromosome spaces need the representation and metrics which would be refinement of those needed for the comparison of the whole chromosomes and their basic classification, where the tuning of the (representations and)

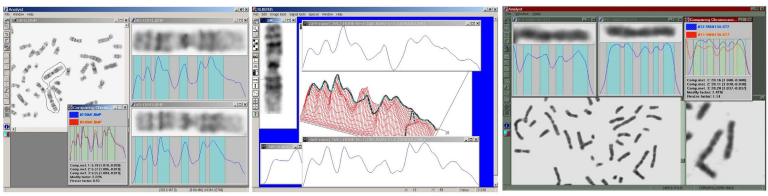


Fig. 7 left: Extraction of the two central chromosomes, their "rectification" - normalization with their photometric representations, exhibiting very high similarity of the objects, which is confirmed with measurements using appropriate fuzzy metrics on chromosomal representations; center: the extracted normalized top chromosome, shown with its photometric manifold and its two different longitudinal sections, illustrating variations in the one dimensional photometric chromosomal representations. Note rather large oscillation in the proportions of local extremes, while proportions of their locations - addresses are very well maintained, which is a good candidate for the object representing invariant. Right: confirmation of threesomy syndrome. The couple of corresponding chromosomes extracted (magnified), resolve one extra band - that one has one local maximum more than the other, as confirmed using appropriate fuzzy metrics.

metrics should not be sensitive to detect the individual fluctuations as those presented in Fig. 7 right. The similarity/approximation determining metrics would be applicable to the spaces of normalized chromosome representations. We begin with the fact that chromosomes have those longitudinal photometric sections which are characteristic. More precisely, as shown before, the whole photometric manifold preserves all details and is a complete chromosomal description. However, individual longitudinal sections can be used instead, which considerably simplifies all involved algorithms. In that way we can take a photometric section, which is always a polynomial to represent a chromosome. The chromosomes appear with more or less fuzzy contour, which means the uncertainty in the localization of its beginning and the end. The measurements are relative, which means that the values in the representation are relative. As a consequence, for a chromosome Ch, the absorption polynomial $P(x)$ should be equivalent to its axial extension/translations $P(ax)$ and $P(x)+b$. Moreover, the relativity of measurement for the comparison of chromosomes does not maintain the proportions of local extremes as algebraic invariants of the representations, hence, for that purpose, the relative position and the type of local extremes is what are the chromosomal invariants, which are exhibited well in $P'(x)$ and $P''(x)$. Finally, it is a kind of characteristic function of a polynomial which can be taken as a proper representation in such case,

$$C_{P(x)} = \begin{cases} 1 \ , \ x \text{ is between two saddle points with a local max of } P(x) \text{ between} \\ -1 \ , \ x \text{ is between two saddle points with a local min of } P(x) \text{ between} \\ 0 \ , \ \text{outside of the range} \end{cases} \tag{7}$$

or combinatorially simpler by

$$C_{1P(x)} = \langle u_i \ : \ i \in I \rangle, \text{ where } u_i = \begin{cases} 1 \ , \ P \text{ has a local max at } x \\ -1 \ , \ P \text{ has a local min at } x \\ 0 \ , \ \text{otherwise} \end{cases} . \tag{8}$$

Here I is the measurement resolution (thus a vector). With C_{2Ch} we can denote the two argument analogue of Ch describing all local extremes of a manifold representing chromosome Ch. The beginning and the end are determined separately with solution for the fuzzy contour. Obviously, for more subtle/detailed analysis, the whole polynomial or manifold representations, which preserve all measurement combinatorial/algebraic invariants, can be used. In this way, depending on the refinement needed in definition of chromosomal representations, if CH is a such representation space we can define basic similarity (fuzzy metrics) as a distance, with the following metrics (or some modifications)

$$d_1(C_P, C_Q) = \sum_i |a_i - b_i|, \tag{9}$$

where a_i, b_i are corresponding polynomial coefficients in the polynomials P and Q, or by

$$d_2(u, v) = \sum_i |u_i - v_i| \tag{10}$$

corresponding to C_{1P} and C_{2Ch} representations, or by

$$d_3(C_P, C_Q) = \min_a \int |C_P(x) - C_Q(ax)| dx, \tag{11}$$

where a is the extension modification coefficient. Applying standard fuzzification (6) to metrics, e.g. d_3, we obtain its fuzzified form

$$M_3(C_P, C_Q, t) = \frac{t}{t + d_3(C_P, C_Q)}. \tag{12}$$

Notice that varying of the parameter t can produce significant impact on the notion of closeness (henceforth to the notion of similarity as well), particularly in the presence of predefined detection/recognition thresholds and the bounded variant of the underlying metric. Recall that each metric d is equivalent (generates the same topology) with the $[0,1]$-valued metric $\bar{d}$ defined by

$$\bar{d}(u, v) = \frac{d(u, v)}{1 + d(u, v)}. \tag{13}$$

When necessary to preserve the original chromosomal structure, the above or similar metrics could be applied on either, non normalized chromosome or at a sequence of small step "rectifications". In Fig. 7 left and right, we show applications of fuzzification M_3 of the above metrics d_3 (using (6)), which provide normalized measurement as well. Such metrics perform well for similarity degree/approximation estimates in various applications involving chromosomes.

In short, as important for similarity assessment here, we have absorption polynomial P coefficients and their corresponding distance, P' and P'' invariants, polynomial shape similarity (by linear or non linear extensions on both axes), topology refinements, determined by involved metrics, extracting algebraic topological and geometrical object characteristic properties - invariants.

Note the similarity of metrics used on chromosome to those used for similarity estimation of spectrogram features. Even more, for curved spectrogram feature recognition/similarity comparison, when no ”rectification”, or a vector of step by step ”rectifications” is needed in feature representations, we use the fuzzified metrics M_2 and M_3 as a very good recognition instrument. This can provide semantic mapping from sensory inputs to the cataloged objects, which, with certain differentia specifica, are entities in a FRDB or some other Fuzzy (discrete information) Structure, thus containing accumulated inventory of characteristic properties. Another way to determine invariants of chromosomal absorptions is to use specific spectroscopic profiles as shown in Fig. 8 left.

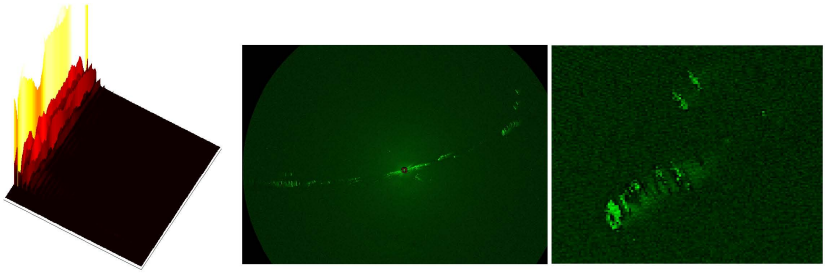


Fig. 8 Left: a chromosome spectrogram, offer invariants which are very similar to the photometric manifolds sections obtained after chromosomal image is transformed into a signal, concatenating chromosomal neighboring longitudinal sections into a locally periodic signal; Center: One example of marine radar surveillance with a complete radar scan; its right NE part is magnified in the right image, where we see the ships in the area, with clutter mainly shown as echo intensity. The objects have fuzzy contours and some have certain shape characteristics. The recognition process includes noise cancellation, contour extraction, kinematics' parameters definition and potential shape classification (for closer/larger vessels).

Problems very similar to those discussed above we meet in the analysis of, e.g., marine radar images, as in the example in Fig. 8, where the noise has to be reduced, small objects and contours of larger objects localized and extracted, kinematics determined (from successive scans), finally, shape identified and ship type classified. Here we deal with fuzzy dots and fuzzy contours. The detailed modeling development involving fuzzy points, fuzzy shape analysis, estimation of fuzzy object distances and application meeting demands like this radar image analysis is presented in [21, 22]. We are applying this method to integrate and best-match data from AIS (automated identification system), radar, other sensors and digital maps.

In noise reduction and reconstruction of biometric data (finger prints), before analysis and matching we applied other techniques of image color defragmentation, followed by alternative reconstruction, which proved well in extraction of fingerprints from materials with certain image deformation or object masking.

2.2 Discrete Information Structurality

In Gödel's final fear the little green evil dwarfs were permanently manipulating and changing formulae and proofs in mathematical books, questioning ontological correctness of mathematical knowledge expressed in theorems, which finally depends on the stability of notation and media used. This threat is becoming more relevant now days. In discrete information structures, DIS, the transition from identity to similarity and approximation seems to be less smooth, demanding more increased scrutiny. As in the case of similarity and approximation in continuous case - metric spaces, we need something to regulate the nearness/distance in order to organize DIS spaces. Mathematics offers two approaches: suitable topologies and some imitations of distance -similarity/approximation hierarchy. We investigate this relationship in related contexts, beginning with examples from molecular biology, programming and mathematical logic. Numerous partial approaches are scattered in the literature (one was presented in [23]).

With just one magnitude higher than the highest magnification used to observe chromosomes, we enter the discrete world, with the discrete organized genetics, where similarity study can still be performed in a fashion like in the section 2.1, in spite of the fact that related domain is discrete. For example, human Y-specific haplotypes, i.e., Y chromosome short tandem repeat loci such as DYS19, DYS385, DYS389I etc. are frequently used in various forensic studies, e.g., such as tribal genetic relationships (see for instance [24]). After standard procedures of extraction, PCR amplification and typing (measurement of each of the studied loci), a row genetic material is transformed into finite subset of $\mathbb{N}^k$. The standard Euclidean metric (restricted to the discrete domain) is used for the statistical expression of similarity and diversity in the analysis of variance framework. Serious drawback of this method is the fact that the smallest Euclidean distance between the two different vectors from $\mathbb{N}^k$ is greater or equal to 1, so the analysis of variance will induce a significant diversity even in the population of very similar individuals. Never the less, natural similarity criterion expressed in terms of the standard fuzzification

$$M_E(u,v,t) = \frac{t}{t + \|u - v\|} \tag{14}$$

of the Euclidean distance, such is for instance

$$u \sim v \Leftrightarrow M_E(u,v,1) \geqslant \varepsilon \tag{15}$$

(ε is the similarity threshold), should be incorporated in the diversity estimation together with the usual statistical measurements. Among the abundant discrete methods in similarity investigations, the **Blast** algorithm has been in extensive use in

molecular biology for a long time, applied generally for estimation of matching quality of two molecular sequences, expressed in percentage of identical coordinates.

Blast(u,v)

$ML = \min(\text{length}(u), \text{length}(v))$
While $i < ML$ begin
If $u(i) = v(i)$ then
$MatchScore = MatchScore + 1$;
end while
return $MatchScore/ML$.

This example shows actually how the DIS similarity is sensitive: measuring the degree of identity is quite apart from similarity and has to be performed with care. Otherwise it can produce misleading and wrong conclusions. If the two sequences are identical, **Blast** will precisely confirm that. However, if the two sequences are e.g. displaced, their matching can be the result of statistical distribution - the chance cycling of the 4 symbol alphabet, without having anything structural in common. That means that **Blast** = 1, or very close to 1, e.g., 0.98 would be significantly good indicator of sequence matching, while 0.62 could mean: the initial 62% segments are identical, or initial 38% is the identical match, while the rest is just the chance match, thus the real match is 38%, or could be even worse.

Applied to software sequences, which are basically of the same semantics, it lacks expected stability. If for example, we insert a single NOP (no-operate instruction) in a program and compare a copy with such mutation with the original, then, depending on the program address space length, we will obtain top score if the mutation is on the end; however, if we move it forward towards the beginning, the Blast match would decrease and offer very unstable matching score, for the instances - copies which should be equally treated. The problem becomes worse in structural hierarchy, involving a tree of functions, instead of linear structure. The identical example we have in Molecular biology. Here, measuring similarity of the syntax with simplest mutations depend strongly on the semantics. Little improved, practically the same algorithm **SbstrBlast(u,v)** would check if one is the substring of the other (not aligning the beginning of sequences); if so, it returns the index in the longer where the shorter begins, and lengths of both sequences. However, this algorithm would suffer of the similar anomalies. This example shows that the notions of identity and similarity, in case of DIS behave differently than in the case of continuous similarity and identity as its special case. We can propose some improvements of **Blast** algorithm:

BF_Blast (Back and Forth Blast)

Find the length of sequences;
From the beginning and the end, find the largest substrings of the shorter sequence which are identical to substrings of the longer;
Return the indices of the beginning of matching sequences and their lengths.

Partitioning of a molecular biology sequence or a computer program into disjoint or embedded functional units is used in a variety of applications. The ways to localize the functionally complete, disjoint segments of code are nontrivial, but in certain cases solvable. It corresponds to the inverse of compilation/linking, and supposedly should generate the structural division of a code into syntactic functional units - functions, localizing each and finding the communication structures, when it was not originally known, or when the functional units have start and end syntax. In Molecular Biology, this is the task of (functional) genomics. Then corresponding matching is

PS_Blast (Partitioned Sequential Blast)

Partition sequences into disjoint contingent subsequences;
Find the matching components, with best Blast matching;
Return the pointers of high score matching.

The above adaptations would indicate structurally identical segments, pointing to the parts of sequences containing mutations. That means that we expanded the tests of identity to substrings, determining the identical subsequences and localizing the parts of the sequences which are substantially different. The best what we can do is not to mix the two in any similarity assessment intention. Rather try to localize separately identical substructures and mutation substructures. That would demand somewhat changed attitude to the problem. Blast variants are numerous.

There are diverse needs for similarity assessments within a single sequence. For example simple palindromes appear frequently in genetics, programming and other diverse contexts. To test if a sequence is a palindrome, e.g. of the form **ABA** we can read it from both sides towards the sequence center, testing the identity of initial and ending sequences, maximizing match before sequence center. In a similar fashion, in mathematical logic, for the sequences which are preverified as correct, we can test if a symbolic sequence has a form $A \to (B \to A)$ and similarly for Polish notation. Denote the test with **Ax1**. If we need to test some other syntax form, like the

$$(\neg B \to \neg A) \to (A \to B) \text{ or } (A \to (B \to C)) \to ((A \to B) \to (A \to C)),$$

we can proceed similarly,

Ax2

divide sequence (counting delimiters); find central connective;
apply same procedure on the remaining parts;
test if left arm sequence begins with "(¬" and if ")" at end, erasing them;
compare (Blast) the left_left with the right_right, and left_right with the right_left subsequences;
return y/n.

In a similar fashion we can build the test for the third sequence form, the **Ax3**. Quite simple is the test for Modus Ponens, applied on two inputs, the $MP(F_1, F_2)$. In this way we have a simple algorithm which scans a sequence of sequences - formulas (earlier tested as correct syntax), and marks all the formulas which are of the form

satisfying tests above, with respectively flags 1,2,3,4. If all formulas are flagged, then the sequence is a (theorem) proof in Sentential Logic.

PrfSL

input $F_1, \ldots, F_n$;
  while $i \leqslant n$
    test(Ax1, Ax2, Ax3, MP)
  end while.

This method easily generalizes to the proofs with hypothesis, thus testing proofs in Theories in Sentential Logic. The test in Predicate Logic with obvious syntactic expansion,

PrfPL

input $F_1, \ldots, F_n$;
  while $i \leqslant n$
    test(Axioms, Rules)
  end while.

In order to test molecular biology sequence syntax forms, we can proceed in a similar manner. The syntactic forms in tests are fixed in the separately built functions. The procedure generalizes to the proof tester from fixed set of hypothesis in predicate logic. These methods verify structural similarity of the sequences with the set of syntactic schemes - forms. Another generalization of syntax form structural similarity compares a set of formulae, in input file, with the scheme-formulae in a separate file, as parameters to the algorithm. The scheme formulae are formed in the same manner as regular formulae (reserving certain variables as syntactic variables for formulae, if comfortable).

Param_syntax_similarity

```
while not eof_input
  get an in_formula from input file;
  construct a tree representing the in_formula structure;
  while not eof_parameters
    get a scheme_formula;
      construct a tree representing the scheme_formula structure;
      localize terminal nodes - occurrences for each of the svf,
      making lists pointers for subsequent substructural identity tests;
      compare the tree structure of in_formula and sch_formula
      fail if not equal;
      check the mutual substructural identities in the in_formula (e.g.
      blast)
      corresponding to a single svf occurrences in sch_formula, for all
      svf,
      fail if not all identities are met;
  end while
end while
```

If small, the parameter file can be read once and kept in memory. This algorithm would work with sentential logic, predicate logic, first order theories or higher order structures, by e.g. supplying hypothesis/non logical axioms as parametric sets. This would provide a general tester for proofs in first order theories. Similar proof testers are developed for other deductive systems, including higher order logics. This approach is applicable to program syntax analysis and analysis of molecular biology sequences. Since they are all built in the fashion of Blast algorithm, they possess not just information on the syntactic form similarity, but can be used to localize structurally all form differences and loci where mutations occur, which can be subjected to independent analysis of mutations, sequence classification and equivalence/similarity estimates. As such this type of investigation is well applicable in other contexts like e.g. molecular biology and programming architectural investigations. Interesting cases are when certain syntax modifications appear within the investigated sequence, for example ABA', where A' is equivalent to A by some algebraic or formal laws; these are the equivalence sort of similarity. Then we can apply similar methods using the equivalence class of A. This happens when we have e.g. (equivalent) variants of the same gene, virus, equivalent formulas or equivalent functions within a program. Then the task can be performed by simple retrieval of the equivalence class of A. In case of Sentential Logic, SL formula testing for e.g. Ax1, with A' a modification of A, for the sequence A' we can put any formula which is formally equivalent to A (provable from the SL axioms) or made such by whatever criteria; we proceed similarly for a modification of a subformula in other axiom forms or with hypothesis subformulae modification, in the proofs from the set of hypothesis, or other variants discussed above. This is all decidable. If we proceed similarly to formal equivalent parts in the First order Logic, FL, and first order theories, it is well possible, however not generally, since we lack general decidability in this case.

There is a well known procedure of Herbrand and later, Alan Robinson for the solution of certain syntax equations. If t_1 and t_2 are first order terms, then the unification algorithm [25] will transform them to identical terms, if possible, using allowed tools. The substitution

$$\tau = \begin{pmatrix} x_1 & \dots & x_n \\ t_1 & \dots & t_n \end{pmatrix}, \tag{16}$$

where $t_1, \dots, t_n$ are terms, is legal if x_i is not present in t_i, for $1 \leqslant i \leqslant n$.

Clearly, composition of substitutions is a substitution. Substitution τ is an instance of τ' if for some substitution σ it can be obtained as it's projection $\tau = \tau'\sigma$; we also say that τ' is more general than τ. For terms t and t' we say that they are unifiable, if there is a substitution τ which makes their derivations identical, i.e. if $t\tau = t'\tau$. It is a well known fact that if two terms are unifiable then they have the most general unifier, a unifying substitution $mgu(t,t')$, such that every other unifier is its instance. The unification test is

Unif(t, t')

For $i = 1$ to $\max(\text{length}(t), \text{length}(t'))$
if $t(i) \neq t'(i)$
if $t(i)$ or $t'(i)$ is a variable, e.g. $t(i)$
if $t(i)$ does not appear in the subterm t" of t' beginning with $t'(i)$
then substitute t" for a variable $t(i)$ in both t, t';
else return 0;
else return 0;
else;
return 1;

The composition of a sequence of individual substitutions in this procedure results with the $mgu(t,t')$. Note that using this algorithm we can test unifiability of SL formulae, which would then submit the solution to the question if the two formulae are of the same syntax form. This is applicable to axiom schemes and to parametric forms which are tested in the PrfSL procedure and its generalization. However, this algorithm is not applicable to the proof tests in PL and first order theories, its application constrained to terms only. Take for example the PL formulae $A \to (B \to A)$ and $A \to (B' \to A)$, both instances of the same sentential axiom, thus very similar. The algorithm would need to unify the whole formulae, which are completely different. Also note that if two terms are equal semantically or provably equal in some formal system, than we can use such algorithms to expand the unification as we did in the above syntax matching procedures. As noted above, semantical equality of terms would be realized by retrieval (of equivalence classes), while for syntactical equality of terms a decision procedure is needed, available in decidable theories and not available in theories generally. For later purposes we will divide unification of (more general) expressions in two steps. The first, which is usually not discerned, would determine the preunification common structure, their meta-form (whose projections they are), the second step proceeding with syntax equations leading to unification of expressions, when it is solvable, or to the denying set of conditions - the residuum, in case it is not solvable. Note that unification can be extended to point all subterms where unification fails, not just the first. Define the whole set of non unifying subterms as Residuum. Unification has a large number of realizations. It is applicable to formal programming context and (some formalization) of molecular biology codes.

Proceeding towards a definition of similarity in DIS spaces we would have to introduce distance/nearness measure of some sort; we can use trees as expression structural representations. For the sake of simpler visualization, we will discuss structural tree representations and the algorithm. Restricting the formalized syntax context to e.g. usual infix notation, we correspond a tree to each expression, with symbols in nodes, with variables or constants as terminal nodes, non variable symbols as non terminal nodes, and downward links between symbols represent the hierarchical order in the expression. Clearly, when we have expressions of certain sort, the correspondence to the representing trees is a bijection, so we can identify an expression e with its representing tree T_e. A tree unification algorithm variant then

can be stated in the following way for terms, but general form for more abstract expressions would be very similar.

Tree_unif(t,t')

get term trees;
if they disagree on a non terminal node, or on a constant terminal, terms are not unifiable;
identify intersection tree; (it can be empty) with dummy variables at terminals;
form equations in terminal variables corresponding to intersection tree terminals;
if any of these equations represents illegal substitution, terms are not unifiable;
else, take those equations in reverse order: they define mgu of the terms.

We define a meta-form $mf(t,t')$, as the intersection tree, with dummy variables or meta variables, obtained in the extension of the above procedure; it presents the involved expressions top common structural form, as discussed earlier; it is not necessary to use tree representation in the formulation of meta-form; as a meta-form we can take simply a meta term, the term with syntactic variables at the appropriate nodes, whose projections are the input terms. Then the initial terms are always instances of their meta-form. If they unify or not is resolved by syntax equations corresponding to the terminal nodes. Hence, unification is a partial operation, resulting in common unifier if the two objects unify, while meta-form extraction is a total operation which could be extended to unification when the unification is possible.

Example 1. Let $t = F(H(x,L(z),y,L(z))$ and $t' = F(H(G(u,v),v),G(u,z),v)$. The term representing trees with the intersection (meta-form) tree indicating points of basic equations

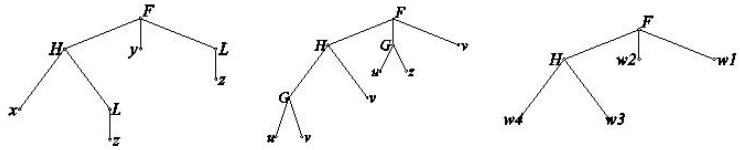


Fig. 9 The left tree and the center tree correspond respectively to terms t and t'. The tree on the right represents their meta-form.

which are listed in solved form, in reverse order

$$w1 : v = L(z)$$
$$w2 : y = G(u,z)$$
$$w3 : v = L(z)$$
$$w4 : x = G(u,v) = G(u,L(z)),$$

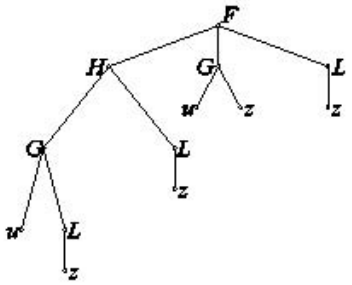


Fig. 10 The unification resulting term tree

with

$$mgu(t,t') = \begin{pmatrix} x & y & z \\ G(u,L(z)) & G(u,z) & L(z) \end{pmatrix}$$

and unification resulting term tree Take $t3 = F(H(x,L(v),y,L(c1)), t4 = F(H(x,L(v),y,L(c2))$. Then their meta-form is equal to term t, whose instances they are, but they are not unifiable.

Unification can be generalized to higher order structures, here discussed in type theory [26]. For the treatment of higher order structures, e.g., Functionals, Relationals refer to [27, 28]. For an account on higher order unification refer to e.g. [29].

For a higher order term, we can suppose to have provided a coloring of its representing tree, which would distinguish the type layers. The syntax within a layer belongs to the same type. The higher order unification works as follows

HOT_Unif(T,T')
top - down
bounding variables within a higher order syntax (like local; global variables in usual programming);
for the corresponding trees,
perform tree unification at a current layer,
marking nonunifiable items;
proceed with next (lower) level.

This procedure can be used to extract the meta-form in HO structures similarly to the way it was done in the case of basic unification algorithm. Clearly, HO Unification can be expanded with a theory (its decision method), set of algebraic laws or a set of semantic laws/rules in a similar way and with similar argumentation as it was the case with basic term unification.

Now we can organize DIS spaces properly. The meta-form can be used to determine the distance in the DIS spaces; We introduce a distance function in a DIS structure, organizing it into a discrete, metric like space, having representation tree structure of expressions. For a set DIS of expressions, denote with $T_{DIS} = \{T_e \ : \ e \in DIS\}$ the set of corresponding tree representations. Introduce a partial

ordering in T_{DIS}, essentially by $T \leqslant T'$ iff T is an initial segment of or equal to T'. Then define similarity degree as a mapping

$$sd : T_{DIS}^2 \longrightarrow T_{DIS} \text{ by } sd(T_e, T_{e'}) = mf(T_e, T_{e'}), \tag{17}$$

where mf stays for meta-form of the input structures. The tree representation is used here for visualization purposes only and it can be omitted with no loses, we can work directly with expressions and their meta-forms, defining $\leqslant$ on expressions in the same way. Such distance will preserve a major portion of desirable properties of a metric and organization of metric spaces and it will apply well in all different contexts discussed above. Still the relation between high level similarity - when the meta-form of expressions becomes very close to the expressions themselves, and the identity of expressions remains somewhat context dependable.

3 Conclusions and Discussion

In the currently evolving large knowledge extraction and synthesis systems, the Cost functions are becoming an important factor, where a demanded piece of knowledge is in some close relationship to already collected and addressable knowledge, when the notions of similarity and approximation of knowledge representation are needed extensively. There is cheap easily deducible/extractible knowledge, but there is knowledge which is very or even extremely expensive in resource consuming; we need to estimate its complexity and, if useful, towards solution, try to use some approximations which were resolved earlier. There are numerous examples when we deal with imprecise knowledge, where similarity and approximation are integrated in the solution process. For two kinds of knowledge, the sensory originating and the formally shaped, we have fundamental methodological division, according to the type of representations used and similarity/approximation assessment methods. For the former, there are numerous examples when metric spaces with certain metrics properly describe the knowledge and these concepts, with smooth metric changes over shifts from identity to approximations. For the later, the other kind of spaces, the discrete information structure - DIS spaces, with discrete similarity degree and approximation, based on tree topologies (expanded with certain knowledge when available), are properly expressing representations, offering the necessary measure of nearness and it is not smooth in the way the metrics in metric spaces are, involving certain non uniformity, case dependant and depending on the point of view. However, both methods meet at the need to organize the sensory based knowledge, within so called semantic mappings, with input consisting of continuous structures, and with discrete/symbolic output, realized as e.g. mappings of sensory data into FRDB, over which other semantic or formal methods apply towards more complex intelligent systems (modeled by hierarchical DIS spaces). Developments in this area are fascinating in the resources integrated, achievements realized, application domain span, of which we mentioned distant examples as an illustration how broad it is and how similar the concepts of knowledge similarity and approximation can be

in those distant contexts, accenting the kinds of mathematical invariants involved in reality.

DIS spaces are involved in extraction of formulae of the same templates, correctness evaluation of program systems and proof procedures, distance of species in evolution, measurements of similarity of molecular biology sequences (and structures), monitoring of mutations and reasonable (evolution) distance calculation in evolutionary systems, e.g. natural languages, investigations of software mutations, in virusology and immunology, music pattern classification -reaching closer to the description of aesthetic, perhaps via classification of non aesthetic components and models. There are other important examples.

The algorithms involved in structural similarity are not fully applicable to molecular biology sequence matching or to recognition of structures in programming and music. There are numerous examples of limiting/forbidding transformations in formalized contexts. There are operations on sequences, such as are already in use in some of these contexts: symbol insertion, deletion, mutation - substitution at individual variable occurrence, or other non variable symbol occurrence replacement by another symbol, most often directly confront basic grammar rules of formalized contexts (partly supported in LISP), generally not supported by reasonable similarity maintaining transformations. However, some contexts exploit those examples generally, most importantly, transformations involved in molecular biology/genetics, programming, linguistics, music. These operations are needed to reduce sequences to a reasonable comparability in certain cases when they correspond to actually realized operations of the same kind in reality. To e.g. motive variations in music should be corresponded specific operations. The use of meta-forms can bridge some of those difficulties. The algebraic like laws applicable in specific context could expand the potential of application of similarity assessing methods. However, this is yet at the beginning.

Acknowledgements. This research has been partly supported by Serbian Ministry of Education and Science, Projects III-41013, TR36001 and 174009.

References

1. del Millan, J.R., Mourino, J., Franze, M., et al.: A local neural classifier for the recognition of EEG patterns associated to mental tasks. IEEE Transactions on Neural Networks 13(3), 678–686 (2002)
2. Cincotti, F., Mattia, D., Babiloni, C., et al.: Classification of EEG mental patterns by using two scalp electrodes and Mahalanobis distance-based classifiers. Methods of Information in Medicine 41(4), 337–341 (2002)
3. Birbaumer, N., Kübler, A., Ghanayim, N., et al.: The thought translation device (TTD) for completely paralyzed patients. IEEE Transactions on Rehabilitation Engineering 8(2), 190–193 (2000)
4. Japundžić-Žigon, N., Milutinović, S., Jovanović, A.: Effects of Nonpeptide and Selective V1 and V2 Antagonists on Blood Pressure Short-Term variability in Spontaneously Hypertensive Rats. Journal of Pharmacological Sciences 95, 47–55 (2004)

5. Spasić, S., Perović, A., Klonowski, W., Djordjević, Z., Duch, W., Jovanović, A.: Forensics of features in the spectra of biological signals. International Journal for Bio Electromagnetism 12(2), 62–75 (2010), http://www.ijbem.net
6. Blinowska, K.: Methods for localization of time-frequency specific activity and estimation of information transfer in brain. International Journal of Bioelectromagnetism 10(1), 2–16 (2008) http://www.ijbem.org
7. Jovanović, A., Perović, A., Klonowski, W., Duch, W., Djordjević, Z., Spasić, S.: Detection of structural features in biological signals. Journal of Signal Processing Systems 60(1), 115–129 (2010)
8. Zatorre, R.J., Halpern, A.R.: Mental concerts: musical imagery and auditory cortex. Neuron. 47(1), 9–12 (2005)
9. Grierson, M.: Composing with brainwaves: Minimal trial P300b recognition as an indication of subjective preference for the control of a musical instrument. In: Proceedings of the ICMC, Belfast (2008)
10. Jovanovic, A.: Brain signals in computer interface. Intelektualnie Sistemi 3(1-2), 109–117 (1998) (in Russian)
11. Klonowski, W., Duch, W., Perovic, A., Jovanovic, A.: Some computational aspects of the Brain Computer Interfaces based on Inner Music. Computational Intelligence and Neuroscience 2009, Article ID 950403, 9 pages (2009), doi:10.1155/2009/950403
12. Babiloni, F., Cincotti, F., Marciani, M., et al.: The estimation of cortical activity for brain-computer interface: applications in a domotic context. Computational Intelligence and Neuroscience 2007, Article ID 91651, 7 pages (2007)
13. Babiloni, F., Cichocki, A., Gao, S.: Brain-computer interfaces: towards practical implementations and potential applications. Computational Intelligence and Neuroscience 2007, Article ID 62637, 2 pages (2007)
14. Cincotti, F., Mattia, D., Babiloni, C., et al.: The use of EEG modifications due to motor imagery for brain-computer interfaces. IEEE Transactions on Neural Systems and Rehabilitation Engineering 11(2), 131–133 (2003)
15. Astolfi, L., Cincotti, F., Babiloni, C., et al.: Estimation of the cortical connectivity by high-resolution EEG and structural equation modeling: simulations and application to finger tapping data. IEEE Transactions on Biomedical Engineering 52(5), 757–768 (2005)
16. Liu, L., Arfanakis, K., Ioannides, A.: Visual field influences functional connectivity pattern in a face affect recognition task. International Journal of Bioelectromagnetism 9(4), 245–248 (2007)
17. Hadžić, O., Pap, E.: Fixed Point Theory in Probabilistic Metric Spaces. Kluwer Academic Publishers, Dordrecht (2001)
18. Hajek, P.: Metamathematics of Fuzzy Logic. Kluwer Academic Publishers (1998)
19. Jovanović, A., Marić, M., Borovčanin, M., Perović, A.: Towards intelligent chromosome analysis. In:Proc. Conf. Bioinformatics Workshop, Soc. for Industrial and Applied Math, SIAM International Conference on Data Mining (SDM 2004), Orlando, Florida (2004)
20. Malkov, S., Vujosevic, M., Jovanovic, A.: One Method for chromosome analysis and comparison. In: Proc. of the Conference Mathematics and Others Sciences. Greek Mathematical Society, Crete
21. Obradović, D., Konjović, Z., Pap, E.: Extending PostGIS by imprecise point objects. In: 2010 IEEE 8th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, September 10–11 (2010)
22. Obradović, D., Konjović, Z., Pap, E., Ralević, N.M.: The maximal distance between imprecise point objects. Fuzzy Sets and Systems 170(1), 76–94 (2011)

23. Perović, A., Stefanović, N., Jovanović, A.: Syntax Zooming. In: Proc. of 3rd Serbian Hungarian Joint Symposium on Intelligent Systems, SISY 2005, Subotica (2005)
24. Stevanović, M., Dobrii̇ć, V., Kecarević, D., Perović, A., Savić-Pavićević, D., Keckarević-Marković, M., Jovanović, A., Romac, S.: Human Y-specific STR haplotypes in population of Serbia and Montenegro. Forensic Science International 171(2-3), 216–221 (2007).
25. Baader, F., Snyder, W.: Unification theory. In: Robinson, A., Voronkov, A. (eds.) Handbook of Automated Reasoning. Elsevier (2001)
26. Wolfram, D.A.: The Clausal Theory of Types. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press (1993)
27. Shoenfield, J.R.: Mathematical logic. Addison Wesley (1967)
28. Hinman, P.G.: Recursion-Theoretic Hierarchies. Springer (1978)
29. Dowek, G.: Higher order unification and matching. In: Robinson, A., Voronkov, A. (eds.) Handbook of Automated Reasoning. Elsevier (2001)

Part V
Applications in Transportation, Network Monitoring, and Localization of Pedestrians in Images

Quay Crane Scheduling for River Container Terminals

Endre Pap, Vladimir Bojanić, Goran Bojanić, and Milosav Georgijević

Abstract. Due to the benefits of container transport, containerization of goods has increased significantly in last two decades. Consequently all aspects of container transport chains should be analyzed. Formulation of mathematical models for container terminals is important for understanding and improving container terminal operations. This paper presents an overview of existing optimization problems in container terminal operations, with the goal of defining a model for quay crane operations in river container terminals. While river container terminals are important nodes in the inland transportation network, they have not yet been addressed in the literature. The efficiency of container cranes, as the lead element in the terminal, is of key importance for the performance of the container terminal. This paper presents scheduling of quay cranes in river terminals for barge reloading.

Keywords: River container terminal, scheduling, quay crane, transport.

1 Introduction

Over the last two decades, the volume of containerized goods has increased significantly and a further increase can be expected in the coming years. Thus there is a constant effort to improve all segments of operations during transportation and

Endre Pap
Faculty of Sciences and Mathematics,
University of Novi Sad,
Trg D. Obradovica 4, Novi Sad, Serbia,
Óbuda university, H-1034 Budapest, Becsi út 96/B, Hungary,
Educons University, 21208 Sremska Kamenica, Serbia
e-mail: pap@dmi.uns.ac.rs, pape@eunet.rs

Vladimir Bojanić · Goran Bojanić · Milosav Georgijević
Faculty of Technical Sciences, University of Novi Sad,
Trg D. Obradovica 6, Novi Sad, Serbia
e-mail: {ydad,gbojanic,georgije}@uns.ac.rs

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 285–300.
DOI: 10.1007/978-3-642-33959-2_15

manipulation of containers. There is also substantial rivalry between competing terminals, operators and others in container supply chains. Container terminals are elements of large logistic networks, and therefore a key factor in the supply chain management system, and operation studies and descriptions of container terminals represent complex problems [15]. Thus, there are an increasing number of scientific papers that deal with problems concerning container reloading and transportation. There are many decision problems related to logistics planning and control issues, and each can be assigned to one of three levels: terminal design, operative planning, and real-time control [7]. One of the early reviews on container terminal operations is presented in [18]. Comprehensive overviews are presented also in [17] and [16]. These papers not only give an overview of the existing literature, but also attempt to create general classifications of existing problems.

Intermodal freight transport is the movement of goods in a loading unit or vehicle using various modes of transport (road, rail and waterway) without any handling of the goods themselves during transfer between modes [11]. Intermodal transport has become more attractive in recent years, mostly because of the problems that characterize road transport, such as traffic congestion and environmental issues. Container terminals are the interface that connects different modes of container transportation. Containers that are delivered by one mode of transportation (rail, waterway or truck) are usually temporarily stored and later shifted out using another or the same mode of transportation.

There are different types of container terminals, with the main differences being the size, layout and reloading equipment. These three elements are interdependent. From a logistical point of view, terminals consist of stock and transport vehicles. The yard stacks, ships, trains and trucks are the stock, while the cranes and transport vehicles for horizontal transport are the transport [17]. Which concept of container terminal is to be implemented depends on several factors. The two main types of container terminal are seaports and inland container terminals. The main difference between them is the working strategy and types of reloading mechanization [14].

2 Seaport Container Terminals

For seaport container terminals (Fig. 1), quay cranes transport containers only from ship to shore and vice versa. A comprehensive overview of container quay crane scheduling is presented in [1]. As a rule of thumb, minimizing the time a ship is at the berth is an overall objective with respect to terminal operations [17].

Since quay cranes are the most expensive type of equipment and their performance directly influences ship turnaround time, quay cranes are always the lead element in these logistic systems. All other transport and storage systems in the terminal are subordinate to the operation of quay cranes.

To speed up ship turnaround time in seaport container terminals it is usual for several quay cranes to operate on the same ship at the same time. The decision as to how many cranes are assigned to a ship is called the crane split. Since the cranes

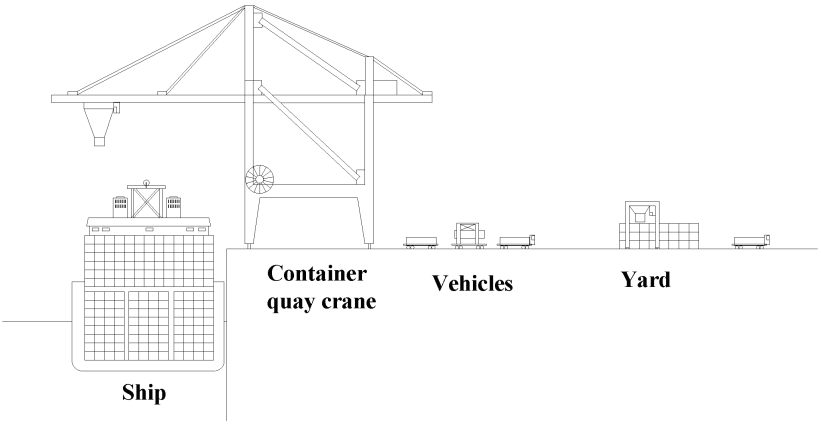


Fig. 1 Seaport container terminal

cannot pass each other because they are positioned on the same track, this raises the problem of container quay crane scheduling.

The goal of the quay crane scheduling problem is to schedule the loading and unloading tasks of a vessel using a set of cranes that minimizes the overall vessel handling time. The basic assumptions of static and dynamic crane scheduling aimed at minimizing delay costs were presented in the first paper [2] in the field of container quay crane scheduling. Since then, many papers have analyzed quay crane scheduling in more or less detail using various exact and approximate methods. Recent papers have analyzed more complex problems, such as non-interference constraints. Lim et al. [10] analyze unidirectional schedules and prove that, when the amount of work in a bay is defined as a single task, the optimal solution is a unidirectional schedule. Kim and Park [8] addressed the problem of container quay crane scheduling by analyzing clusters of containers in the bay of the same type (usually those having the same destination). Meisel and Bierwirth [12] proposed a unified approach to evaluating quay crane scheduling models. Various aggregated models for scheduling quay cranes result from different specifications of the term 'task' (Fig. 2). Tasks can be related to the handling of groups of containers within a bay, all containers within a bay, or all containers within a connected area of bays [12].

One possibility for increasing the reloading capacity of quay container cranes is to implement double cycling. In contrast to single cycling where the unloading tasks in the bay are performed first followed by the loading tasks, double cycling combines unloading of containers from a stack with loading containers to another stack in the same bay. This is a cost-effective operational technique since it does not require any new equipment or investment in infrastructure, such as terminal expansion. A relatively small number of papers have analyzed the possibility of double cycling for container quay cranes. Goodchild and Daganzo [4] proposed a

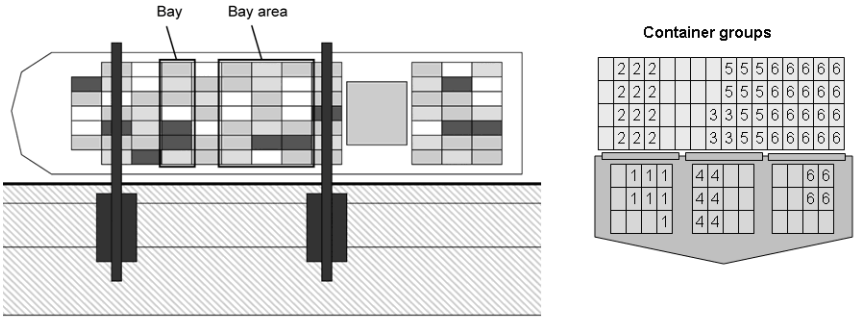


Fig. 2 Different definitions of "tasks"

method for quay crane double cycling with the main aim of reducing the number of operations necessary to complete all tasks for a bay. The problem is solved optimally using Johnson's rule. The amount of work is defined by the number of containers that need to be reloaded, so the main objective is to maximize number of double cycles performed during the reloading of the bay.

The basic formulation can be stated as a two-machine flow shop problem [4].

u_c - number of containers to unload in stack $c \in S$
l_c - number of containers to load in stack $c \in S$
FU_c - completion time of unloading $c \in S$
FL_c - completion time of loading $c \in S$
ω - maximum completion time
X_{kj} - binary variable for ordering unloading jobs (1 if $j \in S$ is unloaded after $k \in S$ and 0 otherwise)
Y_{kj} - binary variable for ordering loading jobs (1 if $j \in S$ is loaded after $k \in S$ and 0 otherwise)
M - a large number

The formulation is: (SP) minimize ω subject to

$$\omega \geq FL_c \quad \forall c \in S,$$
$$FL_c - FU_c \geq l_c \quad \forall c \in S,$$
$$FU_k - FU_j + MX_{kj} \geq u_k \quad \forall j,k \in S,$$
$$FU_j - FU_k + M(1 - X_{kj}) \geq u_j \quad \forall j,k \in S,$$
$$FL_k - FL_j + MY_{kj} \geq l_k \quad \forall j,k \in S,$$
$$FL_j - FL_k + M(1 - Y_{kj}) \geq l_j \quad \forall j,k \in S,$$

$$FU_c \geq u_c \quad \forall c \in S,$$

$$X_{kj}, Y_{kj} = 1, 0 \quad \forall j,k \in S.$$

A reformulation of the quay crane double cycling method using mixed integer programming is presented in [19]. They first analyze sequencing of all stacks in every hatch, and then sequence all the hatches using a heuristic method with the Johnson's rule. A method where rehandles are transferred to another stack in the same bay is presented in [13], while in [4] rehandles are transferred to the shore and later returned to the same stack. Transferring rehandles to another stack is done on the assumption that all rehandles are equal and that is the bay in which they are positioned makes no difference.

Manipulation of containers in the yard is usually performed by rail mounted gantry cranes (RMGs) or rubber tired gantry cranes (RTGs). RMGs are used in modern automated container terminals and RTGs are used in traditional container terminals. RTGs are more flexible, but RMGs are more stable and can travel at greater speeds. When RTGs are used the container blocks are parallel to the quay, while when RMG are used the container blocks are perpendicular to the quay, so the layout of the yard depends on the type of mechanization used to manipulate the containers in the yard. Yard utilization is greater in terminals where RMGs are used. This is because RTGs do not transport containers along blocks in the yard, so there is a free lane alongside every bay in the yard for passage of trucks. In this type of terminal access points are located beside every bay. In automated container terminals there are two access points, one on each side of the block. Internal transport vehicles (usually automated guided vehicles - AGVs) transport containers between the yard and the quay cranes, while external trucks move containers to and from the access point that is further from the quay.

Containers inside the yard are usually classified into three groups. Export containers are containers that arrive by land, are temporarily stored inside the yard, and then sent out by large seagoing vessel (ship). Since the stowage plan and time of arrival of the ship are known in advance, containers in the yard can be arranged in such a way as to minimize the number of rehandles required in the yard prior to the arrival of the ship. Import containers are those brought in by ship and sent out by truck. This case is more complicated since the arrival times of trucks are not usually known in advance. Transshipment containers are brought in and sent out by ship, and usually refers to feeder services. A very important parameter for terminal performance is the number of rehandles required to retrieve a certain number of containers from the yard. The number of rehandles is correlated to the height and width of the container block in the yard. A greater stacking height requires more rehandles but increases yard utilization. Kim [6] proposed a methodology for evaluating the expected number of rehandles needed to pick up a random container in a bay and to pick up all the containers in a bay.

3 Inland Container Terminal

Inland container terminals serve as local hubs for the distribution of containers between container users. While all three types of transportation mode are not necessarily present in every inland container terminal, we assume that all three modes are present in the analyzed model. This type of terminals is also referred to as a river container terminal.

On inland waterways, transportation of containers is performed by barges. Functionally, barges are means of hinterland transportation (like trucks and trains), while operationally they are ships served by container quay cranes [17]. At ports like Rotterdam, Antwerp, Hamburg and Constanta, barges are an important means of transporting containers between terminals in the port and the hinterland. Barges are becoming increasingly attractive due to congestion of the road and rail infrastructure and the growing attention to sustainable mobility [3]. Douma et al. [3] define the barge handling problem as optimizing the sequence in which barges visit the terminals inside the port and determining the order in which the barges will be handled in the terminal. This problem addresses only the optimization of barge performance inside the main port, and not the servicing of barges in the inland terminals. Barge reloading in inland terminals has a large influence on the overall performance of hinterland transportation. The quality of hinterland transport has become increasingly important for the competitiveness of a seaport. Shippers and carriers value not only on the performance of a seaport, but also its accessibility to the hinterland [9].

Containers are stacked inside the barge (usually up to 4 containers in height), and the number of tiers and bays varies depending on the type of barge. Since every section of the river has a maximum allowed depth gauge, the maximum size and capacity of barges on each section of the river is already known, in contrast to sea container terminals, where the size of ships is continually increasing.

The most common reloading machine in typical river container terminals (Fig. 3) is the Rail Mounted Quay Gantry crane (RRMQG) - this abbreviation will be use here for quay cranes in river terminals. The manner in which these cranes operate is functionally different from the quay gantry cranes in sea container terminals. RRMQGs have three different operational movements, in contrast to quay gantry cranes in sea terminals, which have two types of operational movement and one auxiliary movement (gantry movement along the quay).

By design and type of operational movement, RRMQGs are similar to the RMGs used for manipulating containers in container yards of modern automatic container terminals. As mentioned previously, RMGs operate in a container block with access points at both ends. At these points, the crane reloads containers to/from trucks or AGVs (automated guided vehicles). The operation of RRMQGs is more complex and versatile. RRMQGs are used for reloading containers to/from barges, trains, and trucks, and for manipulating containers inside the yard. In comparison with RMGs, RRMQGs usually have greater span and reach due to the requirements of covering a greater area, often the whole river container terminal. This paper deals with RRMQG scheduling in river container terminals for reloading barges.

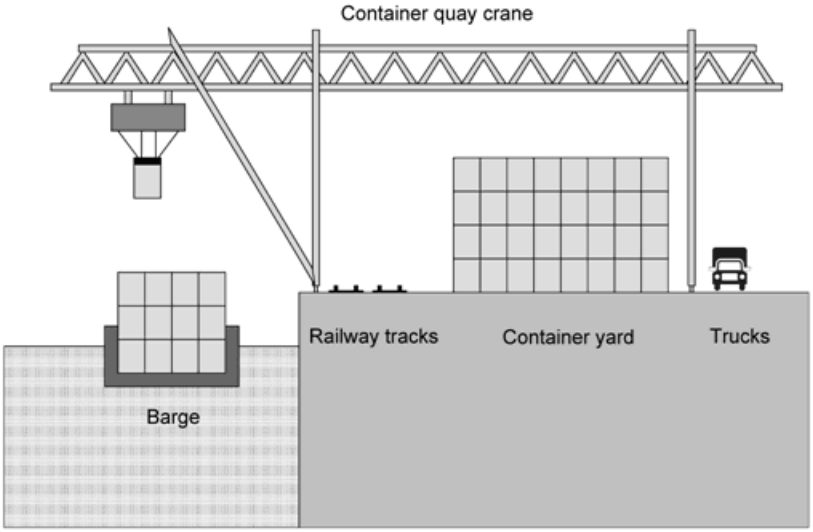


Fig. 3 Typical layout of river container terminal

4 Main Assumptions and Model Definition

The basic movements of the RRMQG are vertical motion of the spreader, trolley motion and gantry motion. Due to safety reasons, all three motions are never performed simultaneously. Thus there are two types of motion. Rectangular motion, where each movement is performed separately, and Chebyshev type motion of the trolley and gantry simultaneously.

Since the destinations of the containers on a barge are usually distributed among several inland terminals, they will be only partially loaded or unloaded at each terminal (Fig. 4). It is assumed, as in [4], that rehandles are replaced in the same stack from which they were removed. The rehandled container is unloaded to the nearest location inside the terminal and loaded back in the same stack, so every rehandle can be represented as one unloading task and one loading task. This is a realistic assumption from the operational aspect, since moving a container to another stack complicates identification and container tracking, and increases the complexity of stowage planning.

For barge reloading, the crane transfers containers directly from the barge to the container yard and vice versa. The most common practice in river container terminals is that containers inside the yard are sorted according to operator, size and destination. The reasons are ease of organization and system tracing and monitoring, so containers from two operators or of two types are not mixed in the same bay. It is assumed that the number of rehandles inside the terminal is the same for every feasible scheduling. Since the arrival of the barge is known at least one day in advance

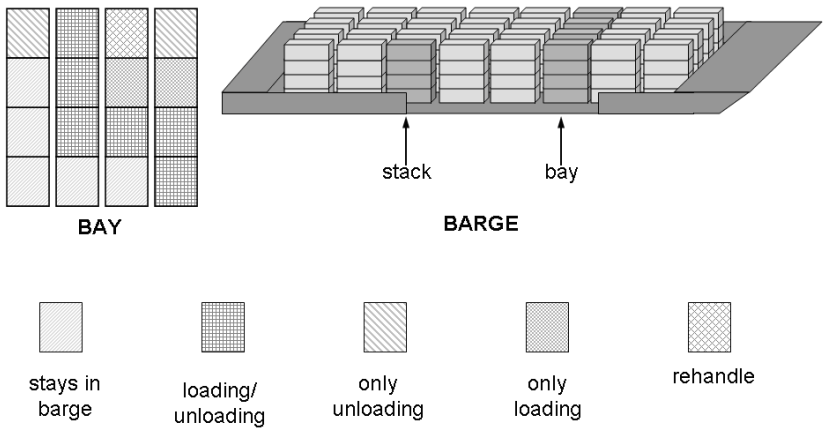


Fig. 4 Different tasks inside one bay of the barge

we assume that the stowage plan of the barge and the location of all containers in the yard is known in advance.

As already mentioned, the RRMQG transports containers directly to the yard and, depending on the spatial planning, it can be assumed that the location of retrieving/disposal for each container can be anywhere in the yard.

The typical procedure for RRMQG scheduling is to use single cycle reloading (Fig. 5). Double cycling consists of unloading one stack and then combining the reloading of that stack with the unloading of another stack (Fig. 6).

Barges have significantly poorer stability than ships. Reloading is performed stack by stack so that stability is not compromised, since the problem of stability should be analyzed separately for each type and size of barge. There is also the problem that the weights of the containers inside the barge are not always known.

Since the vertical motion, frequency of locking/unlocking the spreader and positioning times are the same regardless of whether single or double cycling is implemented, only horizontal motion has to be analyzed to optimize barge reloading, so the goal is to minimize the horizontal motion. It can be seen that for performing one unloading and one loading task, the difference between single cycling and double cycling is the time difference between empty horizontal crane moves in the single cycle and the time of horizontal movements between locations for unloading and loading in the yard. If this difference is defined as time saving, then the optimization problem can be reformulated in terms of maximizing time savings.

When the crane is performing Chebyshev type motion, gantry and trolley movement are performed simultaneously. The time necessary to transport a container from point A to point B is equal to the longer transportation time (gantry or trolley transportation time). The time saving (TS) can be calculated as:

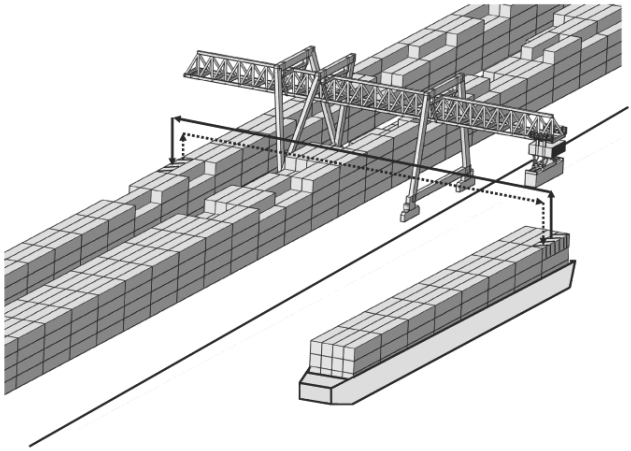


Fig. 5 Single cycle for RRMQG

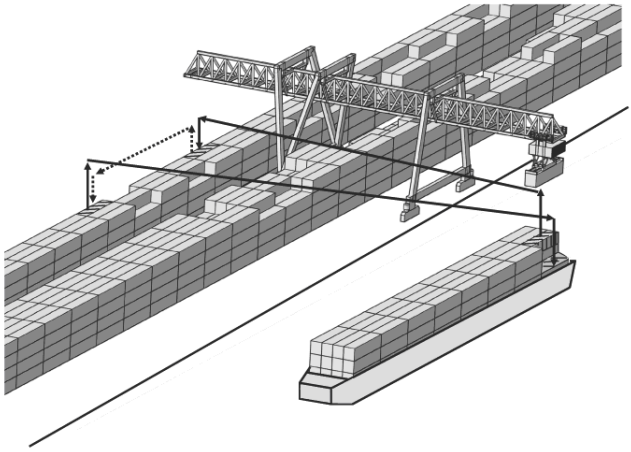


Fig. 6 Double cycle for RRMQG

$$TS = \max(|x_1 - x_0|, |y_1 - y_0|) + \max(|x_2 - x_0|, |y_2 - y_0|) - \max(|x_2 - x_1|, |y_2 - y_1|),$$

where x_i and y_i are time coordinates. x_i represents the time for gantry movement along the quay from the origin of the coordinate system to location i in the terminal, and y_i represents the time for trolley movement along the axis perpendicular to the quay from the origin of the coordinate system to location i in the terminal (Fig. 7).

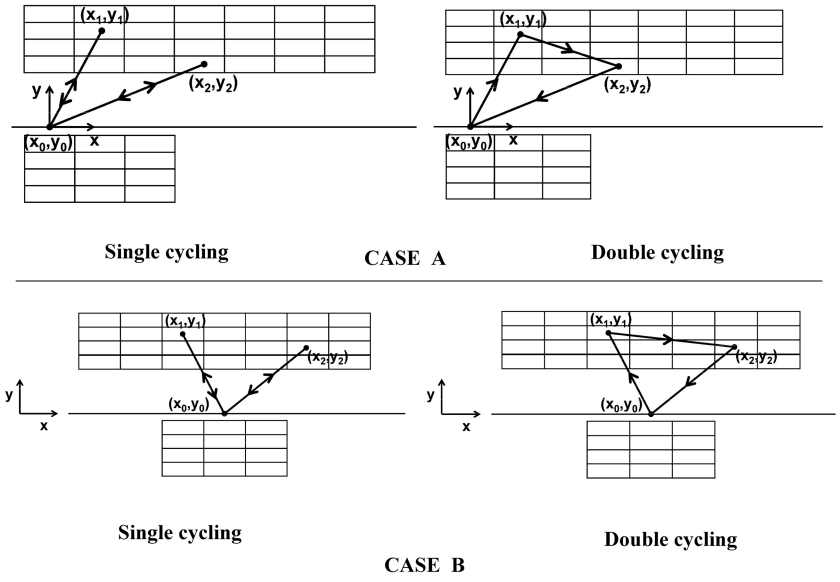


Fig. 7 Horizontal movement for single and double cycle

Theorem 1. *For Chebyshev type movement, the difference in operation time as a result of the application of the double cycle is always positive; i.e., $TS \geq 0$ for all x_i and y_i $(i = 0,1,2)$.*

Proof. We shall investigate the two cases presented in Fig. 7.

Case A

For the case of $x_0 \leqslant x_1 \wedge x_0 \leqslant x_2$ for $\forall x_i > 0$ (Fig. 7-case A), we need to prove the following inequality

$$\max(|x_1 - x_0|, |y_1 - y_0|) + \max(|x_2 - x_0|, |y_2 - y_0|) \geq \max(|x_2 - x_1|, |y_2 - y_1|). \quad (1)$$

Without loss of generality, we move the origin of the coordinate system to the point (x_0, y_0), implying $x_0 = 0, y_0 = 0$. Then, the previous inequality reduces to the inequality

$$\max(x_1, y_1) + \max(x_2, y_2) \geq \max(|x_2 - x_1|, |y_2 - y_1|).$$

The last inequality is a consequence of the following two inequalities:
$\max(|x_1 - x_2|) \leq \max(x_1, x_2)$ and $\max(|y_1 - y_2|) \leq \max(y_1, y_2)$.

Hence, Theorem 1 is proved for case A.

Case B

For the case where $x_1 < x_0 < x_2$ (Fig. 7-case B), we need to prove the inequality (1). Without loss of generality, we can suppose that $y_0 = 0$. Then, we have $y_1, y_2 > 0$ and

$$|x_1 - x_0| + |x_2 - x_0| = |x_2 - x_1|. \tag{2}$$

Therefore, (1) reduces to the inequality

$$\max(|x_1 - x_0|, y_1) + \max(|x_2 - x_0|, y_2) \geq \max(|x_2 - x_1|, |y_2 - y_1|). \tag{3}$$

We now investigate the following cases.

Let $|x_2 - x_1| \geq |y_2 - y_1|$. Then:

With an assumption that $y_1 \leq |x_1 - x_0|$ and $y_2 \leq |x_2 - x_0|$, the inequality (3) by (2) reduces an equality.

With an assumption that $|x_1 - x_0| \leq y_1$ or $|x_2 - x_0| \leq y_2$, we obtain the inequality (3) using (2).

Let $|y_2 - y_1| \geq |x_2 - x_1|$. Then:

With an assumption that $y_1 \geq |x_1 - x_0|$ and $y_2 \geq |x_2 - x_0|$, and using $y_1, y_2 > 0$, the inequality (3) reduces an obvious inequality

$$|y_2 - y_1| \leq y_1 + y_2.$$

With an assumption that $y_1 \leq |x_1 - x_0|$ or $y_2 \leq |x_2 - x_0|$, we obtain inequality (3) using (2). We note that for the case of $y_1 \leq |x_1 - x_0|$ and $y_2 \leq |x_2 - x_0|$, we obtain the equality $|x_2 - x_1| = |y_2 - y_1|$.

Hence, Theorem 1 is proved for case B.

Since we have checked all cases, Theorem 1 is proved.

5 Maximization of TS and Stack Sequencing

A dynamic programming algorithm is designed to maximize time savings. We assume that there are n containers that need to be loaded into a previously unloaded stack and that there are m containers that need to be unloaded from another stack. We also assume that the locations of the containers in the barge and in the container yard are known in advance. Let i represent the label of the container in the stack that needs to be loaded and j represent the label of the container from the stack that needs to be unloaded. Then the matrix of time savings can be created as follows:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & \mathbf{a_{1,j-1}} & a_{1,j} & \cdots & a_{1,m-1} & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & \mathbf{a_{2,j-1}} & a_{2,j} & \cdots & a_{2,m-1} & a_{2,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{a_{i-1,1}} & \mathbf{a_{i-1,2}} & \cdots & \mathbf{a_{i-1,j-1}} & a_{i-1,j} & \cdots & \cdots & \cdots \\ a_{i,1} & a_{i,2} & \cdots & a_{i,j-1} & \underline{a_{i,j}} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \cdots & a_{n-1,j-1} & a_{n-1,j} & \cdots & a_{n-1,m-1} & a_{n-1,m} \\ a_{n,1} & a_{n,2} & \cdots & a_{n,j-1} & a_{n,j} & \cdots & a_{n,m-1} & a_{n,m} \end{bmatrix}$$

where $\mathbf{a_{i,j}}$ is the time saving (TS) that is achieved by combining the loading of container i with the unloading of container j into a double cycle. From matrix A we can create a summary matrix S.

$$S = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & \mathbf{s_{1,j-1}} & s_{1,j} & \cdots & s_{1,m-1} & s_{1,m} \\ s_{2,1} & s_{2,2} & \cdots & \mathbf{s_{2,j-1}} & s_{2,j} & \cdots & s_{2,m-1} & s_{2,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{s_{i-1,1}} & \mathbf{s_{i-1,2}} & \cdots & \mathbf{s_{i-1,j-1}} & s_{i-1,j} & \cdots & \cdots & \cdots \\ s_{i,1} & s_{i,2} & \cdots & s_{i,j-1} & \underline{s_{i,j}} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{n-1,1} & s_{n-1,2} & \cdots & s_{n-1,j-1} & s_{n-1,j} & \cdots & s_{n-1,m-1} & s_{n-1,m} \\ s_{n,1} & s_{n,2} & \cdots & s_{n,j-1} & s_{n,j} & \cdots & s_{n,m-1} & s_{n,m} \end{bmatrix}$$

where element $s_{i,j}$ is calculated as:

$$s_{i,j} = \begin{cases} a_{i,j} & i = 1, \quad j = 1,...m \\ a_{i,j} & j = 1, \quad i = 1,...n \\ c_{max} + a_{i,j} & i = 2,...n, \quad j = 2,...m \end{cases}$$

Here, $c_{\max}$ represents the maximal bolded element from matrix S. Then the total time saving is equal to the maximal element of the last row or last column of S. Maximizing time savings does not necessarily also maximize the number of double cycles, as in [4] for seaport container terminals. This dynamic programming algorithm has more general significance and can also be applied to other problems in which there are two set of tasks with strict precedence constraints and where combining any pair of tasks yields some benefit. Optimization of loading and unloading containers is done using dynamic programming, following which it is necessary to find a sequence for reloading the stacks to maximize time savings. The schedule for stack reloading in a bay directly impacts the amount of time saved during reloading if double cycling is applied.

The model used for stack reload scheduling corresponds to the well-known method for the maximum traveling salesman problem (Max TSP [5]). Max TSP can be solved as a TSP by replacing each edge cost by its additive inverse. A formal definition of the asymmetric traveling salesman problem (ATSP) is as follows. Let $G = (V,H)$ be a given complete digraph, where $V = \{1,\ldots,d\}$ is the vertex set and $H = \{(e,l) : e,l \in V\}$ is the arc set; let t_{el} be the cost associated with arc $(e,l) \in H$ (with $t_{ee} = +\infty$ for each $e \in V$). A Hamiltonian directed cycle (tour) of G is a directed cyclevisiting each vertex of V exactly once, i.e., a spanning subdigraph $\widetilde{G} = (V,\widetilde{H})$ of G such that $|\widetilde{H}| = d$ (the number of elements in finite set $\widetilde{H}$ is denoted by $|\widetilde{H}|$), and $\widetilde{G}$ is strongly connected, i.e., for each pair of distinct vertices $(e,l) \in V, e < l$, both paths from e to l and from l to e exist in $\widetilde{G}$. The ATSP is used to find a Hamiltonian direct cycle $G^* = (V,H^*)$ of G with a minimum cost $\sum_{(e,l)\in H^*} t_{el}$. Without loss of generality, we assume $t_{el} \geqslant 0$ for any arc $(e,l) \in H$. The following integer linear programming formulation of the ATSP is well known:

$$\upsilon(ATSP) = \min \sum_{(e,l)\in H} t_{el} z_{el}$$

subject to

$$\sum_{e\in V} z_{el} = 1 \quad l \in V,$$

$$\sum_{l\in V} z_{el} = 1 \quad e \in V,$$

$$\sum_{e\in Q} \sum_{l\in V\setminus Q} z_{el} \geq 1 \quad Q \subset V : Q \neq \emptyset,$$

$$z_{el} \geq 0 \quad e,l \in V,$$

$$z_{el} \ \ integer \quad e,l \in V,$$

where $z_{el} = 1$ if and only if arc (e,l) is in the optimal tour.

6 Numerical Experiment

First, we analyzed the effect of the container yard length and width. In all examples, we assume that there is a random number of containers in every stack that needs reloading. Results are also presented for different numbers of reloading stacks in the barge and the effect on time saving of the number of containers in a stack. As depicted in Fig. 3, we analyzed river container terminals with a range of possible parameters covering existing systems. The velocities of the gantry and trolley are both 2 m/s. As shown in Fig. 3, the barge is parallel to the quay and without losing generality, two barges berthed alongside each other can be considered one barge (8 stacks per bay). In Table 1 the width of the container yard is 42m. In Table 2, the length of the container yard is 200m. There are two barges (12 bays) that need to be reloaded and there are 4 containers in a stack.

Table 1 Impact of container yard length on time saving

Container yard length [m]	200	500	700
Time saving [s]	9153	20810	28537

Table 2 Impact of container yard width on time saving

Container yard width [m]	24	36	51
Time saving [s]	8978	9072	9270

The length of the container yard has a substantial impact on the time savings achieved, but the effect of width is much less significant because the difference in terminal width has a much smaller range.

In Table 3 and Table 4, results are presented for a container yard with length 550m and width 52m. The impact of different numbers of stacks in the bay is shown in Table 3 (the number of stacks per bay are 4, 6 and 8), where for all three cases there are four containers in a stack. Table 4 presents results of time saving for reloading the barge with different numbers of containers in each stack (for 12 bay for reloading).

Table 3 Impact of different number of the stacks in the bay

Number of stacks in bay	4	6	8
Time saving [s]	8983	15720	22700

Table 4 Impact of different number of containers in the stack

Number of containers in a stack	3	4	6
Time saving [s]	18425	22688	31286

Logically, the impact of time saving using double cycling is greater for larger barges, and the same is true as the number of containers in each stack increases.

7 Conclusion

Increase in the volume of containerized goods has necessitated research on optimization for container reloading and storage in terminals. The efficiency of container cranes, as the lead element in terminals, is of key importance for the performance of container terminals.

In this paper, we give an overview of problems in the existing literature concerning seaport container terminals, with the main goal of a precise definition and determining the differences between seaport and river container terminals. We thus create a framework for river container terminal and RRMQG operations.

Hinterland transportation of containers is of great significance, and inland waterway transportation is an important element of the overall performances of large seaport terminals such as Rotterdam, Antwerp, Hamburg and Constanta.

The technical parameters for container quay cranes are different for seaport and river terminals. There is also a difference in the operational movements of these two types of cranes. Quay cranes are usually the only handling machine in inland terminals and are more versatile machines. In this paper, we present properties and operations of quay cranes for river terminals (RRMQG).

Numerical experiments on RRMQG operations demonstrate the possibility of significant time savings, and indicate the influence of different parameters. Depending on the design of the terminal and the quality of storage planning, time savings of up to 25% can be attained relative to a single cycle operation.

Acknowledgements. Results of this paper were supported by the project: TR - 35036 Ministry of Science and Technological Development (Serbia): Application of information technologies in ports of Serbia - from machine monitoring to connected computer network system with EU environment. The first author was supported by the project MNTRS 174009 and Provincial Secretariat for Science and Technological Development of Vojvodina.

References

1. Bierwirth, C., Meisel, F.: A survey of berth allocation and quay crane scheduling problems in container terminal. European Journal of Operational Research 202, 615–627 (2010)
2. Daganzo, C.F.: The crane scheduling problem. Transportation Research B 23, 150–175 (1989)
3. Douma, A., Schuur, P., Schutten, J.M.J.: Aligning the barge and terminal operations using service-time profiles. Flexible Services and Manufacturing Journal 23, 385–421 (2011)
4. Goodchild, A.V., Daganzo, C.F.: Double-Cycling Strategies for Container Ships and Their Effect on Ship Loading and Unloading Operations. Transportation Science 40, 473–483 (2006)
5. Gutin, G., Punnen, P.A.: The Traveling Salesman Problem and Its Variations. Kluwer Academic Publishers, Dordrecht (2002)
6. Kim, K.H.: Evaluation of the number of rehandles in container yards. Computers and Industrial Engineering 32, 701–711 (1997)
7. Kim, K.H., Gunther, H.O.: Container Terminals and Cargo Systems. Springer, Heidelberg (2007)
8. Kim, K.H., Park, Y.M.: A crane scheduling method for port container terminals. European Journal of Operational Research 156, 752–768 (2004)
9. Konings, R.: Opportunities to improve container barge handling in the port of Rotterdam from a transport network perspective. Journal of Transport Geography 15, 443–454 (2007)
10. Lim, A., Rodrigues, B., Xu, Z.: A m-Parallel Crane Scheduling Problem with a Non-crossing Constraint. Naval Research Logistics 54, 115–127 (2007)
11. Macharis, C., Bontekoning, Y.M.: Opportunities for OR in intermodal freight transport research: A review. European Journal of Operational Research 153, 400–416 (2004)
12. Meisel, F., Bierwirth, C.: A unified approach for the evaluation of quay crane scheduling models and algorithms. Computers and Operations Research 38, 683–693 (2011)
13. Meisel, F., Wichmann, M.: Container sequencing for quay cranes with internal reshuffles. OR Spectrum 32, 569–591 (2010)
14. Pap, E., Bojanic, V., Bojanic, G., Georgijevic, M.: Optimization of Container Quay Cranes Operations. In: Proceedings of the 9th International Symposium on Intelligent Systems and Informatics SISY, Subotica, pp. 137–140. IEEE Catalog Number CFP1184C-CDR (2011a)

15. Pap, E., Bojanic, V., Georgijevic, M., Bojanic, G.: Application of Pseudo-Analysis in the Synchronization of Container Terminal Equipment Operation. Acta Polytechnica Hungarica 8, 5–21 (2011b)
16. Stahlbock, R., Vos, S.: Operational research at container terminals: a literature update. OR Spectrum 30, 1–52 (2008)
17. Steenken, D., Vos, S., Stahlbock, R.: Container terminal operation and operations research - a classification and literature review. OR Spectrum 26, 3–49 (2004)
18. Vis, I.F.A., Koster, R.: Transshipment of containers at the container terminal: An overview. European Journal of Operational Research 147, 1–16 (2003)
19. Zhang, H., Kim, K.H.: Maximizing the number of dual-cycle operations of quay cranes in container terminals. Computers and Industrial Engineering 56, 979–992 (2009)

Network Monitoring Using Intelligent Mobile Agents

Goran Sladić, Milan Vidaković, and Zora Konjović

Abstract. Network monitoring is a critical issue in today's rapidly changing network environment. Existing centralized client-server based network management frameworks suffer from problems such as insufficient scalability, interoperability, reliability, and flexibility, as networks become more geographically distributed. This work describes implementation of the agent-based system for network components and services monitoring. The system is designed in a modular fashion to provide easy and efficient inclusion of diverse monitoring objects. Several types of the intelligent agents are defined to perform efficient monitoring of diverse network components, services and applications using different monitoring protocols. In proposed architecture, we use an ontology for representation of monitoring information to provide semantics for building an underlying knowledge base that not only allows agents to communicate, but also to reason with each other, enabling the desired tasks to be performed collaboratively. The system implementation is based on the XJAF (Extensible Java-based Agent Framework) agent framework and the Java EE technology.

Keywords: Intelligent mobile agent, network management, network component, extensible Java-based agent framework.

1 Introduction

The functionality of a computer network can be described by availability of the services for end users. It can be threatened if there are problems regarding functionality of network elements. These problems involve network elements failure and malfunctioning. Network management is a comprehensive activity whose goal

Goran Sladić · Milan Vidaković · Zora Konjović
University of Novi Sad,
Faculty of Technical Sciences,
Novi Sad, Serbia
e-mail: {sladicg,minja,ftn_zora}@uns.ac.rs

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 301–318.
DOI: 10.1007/978-3-642-33959-2_16

is to minimize network failure and malfunctioning. It is a critical issue in today's rapidly changing network environment. Existing centralized client-server based network management frameworks suffer form problems such as insufficient scalability, interoperability, reliability, and flexibility, as networks become more geographically distributed. Network management implies enforcement of the following processes:

- monitoring,
- notification, and
- problem solving.

Monitoring is the process of data acquisition regarding network elements functionality. Effective monitoring provides state information with the required accuracy to the right places in the network, at minimum cost. *Identification* is the process of failure and malfunction detection, based on data gathered during network monitoring. Notification informs system environment about changes in network functioning. *Problem solving* involves actions required to fix malfunctioning or broken devices.

Due to a lack of support for standardized monitoring and management protocols, heterogeneity of network components, ever-increasing network deployment, and rapid development of new network technologies, flexible mechanisms to monitor network are highly required. Instead of using one centralized application, which is usually complex, a group of simple collaborating software systems can solve the same problem more efficiently. These facts implicate the agent paradigm (agent technology) as a suitable candidate solution for development of network management systems [2, 7] due to their commodity for implementation of small software components which can achieve the common goal in collaboration. They decentralize processing and control and, consequently, reduce the traffic around the management station and distribute processing load.

The use of semantic models eases interoperability between different management domains and applications, not at the level of data exchange, which is mostly solved with standard data models that currently exist (e.g., SNMP MIB), but at the level of knowledge sharing [18]. Achieving this objective enables different administrators and/or management software components to clearly understand the definitions and management rules and goals provided by other administrators (possibly using different management platforms). It also enables the understanding of what the administrators and/or management software components really meant when they provided these definitions and management objectives. The use of this semantic modeling approach also benefits the non-trivial processes related with network management. Examples of these processes are conflict detection and resolution and policy refinement from high-level objectives to low-level configurations.

In this paper we propose extensible network monitoring system based on the mobile intelligent agent architecture and ontology for representation of monitoring results. The proposed architecture is based on the XJAF (Extensible Java-based Agent Framework) [17, 21, 23, 26, 24, 25] agent framework. Several types of the intelligent agents are specified to perform efficient monitoring of diverse network components, services and applications using different monitoring protocols. Thus, system provide easy and efficient inclusion of diverse monitoring objects. We

believe an information and ontological modeling-based approach delivers considerable improvements over existing manually-configured network monitoring solutions due to several reasons. First, it enables a formal representation of monitoring results and supports rich representation of different contextual information. Second, it allows the necessary semantic interoperability between different systems and also enables easy adjustment of the model for use in different systems or for different purposes. Finally, it provides a high degree of inference making by providing additional vocabulary along with a formal semantics to define classes, properties, relations and axioms. The proposed system can be easily adapted to use different ontologies.

The rest of the paper is structured as follows. Section 2 reviews the related work. Section 3 presents an overview of the XJAF agent framework. The system architecture is presented in Section 4. Section 5 concludes the paper and outlines further research directions.

2 Related Work

Ku et al. in [12] propose a framework for an intelligent mobile agents architecture to achieve distributed network management. The proposed architecture reduces the complexity of network management at the managing entity by delegating part of the management responsibility to the managed network entities. The framework includes the mobile agent architecture, travel plan, action plan, management delegation, intelligence and security.

The solution proposed in [2] uses mobile agents to measure network monitoring performance. Mobile agents move across the network and gather responses of SNMP queries. Aglets Software Developer Kit was used as agent development environment. Initial results indicate that the mobile agent approach performs better in network monitoring when the number of network nodes increases significantly.

The system presented in [7] consists of two modules. One provides the agent construction toolkit (the AgentBean Development Kit, ADK) and the other is a plug-in abstraction layer for existing mobile agent platforms enabling execution of mobile agents created with ADK. As a case study for this system, an example of network topology determining is presented. Agents are used for determining topology of the given network. Whenever an agent moves to a new place, it queries the local host routing table using the preconfigured SNMP query. The agent stores the routing table retrieved from the local host. Then, it checks which hosts from the routing table are running one of supported agent platforms. These hosts are stored in the list of hosts to be visited providing they have not already been visited.

A distributed intrusion detection system for ad hoc wireless networks based on mobile agent technology is proposed by Kachirski and Guha [11]. By efficiently merging audit data from multiple network sensors, authors analyze the entire ad hoc wireless network for intrusions and try to inhibit intrusion attempts. Presented approach restricts computation-intensive analysis of overall network security state to a few key nodes. These nodes are dynamically elected, and overall network security is not entirely dependent on any particular node.

Nowak and Bagrij in [16] describe the subject of using the multilevel distributed agent-based architecture for network monitoring purposes, thus making possible to efficiently gather data which may be useful for building comprehensive view of the whole network in automated way. A network model created by proposed approach may find applications in further stages of management process, which tends to increase system dependability, e.g. using it as one of inputs of computer network simulator.

The network performance monitoring using mobile agents is presented in [19]. The proposed solution is used for monitoring following parameters: source address, destination address, number of packets, startup time, down time and bandwidth utilization of the network component.

Aversa et al. in [3] propose a distributed model for delivery of services which exploit pervasive extension composed of different kind of sensors and actuators. The presented solution uses autonomous agents which execute on embedded devices. They can be configured to collect information from close sensors and to check the compliance of user activities with a set of rules.

Ho et al. in [8] present a scalable framework for wireless network monitoring. The proposed framework relies on a distributed set of agents within the network to monitor network devices and store the collected information at data repositories. Its key features are its extensibility for new functionality and seamless support for new devices and agents in the monitoring framework.

The research [20] discusses different attacks in wireless sensor networks and how these attacks were efficiently detected by using the proposed agent based model. The given model identifies the abnormal events in a largely deployed distributed sensor network.

Stephan et al. in [22] present a case study in the design, implementation, and evaluation of a network management platform based on mobile agents. In this framework, a management platform is realized using a number of different types of agents that can be dispatched to a remote site for collection and processing of management information depending on the situation/management event. The given framework uses ontology database to define the scope of the knowledge maintained by family of agents and management database to store all management information as in the case in traditional management platforms.

Jennings et al. in [10] contend that the essence of autonomic network management is the capability of a system to self-govern its behavior, but only within the constraints of the (human-specified) goals that the system as a whole seeks to achieve. Authors propose the use of information and ontological modeling to capture knowledge relating to network capabilities, environmental constraints, and business goals and policies, together with reasoning and learning techniques, to enhance and evolve this knowledge.

Yang and Chang in [27, 28] focuses on how to effectively integrate different networking devices under various enterprize environmental demands to develop a network monitoring management system. Intelligent network management system with ontology-supported multi-agent techniques is proposed. They present a system to collect information through the cooperation of intelligent agent software, in

addition to providing warnings after analysis to monitor and predict some possible error indications among controlled objects in the network.

Carrera and Iglesias in [4] present a multi-agent system architecture for network diagnosis under uncertainty that combines two reasoning processes: semantic reasoning and Bayesian reasoning. Given approach proposes to use Bayesian inference to handle uncertainty inherent in any diagnosis process and semantic inference to discriminate which action is the best one to perform depending on the available data.

An approach to define and execute composite network management processes based on semantic web service technologies is presented in [5]. For this purpose, web services and the semantic web technologies are used. Authors use representation of composite processes with the OWL-S service ontology.

Lopez de Vergara et al. in [14] present an approach that uses OWL to define network and system management information. The paper [13] proposed an algorithm for mapping CIM, as well as other information models, into a common ontology. The given solution uses ontology interaction techniques to improve the network management information interoperability. The proposed method takes into account the semantic contained in the information. With this, a common model can be generated to be used by a manager, independently of the underlying management domains.

Moraes et. al in [15] propose the MonONTO domain ontology for network monitoring. It brings together terms related to the domains of advanced Internet applications and their QoS requirements, to network monitoring environments, and aspects concerning end-users. The MonONTO major goal is to enable the organization of this knowledge and allow for the creation of a knowledge base comprising information about that domain and the implementation of an expert system, which will advise the user or the network administrator about the possible performance of an advanced network application at a given time.

3 Extensible Java-Based Agent Framework

The main goal of this research was to develop agent-related technologies in the field of network monitoring. Since our previous research [17, 21, 23, 26, 24, 25] has produced the fully operational Java-based agent framework, the decision was made to use Extensible Java-based Agent Framework (XJAF) as the basis of this project.

The framework is based on the Java EE technology. The Java EE technology is particularly useful because it comprises a large set of technologies and provides for scalability, reliability and has the large number of implementations. XJAF uses the plug-in technology which provides additional flexibility, since it allows components to be substituted by others without rebuilding the whole framework. XJAF is FIPA [1] compliant agent framework based on the Java technology and introduces the following concepts:

- message exchange,
- agent mobility
- security, and
- agent and service directories.

Also, this framework proposes additional component of the agent framework: the interfacilitator connectivity component which defines how separate facilitators form a network.

The XJAF system consists of *clients*, the *FacilitatorProxy* component and *Facilitator* component (see Figure 1). The clients refer to the facilitators for task execution. The task is being executed by the *agents* recruited by the *Facilitator* component. The *Facilitator* component, which is the main component of the framework is implemented as an EJB component. This component is deployed in the Java EE application server and is not directly used by the client. For this purpose, the *FacilitatorProxy* component is used. *FacilitatorProxy* ensures that the client application can access the facilitator. It also hides all techniques necessary for work with agents from the client. The client only needs to connect to *FacilitatorProxy* and pass it the task or the ACL (Agent Communication Language) [9] message. It also passes the corresponding listener, which would notify it of the execution result. All other details are managed by *FacilitatorProxy*.

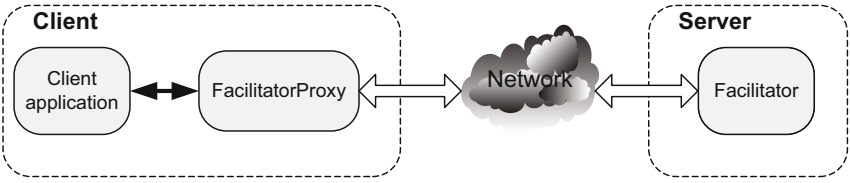


Fig. 1 Communication between client and facilitator

The facilitator forwards parts of its job to the corresponding managers. The managers are instances of classes implementing the corresponding managerial interfaces. The *AgentManager* interface is responsible for allocating and releasing agents. It represents the *Agent Directory Service* component of the FIPA specification. The *TaskManager* interface manages the tasks. It stores tasks, assigns tasks to corresponding agents and provides a way of notifying the client of the results. The *MessageManager* interface is responsible for inter-agent communication and it actually represents the *Transport Message* component of the FIPA specification. The *ConnectionManager* interface manages inter-facilitator connections and relations. The *SecurityManager* interface defines security aspects of the agent framework. The *ServiceManager* interface defines service directory and represents *Service Directory* for the *Services* component of the FIPA specification.

The classes which implement the mentioned interfaces respect the corresponding algorithms for individual functions. The system is designed so that it is possible to choose an arbitrary manager when configuring provided that it implements the given interface. This enables use of arbitrary managers whose existence is not necessary at compile-time, but is at the time of initialization (plug-in concept). Figure 2 lists all the managers in the framework.

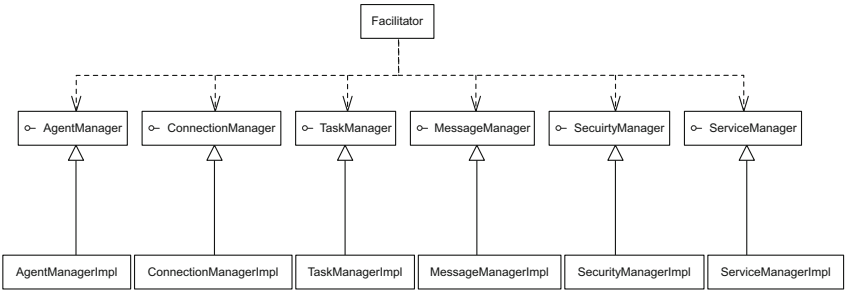


Fig. 2 Functionality of individual parts is assigned to managers

The facilitators are connected into the hierarchically organized agent network (as shown in Figure 3). This network generally has one root node and a number of descendant nodes. Each facilitator is automatically registered on the network at the initialization time. This means that the developer does not have to know the exact address of an arbitrary facilitator and does not have to maintain the list of all available facilitators. Instead, the nodes are registered automatically and the list of all available facilitators is maintained automatically. If each agent framework is connected to a particular system, this automatically makes the network of those systems available.

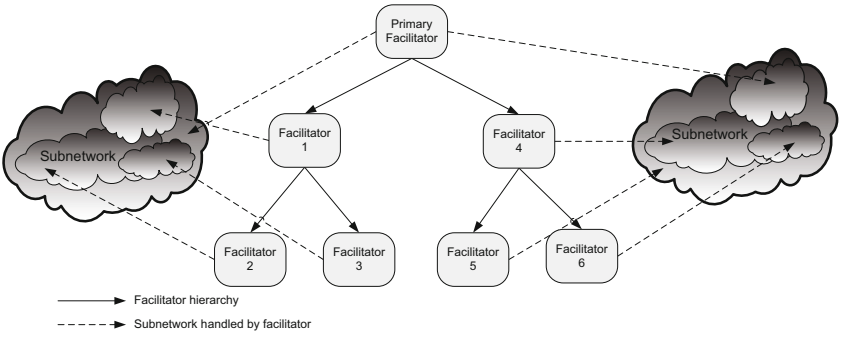


Fig. 3 Example of agent framework network

One example of an agent network used for the computer network monitoring is displayed in the Figure 3. This network is organised as a hierarchical network of agent frameworks. Each node in this agent network is a single agent framework which handles certain subnetwork(s). The node is registered on the agent network at the initialisation time. This organisation provides dynamic network setup since all frameworks register during setup and unregister during shut down.

4 System Architecture

The proposed system supports three basic operations: creation of monitoring or discovery configurations, monitoring specified properties of registered network components or services, and discovery of unregistered network components or services.

The global architecture is presented in Figure 4. A client uses agents in the agents network to store created network monitoring or discovery configurations in the configuration base. Another agents from the agents network use these configurations to perform network monitoring or discovery of network components or services. After monitoring or discovery is done, agents uses obtained results to create ontology individuals and populate them to ontology base. The client can access to the ontology base to get required information.

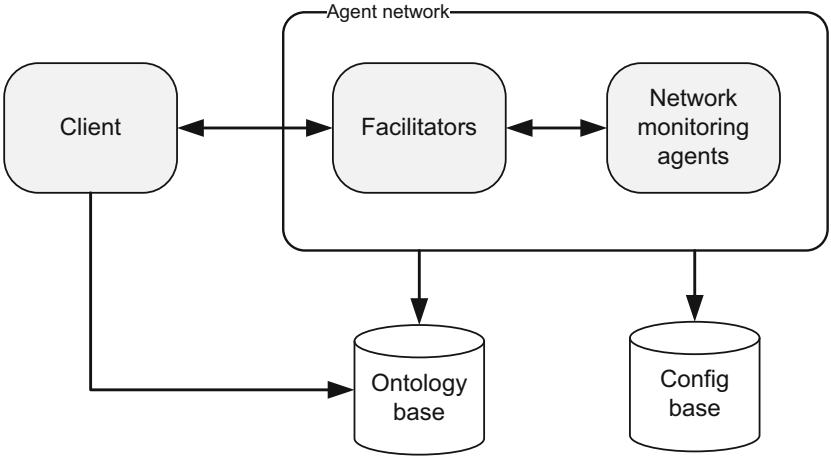


Fig. 4 Global arhitecture

4.1 Client

The client sends a task to the facilitator. The facilitator looks for an appropriate agent and forwards the task to it. Communication between a client and the facilitator is asynchronous. This agent framework uses the concept of listeners for

communication between the client application and the framework. A listener is a Java class which has specialised methods to be invoked when the appropriate type of event occurs. In this case, there are four types of events [26]:

- *action started* event, which is invoked when an agent starts the task execution,
- *action performing* event, which is invoked when an agent partially finishes execution of the task, and
- *action complited* event, which is invoked when an agent finishes the task execution.

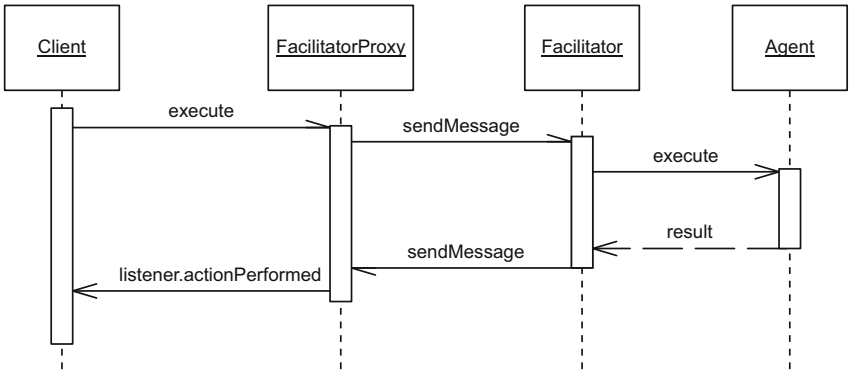


Fig. 5 Task execution sequence diagram

The task can be done in a single step, in which case the first and the third messages appear in the given order. If the task needs more steps to be performed, the *action started* event is generated, then an arbitrary number of the *action performing* events is generated, and at the end, the *action completed* event is generated. The only parameter of these three methods is the instance of the *AgentEvent* class, which contains the result of task execution [26]. Figure 6 shows relation between the client application and the classes used for communication with agents. The class *ConfigTask* models the configuration tasks that client sends to agents. The *MonitoringTask* subclass describes information necessary to perform monitoring, while the *DiscoveryTask* subclass describes information for discovery of new network components or services. The class *AdHocMonitoringTask* provides mechanism by which the client can run the ad hoc monitoring, without creating monitoring configuration, and to immediately receive monitoring results. The *ConfigurationListener* listener class receives results of the *ConfigTask* subtasks execution, while *AdHocMonitoringListener* receives results of the ad hoc monitoring tasks.

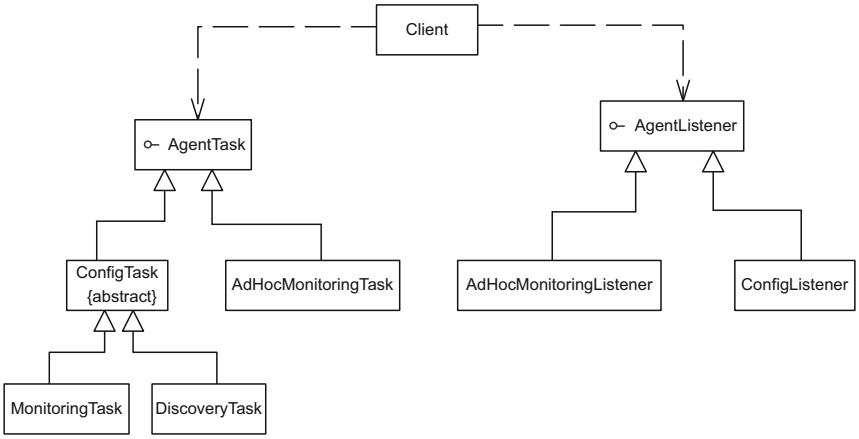


Fig. 6 Client classes for comunication with agents

4.2 Agents

The agents used for network monitoring are presented in Figure 7. The client uses *ClientAgent* for communication with the other agents in the network. Using this agent the client can add a new configuration task in the configuration base or initiate a new ad hoc monitoring. *DispatcherAgent* reads configuration base and manages monitoring and discovering processes. Monitoring is performed by a subclass of the *MonitoringAgent* class. Currently three subclasses are created. The class *SNMPMonitoringAgent* uses SNMP (Simple Network Management Protocol) for monitoring, the class *JMXMonitoringAgent* uses JMX (Java Management Extensions) for monitoring, while the class *WebServiceMonitoringAgent* uses web services to perform monitoring. The subclasses of the *DiscoveryAgent* class are used for network components or services discovery. *OntologyAgent* uses monitoring or discovery results to create ontology individuals ad populate them to the ontology base. The system is designed to support different ontologies. To support a new ontology only a proper implementation of the OntologyAgent subclass should be implemented. The *MonONTOAgent* creates ontology based on the MonONTO [15] ontology. *AggregateOntologyAgent* creates ontology instances by combining results of different monitoring and/or discovery tasks.

In order to execute their task agents rely on the proper services (seee Figure 8). The *ConfigService* class is used for accessing to configuration base. *ClientAgent* uses this class to manage configurations defined by user, while *DispacherAgent* uses *ConfigService* to read configuration tasks. *OntologyService* provides uniform access to the ontology base. Using this class, agents can uniformly access to ontology data in different repositories such as relational databases, XML databases or specialized RDF stores. This class defines methods for the basic CRUD (create, read, update, delete) operations and operations for searching and reasoning ontology. In order to create proper ontology instances the *AgregateOntologyAgent* class uses subclasses

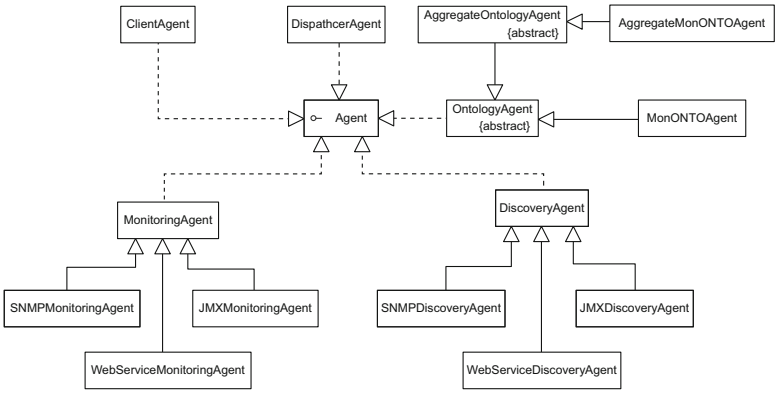


Fig. 7 Agents for network monitoring

of the *OntologyAggregator* class. *OntologyAggregator* represents the implementation of a proper algorithm that uses monitoring and/or discovery results to produce ontology instances or some other information that another *OntologyAggregator* will use to produce ontology instances. The *AgregateOntologyAgent* class can combine more the *OntologyAggregator* subclasses to produce ontology instances. The *AggregatorRegistryServices* class represents the registry that keeps information which *OntologyAggregator* subclasses are needed to create ontology instances for each *AgregateOntologyAgent* subclasses.

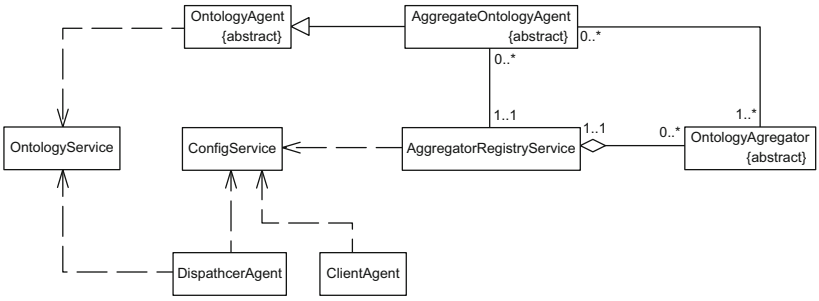


Fig. 8 Services for network monitoring

4.3 Configuration

We use XML Schema to model configuration tasks. The monitoring configuration tasks is presented in Figure 9, while discovery configuration task is presented in Figure 10. The *monitoring_object* (*discovery_object*) element contains name of the network service/component for which monitoring (discovering) configuration is defined. The *monitoring_agent* (*discovery_agent*) element contains the name of the

agent which will perform monitoring (discovery) of the given object. The agent defined by the *ontology_agent* element will be used to produce ontology instance using monitoring (discovery) results. If the condition defined by the *condition* element is defined than task will be executed only if the condition is satisfied. The element *nodes* contains a list of nodes (facilitators) to which agent will migrate and execute monitoring or discovery task. If a condition is assigned to a node, task will be executed on the node only if the condition is satisfied. The element *schedule_period* defines when task should be executed. For each task arbitrary number of parameters can be defined (e.g. how long monitoring should last, on which port is web server, etc.). For each parameter name, type and value is defined.

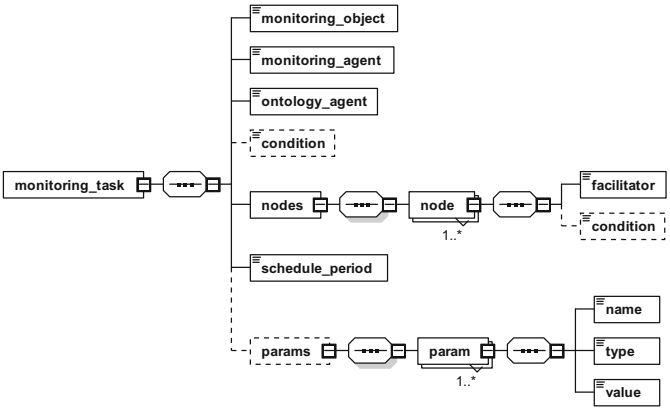


Fig. 9 XML Schema for monitoring task

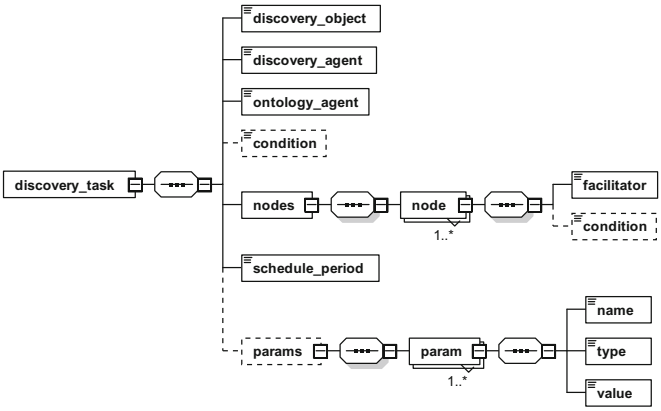


Fig. 10 XML Schema for discovery task

The condition is based on the condition for context defined using ontologies from our earlier research presented in [6]. The condition is defined as a logical expression which may consist of queries for searching ontology (like SPARQL), functions, logical operators $\{\neg, \wedge, \vee\}$ and comparison operators $\{<, \leq, >, \geq, =, \neq\}$. The functions are used to retrieve some current information from a system, like *what is current node*. The condition defined in the EBNF notation is presented in the following Listing.

```
Condition ::=  Query
             | Function
             | Condition BinaryOperator Condition\\
             | UnaryLogicalOperator Condition
             | "("Condition")"
             | "TRUE"
             | "FALSE"
             | String
             | Number
BinaryOperator ::= ComparsionOperator|LogicalBinaryOperator\\
ComparsionOperator ::= "<"|"<="|">"|">="|"=="|"!="
LogicalBinaryOperator ::=\\ "$\&\&$"|"||"
UnaryLogicalOperator ::= "!"
Query ::= "QUERY {" (String|Function)\\{String|Function} "}"
Function ::= "$$"Name"("[Param{","Param}]")$$"
Param ::= String|Number|\\Function|Query
```

4.4 Monitoring Process

The monitoring process is described using sequence diagrams in Figure 11 and Figure 12. The process is initiated by *DispatcherAgent* (see Figure 11) which reads monitoring configurations from the configuration base using the ConfigService class. While task condition is satisfied *DispatcherAgent* verifies a condition of the current node and if that condition is satisfied it sends message with monitoring parameters to *MonitorAgent*. After *MonitorAgent* finishes monitoring it informs *DispatcherAgent* about results. Verification of the condition before a new monitoring message is sent to *MonitoringAgent* provides ontology dependant monitoring process. *DispatcherAgent* agent will continue or stop to sending new monitoring messages based on previous monitoring information from the monitoring base.

When *MonitoringAgent* receives the monitoring message (see Figure 12), it first checks should it migrate on another node. If migration is required the agent finds remote facilitator, by using the local facilitator, migrate to it and perform necessary initialization. After that *MonitoringAgent* monitors the specified object. When monitoring is done the agent sends message with monitoring results to *OntologyAgent* which creates proper ontology individuals and save them to the ontology base using the *OntologyServiceClass*.

The process of discovery of unregistered network components/services is similar tothe process of monitoring of registered objects. Execution of a discovery task is presented in Figure 13. While task condition is satisfied *DispatcherAgent* verifies a condition of the current node and if that condition is satisfied it sends message with monitoring parameters to *DiscoveryAgent*. On receiving message *DiscoveryAgent*

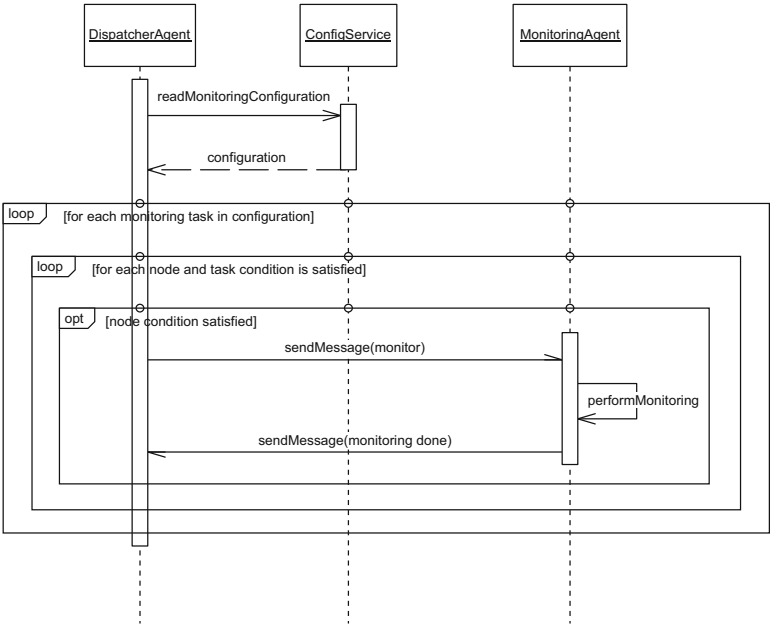


Fig. 11 Dispatching of monitoring tasks

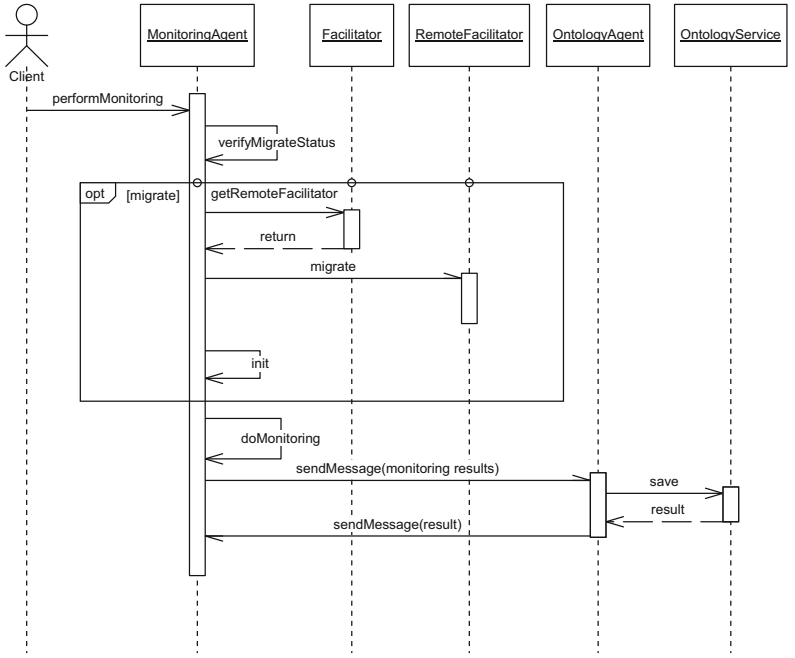


Fig. 12 Monitoring sequence diagram

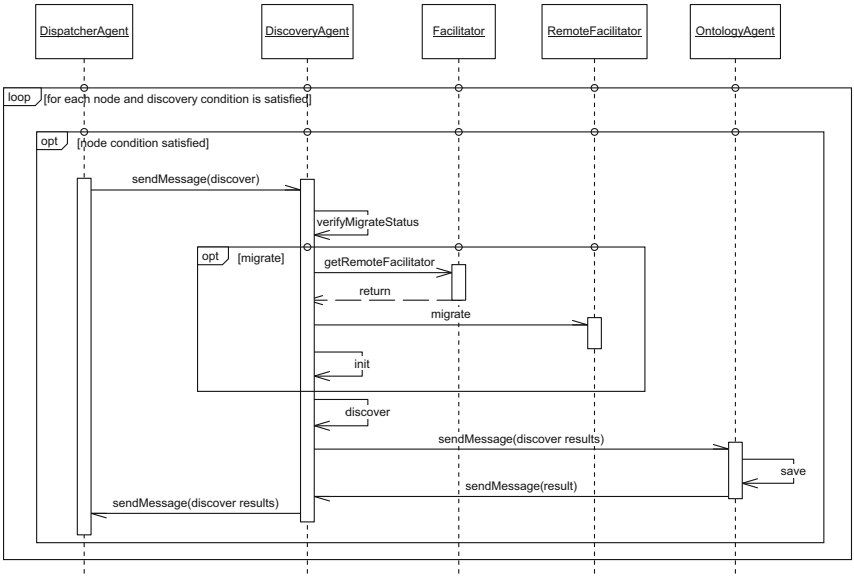


Fig. 13 Discovery sequence diagram

verifies is it requested to migrate on another node. if migration is requested the agent migrates on remote node and perform necessary initialization. *DiscoveryAgent* performs specified discovery procedure and sends result message to *OntologyAgent* which creates proper ontology individuals and save them to the ontology base using the *OntologyServiceClass*. Finally *DiscoveryAgent* informs *DispatcherAgent* about finished job.

5 Conclusion

Disadvantages of centralized network monitoring applications are complexity and the need for monitoring different types of network services. This model of monitoring can be replaced with a large number of small software systems capable of cooperative network monitoring. This distributed model can be more efficient than the centralized one. Agent technology can be utilized as an approach to distributed network monitoring. This paper presents one example of agent-based technology used for network availability and vulnerability.

The architecture based on intelligent mobile agents under the XJAF agent framework for network monitoring is specified. We propose extensible agent-based system for network monitoring which would support monitoring of diverse network objects (including network components, services and applications) using different protocols. An ontology is used for representation of monitoring information to provide rich formal representation of knowledge that not only allows agents to com-

municate, but also allow to reason with each other, enabling the desired tasks to be performed collaboratively. The system is designed so that different ontologies can be used. Future work will include implementation of more sophisticated algorithms for task distribution between agents and support for monitoring using other protocols.

Acknowledgements. This article is part of the research project *Infrastructure for Technology Enhanced Learning in Serbia* supported by the Ministry of Education and Science of the Republic of Serbia [Project No. 47003].

References

1. The Foundation of Intelligent Physical Agents (FIPA), http://www.fipa.org/ (accessed December 26, 2011)
2. Adhicandra, I., Pattinson, C., Shaghouei, E.: Using mobile agents to improve performance of network management operations. In: Proceeding of 4th Annual Symposium of Postgraduate Networking Conference, Liverpool, UK (2003)
3. Aversa, R., Di Martino, B., Venticinque, S.: Distributed agents network for ubiquitous monitoring and services exploitation. In: IEEE International Conference on Computational Science and Engineering, Vancouver, Canada, pp. 197–204 (2009)
4. Carrera, A., Iglesias, C.: Multi-agent Architecture for Heterogeneous Reasoning under Uncertainty Combining *MSBN* and Ontologies in Distributed Network Diagnosis. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Lyon, France, pp. 159–162 (2011)
5. Fuentes, J.M., de Vergara, J.E.L., Castells, P.: An Ontology-Based Approach to the Description and Execution of Composite Network Management Processes for Network Monitoring. In: State, R., van der Meer, S., O'Sullivan, D., Pfeifer, T. (eds.) DSOM 2006. LNCS, vol. 4269, pp. 86–97. Springer, Heidelberg (2006)
6. Gostojic, S., Sladic, G., Milosavljevic, B., Konjovic, Z.: Context-sensitive Access Control Model for Government Services. Journal of Organizational Computing and Electronic Commerce 22, 2 (2012), doi:10.1080/10919392.2012.667717
7. Gschwind, T., Feridun, M., Pleisch, S.: Building mobile agents for network and systems management from reusable components. In: Proceedings of the First International Symposium on Agent Systems and Applications and Third International Symposium on Mobile Agents, Palm Springs, CA, USA, pp. 13–21 (1999)
8. Ho, C., Ramachandran, K., Almeroth, K., Belding-Royer, E.: A scalable framework for wireless network monitoring. In: Proceedings of the 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH 2004), Philadelphia, USA, pp. 93–101 (2004)
9. The Foundation of Intelligent Physical Agents (FIPA), FIPA ACL Message Structure Specification, FIPA standard (2002), http://www.fipa.org/repository/aclspecs.html
10. Jennings, B., Der Meer, S.V., Balasubramaniam, S., Botvich, D., Foghlu, M.O., Donnelly, W., Strassner, J.: Towards autonomic management of communications networks. IEEE Communications Magazine 45(10), 112–121 (2007), doi:10.1109/MCOM.2007.4342833

11. Kachirski, O., Guha, R.: Intrusion detection using mobile agents in wireless ad hoc networks. In: IEEE Workshop on Knowledge Media Networking (KMN 2002), Kyoto, Japan, pp. 153–158 (2002)
12. Ku, H., Luderer, G., Subbiah, B.: An Intelligent Mobile Agent Framework for Distributed Network Management. In: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 1997), pp. 160–164 (1997)
13. de Vergara, L.J., Villagra, V., Berrocal, J.: An ontology-based method to merge and map management information models. In: Proceedings of the HP Openview University Association Tenth Plenary Workshop, Geneva, Switzerland (2003)
14. de Vergara, L.J., Villagra, V., Berrocal, J.: Applying the Web Ontology Language to Management Information Definitions. IEEE Communications Magazine 42(special issue on XML Management), 68–74 (2004)
15. Moraes, P.S., Sampaio, L.N., Monteiro, J.A.S., Portnoi, M.: A Domain Ontology for Network Monitoring and Recommendation for Advanced Internet Applications Users. In: Network Operations and Management Symposium Workshops, Salvador da Bahia, pp. 116–123 (2008)
16. Nowak, K., Bagrij, L.: Using distributed multilevel agent-based monitoring technique for automated network modelling approach. In: International Conference on Dependability of Computer Systems, Szklarska Poreba, Poland, pp. 61–72 (2007)
17. Pesovic, D., Vidakovic, M., Ivanovic, M., Budimac, Z., Vidakovic, J.: Usage of Agents in Document Management. Computer Science and Information Systems 8(1), 193–210 (2011), doi:10.2298/CSIS090608019P
18. Pras, A., Schonwalder, J., Burgess, M., Festor, O., Perez, G.M., Stadler, R., Stiller, B.: Key research challenges in network management. IEEE Communications Magazine 45, 104–110 (2007)
19. Pugazendi, R., Duraiswamy, K.: Mobile agents - A solution for network monitoring. In: International Conference on Advances in Recent Technologies in Communication and Computing, Kerala, India, pp. 579–584 (2009)
20. Sa, M., Rath, A.K.: A simple agent based model for detecting abnormal event patterns in distributed wireless sensor networks. In: International Conference on Communication, Computing and Security (ICCCS 2011), Rourkela, India, pp. 67–70 (2011)
21. Sladic, G., Vidakovic, M., Konjovic, Z.: Agent Based System for Network Availability and Vulnerability Monitoring. In: 9th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, pp. 53–58 (2011)
22. Rayan, S., Pradeep, R., Paramesh, N.: Network management platform based on mobile agents. International Journal od Network Management 14(1), 59–73 (2004), http://dx.doi.org/10.1002/nem.508
23. Vidakovic, M., Konjovic, Z.: EJB based intelligent agents framework. In: Proceedings of the 6th IASTED International Conference on Software Engineering and Applications, Cambridge, Massachusetts, USA, pp. 343–348 (2002)
24. Vidakovic, M., Sladic, G., Konjovic, Z.: Security management in J2EE based intelligent agent framework. In: Proceedings of the 7th IASTED International Conference on Software Engineering and Application, Marina Del Rey, CA, USA, pp. 128–133 (2003)
25. Vidakovic, M., Sladic, G., Zaric, M.: Metadata harvesting using agent technology. In: Proceedings of the 8th IASTED International Conference on Software Engineering and Applications, Cambridge, USA, pp. 489–493 (2004)
26. Vidakovic, M., Milosavljevic, B., Konjovic, Z., Sladic, G.: Extensible Java EE-based agent framework and its application on distributed library catalogues. Computer Science and Information Systems 6(2), 1–28 (2009)

27. Yang, S.-Y., Yi-Yen, C.: A New Network Management System with Ontology-Supported Multi-agent Techniques. In: Proceedings of the International Symposium on Parallel and Distributed Processing with Applications, pp. 275–282 (2010), http://dx.doi.org/10.1109/ISPA.2010.31
28. Yang, S.-Y., Yi-Yen, C.: An active and intelligent network management system with ontology-based and multi-agent techniques. Expert Systems with Applications 38(8), 10320–10342 (2011)

Progressive Pedestrian Localization Using Neural Networks

Markus Gressmann, Günther Palm, and Otto Löhlein

Abstract. The precise localization of pedestrians in images is a difficult problem with many practical applications in the fields of driver assistance, autonomous vehicles and visual surveillance. Localization can be treated as a subsequent step to pedestrian detection that aims at finding the exact position of pedestrians in an input image. In this work, two different approaches for pedestrian localization using neural networks with local receptive fields are presented. The first approach uses a trained ranking classifier to determine the relative order of image windows in regard to their localization quality (coverage) of the pedestrian. Localization is then performed via sampling of the window space in the vicinity of an initial detection. For the second approach, a binary classifier is trained to stepwise move an initial window towards the optimal position. Only few network evaluations are required for this method to converge, making it applicable for real-time detection systems. It is shown how the localization task can be split up into consecutive subtasks, which allows the training of a dedicated classifier for each subtask. This progressive localization scheme improves localization precision and simplifies evaluation of the resulting classifiers. Both approaches are evaluated in detail on the publicly available Daimler Pedestrian Detection Benchmark dataset and the results are compared to a standard detection approach based on non-maximum suppression.

Keywords: Neural network, image, pedestrian, binary classifier.

Markus Gressmann
Institute of Measurement, Control, and Microtechnology,
University of Ulm, Ulm, Germany
e-mail: markus.gressmann@uni-ulm.de

Günther Palm
Institute of Neural Information Processing,
University of Ulm, Ulm, Germany
e-mail: guenther.palm@uni-ulm.de

Otto Löhlein
Group Research, Daimler AG, Ulm, Germany
e-mail: otto.loehlein@daimler.com

E. Pap (Ed.): Intelligent Systems: Models and Applications, TIEI 3, pp. 319–339.
DOI: 10.1007/978-3-642-33959-2_17

1 Introduction

The challenging task of finding humans in images has received significant attention in recent machine learning and computer vision research. It is of particular interest in the automotive domain, where an accurate estimation of a pedestrian's position is the first step towards reliable collision avoidance systems [7, 11, 17]. Most publications on pedestrian detection aim at finding bounding windows in the image that are sufficiently close and of similar scale to manually labeled ground truth windows. A pedestrian is counted as detected, if a similarity measure applied to detection and ground truth window exceeds a predetermined threshold [5]. This similarity measure is usually based on the spatial coverage of the two windows, for instance the commonly used PASCAL measure [8]:

$$a_0(A,B) = \frac{\text{area}(A \cap B)}{\text{area}(A \cup B)}, \tag{1}$$

where a detection is counted as a true positive if a_0 exceeds 50%. This binary evaluation measure does not favor detections that match the ground truth label well, nor does it punish those that only slightly exceed the coverage criterion threshold. Based on this observation, the notion of *Pedestrian Localization* in contrast to *Pedestrian Detection* is introduced in [10].

The task of Pedestrian Localization is defined as finding an image window $\mathbf{w}^*$ that exhibits a high value of a_0 for a given label $\mathbf{w}_l$ in an image I:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{armax}} \quad a_0(\mathbf{w}, \mathbf{w}_l), \tag{2}$$

where the label position $\mathbf{w}_l$ can only be inferred from the image data. The key difference to the detection task is that one wants to find the window that *best* matches the label, instead of finding *any* matching window.

Several applications require or benefit from precise localization of pedestrians in images. Camera based active collision avoidance and mitigation systems in automotive scenarios depend on accurate input from the vision algorithms. Misjudging a pedestrian's position in the image, and therefore in the world, can have fatal consequences. Other sensors, such as radar units or ultrasonic sensors, can be employed to verify or correct the pedestrian's position in the world. But even for such fusion approaches, a reliable estimation of the pedestrian's location in the image is crucial.

This difference between detection and localization is also reflected in the construction of pedestrian detection classifiers, referred to as *training*. During training, the classifier is presented samples annotated with a binary label, declaring the samples either positive (pedestrian) or negative (non-pedestrian). Negative samples are obtained by a sliding window-approach on the input images, while positive samples are generated from ground truth labels. To increase the robustness of the resulting classifier and to account for errors during labeling, positive samples obtained from

slightly shifted labels are usually added to the training dataset. It is clear that such a training procedure, albeit resulting in good *detection* performance, does not lead to satisfactory results for the *localization* task in the sense of (2).

Some authors have proposed to combine the detection and localization tasks. This is achieved by integrating both into the same framework using Structured Learning Support Vector Machines [1] or to apply a detector at a fine grid to obtain points of maximum classifier response [12].

However, this chapter introduces an approach based on [18], that treats detection and localization of pedestrians as completely separate problems that are solved consecutively. An arbitrary detector is used to find an initial estimate of a pedestrian's position and scale in the image. A classifier trained specifically for the localization task, referred to as *localizer* in the following, is then employed to refine the estimate and obtain a high match with the ground truth label.

There are several advantages to treating detection and localization independently. The detector can be chosen to fit the application scenario, while the localizer merely constitutes an additional post-processing step. Detection schemes that are known to work well for an application can be retained. Furthermore, superior performance in both detection and localization is to be expected when a separate dedicated classifier, specifically designed and trained for its purpose, is used for each task. Evaluation of the compound system is also simplified, as it is easier to isolate the effects of the two subsystems.

2 Learning Localization

Pedestrian appearance is highly dynamic in both shape and texture, prohibiting the use of simple image-based localization approaches. While methods such as hough transform or cross correlation work well for finding the position of simple rigid objects, they fail when presented with the high variability of human appearance.

The pedestrian localization task can be made tractable by casting it as a machine learning problem. Given a suitable model function f_p, the goal is to learn the parameter vector p^* from a set of labeled input images, so that f can be used for localization as defined by (2). That is, f_{p^*} applied to an image window $\mathbf{w}$ containing a pedestrian should exhibit a response monotonically increasing with the coverage between $\mathbf{w}$ and the pedestrian's label $\mathbf{w}_l$:

$$f_{p^*}(\mathbf{w}, \mathbf{w}_l) = \xi(a_0(\mathbf{w}, \mathbf{w}_l)), \tag{3}$$

with an arbitrary strictly increasing function $\xi : \mathbb{R} \to \mathbb{R}$. The window best matching the label can then by found by maximizing the value of f_{p^*} over the set of all windows.

2.1 Learning Localization by Regression

A straightforward way to find the desired parameter vector p^* is to choose $\xi(x) = x$ in (3) and apply a regression algorithm to the input data. This amounts to directly learning the coverage values and reduces the localization task to finding the window with coverage of 1.0. However, such an approach does not take the layout of the input space into account. By assuming a fixed aspect ratio, each window in the image can be represented in 3D XYS-space by an x and y position combined with a scale factor relative to an arbitrary base scale, i.e. $\mathbf{w} = (x, y, s)$. The coverage measure in (2) can then be regarded as a projection from XYS-space to one dimensional coverage space.

This non-injective reduction of dimension poses a severe problem for any regression function. Windows that are far apart in XYS-space and therefore differ greatly in their image content, have to be mapped to the same target value if they project to the same point in coverage space. It is also clear that small coverage values are much more common than large values when the input windows are scattered on a discrete grid. This skewed distribution of target values causes the regression algorithm to neglect regions of high coverage, yet these regions are of particular interest for precise localization. Regression appears unsuitable for the localization task, especially when considering that the actual output values of the model function f are of little interest. To satisfy (3), it is sufficient that

$$f(\mathbf{w}_i, \mathbf{w}_l) > f(\mathbf{w}_j, \mathbf{w}_l) \; \forall \, \mathbf{w}_i, \mathbf{w}_j : a_0(\mathbf{w}_i, \mathbf{w}_l) > a_0(\mathbf{w}_j, \mathbf{w}_l). \tag{4}$$

In other words, only the relative *ranking* of image windows induced by f is required to find the window with the best match to the ground truth label. This consideration directly leads to our approach of formulating the localization task as a *ranking* problem.

2.2 Learning Localization by Ranking

Ranking algorithms are usually employed in an information retrieval context, where a user issues search queries to a system. The system then has to resolve each query and present the most informative documents, sorted by their relevance, to the user [3]. For this, the system uses a learned ranking function $r(\mathbf{x_i}) : \mathbb{R}^n \to \mathbb{R}$ to assess the relevance of a document i by features $\mathbf{x_i}$ extracted from the document's contents. Each document is assigned a score by the ranking function and the absolute rank of a document is obtained by comparing this score to the score of all other documents returned by the query [2].

During training of the ranking function, the absolute ranking of documents is given only implicitly by pairwise combination of documents. By considering all training pairs for one query, an absolute ordering of documents related to this query is defined. Training pairs are always constructed from documents pertaining to the

same query and the ranking function is trained with pairs from all queries. Given a ground truth document pair $(\mathbf{x_i}, \mathbf{x_j})$, the ranking function is trained to assign higher rank to $\mathbf{x_i}$, that is $r(\mathbf{x_i}) > r(\mathbf{x_j})$. This ranking framework can be applied to the localization task in the following way: All image windows that overlap with a label window containing a pedestrian can be thought of as the documents returned for one query. The ranking function now needs to score these windows by their relevance for the current query (i.e., label). In the case of localization, this is simply the relation described by (4): a window with a high coverage value a_0 should receive a higher ranking score than a window with low coverage.

The same holds true for window pairs presented to the ranking function during training. Image windows are created for each label on a discrete 3D-grid in XYS-space with its center on the label. The position and scale of those windows is therefore relative to the label position at $[x_0\ y_0\ s_0] = [0\ 0\ 1]$. As mentioned above, there is no need to explicitly specify the absolute ranking of the windows. This ranking can be given implicitly by window pairs p_k, where the first element is taken to be of higher rank and thus of higher coverage in regard to the label window $\mathbf{w}_l$:

$$p_k = (\mathbf{w}_i, \mathbf{w}_j) : a_0(\mathbf{w}_i, \mathbf{w}_l) > a_0(\mathbf{w}_j, \mathbf{w}_l). \tag{5}$$

It is furthermore sufficient to create only pairs for windows that are direct neighbors on the grid in XYS-space. Through the transitive property of rank, an ordering of all windows on the grid is defined. Note that window pairs always belong to the same label, which causes the classifier to focus on *local* differences introduced by shifting or scaling of the input window. Overall, localization by ranking is far more tractable learning approach than localization by regression. Consider for instance the desired output value of the localization function for the label window. For the regression case, the desired target output value is exactly 1.0 for *all* labels, while in the ranking case it is sufficient that the label window merely has a higher score than other windows *in the vicinity* of the label.

2.3 Learning Stepwise Localization

Another approach to find the image window with the highest possible coverage to a given pedestrian label can be derived by differentiating (3) with respect to $\mathbf{w}$:

$$\frac{\partial}{\partial \mathbf{w}} f_{p^*}(\mathbf{w}, \mathbf{w}_l) \sim \frac{\partial}{\partial \mathbf{w}} a_0(\mathbf{w}, \mathbf{w}_l), \tag{6}$$

as ξ is strictly increasing. This result can be interpreted in two ways. First, the window that best matches the label can be found by gradient descent[1] in XYS-space on the learned model function f_{p^*}. This approach is discussed in combination with a ranking function in Section 3.3.

[1] Note that the term gradient descent is used here even in cases where the goal is to find the maximum of a function.

However, a far more robust approach is to skip the preliminary function f_p altogether and to simply learn the sign of the gradient of a_0 with respect to an image window $\mathbf{w}$:

$$\hat{f}_{p^*} = \operatorname{sgn}\left(\frac{\partial}{\partial \mathbf{w}} a_0(\mathbf{w}, \mathbf{w}_l)\right). \tag{7}$$

The function $\hat{f}_{p^*}$ can then be used to implement an efficient iterative stepwise algorithm to find the window that exhibits the highest possible coverage to the unknown label. This approach will be referred to as *stepwise localization* for the remainder of this work:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \hat{f}_{p^*}(\mathbf{w}), \tag{8}$$

where η is the step size in XYS-space. After evaluating $\hat{f}_{p^*}$ at the current window location, the window is moved in the direction returned by $\hat{f}_{p^*}(\mathbf{w})$. If no sub-pixel accuracy of the resulting window is required, η can be chosen in such a way that stepping is always done on a discrete pixel grid.

The main difference to ordinary gradient descent is that the stepping function $\hat{f}_{p^*}$ is learned directly from the input data without any knowledge of the function whose gradient direction it represents. No actual derivatives are employed in the calculation of the stepping function, which makes it robust towards noisy input, as derivatives tend to amplify noise.

By using the sign function in (7), the problem of determining the optimal parameter vector p^* can be cast as a standard binary classification task with the special case of $\hat{f}_{p^*} = 0$. The training procedure of such a classifier is detailed in Section 3.4.

3 Neural Networks with Local Receptive Fields for Localization

To apply a classifier or a ranking function to an image window, suitable features first have to be extracted from the window's raw image data. This is identical to the pedestrian detection case, where a multitude of feature extraction approaches have been published in recent years. The most successful features include histograms of oriented gradients (HOG) [4], local binary patterns (LBP) [20], local receptive fields (LRF) [7] and combinations thereof [22]. Since to our knowledge there are no publications on image features particularly suitable for localization, pedestrian detection features were considered as potential candidates.

Both the HOG and LBP features employed in pedestrian detection build feature value histograms over overlapping spatial regions in the image. The histograms of all regions are then concatenated and normalized, forming the final feature vector. This procedure of sampling histograms on a coarse grid in the image provides a robust representation of the object contained in the window and is one of the reasons why these features perform so well for the detection of pedestrians. For the localization task however, the invariance to slight changes in position and scale afforded by using histograms is actually detrimental.

Instead in this chapter, a neural network approach using local receptive fields (NN/LRF) is explored, as this combined classifier/feature architecture has several properties that are advantageous for localization. The structure of the feature extraction stage of this neural network is adapted to the classification task during training. Thus, no prior knowledge about the input data is required, the features are created data-dependent instead of by manual design. Furthermore, the features operate directly on the raw input pixels. Unlike HOG or LBP, this feature extraction approach preserves detail in the image structure. Finally, the network is resistant to overfitting [7] and can be trained in an online fashion. This is an important property for the localization task, as hundreds of samples can be generated from one label by shifting and scaling, causing the training set to be accordingly large.

3.1 Neural Networks with Local Receptive Fields

In the first layer of this neural network architecture, several small feature maps referred to as *local receptive fields* are shifted in a fine grid over the input image window. A non-linearity is applied to the output of each LRF and the vector of all transformed LRF outputs at each grid position constitutes the input to the second layer of the neural network. The second layer consists of one or more fully connected output neurons, which also apply a transfer function to their input. Whereas in a standard multi-layer perceptron (MLP) the neurons of all layers are fully connected, the NN/LRF employs weight sharing, making the network less susceptible to overfitting. While one hidden unit of an MLP is connected to each input by a *separate* weight, one LRF is applied to all image positions using the *same weights*, reducing the number of adaptable parameters during training [7].

The NN/LRF architecture is a special case of the generic convolutional neural network (CNN) framework [14], which in turn is based on the self-organizing Neocognitron network model [9]. Both these network architectures employ alternating feature extraction layers, where the convolution of several feature maps with the input image is computed, and pooling layers, where feature responses are integrated over a local neighbourhood. In the NN/LRF network, the full convolution is usually replaced by discrete sampling positions in the input image at which feature responses are extracted, requiring less memory accesses and multiply-accumulate operations. Also, the output layer is directly connected to the feature layer, thus no feature pooling is employed [15, 21]. Feature pooling is primarily used in a detection context to achieve invariance to translations of the input, a quality that is detrimental to the localization task, as explained in the previous section.

Let N_p be the number of image positions obtained by shifting LRFs of size $M = S_x \times S_y$ pixels at a step size of D_x and D_y pixels over input windows of size $W \times H$. Furthermore, let $x_{ik} \in \mathbb{R}$ contain the value of the k-th pixel of the image patch at the i-th of N_p positions and let $l_{jk} \in \mathbb{R}$ contain the k-th weight of the j-th local receptive field. Then a NN/LRF with one output neuron applied to the pixel patches $\mathbf{x}$ of an input window can be written as:

$$\text{net}(\mathbf{x}) = h\left(\sum_{i=1}^{N_\text{p}}\sum_{j=1}^{N_\text{lrf}} w_{ij}\, g\left(\sum_{k=1}^{M} x_{ik} l_{jk} + \theta_j\right) + \varphi\right), \tag{9}$$

with bias weights $\theta_j, \varphi \in \mathbb{R}$, output weights $w_{ij} \in \mathbb{R}$ and transfer functions $g : \mathbb{R} \to \mathbb{R}$ and $h : \mathbb{R} \to \mathbb{R}$ applied to layer one and two respectively. The network architecture is illustrated in Fig. 1.

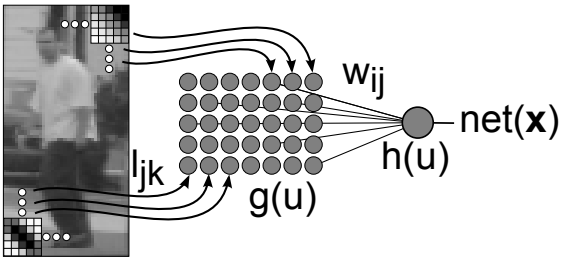


Fig. 1 NN/LRF architecture. The input image is scanned with receptive fields making up the feature layer. The feature layer is fully connected to one or more output neurons. A non-linearity is applied after each layer

3.2 Training the Ranking Localization Network

Training of neural networks for classification is usually done by stochastic gradient descent on a cost function. The cost function is based on the error between the network's output and the target value over all samples. For ranking neural nets, the authors of [2] propose a cross-entropy cost function based on the difference of sample pair network outputs:

$$o_{ij} \equiv f(\mathbf{x}_i) - f(\mathbf{x}_j), \tag{10}$$

$$C(o_{ij}) = -\bar{P}_{ij} o_{ij} + \log(1 + \mathrm{e}^{o_{ij}}), \tag{11}$$

where $\bar{P}_{ij}$ is the probability that sample i is to be ranked higher than sample j. For applications where the relative rank of all training samples is deterministically known, $\bar{P}_{ij}$ can be set to 1.0 when pairs are formed accordingly. Using this cost function, a variation of the popular backpropagation algorithm [16] for the training of ranking two-layer MLPs is derived in [2].

The above cost function can be employed in the same manner to train a *Ranking NN/LRF* by gradient descent [10]. For this, the *Elliot Sigmoid* [6] is used as the non-linearity in the first layer:

$$g(u) = \frac{u}{1 + |u|}. \tag{12}$$

This non-linearity is similar to the standard tanh sigmoid, but much more efficient to compute, since no exponentials have to be calculated, thus speeding up both training and application of the neural network. The output neuron transfer function is chosen as $h(u) = u$, as no squashing of the output activation is required. Since h is the identity, φ is omitted from the network in this section, as rank is not affected by linear transformations of the output. In the following, upper indices 1 and 2 are used to indicate if the current expression pertains to the first or the second sample of a training pair. Taking derivatives of the cost function with respect to the network weights gives:

$$C'(o_{ij}) = \frac{\mathrm{e}^{o_{ij}}}{1+\mathrm{e}^{o_{ij}}} - \bar{P}_{ij}, \qquad g_{ij}^{v} \equiv g\left(\sum_{k=1}^{M} x_{ik}^{v} l_{jk} + \theta_j\right)$$

$$\frac{\partial C}{\partial w_{ij}} = \left(g_{ij}^{2} - g_{ij}^{1}\right) \cdot C', \qquad g_{ij}^{\prime v}, \equiv g'\left(\sum_{k=1}^{M} x_{ik}^{v} l_{jk} + \theta_j\right)$$

$$\frac{\partial C}{\partial \theta_j} = \sum_{i=1}^{N_\mathrm{p}} w_{ij}\left(g_{ij}^{\prime 2} - g_{ij}^{\prime 1}\right) \cdot C', \qquad \frac{\partial C}{\partial l_{jk}} = \left(\sum_{i=1}^{N_\mathrm{p}} w_{ij} g_{ij}^{\prime 2} x_{ik}^{2} - \sum_{i=1}^{N_\mathrm{p}} w_{ij} g_{ij}^{\prime 1} x_{ik}^{1}\right) \cdot C'.$$

3.3 Maximum Rank Search

After the Ranking NN/LRF is trained, it can be used to localize pedestrians in images by finding windows of maximum network output in XYS-space. A straightforward way to find the maximum is to simply apply the network to all windows sampled from the neighborhood of an initial estimate obtained by a detector. However, the fact that the NN/LRF network is a differentiable function of the input pixels allows for a more efficient approach. By taking derivatives of the network function with respect to the coordinates of the image window in XYS-space, the maximum can be found via gradient descent, for instance finding the maximum in y-direction would require:

$$\frac{\partial}{\partial y}\mathrm{net}(\mathbf{x}) = \sum_{i=1}^{N_\mathrm{p}} \sum_{j=1}^{N_\mathrm{lrf}} w_{ij} g'\left(\sum_{k=1}^{M} x_{ik} l_{jk} + \theta_j\right) g\left(\sum_{k=1}^{M} l_{jk} \frac{\partial}{\partial y} x_{ik}\right). \tag{13}$$

Unfortunately, as mentioned in Section 2.3, this approach is very susceptible to noise in the image data and the algorithm can easily get stuck in local maxima. To alleviate these problems, a method to robustly estimate the gradient direction without resorting to derivatives is presented in the next section.

3.4 Training the Stepwise Localization Network

The stepwise localization network introduced in Section 2.3 seeks to find the window position with the highest possible coverage to the unknown label by moving an initial window stepwise in XYS-space. The training objective of the network

is to learn the stepping direction in such a way, that the coverage between current window and label window increases with each step. This can be cast as a classification problem with three-dimensional target values $\mathbf{t} = (t_1, t_2, t_3),\ t_k \in \{-1, 0, +1\}$, where each dimension represents the true stepping direction along the x-, y- and s-axis. Although it is not strictly necessary to include zero target values and their corresponding training samples, as the stepping direction is always either positive or negative, we found that including them did improve performance slightly.

Using a cross-entropy cost function as for the ranking localization network, the sample-wise cost to be minimized during training is:

$$C(\mathbf{x}, \tilde{\mathbf{t}}) = \sum_{v=1}^{3} \tilde{t}_v \log \mathrm{net}_v(\mathbf{x}) + (1 - \tilde{t}_v) \log(1 - \mathrm{net}_v(\mathbf{x})), \tag{14}$$

where $\tilde{\mathbf{t}}$ contains the target values for sample $\mathbf{x}$ converted to probabilities, that is $\tilde{t}_v = (t_v + 1)/2$. The network function net_v is an extension of (9) for multiple output neurons. All outputs share the same feature extraction layer, but employ different output weights:

$$\mathrm{net}_v(\mathbf{x}) = h\left(\sum_{i=1}^{N_\mathrm{p}} \sum_{j=1}^{N_\mathrm{lrf}} w_{ijv} g\left(\sum_{k=1}^{M} x_{ik} l_{jk} + \theta_j \right) + \varphi_v \right). \tag{15}$$

The Elliot Sigmoid (12) is again adopted as the transfer function for the input layer, while the output activation is mapped to probabilities using a logistic function:

$$h(u) = \frac{1}{1 + \mathrm{e}^{-u}}. \tag{16}$$

Taking derivatives of the cost function with respect to the network weights yields:

$$\delta_v \equiv \mathrm{net}_v(\mathbf{x}) - \tilde{t}_v, \qquad \frac{\partial C}{\partial \varphi_v} = \delta_v,$$

$$g_{ij} \equiv g\left(\sum_{k=1}^{M} x_{ik} l_{jk} + \theta_j \right), \qquad g'_{ij} \equiv g'\left(\sum_{k=1}^{M} x_{ik} l_{jk} + \theta_j \right),$$

$$\frac{\partial C}{\partial \theta_j} = \sum_{v=1}^{3} \delta_v \left(\sum_{i=1}^{N_\mathrm{p}} g'_{ij} w_{ijv} \right), \quad \frac{\partial C}{\partial l_{jk}} = \sum_{v=1}^{3} \delta_v \left(\sum_{i=1}^{N_\mathrm{p}} g'_{ij} x_{ik} w_{ijv} \right), \quad \frac{\partial C}{\partial w_{ijv}} = \delta_v g_{ij}$$

The weights of the network can be efficiently trained on large datasets using standard backpropagation [16].

4 Progressive Localization

The localization approaches introduced in the previous section aim at finding the maximum coverage window in XYS-space either by exhaustive search, or by

iteratively stepping towards the maximum. Given an input window, an NN/LRF classifier is used to estimate the goodness of the fit to the unknown label (in the case of ranking localization), or to determine the most likely stepping direction (for stepwise localization). This classifier has be very complex, both in terms of trainable weights and the amount of training samples required, as the input it has to handle is highly variable. Input samples may contain pedestrians that are heavily off-center and out of scale, so robustness in regard to strong deviations in position and scale is required. At the same time the classifier must be sensitive to small changes in the input images, as to facilitate *precise* localization.

To deal with these extremes, the concept of *progressive localization* is introduced in this section. Progressive localization means splitting up the localization task into several consecutive subtasks and training one classifier specifically for each of those subtasks. Assuming that the position of an initial window in the vicinity of the pedestrian label is known (e.g. by scanning a detector over the image), the localization task can be split up into the following:

Coarse Centering. Starting from an initial window, find a window whose center is close to the center of the label, while keeping the size of the window fixed. The classifier used for this task must be able to handle large displacements from the true center. The resulting window does not have to perfectly match the label center, but should be sufficiently close to it.

Fine Centering. Refine the window position obtained from coarse centering by minimizing the distance between the window's center and the true center. The windows presented to the fine centering classifier are all close to the label center due to the previous coarse centering step. The classifier can therefore adapt to small shifts in x- and y-direction to find the optimal position for the center of the window, which is then fed into the next localization stage.

Scaling. Scale the input window while keeping its center position fixed to match the size of the unknown label. As all windows have already been centered on the label in the previous steps, the classifier for this task can focus solely on changes in the image structure induced by scaling.

This concept applies to stepwise localization in a straightforward way. Each initial window is processed successively by every stage in the progressive localization chain, resulting in exactly one window after each stage. Figure 2, (*a*), illustrates the progressive localization chain for stepwise localization. Starting from an initial window, the coverage between the current window and the pedestrian is successively optimized in each stage. The algorithm proceeds to the next stage if the current step yields a position in XYS-space that has previously been visited. All visited windows are sorted by the L2-norm of their classifier activations and the window with the lowest activation, i.e. the window that is probably closest to the label position in XYS-space, is passed to the next stage.

For the ranking based localization approach, the vicinity of the initial window is first finely sampled in XY-space, keeping scale fixed. This set of windows is then passed to the coarse localization stage. The coarse ranking classifier estimates a ranking value for each of those windows and only the κ highest ranking windows

go on to the next stage for fine centering. In this stage, all windows except the highest ranking one are discarded. For the final stage, the centered input window is resampled in different scales and the scaling classifier is applied to every resulting window. Only the highest ranking window is returned as the result of the localization chain.

Progressive localization with ranking can therefore be viewed as consecutive resampling and filtering of several input windows by three different filters as illustrated in Fig. 2, (*b*). Due to its modular nature, progressive localization has several advantages over using a single common classifier for all localization tasks. First and foremost, the local receptive fields learned during training can be specifically adapted to the task at hand. It is reasonable to assume that image features suitable for centering might not necessarily be optimal for determining the proper scale. Experiments in Section 5 will show that this is actually the case.

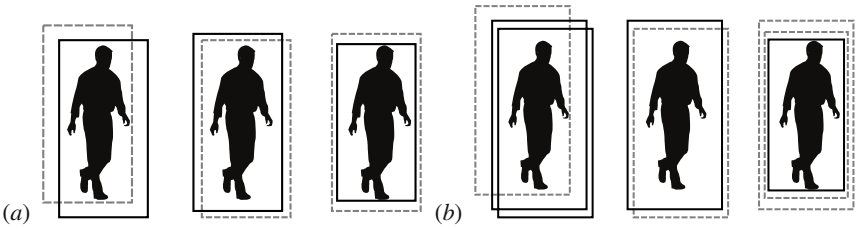


Fig. 2 Exemplary progressive localization chains. The resulting windows from coarse centering, fine centering and scaling are shown as solid boxes from left to right (*a*). Stepwise localization chain. Dashed boxes show the windows that have been discarded in the current stage (*b*). Ranking localization chain. Dashed boxes show the window from the previous step, including the initial window used as the starting point for coarse centering

Furthermore, different architectural parameters, such as the number of LRFs or the granularity of the sampling grid can be optimized for each task. Lastly, progressive localization makes training and evaluation of localization classifiers much more tractable, as each classifier can be handled independently. If the performance of the progressive localization chain is not satisfactory, it is easy to find the classifier which is at fault.

5 Experiments and Results

In this section, the results of the proposed approaches in regard to localization accuracy are presented and evaluation methodology is discussed. All results pertain to the progressive localization approach both with ranking and stepwise localization. We have found this approach to be superior to non-progressive localization in every experiment, so we do not give separate evaluation details for those cases and refer to [10].

5.1 Dataset

For training and evaluation of the Ranking NN/LRF, images of the publicly available *Daimler Pedestrian Detection Benchmark* dataset [7] were used. The training set contains 15,660 pedestrian cut-out images obtained from 3,915 ground truth labels by mirroring and randomly shifting the label box by a few pixels. The test dataset consists of an independent image sequence comprising 21,790 images with 14,132 fully visible pedestrian labels in 259 trajectories. All labels in the dataset have manually been centered on the pedestrians, see [10] for details of this relabeling procedure.

5.2 Classifier Trainings

5.2.1 Pedestrian Detector

To find pedestrians in the input images and obtain an initial estimate of their position, a 13-layer cascaded classifier is employed. The first 12 layers use boosted Haar-like features to discard most non-pedestrian input windows with little computational effort [19]. The final layer utilizes HOG features [4] and a Linear Support Vector Machine (SVM) to classify the remaining windows. Each Haar-feature stage of the cascade is trained on 15,660 positive and negative input windows. Negative windows are extracted at random locations from the full training images. Four positive windows per label are generated by mirroring and randomly shifting the relabeled bounding boxes by 5% of the label width in x- and/or y-direction. For the training of the last layer HOG-SVM, all windows that have passed the previous stages are used. The operating point of the trained classicdloc fier is chosen in such a way that a detection rate of 90% on the test dataset (using the evaluation metrics in [7]) is achieved. Note that only pedestrian labels with a minimum height of 72 pixels are considered for this evaluation.

5.2.2 Ranking Pedestrian Localizer

The first step in training the Ranking NN/LRF for localization is extracting a set of training windows from each label and building pairs from these windows. To this end, windows are sampled on a grid in XYS-space as described in Section 2.2. The sampling volume and granularity relative to the label width, as well as the resulting number of pairs per label is given in Table 1. Note that X/Y sampling is done symmetrical around the center of the label, so sampling from 0% to 8% of the label width also implies sampling from 0% to -8%.

For the coarse centering classifier, large displacements are learned while samples close to the label center are ignored. The fine and scaling classifiers are only presented samples close to the center. All classifiers are trained using identical scale sampling, as scaling is only done in the last stage of the progressive localization chain. Depending on which stage is trained, different pairings of windows according to the function of the stage are employed. The coarse and fine centering classifiers

Table 1 Sampling granularity for localization classifier training and evaluation. All percentages are relative to label box width and sampling distance is symmetrical to zero

Localizer Stage	X/Y Sampling Distance @Granularity	Scale Sampling Factor @Granularity	Windows/Pairs per Label
Coarse Centering	6% - 16% @ 2%	85% - 115% @ 3%	1859 / 3432
Fine Centering	0% - 8% @ 2%	85% - 115% @ 3%	891 / 1584
Scaling	0% - 8% @ 2%	85% - 115% @ 3%	891 / 8100

are presented sample pairs that are directly adjacent in XYS-space and share the same scale. In contrast, sample pairs presented to the scaling classifier may only differ in scale.

The width and height of the pedestrian labels are increased by a factor of one third as training samples are extracted. This additional border is required to enable the classifiers to handle large displacements in location and scale. Before the training windows are presented to the neural network, they are resized to 32x64 pixels and normalized to zero mean and unit variance. LRFs of size 8x8 pixels are shifted on a 4-pixel grid over these normalized windows, resulting in 105 patches (i.e., LRF positions) per window. The feature extraction layer of the network contains 64 LRFs, thus the input dimension of the output neuron is 6720. LRF and output weights are initialized with a zero mean normal distribution. The standard deviation of this distribution is chosen in such a way, that the sum of inputs of each sigmoid is in the sigmoid's linear range [13]. All bias weights are initialized to zero.

The ranking networks are each trained with pairs created from 8 million samples extracted from 3,915 labels for 50 epochs. A learning rate between 10^{-3} and 10^{-4} was found to yield the best results.

5.2.3 Stepwise Pedestrian Localizer

The training parameters for the stepwise pedestrian localizer chain, including the sampling grid and LRF configuration, are identical to the ranking pedestrian localizer. However, the stepwise classifier training uses target values instead of sample pairings to specify the training objective. For the coarse and fine centering stages, the target for each window is a two-dimensional vector whose entries constitute the x- and y-stepping directions, as explained in Section 3.4. The network therefore has two output neurons. The target value for each of the two dimensions is equal to the true stepping direction along that dimension, irrespective of the window scale. For the scaling stage, a scalar target value and a single output neuron is employed. The target value is equal to the true stepping direction along the scale axis, irrespective of the window's x/y position. Just as the ranking networks, the stepwise networks are trained with 8 million samples for 50 epochs.

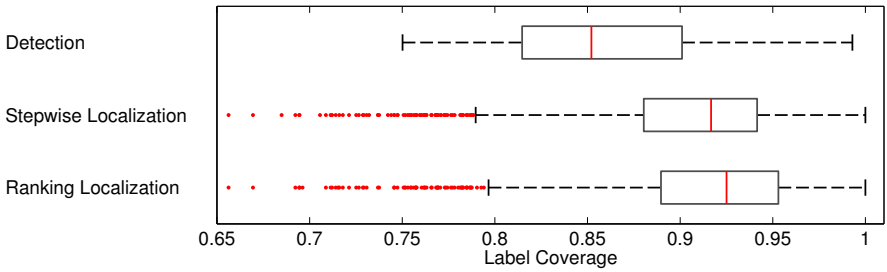


Fig. 3 Results for progressive stepwise and ranking localization chains. The first boxplot shows the coverage of the initial windows generated by the pedestrian detector

5.3 Localization Results

To test the localization performance of both stepping- and ranking-based approaches, the pedestrian detector is first scanned over each image of the evaluation sequence using a fine position- and scale-grid. A confidence-based non-maximum-suppression [7] is used to prune multiple detections on a single pedestrian.

Each remaining detection window in the image is used as a starting location $\mathbf{w}_0$ for the localizer chains. From this initial window, the window best matching the label is estimated as described in Section 4. Both approaches are compared using identical initial detections. The step sizes for the stepping localizers are set to be identical to the granularity of the sampling grid during training (Table 1). All percentages are relative to the width of the initial window. The x/y and scale sampling parameters used during progressive ranking localization are also identical to the training values for each classifier. For each label in the evaluation dataset, the coverage value of the label window and the window estimated by the stepping and ranking localizer chains is calculated. The coverage results are shown in Fig. 3 using boxplots. Red vertical lines mark the median of each sample, while the edges of the boxes are the 25th and 75th percentile. Outliers are drawn as red markers and the whiskers of each column stretch to the most extreme data points not considered outliers. Data points are considered outliers if they exceed a distance of 1.5 interquartile ranges to the first or the third quartile respectively.

The NN/LRF ranking localizer exhibits the best localization performance with a median coverage value of 0.925 while the stepwise localizer achieves a median coverage of 0.917. Both NN/LRF approaches outperform the detector, which achieves a median coverage value of 0.852. This result is not unexpected as the stepwise localization process is far more susceptible to singular errors. While one misranked sample likely has no effect on the result of the ranking localization, a stepping error can cause the stepping localization algorithm to terminate with a solution that is far from the true label value.

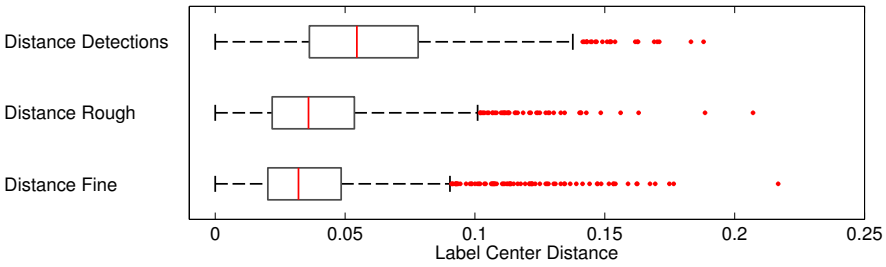


Fig. 4 Distance of window center to label center after the centering stages of the stepwise localization chain from Fig. 3. The distance is given as a percentage of the label width. The first boxplot shows the distance of the initial windows generated by the pedestrian detector

On the other hand, stepping localization requires far less classifier evaluations to estimate the label position. The ranking chain from Fig. 3 needs to compute over 300 classifier evaluations per detection, whereas the stepping classifier requires an average of 12 evaluations. As every evaluation of the classifier involves the computation of each LRF response at each position of the input patch, ranking localization is prohibitively slow for real time applications.

Figure 4 gives detailed results of the two centering stages of the stepwise localization chain from Fig. 3. The first boxplot shows the distance of the initial windows generated by the pedestrian detector. Distances between the center of the resulting window and the label center are plotted as a percentage of the label width. Each centering stage decreases both the median distance to the label, as well as the standard deviation. Due to classification errors, there is a slight increase in outliers in each stage. A similar analysis of the scaling stage of a stepwise localizer is shown in Fig. 5. Here, the scale is given relative to the true label scale. The stage significantly improves the scale of the detection window but also introduces some outliers due to classification errors. These outliers are investigated in the next section. Figure 6 shows the first 36 local receptive fields of each of the stepwise localization classifiers. The LRFs have clearly adapted to the different tasks in the localization chain. Coarse localization seems to require extensive features that respond to oriented contrast. Fine localization also appears to rely on oriented contrasts, but the learned filters are narrower and more sparse than their coarse counterparts. This is not surprising, as fine-scale features are necessary to detect small shifts in position. The scaling LRFs develop high-frequency circular features, likely because image structures are stretched radially by scaling. These findings confirm the main motivation for progressive localization, namely that a monolithic classifier forced to employ the same feature extraction layer for all three localization tasks can not attain optimal performance.

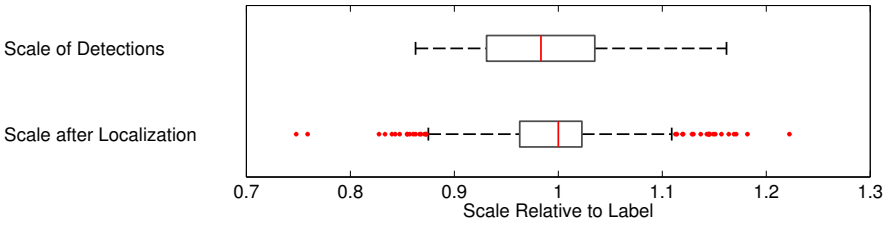


Fig. 5 Scale of the window after the scaling stage of the stepwise localization chain from Fig. 3. The scale is given relative to the true label scale. The first boxplot shows the scale of the initial windows generated by the pedestrian detector

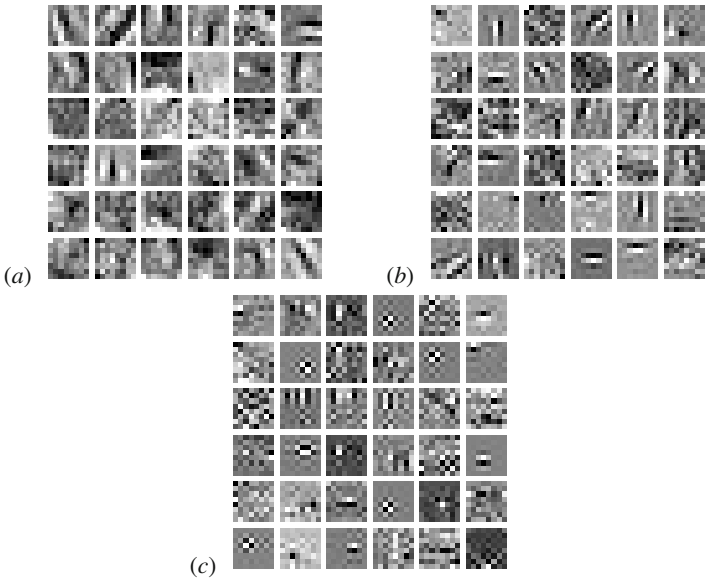


Fig. 6 The first 36 local receptive fields of each of the classifiers from a stepwise localization chain: (*a*) Coarse Centering. (*b*) Fine Centering. (*c*) Scaling

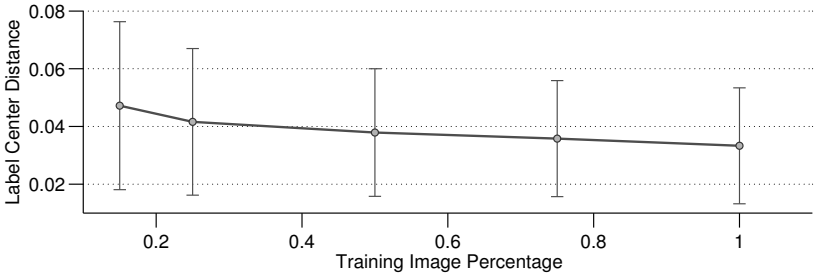


Fig. 7 Performance of fine localization as a function of training dataset size. Y-axis shows median value $\pm\sigma$ of distance to label center as % of label width. Relative size of the dataset is plotted on the x-axis.

5.4 Error Analysis

Although the coverage values of most detection windows are significantly improved by both localization approaches, localization seems to fail for some detections. Several pedestrian images for which the coverage value decreased severely after localization (i.e. outliers in Fig. 3) are shown in Fig. 8. The label windows are marked in green and the windows resulting from progressive localization are displayed in red.

A frequent source of localization errors are pedestrians that are close to each other and of similar scale, as show in Figure 8, (*a*)- (*c*). During the centering stage of localization, the classifier sometimes erroneously moves toward the "background" pedestrian, even if the initial detection is closer to the designated pedestrian. The reason for those errors is likely found in the training dataset which almost exclusively contains fully visible, non-overlapping pedestrians. The above mentioned situation is therefore unknown to the classifier, causing it to focus on the wrong pedestrian. Extending the training dataset to include more images of close standing pedestrians is likely improve performance in those cases. Adding slightly occluded pedestrians to the dataset would also reduce the number of localization errors induced by occlusion in the evaluation images such as in Fig. 8, (*e*).

The classifiers are also prone to errors when applied to images where parts of the pedestrian exhibit little contrast to the image background (Fig. 8, (*f*)), or certain salient image structures surround or cover the pedestrian contour (Fig. 8, (*i*) - (*j*)) A more sophisticated normalization scheme preceding training and application, as well as an increase in training images might alleviate those problems.

The NN/LRF classifiers are very dependant on a sufficient number of training images and at the same time scale very well with large datasets. Figure 7 shows the relationship between the performance of a fine centering stepping localizer on the evaluation dataset and the number of images used to train the corresponding classifier. Classifier performance is in this case given as the median distance (relative to the label width) of the localization result window to the pedestrian label. Whiskers show one standard deviation from the median. The size of the training dataset is plotted on the x-axis as a percentage of the total number of available images (3,915). All classifiers were trained using otherwise identical parameters and evaluated on windows obtained from the coarse stepwise classifier from Fig. 4. The number of local receptive fields was chosen low enough to avoid overfitting by inspecting the validation error progression over 50 epochs of training.

It is evident that even when using the full dataset, the NN/LRF architecture is not saturated in terms of training samples. Both median detection performance and standard deviation decrease linearly for large datasets percentages, motivating the collection and annotation of additional pedestrian data.

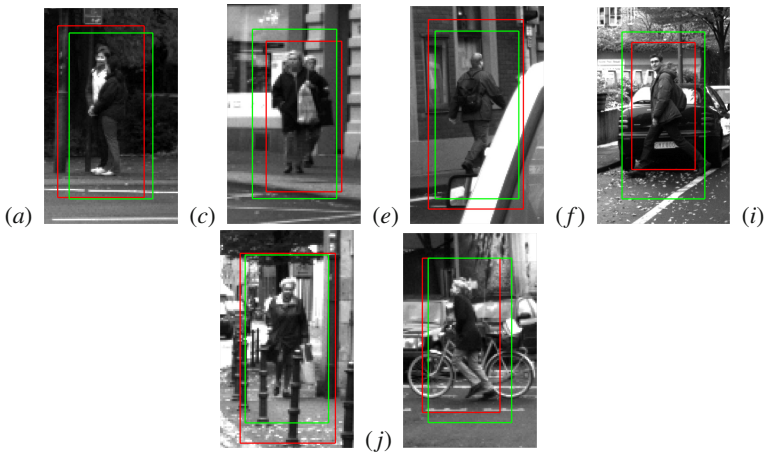


Fig. 8 Several outliers from the results of stepwise localization. The pedestrian label including additional border pixels for training is shown in green and the window resulting from progressive localization in red.

6 Conclusions and Future Research

This chapter introduced two different approaches for the precise localization of pedestrians in images using neural networks with local receptive fields. It was shown why ordinary pedestrian detection methods and simple regression based on label coverage are unsuitable for the localization task. Instead, algorithms based on ranking and stepwise optimization to obtain a suitable localization function were proposed. The resulting neural net based localizers can be trained via gradient descent, learning salient image features from the input data. By dividing the localization task into multiple subtasks, classifiers specifically adapted to centering and scaling can be learned. Both approaches were evaluated using a relabeled version of the Daimler Pedestrian Benchmark dataset [7]. The proposed localizer chains improve significantly on the results of the pedestrian detector. Ranking localization yields less outliers than the stepwise approach, while stepwise localization requires only few classifier evaluations and is therefore suited for real-time applications. In the future we plan to train networks with more than two layers to increase localization performance while reducing the number of required LRFs. Furthermore, we want to investigate how input patches can be normalized to allow the handling of scenes with poor contrast.

References

1. Blaschko, M.B., Lampert, C.H.: Learning to Localize Objects with Structured Output Regression. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 2–15. Springer, Heidelberg (2008)

2. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: Proc. Int. Conf. Machine Learning, pp. 89–96 (2005), doi:10.1145/1102351.1102363
3. Cohen, W.W., Schapire, R.E., Singer, Y.: Learning to Order Things. Journal of Artificial Intelligence Research 10(1), 243–270 (1999)
4. Dalal, N., Triggs, W.: Histograms of Oriented Gradients for Human Detection. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 886–893 (2004), http://eprints.pascal-network.org/archive/00000802/
5. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 304–311 (2009), doi:10.1109/CVPR.2009.5206631
6. Elliott, D.L.: A Better Activation Function for Artificial Neural Networks. Institute for Systems Research. Univ. of Maryland (1993)
7. Enzweiler, M., Gavrila, D.M.: Monocular Pedestrian Detection: Survey and Experiments. IEEE Trans. Pattern Anal. 12, 2179–2195 (2009)
8. Everingham, M., Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision 88(2), 303–338 (2009)
9. Fukushima, K.: Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition. Neural Networks 1(2) (1988)
10. Gressmann, M., Loehlein, O., Palm, G.: Pedestrian Localization. In: Proceedings of 2011 IEEE 9th International Symposium on Intelligent Systems and Informatics, pp. 371–376 (2011), doi:10.1109/SISY.2011.6034355
11. Gressmann, M., Palm, G., Loehlein, O.: Surround view pedestrian detection using heterogeneous classifier cascades. In: Proceedings 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1317–1324 (2011), doi:10.1109/ITSC.2011.6082895
12. Lampert, C., Blaschko, M., Hofmann, T.: Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1–8 (2008)
13. LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.-R.: Efficient BackProp. In: Orr, G.B., Müller, K.-R. (eds.) NIPS-WS 1996. LNCS, vol. 1524, pp. 9–50. Springer, Heidelberg (1998)
14. Lecun, Y., Kavukcuoglu, K.: Convolutional Networks and Applications in Vision. In: Proc. International Symposium on Circuits and Systems, pp. 253–256 (2010), doi:10.1109/ISCAS.2010.5537907
15. Munder, S., Gavrila, D.M.: An experimental study on pedestrian classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(11), 1863–1868 (2006)
16. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature 323(6088), 533–536 (1986)
17. Shashua, A., Gdalyahu, Y., Hayun, G.: Pedestrian Detection for Driving Assistance Systems: Single-frame Classification and System Level Performance. In: Proc. IEEE Conf. Intelligent Vehicle Symp., pp. 1–6 (2004), doi:10.1109/IVS.2004.1336346
18. Vaillant, R., Monrocq, C., Le Cun, Y.: An original approach for the localization of objects in images. In: Proc. Int. Conf. Artificial Neural Networks, pp. 26–30 (1993)

19. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. I-511–I-518 (2001)
20. Wang, X., Han, T.X.: An HOG-LBP Human Detector with Partial Occlusion Handling. In: Proc. IEEE Int. Conf. Computer Vision, pp. 32–39 (2009), doi:10.1109/ICCV.2009.5459207
21. Wohler, C., Anlauf, J.K.: An adaptable time-delay neural-network algorithm for image sequence analysis. IEEE Transactions on Neural Networks 10(6), 1531–1536 (1999)
22. Wojek, C., Schiele, B.: A performance evaluation of single and multi-feature people detection. In: Proc. DAGM Symp. Pattern Recognition, pp. 82–91 (2008), doi:10.1007/978-3-540-69321-5_9

Author Index