Fuzziness Measure in the Pliant System: The Vagueness Measure

J. Dombi
6720 Szeged, Árpád tér 2
Hungary
e-mail: dombi@inf.u-szeged.hu

Abstract: The basic idea of fuzzy sets is the introduction of the membership function, which replaces the classical characteristic function. It is an interesting question to learn how close the membership function is to the characteristic function, when we use a certain class of membership functions. This measure is called the fuzziness measure.

Below we shall present an operator-dependent fuzziness measure called the vagueness measure. We will show that this measure satisfies the usual classical assumptions for the fuzziness measure. In addition, we will show that there is a connection between this measure and the entropy function.

1. Introduction

In our view, one of the most important concepts, because on this basis we can prove "convergence theorems" in the sense that "if there is less fuzziness in the input variables, then there will be less fuzziness in the result".

In the fuzzy literature we do not find such theorems, because membership functions, operators and fuzziness measures are unrelated so it seems hopeless to prove such convergence theorems.

In the Pliant concept we have the distending function instead of the membership function based on the Pliant operator, and now we will define a vagueness measure by using the generator function of Pliant operator. On the basis of this consistent concept we can derive a convergence theorem.

First, we will take a closer look at the fuzziness measure: Let $\mu(x)$ be the membership function and $d(\mu)$ be the fuzziness measure.

Fuzziness measure

We shall give a list of desirable properties for a measure of fuzziness, i.e. the kind of properties we expect it to have. Various combinations of these can be found in the literature. It is remarkable that different authors demand different properties for the d measure.

(P1) *Sharpness*: $d(\mu) = 0$ iff $\mu(x) \epsilon \{0,1\}$ for all $x \epsilon X$ (i.e. μ is "sharp").

(P2) *Maximality*: $d(\mu)$ is maximum iff $\mu(X) = \{\frac{1}{2}\}$.

(P3) *Resolution*: $d(\mu^*) \leq d(\mu)$ if μ^* is any sharpened version of μ, that is:
$\mu^*(x) \leq \mu(x)$ if $\mu(x) \leq \frac{1}{2}$ and $\mu^*(x) \geq \mu(x)$ if $\mu(x) \geq \frac{1}{2}$.

(P4) *Symmetry* (about $\frac{1}{2}$): $d(\mu) = d(\eta(\mu))$,
where $\eta(\mu)$ is the negation function.

(P5) *Valuation*: the function d is a valuation on $[0,1]^X$, i.e.:

$$d(\mu_1 \cup \mu_2) + d(\mu_1 \cap \mu_2) = d(\mu_1) + d(\mu_2),$$

where $\cap$ is the min operator and $\cup$ is the max operator.

(P6) *Generalized additivity*: there exist mappings $s,t : [0,1] \to [0,\infty)$ such that:

$$d(\mu_1 \mu_2) = d(\mu_1) t(P(\mu_2)) + d(\mu_2) s(P(\mu_1)).$$

where

$$P(\mu) = \sum_{i=1}^{n} \mu(x_i) \qquad \text{or} \qquad P(\mu) = \int \mu(x) dx$$

for all $\mu_1 \epsilon [0,1]^X$ and $\mu_2 \epsilon [0,1]^Y$, where X and Y finite sets.

The properties (P1)-(P4) are natural requirements for a measure of fuzziness. All the measures introduced so far satisfy these properties. But the meaning of (P5) and (P6) are not completely clear and therefore including them is debatable.

A survey of the existing fuzziness measures

In this section we will give a brief summary of the various fuzziness measures investigated so far and describe their properties.

1. *DeLuca* and *Termini* (1972) [2]
   Measure:

$$d(\mu) = -K \sum_{i=1}^{N} \{\mu(x_i) log(\mu(x_i)) + (1 - \mu(x_i) log(1 - \mu(x_i)))\}$$

2. *Kaufmann* (1975)

   (a) Measure (using the generalized relative Hamming distance):

$$d(\mu) = \frac{2}{N}\sum_{i=1}^{N} |\mu(x_i) - \mu_{\frac{1}{2}}(x_i)|,$$

   where the $\frac{1}{2}$-cut of μ is defined by:

$$\mu_{\frac{1}{2}}(x) = 0, \qquad \text{if} \qquad \mu(x) < \frac{1}{2},$$

$$\mu_{\frac{1}{2}}(x) = 1, \qquad \text{if} \qquad \mu(x) \geq \frac{1}{2}$$

   (b) Measure (using the generalized relative Euclidean distance):

$$d(\mu) = \frac{2}{N^{\frac{1}{2}}}\left\{\sum_{i=1}^{N}(\mu(x_i) - \mu_{\frac{1}{2}}(x_i))^2\right\}^{\frac{1}{2}}$$

3. *Loo* has proposed a general mathematical form for d, thus creating a large class of measures.

   Measure:

$$d(\mu) = F\left\{\sum_{i=1}^{N} c_i F_i(\mu(x_i))\right\}$$

   where $c_i \epsilon R^+$; for all i and F_i is a real-valued function such that
   $F_i(0) = F_i(1) = 0$,
   $F_i(u) = F_i(1-u)$ for all $u \epsilon$[0,1], F_i is strictly increasing on [0,$\frac{1}{2}$]; and F is a positive, increasing function.

   Properties: (P1)-(P4) are valid and if F is linear, then (P5) also holds.

   The fuzziness measure introduced by DeLuca and Termini can be obtained as special case if $F(x) = x$ and for all $i: \ c_i = K$ and

$$F_i(x) = -(xlog(x) + (1-x)log(1-x))$$

4. *Trillas* and *Riera* (1978) have proposed another large class of fuzziness measures.

   Measure:

$$d(\mu) = \sum_{i=1}^{N} w(x_i)F(\mu(x_i)),$$

   where μ is a real-valued function on [0,1], $F(x) = 0$ iff $x \epsilon\{$0,1$\}$, $F(x)$ is nondecreasing on [0,$\frac{1}{2}$], $F(x)$ is nonincreasing on [$\frac{1}{2}$], and w is a positive weight function from X to R.

5. *Emptoz* (1981) [4]
   Measure:
   $$d(\mu) = \frac{1}{N}\sum_{i=1}^{N} F(\mu(x_i)),$$
   where F is a real-valued function on $[0{,}1]$ such that $F(0) = 0$, $F(x) = F(1-x)$ and F is increasing on $[0,\frac{1}{2}]$.
6. *Ebanks* (1983) [3]
   Measure:
   $$d(\mu) = \sum_{i=1}^{N} \mu(x_i)(1-\mu(x_i)),$$
   where $\mu\epsilon[0{,}1]^X$.

As we are interested in fuzziness measures, we will now introduce the vagueness measure.

2. Vagueness measure induced by Pliant operators

In the Pliant system the logical values basically arise from inequalities. If we are on the border of the inequality, i.e. just the equalities are fullfiled we are not sure whether we are inside or outside a region. If we move away from the border we are more certain to be inside or outside the region. Why are we so vague on the border? Because small changes can radically change the logical value. If we demand stable statements, we should avoid being just on the border. Hence it is important to measure the vagueness and also to know how it depends on the vagueness on the input values.

As we mentioned above the idea and construction of a vagueness measure can be derived from the fuzziness measure. In 1972 DeLuca and Termini [1] introduced a fuzziness measure which is used for the membership function. In the Pliant concept we will use the vagueness measure as the operand of the distending function.

From the property of $d(\mu)$, we can define $\mathcal{V}(x)$ in the following way.

Definition 1. *The Vagueness measure in the Pliant system is*

$$\mathcal{V}^*(\delta(\underline{x})) = \frac{1}{n}\sum_{i=1}^{n} \bar{c}(\delta(x_i),\eta(\delta(x_i))) \tag{1}$$

and

$$\mathcal{V}(x) = \bar{c}(x,\eta(x)) \tag{2}$$

Here $\mathcal{V}(x)$ is called the vagueness function and $c(x,y)$ is the conjunitive operator and $\eta(x)$ is a Pliant negation. I.e.

$$\bar{c}(x,y) = f^{-1}(\frac{1}{2}(f(x) + f(y))) \tag{3}$$

$$\eta(x) = f^{-1}(f(x_0)\frac{f(\nu)}{f(x)}) \tag{4}$$

f is the generator function of the conjunitive operator. [5]

The normalized vagueness measure is

$$\mathcal{V}^N(x) = \frac{1}{\bar{c}(\nu_0,\eta(\nu_0))}\bar{c}(x,\eta(x)) \tag{5}$$

Because

$$\mathcal{V}(x) = \bar{c}(x,\eta(x)) = f^{-1}\left(\frac{1}{2}\left(f(x) + f(\nu_0)\frac{f(\nu)}{f(x)}\right)\right) \tag{6}$$

is the representation of vagueness and if we demand that the maximum value should be at ν_0, we have to find the minimum of:

$$Y = X + \frac{A}{X}, \tag{7}$$

which is at $X = \sqrt{A}$, i.e. $f(x) = \sqrt{f(\nu_0)f(\nu)}$, and the maximum is at ν_0, so $f(\nu_0) = f(\nu)$, and we get

$$\mathcal{V}_{\nu_0}(x) = f^{-1}\left(\frac{1}{2}\left(f(x) + \frac{f^2(\nu_0)}{f(x)}\right)\right) \tag{8}$$

if $f(\nu_0) = 1$

$$\mathcal{V}(x) = f^{-1}\left(\frac{1}{2}\left(f(x) + \frac{1}{f(x)}\right)\right) \tag{9}$$

Now we will give a list of properties for the measure of vagueness:

1. Sharpness (No vagueness) (P1)
   $\mathcal{V}(x) = 0 \quad iff \quad x\epsilon\{0,1\}$

2. Maximality (maximal vagueness) (P2)
   $\frac{1}{\nu_0}\mathcal{V}(x) = 1 \qquad iff \qquad x = \nu_0$
3. Symmetry (P4)
   $\mathcal{V}(x) = \mathcal{V}(\eta(x))$
4. Monotonicity (P3)
   $\mathcal{V}(x_1) < \mathcal{V}(x_2)$ if

   $x_1 < x_2$ and $x_1 \le \nu_0$

   or

   $x_1 > x_2$ and $x_1 \ge \nu_0$

Vagueness measure in the Dombi operator case

As above, the vagueness function is defined in the following way:

$$\mathcal{V}(x) = f^{-1}\left(\frac{1}{2}\left(f(x) + \frac{1}{f(x)}\right)\right)$$

Let $f(x) = \left(\frac{1-x}{x}\right)^{\alpha}$, namely the Dombi operator. Then

$$F_{\alpha}(x) = \frac{1}{1 + \left(\frac{1}{2}\left(\frac{1-x}{x}\right)^{\alpha} + \frac{1}{2}\left(\frac{x}{1-x}\right)^{\alpha}\right)^{\frac{1}{\alpha}}} \tag{10}$$

If $\alpha = 1$ then

$$\mathcal{V}(x) = 2x(1-x) \tag{11}$$

Vagueness measure of the sigmoid function when $\alpha = 1$

The sigmoid function is

$$\delta^{(\lambda)}(x) = \frac{1}{1 + e^{-\lambda x}}$$

So we have

$$\mathcal{V}(\delta^{(\lambda)}(x)) = 2\delta^{(\lambda)}(x)(1 - \delta^{(\lambda)}(x)) \tag{12}$$

$$\mathcal{V}^*(\delta^{(\lambda)}(x)) = 2\int_{-\infty}^{\infty} \frac{(\delta^{(\lambda)}(x))'}{\lambda} dx = \frac{2}{\lambda}\left[\delta^{(\lambda)}(x)\right]_{-\infty}^{\infty} = \frac{2}{\lambda} \tag{13}$$

3. Vagueness measure and entropy

In the case of multiplicative Pliant systems we can define the vagueness function using the conjunctive operator and negation

$$\mathcal{V}(x) = \bar{c}(x,\eta(x)) \tag{14}$$

The normalized $\mathcal{V}(x)$ is

$$\mathcal{V}^N(x) = \frac{1}{\bar{c}(\nu_0,\eta(\nu_0))}\bar{c}(x,\eta(x)) \tag{15}$$

In the additive Pliant case (The bounded or nilpotent operator case)

$$c[x,\eta(x)] = 0$$

and (15) has no meaning.

We will use the following notation:

$$[x] = \begin{cases} 1 \text{ if } x > 1 \\ x \text{ if } 0 \le x \le 1 \\ 0 \text{ if } x < 0 \end{cases} \tag{16}$$

First we try to find a proper vagueness measure for the Lukasiewicz operator case.

$$c[x,y] = [x + y - 1]$$

The generator function of $c[x,y]$ is $f(x) = 1 - x$.

Let us define $c_\alpha[x,y]$ using the $(1-x)^\alpha$ generator function, where $\alpha > 0$.

$$c_\alpha[x,y] = 1 - [((1-x)^\alpha + (1-y)^\alpha)^{\frac{1}{\alpha}}]$$

Therefore

$$c_1[x,y] = [x + y - 1]$$

Let $\eta(x) = 1 - x$, which is the simplest definition of $\eta(x)$. Then

$$c_\alpha[x,1-x] = 1 - ((1-x)^\alpha + x^\alpha)^{\frac{1}{\alpha}} \neq 0$$

Let us define the Vagueness measure for the additive Pliant case based on (15) in the following way:

$$\mathcal{V}[x] = \underset{\alpha\to 1}{lim}\mathcal{V}_\alpha(x) = \underset{\alpha\to 1}{lim}\frac{1-((1-x)^\alpha + x^\alpha)^{\frac{1}{\alpha}}}{1-\left(\frac{1}{2}\right)^{\frac{1}{\alpha}-1}} \tag{17}$$

Theorem 2.

$$\mathcal{V}[x] = \frac{1}{ln(2)}(xlnx + (1-x)ln(1-x)) \tag{18}$$

i.e., we get the ***Shannon entropy*** *as a measure of knowledge disorder and this theorem, which may be drived from the Lukosiewicz system.*

Proof. To calculate the limes of (17) we will use the L' Hospital rule, and we find the derivative at α.

First, we transform the nominator into the following form:

$$\frac{1-e^{\frac{1}{\alpha}\ln((1-x)^\alpha + x^\alpha)}}{1-\left(\frac{1}{2}\right)^{\frac{1}{\alpha}-1}}$$

The derivative of the denominator at $\alpha = 1$ is.

$$-\left(-\frac{1}{\alpha^2}\ln\left((1-x)^\alpha + x^\alpha\right) + \frac{1}{\alpha}\frac{1}{(1-x)^\alpha + x^\alpha}\left[x^\alpha \ln x + (1-x)^\alpha \ln(1-x)\right]\right)\Big|_{\alpha=1} = \\ -\left(x\ln x + (1-x)\ln(1-x)\right)$$

where

$$\frac{d(x^\alpha)}{d\alpha}\Big|_{\alpha=1} = x^\alpha \ln x\Big|_{\alpha=1} = x\ln x \tag{19}$$

The derivative of the denominator is

$$-2\left(\frac{1}{2}\right)^\alpha \ln\left(\frac{1}{2}\right)\Big|_{\alpha=1} = \ln 2$$

■

Theorem 2 can be generalized by substituting x into $f(x)$. Then we get

$$\mathcal{V}[x] = -\frac{1}{\ln(2)}(f(x)\ln(f(x)) + (1-f(x))\ln(1-f(x))) \tag{20}$$

Acknowledgements

This study was partially supported by the TÁMOP-4.2.2/08/1/2008-0008 program of the Hungarian National Development Agency.

References

[1] A. DeLuca, S. Termini. A definition of a non-probabilistic entropy in the setting of fuzzy sets theory. *Inform and Control*, 20:301–312, 1972.

[2] A. DeLuca, S. Termini. Entropy and energy measures of fuzzy sets, in: M. m. gupta, r.ragade, r.r yager, eds. *Advances in Fuzzy Set Theory and Applications*, pages 382–389, 1972.

[3] R. B. Ebanks. On measure of fuzziness and their representations. *Journal of Mathematical Analisis and Applications 94*, pages 24–37, 1983.

[4] H. Emptoz. Nonprobabilistic entropies and indetermination measures in the setting of fuzzy sets theory. *Fuzzy Sets and Systems 5*, pages 307–317, 1981.

[5] J. Dombi. "DeMorgan systems with an infinitely many negations in the strict monotone operator case" *Information Sciences*, 2011, Under print

Modelling the Intelligence Phenomenon

C. Pozna[*,**], R-E. Precup[**], N. Minculete[***], C. Alexandru[****]**

**[*] Department of Informatics, “Széchenyi István” University Győr, Egyetem tér 1
Győr, Hungary, e-mail: pozna@sze.hu,**
**[**] Department of Automation and Applied Informatics, “Politehnica” University
of Timisoara, Bd. V. Parvan 2, 300223 Timisoara, Romania,
e-mail: radu.precup@aut.upt.ro**
**[***] Departament of REI, “Dimitrie Cantemir” University str. Bisericii Ramane 107,
500036, Brasov, Romania, e-mail: minculeten@yahoo.com**
**[****]Department of Product Design and Robotics, “Transilvania” University Bd.
Eroilor 28, 500036 Brasov, Romania, e-mail: calex@unitbv.ro**

Abstract: The paper presents an analysis of the intelligence phenomenon from the point of view of the new Artificial Intelligence approach. The results of the analysis highlight that intelligence is related to the stability, modelling capacity and perturbation management of the agent. The structure of the paper is as follows. It begins with an introduction to the AI approaches, paradigms and knowledge representation. The main part of the paper focuses on an analysis of the intelligence phenomenon. The paper ends with conclusions and a discussion of future work.

Keywords: artificial intelligence, modelling, stability, perturbation

1. Introduction

The goal of Artificial Intelligence (AI) is to construct intelligent machinery (the *agent*). This goal is thought to be possible because scientists (or part of them) believe that it is not only the mechanical attributes of human beings (grasping objects, biped walking etc.) that can be modelled and copied in artefacts, but even the intelligence feature of humans can be understood, modelled and reproduced.

AI is a relatively new science. The first arguments concerning intelligent machinery have been proposed by Allan Turing in 1950 and the AI term has been proposed in 1956 at a conference which took place at Dartmouth College in New Hampshire.

In the earlier stages of development, AI-related knowledge sources have been considered to be philosophy and psychology, because AI researchers focused on explaining and modelling the intelligence. Nowadays, a related science to AI is robotics; here excellent results have been obtained in the aforementioned artefact construction.

Even though it is a new science, AI is classified in several ways: strong or weak; good old fashion (GOFAI) or new. More precisely the differences between strong and weak AI lie in the goal. Strong AI preserves the initial goal of this science and weak AI intends to copy only specific functions of the intelligence phenomenon (learning, decision making etc.). The difference between GOFAI and the new AI lie in intelligence phenomenon understanding.

The GOFAI development consists of a symbol manipulation stage and a connectionist stage. The first one was influenced by Allen Newell and Herbert Simon (1976) – the designers of Physical Symbol System Hypothesis (PSSH), a system which understands the intelligent phenomenon as the ability to syntactically manipulate symbols. The second, initiated by Turing (1940) and proposed by McCulloch and Pitts (1943), understands the intelligence like an emergence of a highly connected network of simple elements which models the biological neurons. The main difference between these stages is *philosophical*: the first considers that intelligence can be modelled by a computer program which is independent of the hard while the soft is running. The second stage considers that intelligence is emergent from the hard itself. The main similarity is that both can compute the same class of calculations as a Turing machine.

It is accepted that the limits of GOFAI are due to the following causes: the designed AI system operates in a simplified environment (micro world) which cannot be scaled to the real world complexity. This is the reason why results from the micro world cannot be used in real world - the robustness of the AI system cannot be compared with the robustness of a biological system, which is why the AI agent is unable to reject noise. The traditional strategy "*sense – model – plan – act*" seems to be inappropriate, which is why the AI agents cannot act in real time. If we try to synthesize the previous reproaches we can affirm that the AI agents are not able to understand and act in the real world. We will remember that we refer to understanding and acting as (approximations of) human actions.

The new AI intends to solve the problems of GOFAI. For this purpose it starts from the following principles: the embodiment is significant in AI design; the AI design must consider a real world environment (the *situatedness*); the AI design must be a bottom – up design, starting with mechanical and electrical subsystems and ending with the intelligent subsystem design. If we try to synthesize the previous principles we can affirm that from the new AI point of view the AI agent is an intelligent robot.

Because our paper will focus on intelligent phenomenon modelling it is important to mention the known paradigm of AI.

In order to prove that intelligent behaviour results from heuristic search, Allan Newell and Herbert Simon (1950-1960) introduced the paradigm of mind modelled by computer programs. The "Logic theorist" created by J. C. Shaw proved theorems in elementary logic. This effort has been continued by Newell and Simon in the program named "General Program Solver" (GPS). The GPS examined the syntactic differences between the current state and the goal state and selected operators which reduced the differences. These methods have been named "weak method problem solving". The “weakness” of these methods consists of the fact that each practical problem involves a big amount of knowledge and different heuristics.

“Strong methods” have been imagined (Brian Smith 1985). These methods use explicit knowledge, combined with particular heuristics of a particular domain. This new paradigm includes the assumption that knowledge is represented propositionally and the behaviour of the intelligent agent can be seen as being formally caused by the proposition in the knowledge base and that this behaviour should be consistent with our agreement about the meaning of the proposition.

The last paradigm in intelligence phenomenon modelling is described as “agent-based”, “embodied” or “emergent” problem solving. These approaches, performed by Brooks (1987), Agre and Chapman (1987), Fatima (2005, 2006), Vieira (2007) have challenged the requirement of having any centralized knowledge base or general inference scheme. On the contrary, problem solvers are designed as distributed autonomous and flexible agents.

The mentioned paradigms are related to information representation. In the following we will describe the main strategies of information representation: the semantic network, the script, the frames and the alternative representations.

The semantic network is grounded on associations theories. These theories, in accord with the empiricist philosophical tradition, define the meaning of an object in terms of a network of associations with other objects. Graphs have proved to be an ideal vehicle for formalizing association theories of knowledge.

A script is a structured representation describing stereotype sequences of events in a particular context (Schank & Abelson 1977). Scripts have been introduced because there is evidence that humans organize knowledge into structures corresponding to typical situations. Scripts are used in organizing a knowledge base in terms of situations which the intelligent agent is to understand.

The frames are another representational scheme similar to scripts that was intended to capture in explicitly organized data structures the implicit connections of information in a problem domain (Minsky 1975). A frame may be viewed as a static data structure used to represent well understood stereotype situations. The concepts of frames and scripts lead to objected oriented programming. Objects are defined as computational systems with an encapsulated state. They have methods associated with this state that support interactions in an environment.

Alternative representations have continued to question the role of explicit representation of knowledge. The subsumption architecture (Brooks 1997, Pozna 2007) is a layered collection of task handlers. Each task is accomplished by a finite state machine that continually maps an input in to an action oriented output. The copycat is a problem solving architecture (M. Mitchell 1993) which supports a semantic network mechanism that grows and changes with continuing experience within its experience. These concepts lead to multi agent systems: a computer program with problem solvers situated in interactive environments which are each capable of flexible autonomous socially organized actions. The difference between objects and agents include the fact that objects rarely exhibit control behaviour and agents do not invoke methods on one another.

Knowing this evolution of AI, and based upon our previous results, we consider that nowadays, when the association of AI with robotics is general accepted, it is useful to reanalyze the concept of intelligence and the possibility of modelling this phenomenon. Using this analysis, the present paper intends to propose a new modelling strategy of the intelligence phenomenon. This strategy belongs to the new AI approach and develops its concepts from the underlying idea that the intelligence phenomenon is related to stability, modelling and perturbation management. The next section of the paper will focus on the intelligence phenomenon analysis. This presentation is initiated with the definition of intelligence and finalized by comparisons with the classical model domain. The paper ends with conclusions and future work specifications.

2. Analysis of the intelligence phenomenon

2.1. Intelligence

A model is an approximation of an object or a phenomenon. Concerning our discussion the model is a result of a modelling process with the purpose of technological knowledge. The model of (human) intelligence will serve in the design and construction of artefacts. Modelling means, in the very first instance, to understand the phenomenon and establish, by cutting the Universe, what is important and what can be excluded in the phenomenon description. Cutting the Universe in two domains (important and unimportant) is a courageous and pragmatic act. It is courageous because it is a subjective act (an art) which belongs to (must be included into) a scientific (objective) representation of the phenomenon. It is pragmatic because (until now) it was successful in the knowledge processes. To be more precise about the unimportant part of the Universe we must specify that this encloses one part which we know to exclude and another which we ignore. To save this mental construction relative to the phenomenon the concept of perturbation has been introduced. The errors, which are the difference between the observed data and the computed data (via model), are generated by perturbations. The mentioned difference has several sources which also include the way in which we treat the important part of Universe (the linearization, the parameter values etc.) but for our discussion we will focus only on the Universe division. There are several ways to treat the perturbations: neglect them or try to diminish them. Diminishing is possible by increasing the model performance (increasing the important part of the Universe, refining the model), predicting them by statistic law (Bayesian or Kalman filters), or constructing closed loop systems which observe - measure (continuously or discretely) the phenomenon. We name the important part of Universe the modelling domain.

We remember that modelling is the way in which we construct our knowledge about phenomena. Modelling is not an exclusively scientific knowledge process. A model can be an explanation (for example an “If … Then” rule) or a black box which implicitly constructs causal rules, for example a neural network or a faith in something. Both cases are accepted like understanding the Universe.

Returning to the AI goal we can say that such a system can be created if the intelligence phenomenon can be modelled. In order to do this we must define intelligence. In the very first moment we must say that this concept seems to be a suitcase concept which includes several meanings. This means that several definitions have been put forward. One of them has been proposed by Piaget: “intelligence is a state of equilibrium to which the successive sensory-motor and cognitive adaptations tend”. This equilibrium state means stability in time and space. Increasing the stability allows the capacity to achieve complex predictions about trajectories of states.

If we corroborate our previous discussion on modelling with this definition we can conclude that increasing the stability of our states is the reason for our adaptations. This stability allows us to produce complex predictions. In order to make predictions we use models about the phenomena from the Universe. This means that intelligence is also the art of perturbations management. Once again, this means the art of constructing complex models in order to diminish the perturbations or simple models and predict - handle significant perturbations.

From the previous definition we detect the following consequences:

> Intelligence refers to a dynamic and elastic link between the agent and the Universe. *Dynamic* means that it depends on past states and *elastic* means that the interaction will modify the state of the Universe but also the state of the agent. More precisely intelligence is about an agent who belongs to a Universe and has a common history with it and not about an isolated agent or a static link;
>
> Intelligence is an adaptation – learning process, this means that it is time dependent and it tries to find local optimum solutions;
>
> Intelligence is caused by the Universe – agent interactions. Specific interactions create specific adaptations. For a human we can also speak of cultural interactions. The cultural interactions are possible by symbol manipulations under the conditions of symbol meaning achievement;
>
> A characteristic of intelligence is that the adaptations are reinforced by predictions via modelling; modelling is based on knowledge, acquired by cognitive processes. The main source of knowledge is the experience where the named interaction between the Universe and the agent take place.
>
> Prediction means to simulating a model and understanding the simulation result; it is a mental operation where experiments are not needed; knowing that modelling creates approximations of the Universe in order to obtain good prediction performances in perturbation management is required here.

Because perturbation management seems to be so important in the intelligence explanation the previous analysis is continued with this concept. Perturbation management in artificial intelligence achievement means to:

> Construct the appropriate model; find a balanced ratio between the complexity of the model and its utility. From the sensory – motor point of view we have the "keep it smart and simple" rule of thumb. From the cognitive point of view we know the razor of Occam rule.
>
> Associate the model (outputs) predictions with an indicator, the degree of trust. Perform observations when the degree of trust reaches a lower admitted limit value;
>
> Construct a compliance mechanism, from the sensory – motor point of view (e.g. having a redundant degree of freedom or elastic coupling etc) and from the cognitive point of view (e.g. having alternative behaviours).
>
> Decrease the reaction time. From the sensory – motor point of view, this means to create a hierarchical control system and solve with low level control loops as many tasks as possible, etc. From a cognitive point of view, this means to replace as many times as possible the chains of sense – model – plan – act with input – output behaviours (neural network or state machine).

Preliminary conclusions are needed: intelligence is a phenomenon which refers to the agent states stability; the equilibrium point is reached by adaptation; a higher level form of adaptation is the prediction where models are simulated; modelling needs knowledge acquirement (cognition); predictions are related to perturbation management.

Intelligence is possible because of the specific structure of the agent. In *Figure 1* we propose a graphical representation of the mentioned structure. The agent is captured by its sensory – motor and cognitive components and the adaptation is represented by its two components: the assimilation and the accommodation (Piaget).

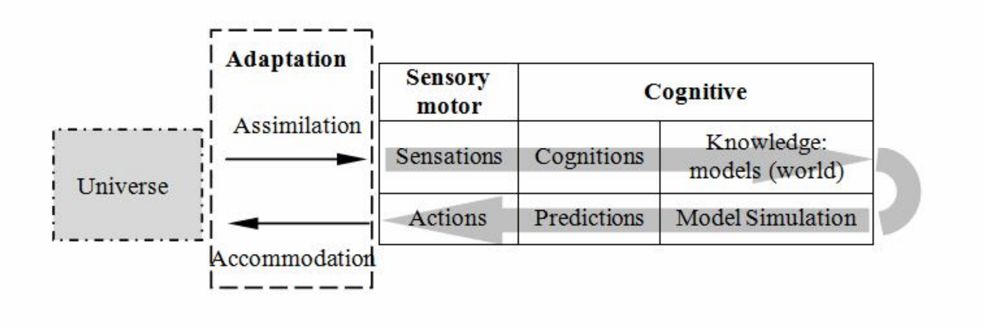


Figure 1. The Intelligence phenomenon diagram

The assimilation is possible by sensations which give information for mental processes (cognition). These processes are the perceptions, the observations, the experience, the memory etc. By cognition the intelligent agent will construct knowledge: models about part of universe. (The collection of these models is usually named *the world*.)

The reason for model construction (this investment) is the anticipation possibility, the prior knowledge generation. This implies simulating models and establishing predictions and ultimately generating actions. An intelligent agent creates, by the described mechanism, a stable vicinity of its state. Using the strategy of *modelling – simulation - prediction* the intelligent agent avoids the necessity of short time reactions.

2.2. Copying intelligence

In the previous section we have argued that intelligence is a high level of adaptation which is possible due to a particular structure of the (intelligent) agent. A possible way to design an artefact which will have the attribute of intelligence (the goal of AI) is to copy this structure. In this way we consider that intelligence will emerge from this structure. In *Figure 2* a block diagram of elements involved in the artificial intelligence phenomenon is presented. The block diagram is in conformity with the new AI concept whereby the intelligence is a result of the interactions between the world and the agent.

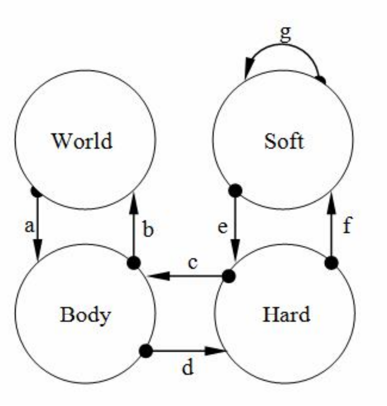


Figure 2. Elements involved in the artificial intelligence phenomenon

The world is defined here as a part of the Universe with the feature that it can be known. The body is an embodiment of the intelligent agent. The body allows interactions between the intelligent agent and the world; it is a material structure able to act on the world and to receive sensations caused by interactions. The prototype image of the body is a robot composed of mechanical elements, electrical drivers and sensors. The hard is the material support of the soft and its prototype image is a processor or a neural network electrical circuit. The soft is the cognitive centre. By cognition (a set of processes like memory, reasoning, learning etc.) the perceptions are transformed into knowledge. Motivations like expectation, wishes etc are transformed into desires and in the end, according to a specific set of rules, behaviours etc. transformed into tasks. The links between these elements can be defined in the following way:

a. Constraints which are the world influence on the body; the constraints mean the available links (contacts) between the world and the body (including the available input messages into a language);
b. Actions which represent the body influence on the world; the actions imply changes in the world, the transformation of the constraints (including output messages into a language);
c. The hard influence on the body is represented by the commands;
d. Sensations which are the world influence on the body; the sensations are a proto knowledge about the world obtained from constrains;
e. The task, which is the soft influence on the hard; the task is an objective imposed by the soft;
f. Perceptions, which are the hard influence on the soft; the perception is a first stage of knowledge here - the world is decomposed in intelligible objects;
g. Reflection is the soft influence on himself; reflection allows for changes in the soft structure;

3. Conclusions

In the first section of this paper we have described the evolution of AI. This part illustrate the paradigms of this science and concludes that nowadays we accept that in order to design intelligent artefacts we must provide the interaction between these artefacts and the Universe. It is accepted that this interaction will ensure the understanding of the meaning of knowledge.

We have considered that in this circumstance a reanalysis of the intelligence phenomenon is useful because it will refresh the background of our understanding and will revive new ideas on intelligent artefact construction.

The main result of this analysis is the idea that the mentioned interaction is not only the possibility of intelligence but it is the cause for intelligence. More precisely, intelligence is the phenomenon of stability acquisition from an agent - Universe relation. Another result was the idea that a specific structure of the system ensures the possibility of intelligence. This structure ensures the knowledge acquisition (model of the Universe), simulation of models and prediction. The reason for model construction is the anticipation possibility, the prior knowledge generation. Using the strategy of *modelling – simulation – prediction* the intelligent agent avoids the necessity of short time reactions.

References

[1] T. L. Griffiths, C. Kemp, J. B. Tenenbaum: “Bayesian models of cognition,” in *The Cambridge Handbook of Computational Psychology*, R. Sun, Ed. Cambridge: Cambridge University Press, 2008, pp. 59–100.

[2] http://www.fp6.cordis.lu/fp6/call.

[3] J. L. McClelland, D. E. Rumelhart, and the PDP Research Group: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 2 Psychological and Biological Models. Cambridge, MA: MIT Press, 1986.

[4] D. E. Broadbent: *Perception and Communication*. London: Pergamon Press, 1958.

[5] A. M. Treisman, G. Gelade: "A feature integration theory of attention," *Cognitive Psychology*, vol. 12, pp. 97–136, Jan. 1980.

Comparison of a Neural Network Based on Fuzzy Flip-Flops and an MLP Robustness in Function Approximation

Robustness of Fuzzy Flip-Flop Based Neural Networks

R. Lovassy[1], L. T. Kóczy[2,3], L. Gál[2,4]

Abstract: In this paper two types of neural networks, namely the “traditional” *tansig* based neural networks and the multilayer perceptrons based on fuzzy flip-flops (F^3NN) trained by the Bacterial Memetic Algorithm with Modified Operator Execution Order (BMAM) are tested and compared on their robustness to test functions outliers. The robust design of the F^3NN is presented, and the best suitable fuzzy neuron type is emphasized. As our major motivation in these investigations was to construct a technology for the creation of real hardware MLPs and for this reason the fuzzy flip-flop based F^3NNs obviously offered much simpler and cheaper possibility for hardware implementation compared to a relatively complicated *tansig* type neural network.

Keywords: multilayer perceptrons based on fuzzy flip-flops, Bacterial Memetic Algorithm with Modified Operator Execution Order, fuzzy neural networks robustness to outliers

1. General remarks

Fast and good function approximation is a critical problem to solve in many applications ranging from system identification to pattern classification. It is also known, that the use of function approximation has an essential role in control system [26]. Investigations on the universal approximator properties of fuzzy systems and neural networks have been one of the most interesting aspects of these fields.

In the years 1990-92 papers by Dubois et al. [4], Kosko [16], furthermore Wang and Mendel [27], [28] proved almost simultaneously that fuzzy systems were universal approximators. In 1997 Klement et al. [15] argued that fuzzy systems could only be universal approximators in a rather restricted sense, because of the limits set by computational complexity. The authors also exemplified the main approaches to realize the idea of controlling real world processes by means of linguistic variables.

In the field of artificial neural networks mathematical function approximation using input-output data pairs from a set of examples is the object of study in different applications such as applied mathematics, and computer science. The paper of Hornik et al. [13] established that standard multilayer feedforward networks with a single hidden layer constituted a class of universal approximators. They gave a general investigation of the capabilities and properties of multilayer feedforward networks, without any suggestion to the number of hidden units needed to achieve a given accuracy of approximation. The function approximation capability of multilayered neural networks was studied in detail by Ciuca [2], Cybenko [3], Funahashi [5], Hecht-Nielsen [10] and Ito [14]. They proved that any continuous function could be approximated by a three-layered feedforward network with hidden sigmoid units and one linear output unit. The use of four-layered neural networks (with two sigmoid unit layers) as universal approximators of continuous functions was investigated by Funahashi [5], Girosi and Poggio [8] and Hecht-Nielsen [10]. Kurkova [17] studied also multilayer feedforward networks with sigmoid activation function approximation capabilities, analyzing also their computational complexity issues. Blum and Li showed [1] that four-layered feedforward networks with two hidden layers of semi linear units and with a single linear output unit were also universal approximators.

Hirota introduced the concept of fuzzy flip-flops (F^3) [11], as the extension of the binary counterparts. He started his investigations paralelly from the conjunctive and disjunctive minimal forms, thus defining set and reset type F^3s. As there exist infinite many different fuzzy logic operations (and this is true even for conjunctions, disjunctions and negation) obviously there are infinite many ways to define fuzzy flip-flops. These units obviously show a rather wide scope of behaviour. Hirota's group did various hardware implementations as well, investigating the behaviour of these fuzzy circuits [12]. A unified equation proposed in [24] led to a family of circuits with more symmetrical behaviour.

Fuzzy Flip-Flop based Neural Networks (F^3NN) were defined and their learning algorithm was proposed in [21], furthermore giving in this way a possible starting point for many researchers in the field of hybridization of fuzzy logic with artificial neural networks, and in function approximation for learning control.

This paper investigates the problem of stability analysis of F^3NNs. The sensitivity of fuzzy neural networks to the fuzzy neuron type is evaluated as approximator of different test functions. Performance of the F^3NNs and a simulated MLP is compared in order to evaluate robustness on the various test functions with respect to the outliers. When developing a robust function approximator, an important point is to maintain stability even in the presence of unexpected outliers or external disturbances applied to the network.

This paper is organized as follows. Section 2 summarizes some of our previous works and offers fundamental results for providing our results. In Section 3 the design of the F^3NN is presented. In Section 4 illustrative examples are given to study stability and to evaluate the robustness of simulated and F^3 based neural networks. It turned out that the Łukasiewicz type F^3NNs had a much more robust behaviour than the standard *tansig* based simulator. Finally, some concluding remarks are given.

2. Fuzzy neuron activation functions, review of previous works

One of the common forms of the transfer function used in the construction of neural networks is the sigmoid activation function. The starting point of our own investigations was the fuzzy flip-flops [11]. In our previous papers [21], [22] the unified fuzzy J-K flip-flop based on various norms combined with the standard negation, was analyzed in order to investigate, whether and to what degree they present more or less sigmoid (s-shaped) $J \to Q_{out}$ characteristics in particular cases, when $K = 1 - Q$ (unified fuzzy J-K flip-flop with feedback), $K = 1 - J$ (new fuzzy D flip-flop derived from the unified fuzzy J-K one) with fixed values of Q. We conducted extensive investigations and found that the $J \to Q_{out}$ transfer characteristics of fuzzy J-K flip-flops with feedback based on Łukasiewicz, Yager, Dombi and Hamacher norms, further the $D \to Q_{out}$ characteristics of fuzzy D flip-flops of Łukasiewicz, Yager and Hamacher operations show quasi sigmoid curvature for selected Q and fuzzy operations parameter value pairs.

A new pair of conjunction and disjunction, the triangular t-norm and t-conorm was introduced in [7]. These new fuzzy operations, furthermore the Łukasiewicz and Dombi norms combined with the standard negation were applied in a practical problem, namely, they were proposed as suitable triangular norms for defining fuzzy flip-flop based neurons. Table 1 shows the above mentioned two widely used fuzzy operations, and the new triangular t-norm and t-conorm expressions.

Table 1. Some selected t-norms and t-conorms

Fuzzy Operation	t-norm $i\ (x, y)$	t-conorm $u\ (x, y)$
Łukasiewicz	$\max(0, x+y-1)$	$\min(1, x+y)$
Dombi	$\frac{1}{1+\left[(1/x-1)^{\alpha}+(1/y-1)^{\alpha}\right]^{1/\alpha}}$	$\frac{1}{1+\left[(1/x-1)^{-\alpha}+(1/y-1)^{-\alpha}\right]^{-1/\alpha}}$
Trigonometric	$\frac{2}{\pi}\cdot\arcsin\left(\sin\left(x\frac{\pi}{2}\right)\cdot\sin\left(y\frac{\pi}{2}\right)\right)$	$\frac{2}{\pi}\cdot\arccos\left(\cos\left(x\frac{\pi}{2}\right)\cdot\cos\left(y\frac{\pi}{2}\right)\right)$

The sigmoid activation functions are given by the expressions (1)-(3), which are the characteristic equations of the fuzzy J-K flip-flops with feedback based on Dombi and trigonometric norms, furthermore the fuzzy D flip-flop (derived from the J-K one) based on Łukasiewicz operations. For simplicity the t-norm is denoted by i (intersection), and the t-conorm by u (union), where the subscripts refer to the initial of the type of the norm: e.g., in case of the Dombi norms: $i_D\ (x,y) = x\ i_D\ y$ and $u_D\ (x,y) = x\ u_D\ y$.

The unified $Q(t+1)$ definition based on Dombi norms is

$$Q(t+1) = \left(J\ u_D\ Q\right)\ i_D \left(J\ u_D\ Q\right)\ i_D\ \left(Q\, u_D \left(1-Q\right)\right) = \tag{1}$$

$$= 1/\left(1 + \left(\left(2^{1/\alpha}\left(\left(\left(\left(-1+\frac{1}{J}\right)^{-\alpha} + \left(-1+\frac{1}{Q}\right)^{-\alpha}\right)^{-1/\alpha}\right)^{\alpha}\right)^{1/\alpha}\right)^{\alpha} + \left(\left(\left(-1+\frac{1}{1-Q}\right)^{-\alpha} + \left(-1+\frac{1}{Q}\right)^{-\alpha}\right)^{-1/\alpha}\right)^{\alpha}\right)^{1/\alpha}\right)$$

The characteristic equation of the fuzzy J-K flip-flop with feedback based on trigonometric norms is

$$Q(t+1) = \left(J\ u_T\ Q\right)\ i_T \left(J\ u_T\ Q\right)\ i_T\ \left(Q\, u_T \left(1-Q\right)\right) = \tag{2}$$

$$= 2\,\mathrm{a}\sin\left(\sin\left(\mathrm{a}\cos\left(\frac{1}{4}\cos^2 \pi^2 \left(1-Q\right)Q\right)\right)\sin\left(\mathrm{a}\sin\left(\sin\left(\mathrm{a}\cos\left(\frac{1}{4}\cos^2 \pi^2 JQ\right)\right)^2\right)\right)\right)/\pi$$

The next equation is defined by proposing the Łukasiewicz norms in the maxterm form of the new fuzzy D flip-flop formulae

$$Q(t+1) = \left(D\ u_L\ D\right)\ i_L\ \left(D\ u_L\ Q\right)\ i_L\ \left(D\ u_L\ \left(1-Q\right)\right) = \tag{3}$$

$$= \max\left(0,\ -1+\max\left(0,\ -1+\min\left(1,\ 2D\right)+\min\left(1,\ D+Q\right)\right)+\min\left(1,\ 1+D-Q\right)\right)$$

These sigmoid function generators can be implemented rather simply in hardware because they require only primitive mathematical operations, and therefore, they are inexpensive in terms of memory and computational requirements. The fuzzy flip-flop based neurons simple internal structure was given in [20]. The clocked fuzzy J-K and D flip-flop neuron circuits were built up using hardware blocks in order to realize various t-norms, t-conorms and fuzzy negations [29]. Since t-norms and t-conorms are functions from the unit square into the unit interval, the fuzzy J-K flip-flop block diagram differs from the binary J-K flip-flop structure. The input *J* is driven by a synchronized clock pulse in the sample-and- hold (S/H) circuit. FPGA technology was applied for hardware implementation of fuzzy D flip-flop neurons based on Łukasiewicz norms published in [19]. Simulation results, the hardware resources, number of logical gates, logic levels required, I/O used and timing delay consumed by Łukasiewicz operations and overall fuzzy D flip-flop neurons were presented. The proposed circuit was built up using core blocks which realize Łukasiewicz t-norm and t-conorm. The implementation was an 8 bit data width architecture and additionally two bit as selection for operations. These blocks can also be chained to form more complex structures such as Fuzzy Neural Network.

3. Multilayer perceptrons based on fuzzy flip-flops

The most commonly used network architecture is the multilayer feedforward network. The network topology consists of the connection between neurons, the input and the output structure. The neuron output signal is computed as the weighted sum of the input signals transformed by the transfer function. The deployment of the Levenberg-Marquardt algorithm (LM) [18], [23] and the Bacterial Memetic Algorithm with Modified Operator Execution Order Algorithm (BMAM) [6], was proposed and applied for F^{3}NN variables optimization and training. The four-layered F^{3}NN with hidden nodes defined by fuzzy flip-flop neurons (as sigmoid function generators) and a linear output node presented good function approximation properties. The network architecture built up from fuzzy D neurons (derived from the J-K one) is shown in Figure 1. In this approach the weighted input values are connected to inputs *J* and *K* of the new fuzzy flip-flop neuron based on a pair of t-norm and t-conorm, having quasi sigmoid transfer characteristics. The output signal is then computed as the weighted sum of the input signals transformed by the transfer function.

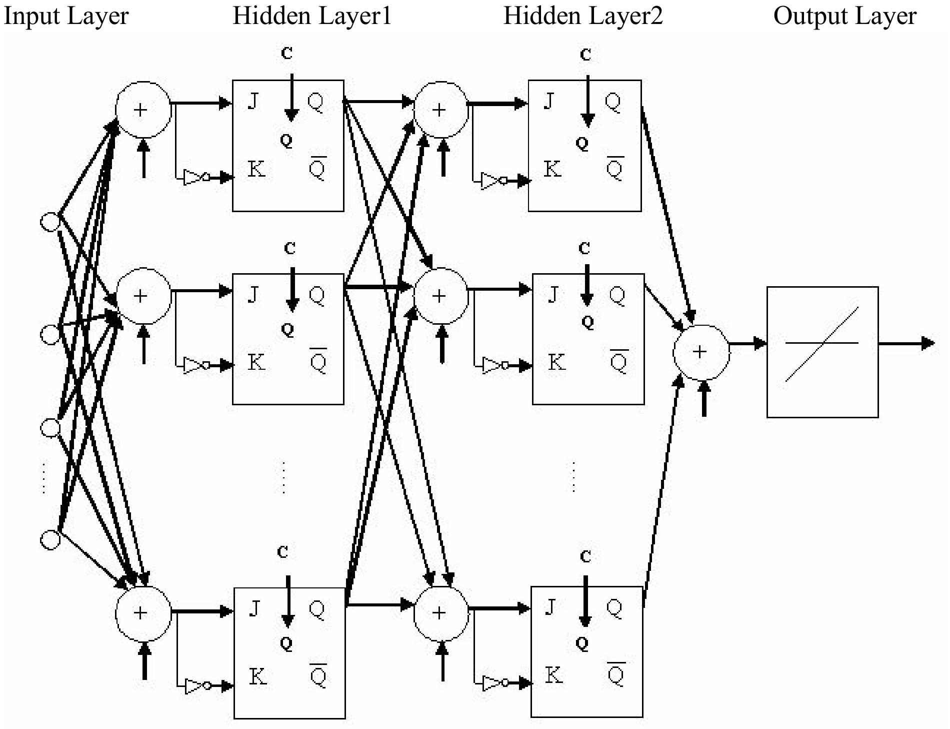


Figure 1. Fuzzy flip-flop based neural network

The most suitable types of F^3 neurons for constructing fuzzy neural networks were the Dombi and trigonometric type fuzzy J-K flip-flop neurons and Łukasiewicz type fuzzy D flip-flop neurons. Our investigations included comparative experiments using the

MATLAB Neural Networks Toolbox applying *tansig* based characteristic functions for the simulated neurons that we always used as reference [21], [22].

Our experiments made it obvious that the F^{3}NNs based on even the best suitable fuzzy operations always led to greater Mean Square of Error – MSE when the "idealistic" *tansig* based simulator had exactly the same structure and number of neurons. However the best type F^{3}NNs produced MSEs in the same order of magnitude, thus we concluded that F^{3}NNs presented a realistic alternative to simulated MLPs.

As our major motivation in these investigations was to construct a technology for the creation of real hardware MLPs and F^3 based NNs obviously offered much simpler and cheaper possibility for hardware implementation compared to a rather complicated approximated *tansig* type neural network.

In the next section we will show that the fuzzy flip-flop based neural networks with a given structure and neuron type posses the ability to approximate robust the proposed test functions with and without outliers. It will guarantee that a certain level of approximation accuracy is possible, this is what the stability and performance of our systems typically depends on. Illustrative examples are presented, in order to demonstrate the success of this work in terms of the function approximation capability and the robustness of the proposed fuzzy neural network.

4. Comparison of the robustness F^{3}NNs and *tansig* based neural networks

In the next, the robustness of three different types of F^{3}NN and the *tansig* based neural networks is compared from the various test function approximation goodness points of view. The Dombi and trigonometric type fuzzy J-K flip-flop neurons, furthermore the Łukasiewicz type fuzzy D flip-flop neurons based fuzzy neural networks (DJKFNN, TJKFNN, LDFNN) and the simulated neural networks are studied. Here the robustness of the models can be defined in such a way that the set of selected models should be robust in the sense that they are indifferent to the hidden layers neuron numbers and also they do not overfit in case of outliers.

The one and two dimensional test functions used for NNs robustness comparison will be described in the next.

4.1. Data description

Modified *pH* benchmark problem. Originally, the one dimensional *pH* problem [25] is the task to approximate the following function:

$$f_1(x) = pH^{-1} = -\frac{\log\left(\sqrt{\frac{y^2}{4}+10^{-14}}-\frac{y}{2}\right)+6}{26}; \qquad (4)$$

$$y = 2\cdot 10^{-3}\cdot x - 10^{-3};\ x \in [0,\ 0.75]$$

The pattern set consists of 101 datasets, with very uneven distribution:

Domain: [0.034914, 0.743401]

Range: [0.0001, 1.0000]

No data in (0.19, 0.38); (0.39, 0.59); etc.

In our approach, we modified the pattern set by generating intentionally four outlier points at the beginning and at the end of the domain.

Superposition of two sine waves. A combination of two sine wave forms with different period lengths described with the equation

$$f_2(x) = \frac{\sin(c_1 \cdot x)\cdot \sin(c_2 \cdot x)}{2} + 0.5 \tag{5}$$

The values of constants c_1 and c_2 were selected to produce a frequency proportion of the two components 1:0.35. The test function is represented by 100 input/output data sets.

Two two - input trigonometric functions. The two dimensional test functions are

$$f_3(x_1, x_2) = \frac{\sin(c_1 \cdot x_1)^5 \cdot \sin(c_2 \cdot x_2)^3}{2} + 0.5 \tag{6}$$

and

$$f_4(x_1, x_2) = \left(\cos\left(\arctan^*\left(\frac{x_1}{x_2}\right)\right) + 1\right) \cdot e^{-\frac{r^2}{50}} \cdot \sin^5\left(\frac{r}{10}\right) \tag{7}$$

$r = \sqrt{x_1^2 + x_2^2}$, $x_1, x_2 \in [-20, 20]$; arctan* is the four-quadrant inverse tangent function. The two-input functions are both represented by 1600 input/output data sets.

4.2. Training algorithm

The Bacterial Memetic Algorithm with Modified Operator Execution Order Algorithm [6] was applied for network training. In this application according to the network size a population was initialized. During the algorithm each individual is selected one by one, and 30 generations of 5 individuals with 5 clones are chosen to obtain the best fitting variable values, with the lowest performance. The same part or parts are selected randomly from the clones and mutated. The LM method nested into the evolutionary algorithm is applied for 5 times for each clone. Several tests have shown that it is enough to run 3 to 5 of LM iterations per mutation to improve the performance of the whole algorithm. The best clone is selected and transferred with its all parts to the other clones. Choosing-mutation-LM-selection-transfer cycle is repeated until all the parts are mutated, improved and tested. The best individual is remaining in the population all other clones are deleted. This procedure is repeated until all the individuals are taking part in the modified bacterial mutation. As a second main step, the LM is applied 7 times for each individual executing several LM cycles during the bacterial mutation after each mutation step. In the applied algorithm the gene transfer operation is completely excluded.

4.3. Results and discussion

The outlier points introduced in our data sets are numerically distant from the rest of data. Usually, this clear deviations from other data sample indicates that they do not fit the model under study, can occur by change in any distribution, input/output data pairs with very uneven distribution, or an error in measurement.

Next it is tested how well the simulated and the F^3 based neural networks function approximation handle the test functions outlier points. This section analyses the function approximation performance of these types of network in terms of accuracy, hidden layer neuron number, stability and robustness.

During the simulations the *tansig* based neural network is compared with all combinations of fuzzy J-K and D type F^{3}NNs which had been covered with all three fuzzy operation pairs to approximate the test functions listed in subsection 4.1, eq. 4-7.

Table 2. MSE median values, one dimensional test functions

	1D-1			
NN type	Neural network size			
	1-2-2-1	1-3-3-1	1-4-4-1	1-6-6-1
tansig	3.161x10^{-3}	2.996x10^{-3}	2.243x10^{-3}	1.242x10^{-3}
DJKFNN	3.171x10^{-3}	3.154x10^{-3}	3.080x10^{-3}	3.019x10^{-3}
TrigJKFNN	3.305x10^{-3}	3.222x10^{-3}	3.062x10^{-3}	3.029x10^{-3}
LDFNN	3.225x10^{-3}	3.067x10^{-3}	2.879x10^{-3}	2.298x10^{-3}
	1D-2			
NN type	Neural network size			
	1-2-2-1	1-4-4-1	1-6-6-1	1-8-8-1
tansig	7.112x10^{-3}	7.394x10^{-7}	2.141x10^{-7}	6.032x10^{-9}
DJKFNN	2.464x10^{-2}	1.563x10^{-3}	9.665x10^{-4}	3.764x10^{-6}
TrigJKFNN	6.626x10^{-3}	1.798x10^{-3}	3.025x10^{-4}	2.498x10^{-6}
LDFNN	*6.888x10^{-3}*	*2.043x10^{-3}*	*1.969x10^{-3}*	*6.858x10^{-4}*

Table 2 presents the 5 runs average approximation goodness, by indicating the median MSEs (mean squared error) of various networks training values. During evaluation we compared the median MSE values, considering them as the most important indicators of trainability. The median is a robust estimate of the centre of a data sample, since outliers have little effect on it. The median represents robust statistic, while the mean does not. In the next, we study how the network function approximation goodness depends on the number of hidden layers neurons, furthermore we show how much the simulator is better than the emulated hardware, and how many more F^3 neurons should be applied in order to achieve similar training properties. We denoted by 1D-1 the modified *pH* problem, by 1D-2 the one dimensional sine waveforms, by 2D-1 the $f_3(x_1, x_2)$, and finally by 2D-2 the $f_4(x_1, x_2)$ test functions.

Comparing the simulation results we concluded that increasing the neuron numbers in the hidden layers the networks function approximation accuracy had been increased. This research was concluded as an essentially successful one, especially so because one of the reference problem (modified *pH* problem) showed that F^3NNs had a much more stabile behaviour than the *tansig* based simulator: “Rogue” data (outliers) caused overfitting with the MATLAB simulator while F^3NNs kept the approximated function in a relatively smooth form, (see Figure 2.) even so the different networks function approximation goodness are in the same order of magnitude. In this case the simulator it turned out to be the same like the emulated hardware, furthermore the same number of F^3 neurons should be applied in order to achieve similar training properties.

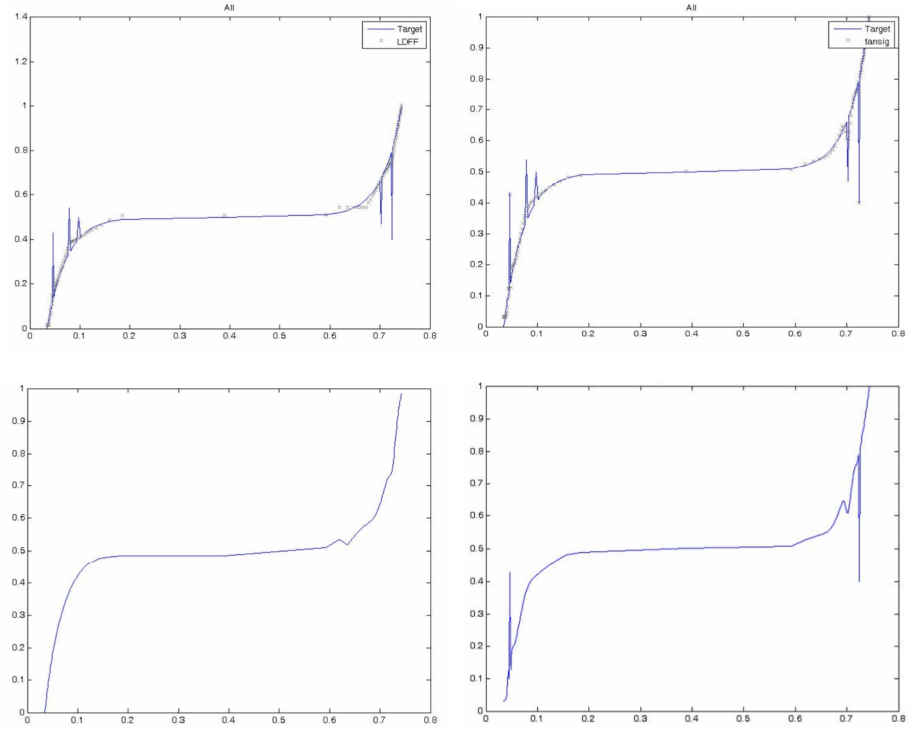


Figure 2. Łukasiewicz type F^3NN and tansig based neural network pH benchmark problem, 1-6-6-1 network size

When instead of the benchmark data set a more complex wave form (eq. 5) was used as test function the training MSE median values in case of 2-2 neurons in the hidden layers present almost the same values. Extending the network size, we have to apply approximately twice more F^3 neurons in order to obtain the same function approximation goodness as the *tansig* based neural network. The exception is the Łukasiewicz type fuzzy D flip-flop neuron based neural network, where several times more neurons have to be used. The simulation results ensure that this type of F^{3}NNs possesses a stabile property – the MSE median values are very close to each other. Figure 3 shows the simulation results.

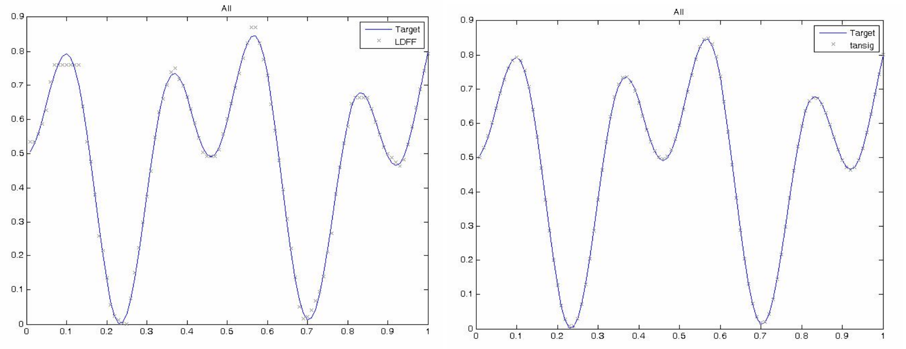


Figure 3. Łukasiewicz type F^{3}NN and tansig based neural network two sine waves, 1-4-4-1 network size

Table 3 summarizes the simulation results in case of the two dimensional test functions (eqs. 6 and 7).

Table 3. MSE median values, two dimensional test functions

	2D-1			
NN type	Neural network size			
	1-4-4-1	1-8-8-1	1-15-15-1	1-20-20-1
tansig	1.782x10^{-3}	9.077x10^{-5}	2.228x10^{-6}	9.071x10^{-7}
DJKFNN	5.519x10^{-3}	6.785x10^{-5}	2.131x10^{-5}	8.754x10^{-6}
TrigJKFNN	1.664x10^{-3}	1.557x10^{-4}	7.921x10^{-5}	1.033x10^{-5}
LDFNN	*4.038x10^{-3}*	*1.496x10^{-3}*	*9.672x10^{-4}*	*7.481x10^{-4}*
	2D-2			
NN type	Neural network size			
	1-4-4-1	1-8-8-1	1-15-15-1	1-20-20-1
tansig	3.735x10^{-7}	4.344x10^{-8}	6.229x10^{-9}	3.889x10^{-9}
DJKFNN	7.097x10^{-6}	1.501x10^{-6}	2.672x10^{-7}	5.732x10^{-8}
TrigJKFNN	1.241x10^{-5}	1.234x10^{-6}	5.343x10^{-7}	1.879x10^{-7}
LDFNN	*4.246x10^{-5}*	*1.714x10^{-5}*	*3.702x10^{-6}*	*1.046x10^{-6}*

Independently from the input function complexity, in these cases also the function approximation accuracy is directly proportional to the neuron numbers. Only the Łukasiewicz type F^{3}NNs are robust in the sense that that the network output does not change significantly under the influence of changes in the hidden layers. These types of F^{3}NNs are more robust to variations of neuron numbers. Figure 4 shows the function approximation goodness of Łukasiewicz type F^{3}NNs and the *tansig* based neural network in case of two dimensional test functions (eqs. 6, 7).

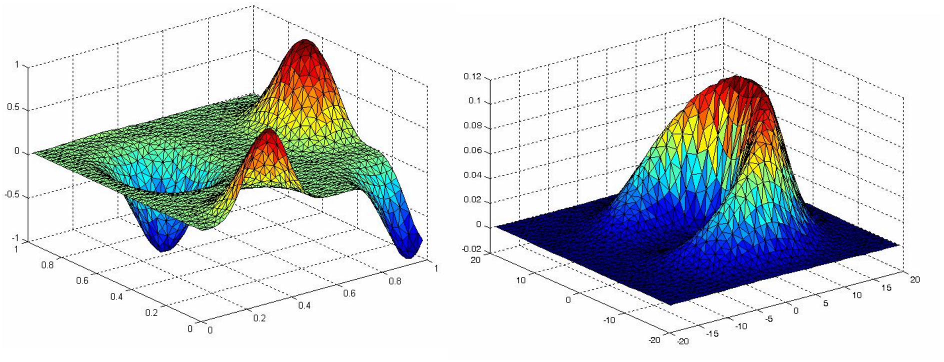


Figure 4. Łukasiewicz type F^{3}NN and tansig based neural network two dimensional test functions, 1-15-15-1 network size

Conclusion

In this paper we proposed the design of a robust function approximator for a general class of multi-input and one output systems. One of our test data sets contained a number of outlaying items. Taking both theory and practice of the Evolutionary Algorithms (EAs) [9] into account, it is safe to anticipate that the solutions obtained by the application of a conventional evolutionary optimization algorithm will be the most performing, not the most robust, that is the reason why we proposed the BMAM algorithm for the NNs training. In addition, selecting the most relevant rules (maxim of relation) the Łukasiewicz type F^{3}NNs yield more compact and robust systems. This kind of neural networks are robust enough for hardware implementation with general purpose (unknown application) and are more suitable to avoid overfitting than customary (e.g. *tansig* based) neural networks.

Further research will be dedicated to the mathematical analyses of why Łukasiewicz type F^{3}NNs present a better stability in terms of the number of neurons than the reference simulator. Next steps would be to prove stability and evaluate the robustness of these approaches. The idea is to avoid unnecessary retraining, and as consequence, we would like also to avoid any particular overtraining problem. Additionally, besides the often mentioned advantage of complexity reduction when using fuzzy logic in practical applications we found robustness being also present.

References

[1] Blum, E. K., Li, L. K.: *Approximation theory and feedfordward neural networks* Neural Networks, 4(4), 1991, pp. 511-515.

[2] Ciuca, I., Ware, J. A.: *Layered Neural Networks as Universal Approximators*, Fuzzy Days Dortmund, Germany, 1997, pp. 411-415.

[3] Cybenko, G.: *Approximation by superposition of sigmoidal functions*, Math. Contr. Signals. Syst. 2, 1989, pp. 303-314.

[4] Dubois, D., Grabisch, M., Prade, H.: *Gradual rules and the approximation of functions*, in Proc. of the 2nd International Fuzzy Systems Association Congress Iizuka, Japan, 1992, pp. 629-632.

[5] Funahashi, K. I.: *On the approximate realization of continuous mapping by neural networks*, Neural Networks, 2, 1989, pp. 183-192.

[6] Gál, L., Botzheim, J., Kóczy, L. T.: *Improvements to the Bacterial Memetic Algorithm used for Fuzzy Rule Base Extraction*, in Proc. of Computational Intelligence for Measurement Systems and Applications, CIMSA, Istanbul, Turkey, 2008, pp. 38-43.

[7] Gál, L., Lovassy, R., Kóczy, L. T.: *Function Approximation Performance of Fuzzy Neural Networks Based on Frequently Used Fuzzy Operations and a Pair of New Trigonometric Norms*, Proc. of IEEE World Congress on Computational Intelligence, WCCI 2010, Barcelona, Spain, pp.1514-1521.

[8] Girosi, F., Poggio, T.: *Representation properties of network: Kolmogorov's theorem is irrelevant*, Neural Computation 1, 1989, pp. 456-469.

[9] Goldberg, D. E.: *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley, Reading (1989).

[10] Hecht-Nielsen, R.: *Theory of the backpropagation neural network*, in Proc. of the Neural Networks, IJCNN, International Joint Conference on Neural Networks, 1989, pp. 593-605.

[11] Hirota, K., Ozawa, K.: *Concept of fuzzy flip-flop*, Preprints of 2nd IFSA Congress, Tokyo, 1987, pp. 556-559.

[12] Hirota, K., Ozawa, K.: *Fuzzy flip-flop as a basis of fuzzy memory modules*, In: M. M. Gupta et al., eds., Fuzzy Computing. Theory, Hardware and Applications, North Holland, Amsterdam, 1988, pp. 173-183.

[13] Hornik, K., Stinchcombe, M., White, H.: *Multilayer Feedforward Networks are Universal Approximators*, Neural Networks, 2, 1989, pp. 359-366.

[14] Ito, Y.: *Approximation Capability of Layered Neural Networks with Sigmoid Units on Two Layers*, Neural Computation, 6(6), 1994, pp. 1233-1243.

[15] Klement, E. P., Kóczy, L. T., Moser, B.: *Are fuzzy systems universal approximators?* International Journal of General Systems 28 (2–3), 1999, pp. 259 – 282.

[16] Kosko, B.: *Fuzzy Systems are Universal Approximators*, in Proc. of the IEEE International Conference on Fuzzy Systems, San Diego, CA, 1992, pp.1153-1162.

[17] Kurkova, V.: *Kolmogorov's theorem and multilayer neural networks*, Neural Networks, 5, 1992, pp. 501-506.

[18] Levenberg, K.: *A method for the solution of certain non-linear problems in least squares*, Quart Appl. Math. 2(2), 1944, pp. 164-168.

[19] Lovassy, R., Zavala, A. H., Gál, L., Nieto, O. C., Kóczy, L. T., Batyrshin, I.: *Hardware Implementation of Fuzzy D Flip-Flop Neurons Based on Łukasiewicz Norms*, in Proc. of the 9th WSEAS Int. Conference on Applied Computer and Applied Computational Science, Penang, Malaysia, 2010, pp.196-201.

[20] Lovassy, R., Kóczy, L. T., Gál, L.: *Optimization in Fuzzy Flip-Flop Neural Networks*, Computational Intelligence in Engineering; Studies in Computational Intelligence (eds.: I. J. Rudas, J. Fodor, J. Kacprzyk), Springer, Vol. 313, 2010, pp.337-348.

[21] Lovassy, R., Kóczy, L. T., Gál, L.: *Function Approximation Capability of a Novel Fuzzy Flip-Flop Based Neural Network*, Proc. of International Joint Conference on Neural Networks, IJCNN 2009, Atlanta, USA, 2009, pp. 1900-1907.

[22] Lovassy, R., Kóczy, L. T., Gál, L.: *Applicability of Fuzzy Flip-Flops in the Implementation of Neural Networks*; Proc. of CINTI 2008, 9th International Symposium of Hungarian Researchers on Computational Intelligence, Budapest, Hungary, 2008, pp. 333-344.

[23] Marquardt, D.: *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*, SIAM J. Appl. Math. 11, 1963, pp. 431-441.

[24] Ozawa, K., Hirota, K., Kóczy, L. T.: *Fuzzy flip-flop*, In: M. J. Patyra, D. M. Mlynek, eds., Fuzzy Logic. Implementation and Applications, Wiley, Chichester, 1996, pp. 197-236.

[25] Ruano, A E., Cabrita, C., Oliveira, J. V., Tikk, D., Kóczy, L. T.: *Supervised training algorithms for B-spline neural networks and fuzzy systems*, 9th IFSA World Congress and 20th NAFIPS International Conference (IFSA NAFIPS) Vancouver, 2001, pp. 2830-2835.

[26] Spooner, J. T., Maggiore, M., Ordonez, R., Passino, K. M.: *Stable adaptive control and estimation for nonlinear systems: neural and fuzzy approximator techniques*, John Wiley and Sons, 2002.

[27] Wang, L. X.: *Fuzzy systems are universal approximators*, in Proc. of the IEEE International Conference on Fuzzy Systems, San Diego, CA, 1992, pp. 1163-1169.

[28] Wang, L. X., Mendel, J. M.: *Fuzzy basis functions, universal approximations and orthogonal least-squares learning*, IEEE Transactions on Neural Nets, 3, 1992, pp. 807-814.

[29] Zavala, A. H., Nieto, O. C., Batyrshin, I., Vargas, L. V.: *VLSI Implementation of a Module for Realization of Basic t-norms on Fuzzy Hardware*, Proc. of FUZZ-IEEE 2009, IEEE Conference on Fuzzy Systems, Jeju Island, Korea, August 20-24, 2009, pp. 655-659.

Simulation Studies on Various Tuning Methods for Convergence Stabilization in a Novel Approach of Model Reference Adaptive Control Based on Robust Fixed Point Transformations

József K. Tar[†], Kristóf Eredics[‡]

Óbuda University
[†]Institute of Intelligent Engineering Systems,
John von Neumann Faculty of Informatics,
[‡]Donát Bánki Faculty of Mechanical Engineering and Security Technology

Abstract: The concept of “*Artificial Intelligence (AI)*” also contains the adaptive controllers that are able to observe the behavior of the *a priori only insufficiently known physical systems* under their control, and automatically can adjust themselves in order to achieve *precise control*. Regarding their implementations, certain approaches use the rather “conventional” means of AI as rule bases, fuzzy, neural or neurofuzzy systems, others can more strictly utilize the specialties of the available analytical models. In this paper the behavior of a novel version of the “*Model Reference Adaptive Controllers*” is investigated. In contrast to the traditional approaches the design of these controllers does not need the use of the difficult technique of Lyapunov’s “direct” method. Instead of the use of Lyapunov functions that can guarantee global asymptotic stability it applies “*Robust Fixed Point Transformations*” that work with a local, bounded basin of convergence of the iteration that converges to the solution of the control task. The method applies only three control parameters that in the most of the cases can be fixed. It is shown that by properly tuning only one of the three parameters the convergence of the controller can be stabilized. The theory does not uniquely define the details of this tuning in which we have a great freedom. The operation of various tuning strategies were investigated for the adaptive control for two interesting paradigms: an underactuated mechanical system and an other mechanical system that contains a dynamically coupled internal degree of freedom neither observed nor directly manipulated by the controller.

Keywords: Model Reference Adaptive Control, Lyapunov’s Direct Method, Robust Fixed Point Transformations, Contractive Mapping, Cauchy Sequence

1. Introduction

The subject area of Artificial Intelligence includes the set of adaptive controllers. These systems are also intelligent in the sense that they observe the behavior of the *a priori only insufficiently known systems* under their control and automatically adjust themselves in order to achieve *precise control* in spite of the missing *a priori* information. Regarding their implementations, certain approaches use the rather "conventional" means of AI as rule bases, fuzzy, neural or neurofuzzy systems. For instance, the mathematical model of the turbo jet engines (e.g. [1]) allows a kind of "situational control" [2] in which only typical control situations are of interest that can be arranged in a hierarchical structure. The method can be extended for dealing with large scale systems, too [3]. Other implementations can more strictly utilize the specialties of the available analytical models. For realization of *adaptive control of dynamical systems* generally two "traditional" groups of problem tackling exist. One of them uses the dynamical model of the system to be controlled with imprecise model parameters that are precisely tuned by observing its state propagation. This set can be referred to as the group of *"model parameter adaptive controllers"*. Classical representatives of these groups are the "*Adaptive Inverse Dynamics Controllers (AID)*" and the "*Adaptive Slotine–Li Controller (ASLC)*" that use formally exact analytical models, but other types of models based on universal approximators can also be designed in this group. These controllers work well only if it is guaranteed that the observed behavior of the system can be attributed exceptionally to its own dynamics, therefore the presence of unknown external disturbances and dynamically coupled subsystems not involved in the initial model can deceive their model–based tuning process. The "fundamental" significance of this classical approach is well substantiated by the homepage of the *"Nonlinear Systems Laboratory"* of the *"Massachusetts Institute of Technology"* [4] that, according to the *"U.S.News & World Report"* [5], is the World's leading educational institution in the "Engineering and IT" category. This home pages recommends only two fundamental books from 1986 [6] and 1991 [7] and about 90 journal papers written in the era from 1991 to 2010.

The members of the other traditional approach also use imprecise initial dynamic models but instead trying to tune the models' parameters they tune some controllers' parameters. Such kind of controllers can be referred to as *"signal adaptive controllers"* since their tuning processes directly influence the actions of the drives/actuators that realize the control. The "*Model Reference Adaptive Control (MRAC)*" is a typical representative of this group. It is a popular approach from the early nineties to our days (e.g. [8], [9], [10], [11]). The essence of the idea of the MRAC is the transformation of the actual system under control into a well behaving reference system (the *"reference model"*) for which simple controllers can be designed. In the practice the reference model used to be stable linear system of constant coefficients, but in principle it can be any type of prescribed "nominal" system having some "decent behavior". In [9] e.g. C. Nguyen presented the implementation of a joint-space adaptive control scheme that was used for the control of a non-compliant motion of a Stewart platform-based manipulator that was used in the Hardware Real-Time Emulator developed at Goddard Space Flight Center to emulate space operations. In [10]

Somló, Lantos, and Cát suggested and investigated via simulations a local, robust MRAC axis control for robots. The method is also attracting in the control of teleoperation systems [11].

The above mentioned examples of MRAC controllers as well as other appearances of the same idea in the mainstream of control literature applied Lyapunov's "direct method" that originally was elaborated for the investigation of the stability of dynamical systems in his PhD dissertation in 1892 [12], [13]. This method is quite ingenious because it allows to guarantee the stability of the motion of the controlled systems —either global or local, common, exponential or asymptotic stability— on the basis of relatively simple estimations without needing to obtain and study the solutions of the equations of motion. (It is well known that the majority of the practically important problems do not have analytical solutions in closed form, while the numerical solutions are normally valid only for the limited time-span of investigations and without deeper mathematical background their results cannot be extrapolated.) However, in spite of the essential conceptional simplicity of the Lyapunov function technique its practical use is rather an "art" than a simple procedure that could easily be automated. Finding the appropriate Lyapunov function candidate and making the proper mathematical estimations required for the proof of convergence need great mathematical skills and practices, and these difficult proofs normally take pages of papers and generate complicated, nontrivial restrictions to be met. For example in [14] 10 pages of the 14 pages long paper is consumed up with the description of necessary special assumptions and the intricate proof of convergence with a Lyapunov function consisting of 4 components. Such mathematical complications made it reasonable to seek some alternative approach to the realization of the idea of the adaptive controllers. For highlighting these difficulties two, relatively simple examples will be analyzed in the next subsection.

1.1. The Adaptive Slotine–Li Controller

The here presented observations were detailed in [15]. This control approach utilizes subtle details of the Euler–Lagrange equations of motion, namely that the terms quadratic in the generalized velocity components are not independent of the inertia matrix: they can be deduced from the inertia matrix, and according to their special position in the equations of motion they can be symmetrized. In this approach the exerted generalized torque/force components are constructed by the use of the actual model as follows:

$$\begin{gathered} Q = \hat{H}(q)\dot{v} + \hat{C}(q,\dot{q})v + \hat{g} + K_D r, e := q^N - q, v := \dot{q}^N + \Lambda e, \\ r := \dot{e} + \Lambda e, \tilde{p} := \hat{p} - p, C_{ij} = \frac{1}{2}\sum_z \dot{q}_z \left(-\frac{\partial \hat{H}_{zj}}{\partial q_i} + \frac{\partial \hat{H}_{ij}}{\partial q_z} + \frac{\partial \hat{H}_{iz}}{\partial q_j}\right) \\ Q = Y(q,\dot{q},v,\dot{v})\hat{p} + K_D r \end{gathered} \tag{1}$$

in which q^N and q denote the generalized co–ordinates of the *nominal* and the *actual* motion, K_D and Λ are symmetric positive definite matrices, matrices $\hat{H}$, $\hat{C}$, and $\hat{g}$ are the actual models of the system's inertia matrix, the Coriolis, and the gravitational terms, respectively. The possession of the exact form of the dynamical model makes it possible to linearly separate the system's dynamic parameters (the array p in the expression of the physically interpreted *generalized forces* Q) by the use of matrix Y that exclusively consists of precisely known kinematical data. The Lyapunov function of this method is $V = r^T Hr + \tilde{p}^T \Gamma \tilde{p}$, with *positive definite symmetric matrix* Γ. *For guaranteeing asymptotic stability of the control negative derivative is needed for the Lyapunov function.* For this purpose the *skew symmetry* of the matrix C_{ij} and the parameter tuning rule $\dot{\hat{p}} = \Gamma^{-1} Y^T r$ are utilized. *This method has the following difficulties. For a complex dynamic system the array* p *may have many, say* m *elements, to which a matrix* Γ *of size* $m \times m$ *containing* $m + (m^2 - m)/2$ *independent, arbitrary control parameters belongs. These parameters cannot optimally be set, they are present in the tuning rule and make the process of tuning clumsy. Unknown perturbations and coupled, not modeled dynamics can deceive the tuning process. The control method itself requires the use of the complicated analytical expressions giving the arrays* $\hat{H}$*,* $\hat{C}$*, and* $\hat{g}$. The next example is a simple version of the traditional MRAC controllers.

1.2. A Possible MRAC Controller Designed by the Use of a Lyapunov Function

The here presented realization of the traditional MRAC design was studied in details in [16]. The essence of the idea of the *MRAC* is the transformation of the actual system under control into a well behaving reference system (reference model) for which simple controllers can be designed. In the practice the *reference model* used to be stable linear system of constant coefficients. To achieve this simple behavior normally special adaptive loops have to be developed. For comparison we choose a relatively simple implementation containing integrated feedback in the tracking error. Let the tracking error be denoted as $e := q^N - q$ and let $\xi(t) := \int_0^t e(\tau) d\tau$ (q^N denotes the *nominal*, q stands for the actual *generalized coordinates*). The kinematically prescribed trajectory tracking can be defined by the positive definite matrix Λ and the "error metrics" of the VS/SM controllers as $S := \left(\frac{d}{dt} + \Lambda\right)^3 \xi(t)$ that is required to be zero. If this prescription is realized the integrated tracking error has to vanish exponentially. (According to Barbalat's lemma [7] for uniformly continuous error e it also means that $e \to 0$. Uniform continuity is not an extreme restriction for smooth dynamical systems and uniformly continuous nominal trajectories.) The result is a "*desired 2nd derivative*" $\ddot{q}^D := \ddot{q}^N + \Lambda^3 \xi + 3\Lambda^2 e + 3\Lambda \dot{e}$. Let the reference model consist of two symmetric positive definite constant matrices M^{Ref} and B^{Ref} as $M^{Ref} \ddot{q}^D + B^{Ref} \dot{q} = Q^{Ref}$ where Q^{Ref} corresponds to the force/torque need for the reference model in the actual state of the system defined by the q, and $\dot{q}$ values. Let $H(q)\ddot{q} + h(q,\dot{q}) = Q$ be the actual system's equation of motion. By "copying" the idea of the *Adaptive Inverse Dynamics Controller* let the exerted force/torque be $M^{Ref} \ddot{q}^D + B^{Ref} \dot{q} + D = H(q)\ddot{q} + h(q,\dot{q}) = Q$ in which D corresponds to an additive force to be determined by the MRAC controller. Via subtracting $M^{Ref} \ddot{q}$ from both sides we can express the known difference of the desired and actual joint accelerations as $\ddot{q}^D - \ddot{q} = {M^{Ref}}^{-1} \left[\left(H(q) - M^{Ref} \right) \ddot{q} + h(q,\dot{q}) - B^{Ref} \dot{q} - D \right]$. By the introduction of the arrays $x := \left[\xi^T,e^T,\dot{e}^T\right]^T$ and $\dot{x} = \left[e^T,\dot{e}^T,\ddot{e}^T\right]^T$ the following equation of motion holds:

$$\dot{x} = Ax + \Phi \quad \texttt{with} \quad A := \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ -\Lambda^3 & -3\Lambda^2 & -3\Lambda \end{bmatrix} \tag{2}$$

and

$$\Phi := \left[\ 0^T,0^T,[M^{Ref^{-1}}\left(\left(H - M^{Ref}\right)\ddot{q} + h - B^{Ref}\dot{q} - D\right)]^T\ \right]^T. \tag{3}$$

in which it can also be written that $\Phi = [0^T,0^T,(\ddot{q}^D - \ddot{q})^T]^T$. With a positive definite matrix P the Lyapunov function $V := x^T P x$ can be introduced with the desired negative time derivative $\dot{V} = x^T\left(A^T P + PA\right)x + 2x^T P\Phi < 0$. By solving the Lyapunov equation $A^T P + PA = -R$ with a *prescribed positive definite symmetric matrix* R appropriate P can be obtained. So the term quadratic in x can be made negative. Therefore, it is sufficient to guarantee the non–positive nature of the remaining one, $x^T P\Phi$. Since $\ddot{q}^D - \ddot{q}$ is known (measurable) the remaining term consists of the sum of *known* and *unknown* parts as

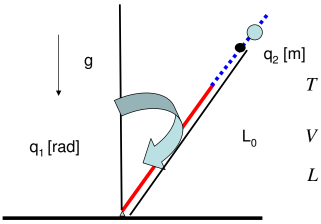
$$\begin{gathered} z_{meas} := x^T P\Phi = x^T P(\ddot{q}^D - \ddot{q}) = \\ x^T P\left[0^T;0^T;M^{Ref^{-1}}\right]\left\{\left(H - M^{Ref}\right)\ddot{q} + h - B^{Ref}\dot{q}\right\} - \\ -x^T P\left[0^T;0^T;M^{Ref^{-1}}\right]D < 0 \end{gathered} \tag{4}$$

in which the 2nd row contains the *unknown part* denoted by u in the sequel, while the *known part* in the 3rd row shortly can be denoted as a product as $w^T D$ with $w^T := x^T P\left[0^T;0^T;M^{Ref^{-1}}\right]$. Let us seek D in the form of $\alpha(t)w$! Then the condition $z_{meas} = u - \alpha(t)w^T w < 0$ should be met. Since $w^T w \geq 0$, in the possession of z_{meas} we have idea if $\alpha(t)$ must be increased or decreased. Let us apply a tuning rule with $\kappa > 0$ as follows: $\dot{\alpha} = \kappa z_{meas}$. With properly great κ and P this tuning can soon lead to decreasing Lyapunov function, i.e. to stable control. *This method has the following weak points: the additional control force D is directly varies with the controlled quantity α. This requires slow, cautious tuning. The tuning rule again contains the numerous arbitrary elements of P (inherited from matrix R) that are not optimally chosen.* Its advantage is that this controller (in contrast to that developed by Slotine & Li) already is able to compensate the effects of unknown external disturbances and coupled, not modeled dynamics

The use of the Lyapunov function technique was found to be avoidable by the application of "*Robust Fixed Point Transformations (RFPT)*" in the design of MRAC controllers not only for "*Single Input – Single Output (SISO)*" systems [17] but for "*Multiple Input – Multiple Output (MIMO)*" systems in [18]. This new adaptive approach in principle can compensate the effects of not modeled coupled dynamics and persistent external disturbances on which we can obtain information only by observing the motion of the controlled system.

In comparison with Lyapunov's "direct method" this novel approach has only one weak point: its stability is based on a convergent tuning process having only a local, bounded basin of attraction. Therefore, no global stability can be guaranteed by the new method, and its sensitivity to external disturbances and coupled dynamics can be studied via simulations. (However, if numerical details are also needed then the same numerical analysis must be applied in the approaches using some Lyapunov function, since it can guarantee global asymptotic stability without revealing easily interpretable numerical details.) Whenever the external disturbances and the coupled dynamic subsystems show strong time–varying nature this deficiency may be significant.

The Dynamic Model of the Paradigm



The Euler-Lagrange Equations of Motion

$$T=\frac{1}{2}\Theta\dot{q}_1^2+\frac{1}{2}m\left\{(L_0+q_2)^2\dot{q}_1^2+\dot{q}_2^2\right\}$$

$$V=[C+mg(L_0+q_2)]\cos q_1+\frac{1}{2}kq_2^2$$

$$L:=T-V$$

$$\frac{d}{dt}\frac{\partial L}{\partial\dot{q}_i}-\frac{\partial L}{\partial q_i}=Q_i$$

For determining the „actual" joint accelerations:

$$\begin{bmatrix}Q_1\\Q_2\end{bmatrix}=\begin{bmatrix}\Theta+m(L_0+q_2)^2 & 0\\ 0 & m\end{bmatrix}\begin{bmatrix}\ddot{q}_1\\\ddot{q}_2\end{bmatrix}+\begin{bmatrix}2m(L_0+q_2)\dot{q}_1\dot{q}_2-[C+mg(L_0+q_2)\sin q_1]\\-m(L_0+q_2)\dot{q}_1^2+kq_2+mg\cos q_1\end{bmatrix}$$

$$Q_2\equiv 0$$

Figure 1: The model of the pendulum with uncertain mass center point: the rotary axle q_1 is directly controlled while the mass m can freely move within the arm of the pendulum.

As a remedy, for the example of the adaptive control of an underactuated Classical Mechanical System with singular equations of motion an ancillary tuning strategy was introduced in [19] in which only one of the three adaptive parameters of the novel MRAC controller was adaptively tuned. The tuning strategy applied was quite heuristic, and it may have countless variants.

The aim of the present paper is to analyze various tuning possibilities for the novel MRAC control for two interesting, strongly nonlinear physical paradigms as follows: a) a pendulum of uncertain mass center point that seemingly behaves as a SISO system with a not modeled coupled internal degree of freedom the actual state of which drastically influences the behavior of the observed and controlled degree of freedom [Fig. 1]; b) the underactuated system already studied in [19] [Fig. 2].

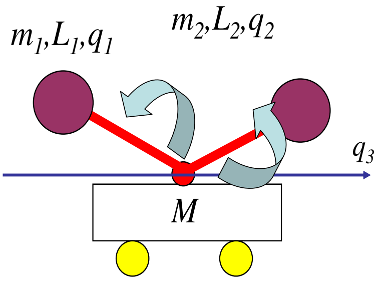


Figure 2: The structure of the underactuated system: a cart + double pendulum (q_3 does not have own drive, the motion along it can be controlled by directly controlling the motion of the counterweights m_1 and m_2 along their rotary axles q_1 and q_2).

The analysis happened by numerical computations by the use of the SCILAB-5.1.1 version and its SCICOS ver. 4.2 co-simulator packages. They can freely be used for research purposes. These software packages automatically select the appropriate numerical integration method depending on the observed stiffness of the problem under consideration. The paper is structured as follows. In Section 2 the controller and the stabilization tuning are briefly described. In Section 3 the simulation results are presented, and the paper is completed by the conclusions in Section 4.

2. The RFPT-based Adaptive Controller with Parameter Tuning

The basic idea of the novel MRAC controller is outlined in Fig. 3.

Assume that on purely kinematical basis we prescribe a trajectory tracking policy that needs a desired acceleration for certain axes as $\ddot{q}^D$. From the behavior of the reference model for that acceleration we can calculate the physical agent (force or torque) that could result in the response Q^D for the reference model. The direct application of this Q^D value for the actual system could result in different response since its physical behavior differs from that of the reference model. Therefore it can be "deformed" into a "*required*" Q^{Req} value that directly can be applied to the actual system. Via substituting the realized response of the actual system $\ddot{q}$ into the reference model the "realized control action" Q^R can be obtained instead of the "desired one" Q^D [Fig. 3]. Our aim is to find the proper deformation by the application of which Q^R well approaches Q^D while the prescribed nominal trajectory is precisely tracked, that is *at which the controlled system seems to behave as the reference system and the trajectory tracking is precise, too*. The proper deformation may be found by the application of an iteration as follows.

Consider the iteration generated by some function as $Q^{Req}_{n+1} = G(Q^{Req}_n, Q^R_n, Q^D_{n+1})$ in which n is the index of the control cycle. For slowly varying scenario Q^D can be considered to be constant. In this case the iteration is reduced to $Q^{Req}_{n+1} = G(Q^{Req}_n, Q^R_n, Q^D)$ that must be made convergent to some Q^{Req}_*. One possibility for that is the application of contractive maps in the arrays of real numbers that result in Cauchy Sequences that are convergent in a complete linear normed space. By using the norm–inequality, for a convergent iterative sequence $x_n \rightarrow x_*$ it is obtained that $Q^{Req}_* = G(Q^{Req}_*, Q^R(Q^{Req}_*), Q^D)$ since

$$\begin{aligned} ||G(x_*,Q^D) - x_*|| = ||G(x_*,Q^D) - x_n + x_n - x_*|| \leq \\ \leq ||G(x_*,Q^D) - x_n|| + ||x_n - x_*|| = \\ = ||G(x_*,Q^D) - G(x_{n-1},Q^D)|| + ||x_n - x_*||. \end{aligned} \tag{5}$$

It is evident from (5) that if G is continuous then the desired fixed point is found by this iteration because in the right hand side of (5) both terms converge to 0 as $x_n \rightarrow x_*$. The next question is giving the necessary or at least a *satisfactory condition* of this convergence. It also is evident that for this purpose contractivity of $G(\bullet)$, i.e. the property that $||G(a) - G(b)|| \leq K||a - b||$ with $0 \leq K < 1$ is satisfactory since it leads to a *Cauchy Sequence* ($||x_{n+L} - x_n|| \rightarrow 0 \; \forall L \in \mathbb{N}$):

$$\begin{aligned} ||x_{n+L} - x_n|| = ||G(x_{n+L-1}) - G(x_{n-1})|| \leq ... \\ \leq K^n||x_L - x_0|| \rightarrow 0 \;\; as \quad n \rightarrow \infty \end{aligned} \tag{6}$$

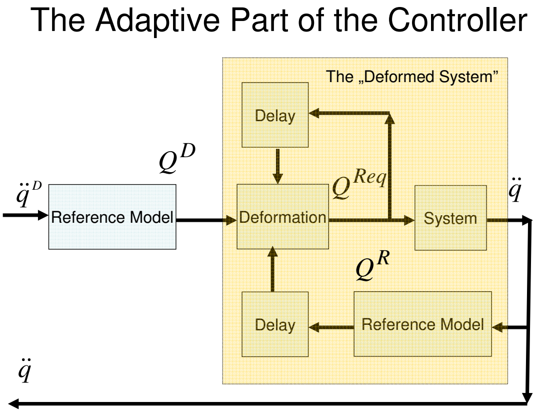


Figure 3: The block scheme of the "novel" MRAC controllers.

For the role of function $G(x,Q^D)$ a novel fixed point transformation was introduced for *Single Input – Single Output (SISO) Systems* that is rather "robust" as far as the dependence of the resulting function on the behavior of $f(\bullet)$ is concerned (7). This robustness can approximately be investigated by the use of an affine approximation of $f(x) := Q^R(x)$ in the vicinity of x_* and it is the consequence of the strong nonlinear saturation of the sigmoid function $\tanh(x)$:

$$\begin{gathered} G(x,Q^D) := (x+K) \times \left[1 + B\tanh(A[f(x) - Q^D])\right] - K \\ \texttt{if} f(x_*) = Q^D \;\texttt{then}\; G(x_*|Q^D) = x_*, \quad G(-K,Q^D) = -K, \\ G(x_*,Q^D)' := (\partial G/\partial x \;\texttt{at}\; x = x_*) = (x_* + K)ABf'(x_*) + 1. \end{gathered} \tag{7}$$

Equation (7) evidently has a proper (x_*) and a false ($-K$) fixed point, but by properly manipulating the control parameters A, B, and K the condition $|G'(x_*,Q^D)| < 1$ can be guaranteed and the good fixed point can be located within the basin of attraction of the procedure. This means that the iteration can have considerable speed of convergence even nearby x_*, and the strongly saturated $\tanh$ function can make it robust in its vicinity, that is the properties of $f(x)$ have not too much influence on the behavior of G. [It can be noted that instead of the $\tanh$ function any sigmoidal function with the property of $\sigma(0) = 0$, e.g. $\sigma(x) := x/(1 + |x|)$ can be similarly applied, too.]

The idea of keeping the iteration convergent by manipulating the width of the basin of attraction to the good fixed point can be developed in the following manner for SISO systems. On the basis of the available rough system model a simple PID controller can be simulated that reveals the order of magnitude of the occurring responses. Parameter K can be so chosen for which the $x+K$ values are considerable negative numbers. Depending on $sign(f')$ let $B = \pm 1$ and let $A > 0$ be a small number for which $|\partial G(x,Q^D)/\partial x| \approx 1 - \varepsilon_{goal}$ for a small $\varepsilon_{goal} > 0$. For Q^D varying in time the following estimation can be done in the vicinity of the fixed point when $|x_n - x_{n-1}|$ is small: $x_{n+1} - x_n = G(x_n,Q^D_n) - G(x_{n-1},Q^D_{n-1}) \approx \frac{\partial G(x_{n-1},Q^D_{n-1})}{\partial x}(x_n - x_{n-1}) + \frac{\partial G(x_{n-1},Q^D_{n-1})}{\partial Q^D}(Q^D_n - Q^D_{n-1})$. Since from the analytical form of $\sigma(x)$ the term $\frac{\partial G(x_{n-1},Q^D_{n-1})}{\partial Q^D}$ is known, and the past "desired" inputs as well as the arguments of function G are also known, this equation can be used for real–time estimation of $\frac{\partial G(x_{n-1},Q^D_{n-1})}{\partial x}$. The quantity ε_{goal} can be tried to be fixed around -0.5 by tuning parameter A for which various possibilities are available. For the $\sigma(x) := x/(1 + |x|)$ choice a "*moderate tuning strategy*" according to (8)

$$\begin{array}{l} d_n := x_n - x_{n-1}, \;\; d_{n-1} := x_{n-1} - x_{n-2} \;\; h_n := BA\sigma'\left(A(f_{n-1} - x^d_{n-1})\right) \\ d^d_n := Q^D_n - Q^D_{n-1} \;\; \varepsilon := \left(d_n + (x_{n-1} + K)h_n d^d_n\right)/d_{n-1} - 1 \\ \texttt{if}\; \varepsilon - \varepsilon_{goal} < 0 \;\texttt{then}\quad \dot{A} = \kappa_1\alpha_1\left(\sigma(\varepsilon - \varepsilon_{goal}) + \kappa_2 sgn(\varepsilon - \varepsilon_{goal})\right) \\ \texttt{else}\; \dot{A} = \alpha_1\left(\sigma(\varepsilon - \varepsilon_{goal}) + \kappa_2 sgn(\varepsilon - \varepsilon_{goal})\right) \end{array} \tag{8}$$

with $\kappa_1 > 1$, $\kappa_2,\alpha_1 > 0$ parameters [$\sigma'(x)$ refers to the derivative of the function $\sigma(x)$], and a more drastic "*exponential tuning*" according to (9)

$$\texttt{if } \varepsilon - \varepsilon_{goal} < 0 \texttt{ then } \dot{A} = -\kappa_3\alpha_2 A \texttt{ else } \dot{A} = \alpha_2 A \tag{9}$$

with $\kappa_3 > 1$, $\alpha_2 > 0$ parameters can be prescribed. In (8) κ_1 and in (9) κ_3 corresponds to faster decrease than increase in A since it was observed that decreasing A introduces little fluctuations in the consecutive x_n values in a discrete approach, therefore it is more expedient to quickly step over this fluctuating session by fast decrease in A. In the next section simulation results will be presented for the application of these tuning strategies.

3. Simulation Results

In the simulations the maximum allowed discrete time-step of integration was set to $10^{-4}s$, while the cycle–time of the controller was assumed to be $1ms$. The “digital nature” of the controller was modeled by the use of event clocks and sample holders. The nominal trajectory was a 3rd order periodic spline function of time.

At first the control of the underactuated system in Fig. 2 is considered in which the horizontal position of the arms of the counterweights is critical since their motion does not generate horizontal force components. Therefore one of the weights were kept at vertical position while the other weight’s reaction force was used for motion control. Whenever the active weight approached the critical position it was directed back to the horizontal direction and the other one’s reaction force was used for active control. The here applied solution used a finer combination of the reaction forces of the counterweights than that in [19]. The *actual numerical values* were: $M = 20\,kg$, $m_1 = 8\,kg$, $m_2 = 8\,kg$, $L_1 = L_2 = 2\,m$, $g = 9.81\,m/s^2$. The reference model had the parameters as $\hat{M} = 18\,kg$, $\hat{m}_1 = 4\,kg$, $\hat{m}_2 = 6\,kg$, $\hat{L}_1 = \hat{L}_2 = 2\,m$, $\hat{g} = g$. In the forthcoming figures the following “*naming conventions*” are used: the “*nonadaptive*” version uses the approximate dynamic model with a simple kinematically designed PID controller; the “*classic*” version uses the rule defined in (8) with $\kappa_1 = 2$, $\alpha_1 = 2.5/s$, and $\kappa_2 = 0$; the “*speedy*” version uses the rule defined in (8) with $\kappa_1 = 2$, $\alpha_1 = 2.5/s$, and $\kappa_2 = 0.1$ (that also yields some finite tuning speed in the close vicinity of zero error; and the “*exponential*” version corresponds to (9) with $\alpha_2 = 2.5/s$ and $\kappa_3 = 3$.

Figure 4 describes the trajectory tracking properties of the controllers. The tracking errors of the nonadaptive solution reveal the significance of the modeling errors that are finely reduced by the adaptive solutions. The same holds for the phase trajectory tracking and acceleration tracking given in Figs. 5 and 6.

The realization of the “*MRAC principle*” is well revealed by Fig. 7 and in details by Fig. 8. In the non-adaptive case (and in the first $0.1\,s$ long phase of the adaptive ones before switching on adaptivity) $Q^D \equiv Q^{Req}$, that is the desired values are exerted to the system without any deformation. Accordingly the lines of the “exerted (required)” components as Q_1^{Req}=(longdash dot: — . — . —), Q_2^{Req}=(bigdash dot: — .. — .. —) cannot be seen in the non-adaptive phase: they are exactly covered by the “desired” Q_1^D=(solid: —), Q_2^D=(bigdash longdash: — - — -) curves. After reaching the adaptive phase the “pairs”

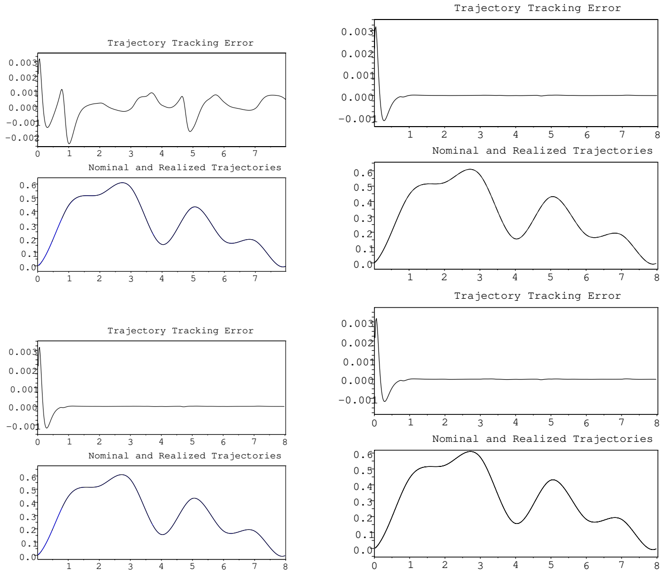


Figure 4: Trajectory tracking of the underactuated system ($q_3\,m$ vs. time in s units) of the "novel" MRAC controller with various tuning methods: "*nonadaptive*": upper left, "*classic*": upper right, "*speedy*": lower left, "*exponential*": lower right charts.

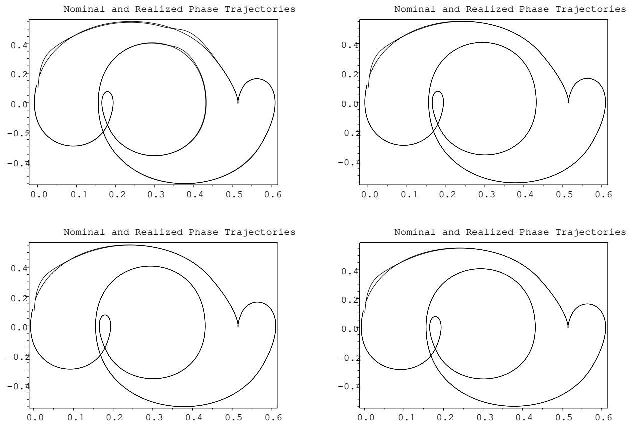


Figure 5: Phase trajectory of the underactuated system ($\dot{q}_3\,m/s$ vs. $q_3\,m$) tracking of the "novel" MRAC controller with various tuning methods: "*nonadaptive*": upper left, "*classic*": upper right, "*speedy*": lower left, "*exponential*": lower right charts.

are reorganized: the $Q^D \approx Q^R$ situation is finely approached (the "recalculated" values well approximate the "desired" ones). Accordingly the Q_1^D=(solid: —) curve drains into itself the recalculated Q_1^R=(dash: — — —) curve, as well as the Q_2^D=(bigdash longdash: — - — -) curve attracts the Q_2^R=(dash dot: - - -) line, while the seriously deformed "exerted" curves Q_1^{Req}=(longdash dot: — . — . —) and Q_2^{Req}=(bigdash dot: — .. — .. —) remain "single".

Figure 9 reveals that the parameter tuning is more or less consistent (no exact setting is necessary for A), and the names of the tuning methods well mirror their properties.

The other system under consideration, i.e. the pendulum of Fig. 1 shows different kind of problem with its dynamically coupled subsystem not known by the controller. In its control higher dynamic ranges have to be covered for which a kind of noise–filtering technique would be useful. (While nearby the fixed point (7) is not very sensitive to these noises, the tuning rules (8) and (9) that contain different past values evidently may be noise–sensitive.) In the case of a digital controller these noises can be modeled by adding random disturbance terms to the simulated/observed second time–derivatives. Any linear noise filter can be modeled by an integral or a sum in the discrete approximation as

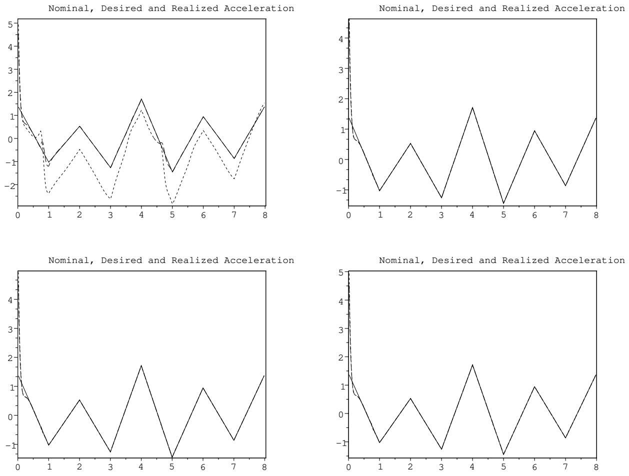


Figure 6: Acceleration tracking of the underactuated system ($\ddot{q}_3\ m/s^2$ vs. time in s units) of the "novel" MRAC controller with various tuning methods: "*nonadaptive*": upper left, "*classic*": upper right, "*speedy*": lower left, "*exponential*": lower right charts; {line attributes: nominal $\ddot{q}_3^N$=(solid: —), desired $\ddot{q}_3^D$=(dash: — — —), realized $\ddot{q}_3$=(dash dot: - - -) }.

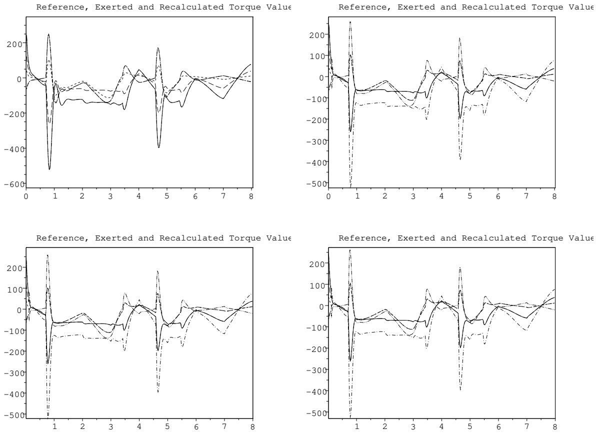


Figure 7: The torque components of the underactuated system (in Nm unit) vs. time (in s unit) of the "novel" MRAC controller with various tuning methods: "*nonadaptive*": upper left, "*classic*": upper right, "*speedy*": lower left, "*exponential*": lower right charts; $\{$line attributes: "*desired torque components*" Q_1^D=(solid: —), Q_2^D=(bigdash longdash: — - — -); "*recalculated torque components*" Q_1^R=(dash: — — —), Q_2^R=(dash dot: - - -); "*exerted (required) torque components*" Q_1^{Req}=(longdash dot: — . — . —), Q_2^{Req}=(bigdash dot: — .. — .. —)$\}$.

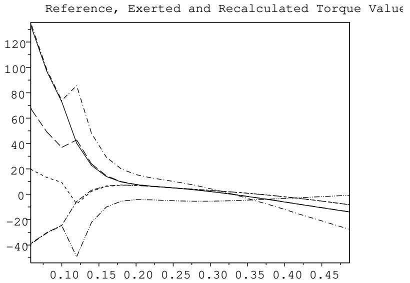


Figure 8: Fine details of the torque components of the underactuated system (in Nm unit) vs. time (in s unit) of the "novel" MRAC controller for the "*classic*" tuning method $\{$line attributes: "*desired torque components*" Q_1^D=(solid: —), Q_2^D=(bigdash longdash: — - — -); "*recalculated torque components*" Q_1^R=(dash: — — —), Q_2^R=(dash dot: - - -); "*exerted (required) torque components*" Q_1^{Req}=(longdash dot: — . — . —), Q_2^{Req}=(bigdash dot: — .. — .. —)$\}$.

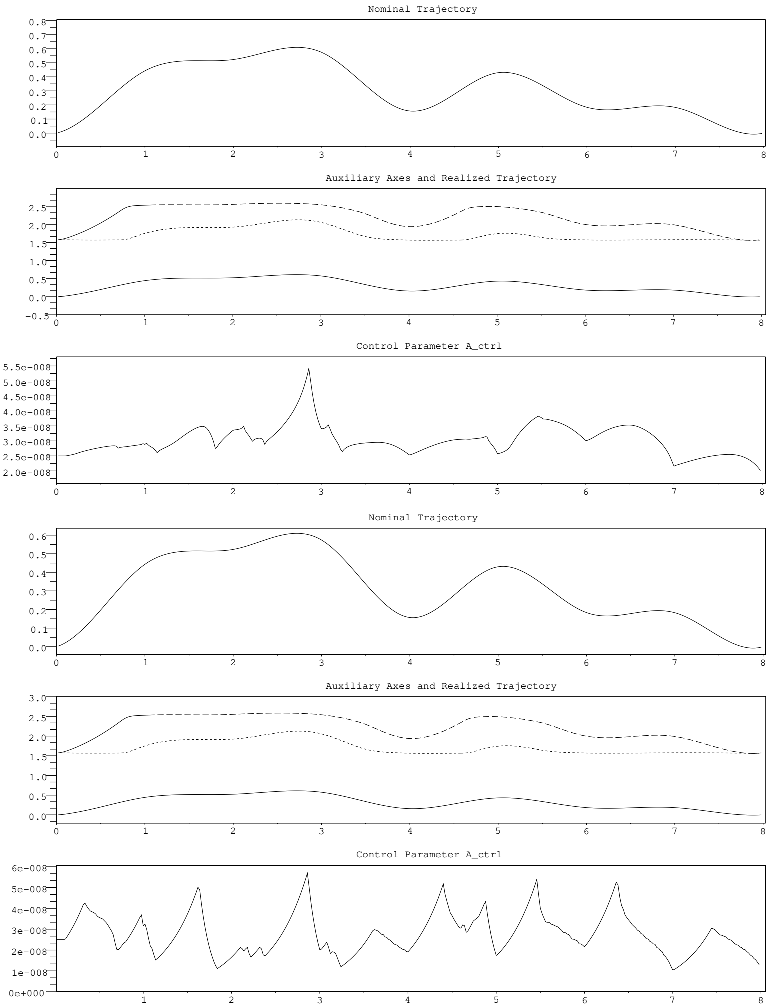


Figure 9: The motion of the axles of the underactuated system and the tuned parameter vs. time (in s unit) of the “novel” MRAC controller with two typical tuning methods: the “*classic*”, and the ‘*exponential*”tuning {line attributes: nominal trajectory=(solid: — in the upper segments), realized trajectory=(solid: —), auxiliary trajectories: (dash: — — —) and (dash dot: - - -) in the middle segments, the control parameter: (solid: —) line in the lower segments of the graphs.

$$\tilde{f}(t) := \int_0^\infty F(\tau)f(t-\tau)d\tau \;\; \mathtt{or} \;\; \tilde{f}_k := \sum_{i=0}^\infty F_i f_{k-i} \tag{10}$$

with some monotone decreasing function $F(\tau)$ or discrete weights F_k that normally converge to zero as $\tau, k \to \infty$. Normally some weighted average can be calculated for a period corresponding to shorter or longer "memory". In the case of a discrete controller the smallest memory is needed for the very simple solution $F_k := \beta^k(1-\beta)$ with $0 < \beta < 1$ that can be calculated by a single buffer P according to the updating rule $P_{n+1} = \beta P_n + f_{k+1}$, $\tilde{f}_{k+1} = (1-\beta)P_{k+1}$. The actual value of β directly influences the "memory" of the system: larger value corresponds to longer memory than a smaller one. In the forthcoming simulation results the $\beta = 0.85$ value was chosen.

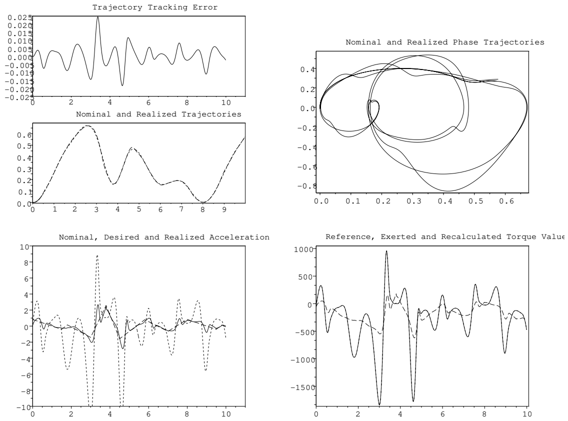


Figure 10: Some results belonging to the nonadaptive control of the pendulum {line attributes: tracking error=(solid: —), nominal motion q_1^N=(dash: — — —), realized motion q_2=(dash dot: - - -); realized acceleration $\ddot{q}_1$=(solid: —), desired acceleration $\ddot{q}_1^D$=(dash dot: - - -), nominal acceleration $\ddot{q}_1^N$=(dash: — — —) line; desired torque from the reference model Q^D=(solid: —), the deformed exerted (required) torque Q^{Req}=(dash dot: - - -) (exactly covering each other), and the torque recalculated from the reference model Q^R=(dash: — — —)}.

Figure 10 describes the operation of the simple PID controller without adaptive deformation. It can well be seen that the difference between the "exact model" and "reference model" effects significant tracking error. (According to Fig. 1 $\Theta = 30\,kg \cdot m^2$, $C = 50\,kg \cdot m^2$, $m = 50\,kg$, $k = 1000\,N/m$, $g = 9.81\,m/s^2$, $\mu = 0.1\,N/(m/s)$ –a viscous friction term not listed in the figure–, and $L_0 = 2\,m$, and $\hat{\Theta} = 50\,kg \cdot m^2$, $\hat{C} = 70\,kg \cdot m^2$, $\hat{m} = 20\,kg$, $\hat{k} = 100\,N/m$, $\hat{g} = 10\,m/s^2$, $\hat{\mu} = 0.01\,N/(m/s)$ and $\hat{L}_0 = L_0 = 2\,m$.)

Figure 11 corresponds to the tuning defined in (8) with $\kappa_1 = 2.5$, $\alpha_1 = 10/s$, and $\kappa_2 = 1$. Its drastic improvement in the trajectory tracking and the realization of the MRAC idea is evident.

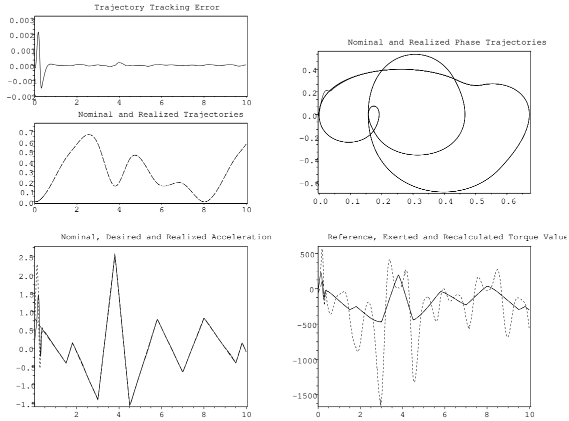


Figure 11: Some results belonging to the adaptive control of the pendulum with *“moderate tuning”* {line attributes: tracking error=(solid: —), nominal motion q_1^N=(dash: — — —), realized motion q_2=(dash dot: - - -); realized acceleration $\ddot{q}_1$=(solid: —), desired acceleration $\ddot{q}_1^D$=(dash dot: - - -), nominal acceleration $\ddot{q}_1^N$=(dash: — — —) line; desired torque from the reference model Q^D=(solid: —), the deformed exerted (required) torque Q^{Req}=(dash dot: - - -) (now quite different lines), and the torque recalculated from the reference model Q^R=(dash: — — —) (now almost exactly covering the line of Q^D)}.

Figure 12 belongs to the tuning method described in (9) with $\alpha_1 = 10/s$ and $\kappa_3 = 2$. To same holds for this result as to that described in Fig. 11.

Figure 13 reveals the variation of the tuned parameter in the case of the moderate and fast tuning for which we got comparable results again. It is important to note that the tuned parameter A varies in a great dynamic range.

It is interesting to investigate what happens in the case of the pendulum if A is set to be constant. The simulation results in Fig. 14 reveal that the active tuning really can stabilize the controller. It can be seen that at several regions strong fluctuations occur that cause considerable tracking errors. An efficient tuning can solve the problem that can be caused by an improperly chosen initial value for A. Since the proposed control method deals with observing the system's response to the excitation by the controller that depends on the properties of the nominal trajectories and external disturbances this tuning ability is a significant value.

4. Concluding Remarks

In this paper the operating principle of a novel version of the Model Reference Adaptive Controllers was briefly outlined in formal comparison with two typical representatives of the "traditional adaptive approaches". The advantage of the new approach is that for design purposes it does not require the use of Lyapunov's ingenious "direct method" that normally needs very good mathematical skills and invention on the part of the designer. In contrast to the numerous arbitrary control parameters of the traditional approaches it applies only three parameters that can easily be set via simple simulations. The deficiency of the method is that normally it cannot guarantee global stability. It works with a local basin of attraction that can be left by the controlled system's response. To relax/amend this deficiency a simple tuning procedure was suggested for continuously modifying only one of the three adaptive parameters of this controller to keep its operation nearby the center of the basin of its convergence. For such a tuning various open possibilities are available.

In the present paper various possible tuning methods were investigated on two interesting dynamical systems. One of them was an underactuated cart and double pendulum system, the other one contained a "hidden" internal degree of freedom dynamically coupled to that under direct observation and control.

The results of the above investigations reinforced the expectation that the novel variant of the Model Reference Adaptive Controllers designed by the use of Robust Fixed Point Transformations could successfully be stabilized by tuning only one of the three control

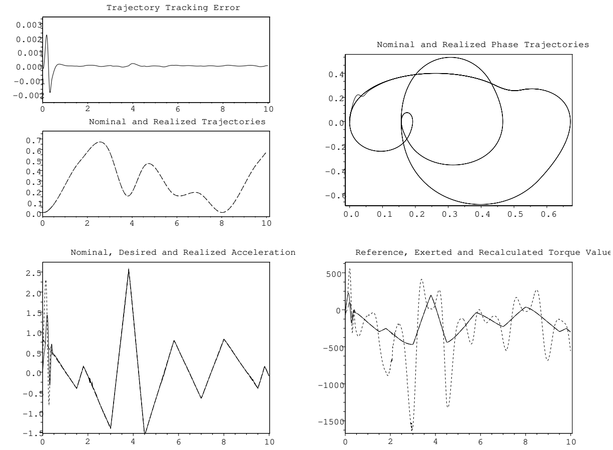


Figure 12: Some results belonging to the adaptive control of the pendulum with *"fast tuning"* {line attributes: tracking error=(solid: —), nominal motion q_1^N=(solid: —), realized motion q_2=(dash dot: - - -); realized acceleration $\ddot{q}_1$=(solid: —), desired acceleration $\ddot{q}_1^D$=(dash dot: - - -), nominal acceleration $\ddot{q}_1^N$=(dash: — — —); desired torque from the reference model Q^D=(solid: —), the deformed exerted (required) torque Q^{Req}=(dash dot: - - -), and the torque recalculated from the reference model Q^R=(dash: — — —) (almost exactly covering the line of Q^D)}.

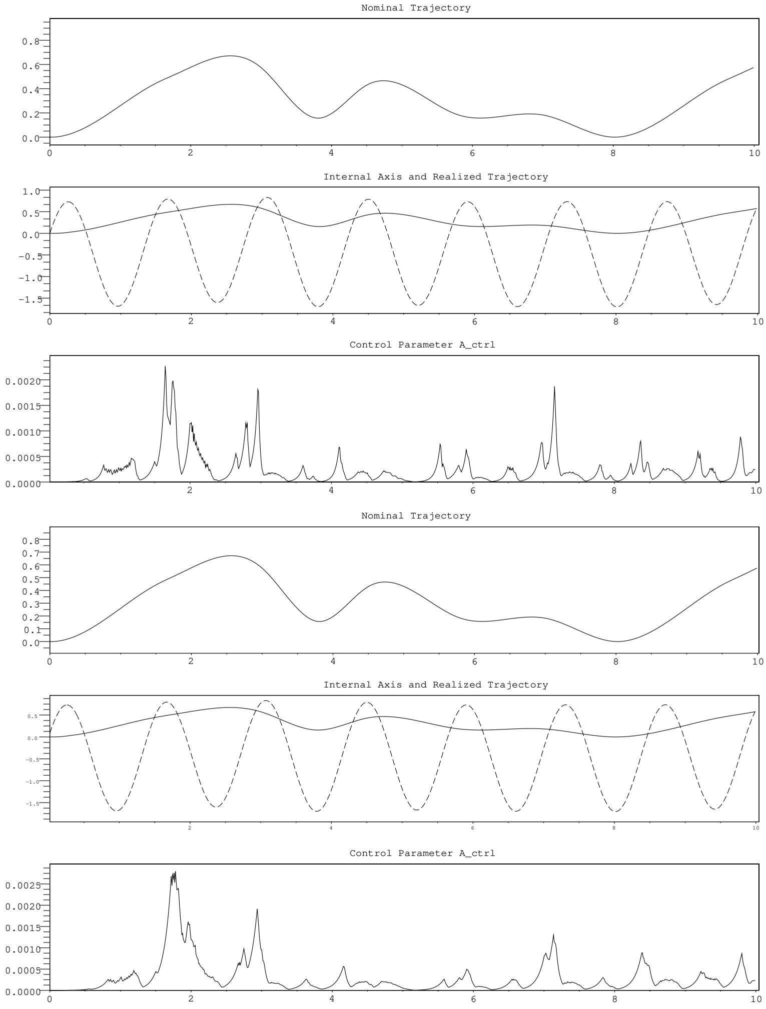


Figure 13: Some results belonging to the adaptive control of the pendulum with *"moderate tuning"* and *"fast tuning"* {line attributes: nominal trajectory: q_1^N=(solid: —), realized trajectory: q_1=(solid: —), motion of the coupled 2^{nd} axis q_2=(dash: — — —), the control parameter A=(solid: —) lines}.

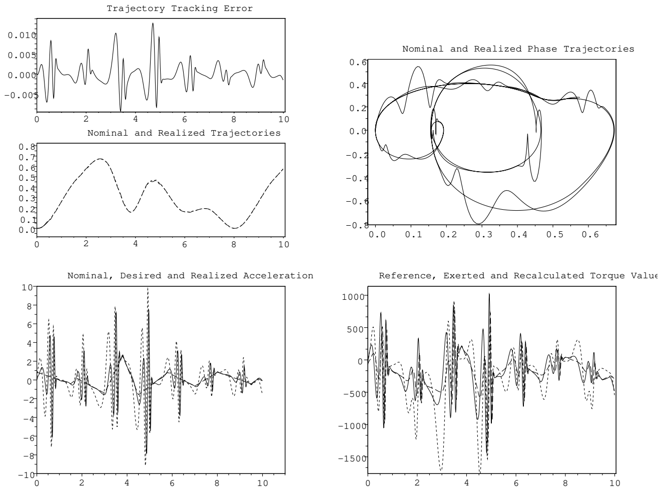


Figure 14: Some results belonging to the adaptive control of the pendulum without tuning ($A = 10^{-6} = const.$) $\{$line attributes: tracking error=(solid: —), nominal motion q_1^N=(dash: — — —), realized motion q_2=(dash dot: - - -); realized acceleration $\ddot{q}_1$=(solid: —), desired acceleration $\ddot{q}_1^D$=(dash dot: - - -), nominal acceleration $\ddot{q}_1^N$=(dash: — — —) line; desired torque from the reference model Q^D=(solid: —), the deformed exerted (required) torque Q^{Req}=(dash dot: - - -), and the torque recalculated from the reference model Q^R=(dash: — — —)$\}$.

parameters. In certain cases no any tuning is needed, in other cases fast tuning is required for stabilization.

In the next step of the research modeling and investigation of various noise components would be expedient that may concern the operation of the parameter tuning applied.

Acknowledgement

The authors gratefully acknowledge the support by the National Office for Research and Technology (NKTH) using the resources of the Research and Technology Innovation Fund within the project OTKA No. CNK–78168.

References

[1] Főző, L., Andoga, R., Madarász, L.: *Mathematical Model of a Small Turbojet Engine MPM–20*, In: Studies in Computational Intelligence., 2010 Springer-Verlag Berlin Heidelberg, Vol. 313, pp. 313–322, 2010.

[2] Madarász, L., Andoga, R., Főző, L.: *Intelligent Technologies in Modelling and Control of Turbojet Engines*, In: New Trends in Technologies: Control, Management, Computational Intelligence and Network Systems., Meng Joo Er (ed.), SCIYO, pp. 17–38, 2010.

[3] Madarász, L., Andoga, R., Főző, L., Lazar, T.: *Situational control, modeling and diagnostics of large scale systems*, In: Towards Intelligent Engineering and Information Technology., 2009 Springer-Verlag Berlin Heidelberg, No. 143, pp. 153–164, 2009.

[4] `http://web.mit.edu/nsl/www/` (in Janaury 2011)

[5] `http://www.usnews.com/articles/education/worlds-best-universities/` (in January 2011)

[6] Asada, H., and Slotine, J.-J.E., *Robot Analysis and Control*, John Wiley & Sons, New York, 1986.

[7] Slotine, J.-J.E., Li, W.: *Applied Nonlinear Control*, Prentice–Hall International, Inc., Englewood Cliffs, New Jersey, 1991.

[8] Isermann, R., Lachmann, K.H., and Matko, D.: *Adaptive Control Systems*, New York DC, Prentice–Hall, USA, 1992.

[9] Nguyen, C.C., Antrazi, Sami S., Zhou, Zhen-Lei, Campbell, Charles E. Jr: *Adaptive control of a stewart platform-based manipulator*, Journal of Robotic Systems, volume 10, no. 5, pp. 657-687, 1993.

[10] Somló, J., Lantos, B., Cát, P.T.: *Advanced Robot Control*, Akadémiai Kiadó, Budapest, Hungary, 2002, p. 268.

[11] Hosseini-Suny, K., Momeni, H. and Janabi–Sharifi, F.: *Model Reference Adaptive Control Design for a Teleoperation System with Output Prediction*, J Intell Robot Syst, DOI 10.1007/s10846-010-9400-4, pp. 1–21, 2010.

[12] Lyapunov, A. M.: *A general task about the stability of motion*(in Russian), PhD Thesis, 1892.

[13] Lyapunov, A. M.: *Stability of motion*, Academic Press, New-York and London, 1966.

[14] Mirkin, B., Gutman, Per-Olof, Shtessel Y: *Robust adaptive tracking with an additional plant identifier for a class of nonlinear systems*, Journal of the Franklin Institute, vol. 347 pp. 974–987, 2010.

[15] Tar, J.K., Rudas, I.J., Gáti, J.: *Improvements of the Adaptive Slotine & Li Controller - Comparative Analysis with Solutions Using Local Robust Fixed Point Transformations*, Proc. of the 14th WSEAS International Conference on APPLIED MATHEMATICS (MATH'09), Puerto De La Cruz, Canary Islands, Spain, December 14–16, 2009, pp. 305–311, 2009.

[16] Tar, J.K., Bitó, J.F., Rudas, I.J., Eredics, K: *Comparative Analysis of a Traditional and a Novel Approach to Model Reference Adaptive Control*, Proc. of the 11th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, November 18-20, 2010, Budapest, Hungary, Óbuda University, pp. 93–98, 2010.

[17] Tar, J.K., Bitó, J.F., Rudas, I.J.: *Replacement of Lyapunov's Direct Method in Model Reference Adaptive Control with Robust Fixed Point Transformations*, Proc. of the 14th IEEE International Conference on Intelligent Engineering Systems 2010, May 5–7, 2010, Las Palmas of Gran Canaria, Spain, pp. 231–235, 2010.

[18] Tar, J.K., Rudas, I.J., Bitó, J.F., Kozłowski, K.R., and Pozna, C.: *A Novel Approach to the Model Reference Adaptive Control of MIMO Systems*, Proc. of the IEEE 2010 Robotics in Alpe–Adria–Danube Region (RAAD 2010) Conference, June 23–25 2010, Budapest, Hungary, 2010.

[19] Tar, J.K., Rudas, I.J., Bitó, Preitl, S., Precup, R.-E.: *Convergence stabilization by parameter tuning in Robust Fixed Point Transformation based adaptive control of underactuated MIMO systems*, in the Proc. of the International Joint Conference on Computational Cybernetics and Technical Informatics (ICCC-CONTI), 2010, 27–29 May 2010, Timişoara, Romania, pp. 407–412, 2010.

Complexity Comparisons in Logic Testing

J. Sziray

Department of Informatics, Széchenyi University
Egyetem tér 1, H-9026 Győr, Hungary
E-mail: sziray@sze.hu

Abstract: The paper is concerned with analyzing and comparing two exact algorithms from the viewpoint of computational complexity. Both serve for calculating fault-detection tests of digital circuits. The first one is the so-called composite justification, and the second is the D-algorithm. The analysis will be performed on combinational logic networks at the gate level. Here single and multiple stuck-at logic faults will be considered. As a result, it is pointed out that the composite justification requires significantly less computational step than the D-algorithm and its modifications. From this fact it has been conjectured that possibly no other algorithm is available in this field with fewer computational steps. If the claim holds, then it follows directly that the test-calculation problem is of exponential time, and so are all the other NP-complete problems. It may also be expected that the minimal complexity of composite justification applies to any modeling level (either low or high) of digital circuits, just like the exponential-time solution.

Keywords: Computational complexity, test-pattern calculation, logic networks, multi-valued logic, NP-complete problems.

1. Introduction

As known, all computational problems can be classified into two categories: those that can be solved by algorithms, and those that cannot. The first category incorporates problems for which there exists a Turing machine that is guaranteed to halt on all possible inputs, and all such machines will be called algorithms. This principle is known as the Church-Turing thesis [1]. The second category includes problems known to be undecidable, i.e., for which there exists no Turing machine that is guaranteed to halt ever on all possible inputs.

With the enormous advances in computer technology of the past decades (see, e.g., the Moore's law), one may expect that each problem in the first category can today be solved in a satisfactory way. Unfortunately however, computing practice reveals that many problems, although in principle solvable, cannot be solved in any practical sense by computers, due to excessive time requirements [1]–[3].

There are a great number of such problems, and most of them have real practical importance. The common feature of them is that they all are NP-complete. (NP: nondeterministic polynomial.) As known, these problems have a computational complexity for which probably there exists no upper bound by a finite-degree polynomial of the problem size. It means actually that the number of computational steps is finite, but absolutely unpredictable [1]–[3].

Let the problem size be denoted by an integer ***n***. Now we can establish a polynomial of the finite-degree integer ***k***, as follows:

$$P_k(n) = a_k \cdot n^k + a_{k-1} \cdot n^{k-1} + ... + a_1 \cdot n + a_0 ,$$

where $k \geq 1$, and $a_i \geq 0$ is an integer for $i = 0, 1, 2, ... , k–1$, whereas $a_k \geq 1$ is also an integer value.

The computation of solutions for a problem is said to be polynomially bounded if and only if there exists an algorithm which has no more steps than $P_k(n)$, for each possible instance of the problem. In terms of time spent by a computer, this notion can also be expressed as polynomial-time algorithm. Unfortunately, no one has found such an algorithm for any NP-complete task so far, and it is widely believed that there is no algorithm of this category. Instead, it is expected that any algorithm solving the problem has a computational complexity of an exponential nature.

As a matter of fact, all the exact solutions found so far for NP-complete problems are really of exponential complexity. It means that the problem size ***n*** determines the number of algorithmic steps ***S(n)*** in the following way:

$$S(n) = a \cdot b^n,$$

where $a \geq 1$, $b \geq 1$ are constant integers.

At present, the question whether there exists a polynomial solution to NP-complete problems is undecided. The proven answer in this case is one of the most intriguing demands in the field of computation theory. This is all the more important, since if it were a polynomial algorithm to any individual NP-complete problem, then it could be transformed to all other NP-complete problems, as a solution, while preserving its polynomial property [1]–[3].

One of these important problems is the generation of fault-detection tests for digital circuits [4]–[7]. Here the problem size is an integer number, and can be expressed, among others, by the number of gates or transistors in a circuit. In our approach, the number of gates will be used, and denoted by ***n***.

This paper is meant for analyzing and comparing two concrete algorithms from the viewpoint of computational complexity. Both are for calculating fault-detection tests for digital circuits. The first one is the so-called ***composite justification*** [8]–[10], and the second is the ***D-algorithm*** [6], [7] developed by Roth. The analysis will be performed on combinational logic networks at the gate level. Here single and multiple stuck-at logic faults will be considered at the primary inputs and the gate outputs.

2. Treatment of single faults

Let the vector of primary input and output variables for a general logic network be $\overline{x} = (x_1, x_2, ..., x_n)$ and $\overline{z} = (z_1, z_2, ..., z_m)$, respectively. Let the set of possible logic values in the network be $V = \{v_1, v_2,..., v_s\}$. In addition to the elements of V, the indifferent or don't care value ***d*** will be applied.

In case of **composite justification** the computations are carried out simultaneously in the normal and faulty network, i.e., in the normal and the faulty domain. Logic values simultaneously representing signal values in both the normal and the faulty networks are called **composite values.** Line justification performed in terms of composite values is referred to as **composite justification** [8], [9]. The two components of a composite value will be separated by a slash, with the normal component preceding the faulty one. The actual logic value of the ***i***-th line in the network will be denoted by ***v(i)***. Then, for example, a composite value of line ***i*** is

$$v(i) = v_a / v_b$$

Let the set of lines with stuck faults be denoted by ***SL***, where the number of lines belonging to *SL* is q. If the stuck value at line *i* of *SL* is *s(i)*, then the initial set of the logic values that are to be justified will be as follows:

$z_j = \alpha / \beta$ for a selected primary output, where

$$\alpha \in V, \quad \beta \in V, \text{ and } \alpha \neq \beta, \text{ and}$$

$$v(i) = d / s(i) \quad \text{for each} \quad i \in SL.$$

In the justification process, the value ***d*** need not be justified, whereas the stuck values *s(i)* must not be justified. The reason for *d* is obvious, while the stuck values are self-consistent.

Below we introduce four more notations:

Let the **stuck-at-α** fault of the **i-th line** be denoted by **i(α)**. The apostrophe sign will stand for logic inversion. The network path containing the series of lines i, j, …, p will be denoted by

P(i – j – … – p).

Network lines that carry signals from the faulty lines to z_j. are referred to as **potentially active lines** or **PAL's**. Here **PAL(i – j)** denotes the set of potentially active lines between lines ***i*** and ***j***.

As an example, consider the combinational network shown in Figure 1, where a test is to be calculated for fault **7(1)**. In this case

PAL(7 – 16) = {7, 9, 10, 11, 12, 14, 16}.

The potentially active lines are marked heavy in the schematic. The figure indicates the logic values assigned to the lines in the network upon completion of the calculation process.

When starting, at first z_j is set to 0 / 1, and whenever a contradiction occurs at the fault site, we interchange the components of each composite value, and proceed with the calculation in the same way as before [8].

The obtained test consists of the 0 or 1 components of the final composite values at the primary input lines:

$$\overline{x}_t = (d, 0, 0, d, d, 0).$$

As it can be seen, the test simultaneously sensitizes paths P(7-10-11-14-16) and P(7-10-12-16), while it leaves path P(7-9-14-16) necessarily unsensitized. The sensitized paths propagate the faulty signals towards the primary output z_1.

At the end of the calculations, those lines which actually carry the fault signal to z_j will be the real **active lines**. These lines have the so-called **active composite values** which are equivalent with the fault signals [8].

In the following an evaluation of the composite justification algorithm is given, in comparison with the D-algorithm. From a computational point of view, it is sufficient to take into account only one primary output of fault detection, say z_j.

In the event of a single fault i(α) the D-algorithm attempts to simultaneously sensitize a combination of paths leading from *i* to z_j. If successful, line justification is performed for the initially established set of sensitizing logic values. In the worst case each possible combination of paths must be processed in the same way until a consistent a test pattern is found.

If the number of different single paths leading from i(α) to z_j is ***p***, then the D-algorithm is to be performed $2^p - 1$ times in worst case. This number is $2^3 - 1 = 7$ for the network in Figure 1. As opposed to this, composite justification has only **one single** computational process, independently of the number of paths.

It has been pointed out by Muth in [11] that the use of Roth's five-valued logic (i. e., 0, 1, D, D', and d) in the D-propagation phase means overspecification which may lead to a large amount of wasteful computation. The proposed modification in [11] was to use nine-valued logic in the D-algorithm, instead of the original five-valued. This logic is based upon composite values. In worst case the modified D-algorithm attempts to propagate the faulty signal via each possible single path from *i* to z_j, and performs line-value justification in each case. Thus, the amount of computation in both the original and the modified D-algorithm is greatly influenced by the number of paths which may occur due to reconvergence. In our example in Figure 1, this number is **three**.

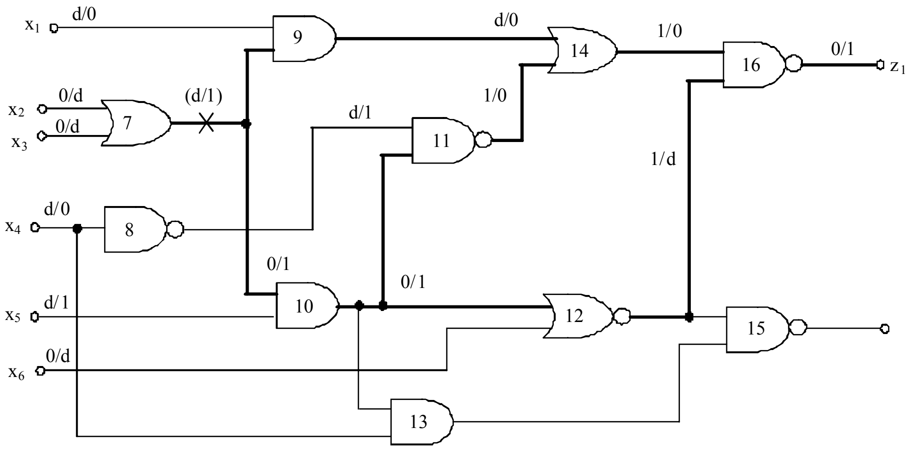


Figure 1. Composite justification for a single stuck-at fault

In contrast with these solutions, the composite justification algorithm leaves out the path sensitization phase along with the successive attempts to justify the different fault-propagation possibilities. Instead, it performs only **one justification process**, thus finding a test as a direct result. Moreover, this solution is affected by the presence of reconvergent paths to a much lesser extent than even the modified D-algorithm. The number of paths between *i* and z_j affects the amount of potentially active composite values used during justification. These are the values on the potentially active lines. For lines that do not belong to these paths, the determined logic values (i.e., 0 and 1) must not differ in the two domains. They are called **passive lines** in [8]. This fact results in a considerable reduction in the amount of possible decisions and backtracks.

3. Treatment of multiple faults

In the next example a test is to be found for the multiple fault 5(1), 6(0), and 9(1) as shown in Figure 2. The PAL's are also marked heavy in the schematic. When applying composite justification, the final result shows that the test vector$\overline{x}_t$ = (0, 1, 1, 0) propagates fault 6(0) along P(6-8-9-10) and fault 9(1) along P(9-10) simultaneously, whereas fault 5(1) remains blocked from propagation.

Figure 3 presents another solution for this same task. Here the obtained test vector $\overline{x}_t$ = (1, 0, d, d) propagates only fault 5(1) along P(5-7-10) to the primary output. The other two faults will not be propagated by the test.

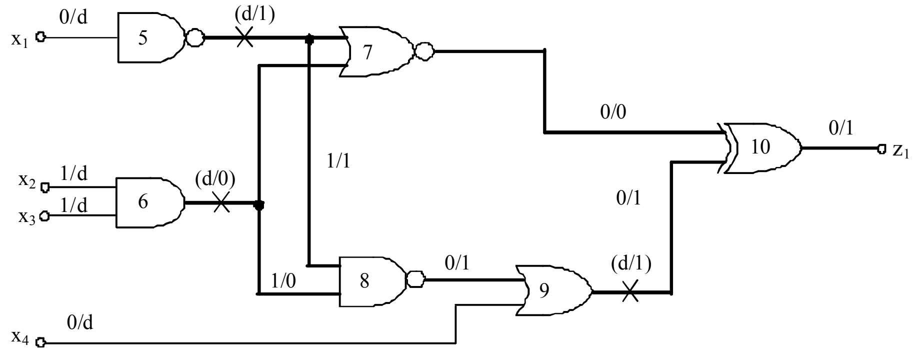


Figure 2. Multiple-fault test calculation. Result 1.

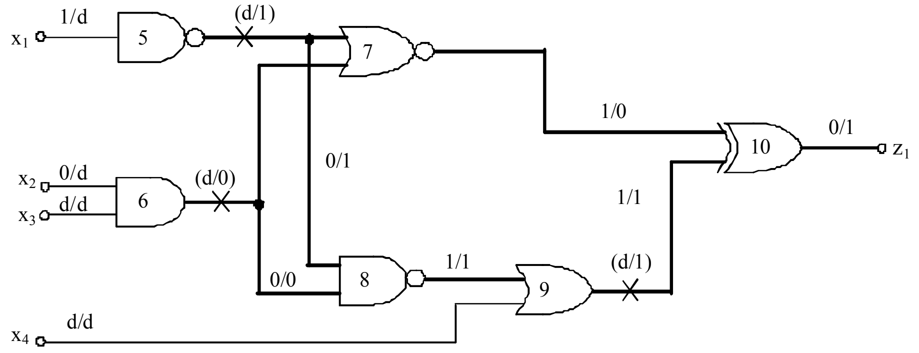


Figure 3. Multiple-fault test calculation. Result 2.

In order to find a test for a multiple fault, attempts are made to propagate different combinations of the individual faults to z_j when using the D-algorithm [6], [7]. If the number of faults is ***q***, a maximum of $2^q - 1$ attempts are required in worst case. In our example, this number is $2^3 - 1 = 7$. As a matter of fact, it should be remembered that within a single attempt, the D-algorithm has to be repeated $2^p - 1$ times, even for a single fault, where ***p*** is the number of the different paths from the fault site towards z_j. In our example this number for 5(1) is 3, for 6(0) is 3, and for 9(1) is 1, which results in the sum of 7. Thus, just the separate single-path propagations would require a maximum of 7 × 7 = 49 attempts in worst case. And so far we have not counted the possible multiple-fault propagations.

On the other hand, the D-algorithm becomes more complicated if more than one fault effect must be propagated simultaneously. In this case, the individual path combinations for the participant faults will be summed up, thus yielding the possible maximum number of attempts. Anyway, in general, the overall sum of computational steps may be a significant multiple of $2^q - 1$ in worst case.

For the modified D-algorithm it is sufficient to attempt to propagate the individual faults separately. Hence, this approach implies a maximum of ***q*** fault-propagation attempts [11]. In our example it is **three.**

The composite justification for multiple faults needs to be performed at first for $z_j = 0 / 1$, and if it fails then repetition for $z_j = 1 / 0$ is required. It means that the number of attempts in worst case is only **two,** independent of the multiplicity of faults. Nevertheless, this algorithm is only slightly affected by the number of faults. In addition, we do not have to iterate through all the possible fault propagation combinations. Instead, a combination necessarily results from the justification process itself, as illustrated in Figure 2 and 3.

4. The thesis of minimal complexity

The efficiency of composite justification is based on the fact that it establishes **the minimal necessary and sufficient set of logic values** which yield the test conditions for the faults. As seen, the tests are obtained only by justifying this initially set logic conditions. This fact implies **the total absence of the fault propagation phase,** which is a salient advantage of composite justification

There exist other test generation algorithms like PODEM [13] and FAN [14] that have been proved to be more efficient than the D-algorithm for single faults. As a matter of fact, the acceleration results in the PODEM and FAN are relied on some clever improvements on the original D-approach, by using structural and logic analyses, while the overall fault propagation and justification principle remained unchanged.

When multiple faults are considered, the complete procedure of the D-algorithm has to be repeated $2^q - 1$ times in worst case, for one primary output, where q is the multiplicity of faults. In this case, attempts are made to propagate different combinations of the individual faults. The modified D-algorithm has ***q*** attempts in worst case. On the other hand, composite justification requires only **two iterations** in worst case, also for one primary output.

From the above results the following **thesis** can be conjectured:

The composite justification algorithm has an **absolute minimal** computational complexity in generating a test for any single or multiple stuck-at faults in logic networks at the gate level. In other words, there exists no other exact algorithm which would need fewer computational steps.

Under the assumption that this thesis holds, the following consequences will be implied:

As known, the justification process itself is nothing else than traversing downward and upward along a given decision tree, until a solution is found. All this means that the process has an exponential complexity, depending on the number of nodes within the tree. On the other hand, since all NP-complete problems have the same property in terms of complexity, each of them can only be solved by some exponential-time algorithm in general. If so, it would mean that no polynomial-time solution is available for any of the NP-complete problems in general.

As far as the computer implementation is concerned, only line justification is to be accomplished in the presented principle, which is also an advantage. In order to perform line-value justification, the **inverse models** of the building elements in the network are required. An inverse model defines the set of possible input patterns which result in a given output pattern.

After all, the process may also be called **inverse simulation**, as opposed to the normal one, where the outputs are calculated with knowledge of the inputs. Thus, the two simulation approaches have opposite directions in treating the logic elements within a circuit.

5. Application of different modeling levels

The fact that composite justification is in essence an inverse simulation algorithm which avoids fault propagation, implies the following consequences:

1) The algorithm enables the user to model the digital circuits at various levels, namely:

- **Switch level for CMOS circuits:** the building elements are transistors.
- **Gate level:** logic gates are applied exclusively.
- **Functional level:** the logic values of an element are calculated with knowledge of its external functional behavior. For this modeling purpose, high level **hardware-description languages (HDL's)** can be applied, for instance, VHDL.
- **Register-transfer level (RTL):** the behavior of the digital system is described by means of bit vectors that are processed and transferred among various building blocks.

2) The feature that composite justification avoids direct fault propagation makes the approach extremely flexible in terms of circuit modeling and fault classes [9], [10], [12]. As known, D-propagation is an inherent part of the D-algorithm, where this phase implies serious difficulties for functional level models and multiple faults [7]. Functional algorithms for constructing computational tools of complex logic modules have been presented in [15]. This paper clearly illustrates the problems encountered in this topic.

As a matter of fact, the acceleration results in the PODEM and FAN are relied on the use of gate-level network structure. However, if functional level modeling is considered then the structural and logic analyses performed in both algorithms become extremely cumbersome and hardly feasible. In contrast, the approaches based on composite justification are not really limited by the way of network modeling. The same applies to the network types, i.e., combinational or sequential, and also, the types and multiplicity of the faults [9], [12].

At this point, it should be mentioned that in case of sequential circuits, multiple-fault calculation is necessary even for a single fault which may propagate to various storage elements, and there manifests itself in a virtual appearance.

For a sequential logic element, an inverse model defines the set of possible input patterns which result in a specific **state** or an **output** pattern [15], [16]. For this purpose, high level hardware-description languages, such as VHDL or any RTL language can also be applied.

3) It is known that generating fault-propagation tests is equivalent with generating tests for delay faults [17], [18]. It means that all the advantages related to composite justification are equally valid for delay testing as well [12].

4) As for transistor-level models, [19] has shown how the D-algorithm can be extended to CMOS circuits, while [20] has demonstrated the same with composite justification.

5) Finally, as a result of the above considerations, it may also be expected that the **thesis of minimal complexity** conjectured in Section 4 applies in general to any categories of test calculation problems for logic networks, independent of the way of modeling and fault type.

6. Conclusions

This paper has been meant for showing how the test calculation algorithm first published in [8] can be compared with the D-algorithm and its modifications. The comparison results demonstrate that the former one requires significantly less computational step than the D-algorithm and its improvements. From this fact it has been conjectured the thesis that possibly there is no other algorithm for the given problem with fewer computational steps. The implication of the thesis is that test calculation is definitely of exponential time, and so are all the other NP-complete problems.

As a result of further analyses in the paper, it may also be expected that the thesis of minimal complexity applies in general to any categories of test-generation problems for logic circuits, independent of the modeling level and fault types. From this it follows that the solutions to be obtained are also of exponential time.

The importance of exact algorithms manifests itself in the necessity of solving a practical problem at the cost of an enormous but still acceptable computation time. Such a requirement is to reach a high-degree fault coverage of a test set for digital circuits. Usually, it is far above 90%.

As known, there are random and heuristic methods that perform well in producing test sets. In general, these approaches are necessarily combined with fault simulation. Recently, there have been published new heuristic approaches which proved to be very efficient in treating VLSI (Very Large-Scale Integration) circuits, and a wide range of fault models [21], [22]. The paper [21] provides a comprehensive framework for treating various defects in a logic circuit, by applying an approach based on Cause-

Effect analysis [7]. As far as the paper [22] is concerned, it presents two new methods for the generation of test sets with a small number of specified bits.

However, if the fault coverage achieved is not satisfactory, as is often the case with heuristic methods, an additional test-generation attempt is required for the undetected faults. In this situation, only an exact algorithm can be taken into account.

Acknowledgements

This work was supported in part by the project TAMOP 4.2.1.B at the Széchenyi University of Győr.

References

[1] H. R. Lewis, Ch. H. Papadimitriou: *Elements of the Theory of Computation,* Prentice-Hall, Inc., USA, 1998.

[2] J. E. Hopcroft, R. Motwani, J. D. Ullman: *Introduction to Automata Theory, Languages, and Computation,* Second Edition, Addison-Wesley Publishing Company, USA, 2001.

[3] T. H. Cormen, Ch. E. Leiserson, R. L. Rivest, C. Stein: *Introduction to Algorithms,* McGraw-Hill Publishing Company, USA, 2001.

[4] O. H. Ibarra, S. K. Sahni: *Polynomially complete fault detection problems,* IEEE Trans. On Computers, Vol. C-24, pp. 242–249, March 1975.

[5] H. Fujiwara: *Logic Testing and Design for Testability,* MIT Press, USA, 1985.

[6] J. P. Roth: *Diagnosis of automata failures: A calculus and a method,* IBM Journal of Research and Development, Vol. 10, pp. 278–291, July 1966.

[7] M. Abramovici, M. A. Breuer, A. D. Friedman: *Digital Systems Testing and Testable Design,* Computer Science Press, USA, 1990.

[8] J. Sziray: *Test calculation for logic networks by composite justification,* Digital Processes, Vol. 5, No. 1–2, pp. 3–15, 1979.

[9] J. Sziray: *A comprehensive method for the test calculation of complex digital circuits,* Periodica Polytechnica, Budapest Techn. Univ., Series of Electronic Eng., Vol. 41, No. 4, pp. 251–257, 1998.

[10] J. Sziray: *Functional level test calculation and fault simulation for logic networks,* Discrete Simulation and Related Fields (Ed. by A. Jávor), pp. 223–234, North-Holland Publishing Company, Amsterdam, 1982.

[11] P. Muth: *A nine-valued circuit model for test generation,* IEEE Trans. on Computers, Vol. C-25, pp. 630–636, June 1976.

[12] J. Sziray: *Test Calculation for Logic and Delay Faults in Digital Circuits,* IEEE Microprocessor Test and Verification Workshop, (MTV-06), Proceedings, pp. 20–29, Austin, Texas, USA, December, 2006.

[13] P. Goel: *An implicit enumeration algorithm to generate tests for combinational logic circuits,* IEEE Trans. on Computers, Vol. C-30, pp. 215–222, March 1981.

[14] H. Fujiwara, T. Shimono: *On the acceleration of test generation algorithms,* IEEE Trans. on Computers, Vol. C-32, pp. 1137–1144, December 1983.

[15] M. A. Breuer, A. D. Friedman: *Functional level primitives in test generation,* IEEE Trans. on Computers, Vol. C-29, pp. 223–235, March 1980.

[16] J. Sziray, Zs. Nagy: *OPART: A hardware-description language for test generation,* Microprocessing and Microprogramming, (Ed. by P. Nunez), pp. 525–530, North-Holland Publishing Company, Amsterdam, 1991.

[17] C. J. Lin, S. M. Reddy: *On delay fault testing in logic circuits,* IEEE Trans. on Computer-Aided Design, Vol. CAD-6, pp. 694–703, September 1987.

[18] I. Pomeranz, S. M. Reddy: *A generalized test generation procedure for path delay faults,* FTCS-28, International Symposium on Fault-Tolerant Computing, Proceedings, pp. 274–283, Munich, June 1998.

[19] S. K. Jain, V. D. Agrawal: *Modeling and test generation algorithms for MOS circuits,* IEEE Trans. on Computers, Vol. C-34, pp. 426-433, May 1985.

[20] J. Sziray: *Switch-Level Test Calculation for CMOS Circuits,* IEEE 10th International Workshop on Microprocessor Test and Verification, (MTV-09), Proceedings, pp. 58–65, Austin, Texas, USA, December 7–9, 2009.

[21] A. Bosio, P. Girard, S. Pravossoudovitch, A. Virazel: *A Comprehensive Framework for Logic Diagnosis of Arbitrary Defects,* IEEE Trans. on Computers, Vol. C-59, pp. 289–300, March 2010.

[22] S. N. Neophytou, M. K. Michael: *Test Set Generation with a Large Number of Unspecified Bits Using Static and Dynamic Techniques,* IEEE Trans. on Computers, Vol. C-59, pp. 301–316, March 2010.

Robot Systems with Fuzzy Signature Rule Bases

Á. Ballagi[1], L. T. Kóczy[2,3], C. Pozna[4]

[1]Department of Automation, Széchenyi István University,
H-9026, Győr, Egyetem tér 1., Hungary
Email: aronball@gmail.com
[2]Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics,
H-1117, Budapest, Magyar tudósok krt. 2., Hungary
Email: koczy@tmit.bme.hu
[3]Inst. of Mechanical, Electrical Engineering and Information Technology,
Széchenyi István University,
H-9026, Győr, Egyetem tér 1., Hungary
[4]Departament of Design and Robotics, University Transilvania of Brasov
B-dul Eroilor 29 Brasov Romania
Email: cp@unitbv.ro

Abstract: This paper presents some examples for fuzzy communication and intention guessing from the real life to the cooperation of intelligent mobile robots. In a special experimental environment a new communication approach is investigated for intelligent cooperation of autonomous mobile robots. Effective, fast and compact communication is one of the most important cornerstones of a high-end cooperating system. In this paper we propose a fuzzy communication system where the codebooks are built up by fuzzy signatures. Fuzzy signature can be considered as special multidimensional fuzzy data. Some of the dimensions are interrelated in the sense that they form sub-group of variables, which jointly determine some feature on higher level. Thus, complex objects and situations can be described by fuzzy signatures. We use cooperating autonomous mobile robots to solve some logistic problems.

Keywords: fuzzy signature :fuzzy communication :mobile robotics.

1. Introduction

Intelligent cooperation of two or more robots has been tackled in different ways since at least the late 1980's. If one intends to build a cooperating robot system with intelligent behavior, it is impossible to foresee all scenarios potentially occurring, thus effective, fast and/or compact communication is one of the most important cornerstones of such a cooperating system. Communication is sometimes quite expensive, so very compact solutions have often priority in comparison to very precise ones. We suggest building up a context dependent knowledge base and a common codebook in the distant on-board controller of each robot. By observation of the scenario and by "intention guessing" (i.e. analyzing the other robots' behavior), it is possible to shorten the actual communication process and to essentially reduce the amount of information that needs to be transmitted from one agent to another.

We have proposed a fuzzy communication philosophy and implementation technique, where the codebooks are built up by fuzzy signatures or fuzzy signature sets. Fuzzy signatures structure data into nested vectors of fuzzy values, each value being a further fuzzy vector. Fuzzy signatures are suitable for describing cases with different numbers of data components, with even some of the components missing, i.e. fuzzy signatures of varying structure are used. Fuzzy signature can be considered as special multidimensional fuzzy data. Some of the dimensions are interrelated in the sense that they form sub-group of variables, which jointly determine some feature on higher level. Thus, complex objects and situations can be described by fuzzy signatures. This is a useful tool for building the codebook in a context dependent reconstructive communication situation. Fuzzy communication is based on the comparison and reconstruction of fuzzy signatures [1, 2].

At the end some real scenarios of autonomous mobile robot cooperation are presented. A group of autonomous intelligent mobile robots is supposed to solve simple transportation problems, where always two robots must cooperate. Exact instructions are given only to the Robot Foreman (R0) who starts behaving in a way indirectly indicating the job to be done. The other robots have no direct communication links with R0 and all the others, but they try to solve the task by intention guessing from the actual movements and positions of other robots, even though these might sometimes be ambiguous.

The research towards extending this fuzzy communication method to more complex robot cooperation is going on currently in the frame of a large EEA grant.

2. Fuzzy Communication

There are orthogonally contradicting interpretations of the idea of Fuzzy Communication. In [3] a scenario is described where the lack of precise and sufficient information in a business environment leads to employees creating their own fictive scenarios where they fill the information vacuum with conjecture and wrong assumptions, which eventually leads to catastrophic results. Another kind of scenario is taken from the Laboratory for International Fuzzy Engineering Research (Yokohama) that operated between 1989 and 1995 as the spiritual center of applied fuzzy research in Japan [4]. This latter gives a positive example for using fuzzy elements for compressed and effective communication between humans.

Scenario (based on a presentation at LIFE [4])

> *Director Tanaka receives a new secretary, Ms. Sato, on Monday. When Mr. Tanaka returns from lunch, he calls Ms.Sato and the next conversation follows:*

'Ms. Sato, I would like to have a cup of tea.'
'Yes, Mr. Tanaka. Do you prefer hot or cold tea?'
'Hot tea, please.
'Do you prefer black or green tea?'
'Give me black tea.'
'Do you need sugar?'
'No sugar, please.'
'Any milk to the tea?'
'No, thank you.'
So, Ms. Sato prepares the tea according to the request.
On Tuesday, after lunch the director calls Ms. Sato again.

'May I have a cup of tea?'
'Yes, Mr. Tanaka. Black tea, again?'
'Yes.'
'No sugar, no milk?'
'Exactly as you say.'
Now fewer questions led to the same action by the secretary.
On Wednesday, when Mr. Tanaka arrives, he does not say
anything but
'May I?'
'The usual tea?'
'Yes.'
On Thursday, when the director comes in after lunch, Ms.
Sato asks him:
'May I prepare your usual tea?'
'Yes, thank you.'

The Monday conversation consisted of a request and four questions and informative answers (9 sentences altogether). On Thursday there was no more need for any request, the fact that Mr. Tanaka arrived after lunch triggered the single yes/no question and after confirmation the tea was prepared (two sentences only). In the first example the story is about some employees who misinterpreted the behavior of the new chairman, having a certain "codebook" in the sense of fuzzy communication in [5], which contained rules that were not valid any more for the new situation. Thus meta-communication leads to false assumptions. In the "tea example" the codebook was built up by learning in the head of Ms. Sato, and it was adequate for the situation thus communication with Mr. Tanaka evolved into a very effective one.

In the tea scenario, the formal determination of the codebook contents could be something like "If Mr. Tanaka arrives after lunch (T), he likes to drink hot (H) black tea (B) without milk $(\neg M)$ and without sugar $(\neg S)$." With logical symbols it is $T \rightarrow B \,\&\, T \rightarrow H \,\&\, T \rightarrow M \,\&\, T \rightarrow \neg S$

If this codebook remains valid (Mr. Tanaka does not change his tea drinking habits), the simple formal model might be sufficient for the future. We might assume that this was not Ms. Sato's first secretary position and she had learned various other contexts valid in different environments, which she essentially discarded at the point of entering Mr. Tanaka's office. In addition to the newly learned codebook elements, she has the general background knowledge base (common knowledge) that contains information like "After lunch people often drink coffee, tea, water or cold drinks", "There are hot and cold tea", "There are short, long and cappuccino style coffee", "People drinking tea or coffee sometimes add sugar and/or milk", etc. Concerning the codebook learning procedure, the next continuation of the scenario will clarify that a more structured codebook model should be applied more appropriately in order to keep the codebook flexible enough for accepting changes and additions easily. It should be remarked that expressions like often or sometimes might be formally interpreted by Precisiated Natural Language introduced by L. A. Zadeh [6] as a tool for describing verbally non-exact adverbs, modifiers, hedges, etc. that might play important roles in modeling complex imprecise phenomena, both in everyday life and in the applied sciences. Next a possible continuation of the above scenario is given.

Scenario (continued) One day Mr. Tanaka arrives from lunch and Ms. Sato asks the everyday question:

'May I prepare your usual tea?'
This time the answer is
'No, today I prefer coffee.'
'Do you drink it long or short?'
'Short please.'
'Any sugar?'
'No, thank you.'
'Any milk?'
'Yes, some milk, please.'

This way the knowledge base in the codebook might be formulated as the following:

"If Mr. Tanaka arrives after lunch (T), he usually likes to drink hot black tea (B) without milk $(\neg M)$ and without sugar $(\neg S)$, and sometimes he prefers short (E) coffee (C) with some milk (M) but no sugar $(\neg S)$." Formally it could be $(T[usually] \to B \,\&\, T[usually] \to H \& T[usually] \to \neg M \& T[usually] \to \neg S,$

$T[sometimes] \to C \,\&\, T[sometimes] \to E \,\&\, T[sometimes] \to M \,\&\, T[sometimes] \to \neg S)$ However, it is much more reasonable to order the information into a structured way:

$$(T[usually] \to B \& T[sometimes] \to E; B \to H \& \neg M \& \neg S; C \to E \& M \& \neg S)$$

Such codebook elements can be quite well formulated by using PNL, fuzzy logic and fuzzy sets. In the continuation of the scenario obviously the context has not changed in the sense that its already existing elements were not discarded but further elements had to be added. Generally it may not be expected that in a context like in the above scenario, where a human environment gives the base for the context, and human interactions are involved, the codebook could ever be considered as final or completed.

Both examples tell about communication between two or more humans, i.e. "man-man" communication. In both cases, from meager information the original contents should and in the latter case it also could be reconstructed. Often the problem to solve involves "man-machine" or even, "machine-machine" communication. It is possible that a similar communication channel can be opened between man and machine or between two machines where just the "skeleton" or even the "approximate skeleton" of the information that is supposed to be put through will be or can be transmitted. It is an open question, whether that kind of context dependent reconstructive type fuzzy communication that was used in the tea and coffee scenario could be utilized for engineering or other applied science purposes in the practice.

3. Context dependent Reconstructive Communication with Fuzzy Signatures

There is definitely a wide area of possible applications of compressed communication transmission, if the reconstruction at the receiving end is possible. Communication itself is very expensive. Generally speaking, it is much more advisable to build up as big as possible contextual knowledge bases and codebooks in distant computers in order to shorten their communication process if it essentially reduces the amount of information that must be transmitted from one to another, than to concentrate all contextual knowledge in one of them and then to export its respective parts whenever they are needed in the other(s) It seems to be very important in the cooperation and communication of intelligent robots or physical agents that the information exchange among them is as effective and compressed as possible. There are also some other areas where this kind of compressed communication might help to solve difficult tasks. The next example is also taken from the partly unpublished research projects at LIFE [4]. The following example was actually implemented at LIFE in two alternative forms (hardware and simulation software), and some partly encouraging results could be observed.

It must be remarked, however, that the new considerations and approaches concerning an alternative solution of the robot communication problem will be first discussed here by the authors.

In the next sections some formal methods will be proposed for building codebooks for Context Dependent Reconstructive Communication (CDRC) where fuzzy components and PNL elements can also be considered.

3.1. Fuzzy Signatures

The original definition of fuzzy sets was $A:X \rightarrow [0,1]$, and was soon extended to *L-fuzzy sets* by Goguen [7]

$$A_s : X \rightarrow \left[a_i\right]_{i=1}^{k}, a_i = \left\{ \frac{[0,1]}{\left[a_{ij}\right]_{j=1}^{k_i}} \right., a_{ij} = \left\{ \frac{[0,1]}{\left[a_{ijl}\right]_{l=1}^{k_{ij}}} \right. \qquad (1)$$

$A_L : X \to L$, L being an arbitrary algebraic lattice. A practical special case, Vector *Valued Fuzzy Sets* was introduced by Kóczy [8], where $A_{V,k} : X \to [0,1]^k$, and the range of membership values was the lattice of k-dimensional vectors with components in the unit interval. A further generalization of this concept is the introduction of fuzzy signature and signature sets, where each vector component is possibly another nested vector (right).

Fuzzy signature can be considered as special multidimensional fuzzy data. Some of the dimensions are interrelated in the sense that they form sub-group of variables, which jointly determine some feature on higher level. Let us consider an example. Figure 1shows a fuzzy signature structure.

The fuzzy signature structure shown in Figure 1 can be represented in vector form as follow:

$$x = \begin{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} \\ \begin{bmatrix} x_{21} \\ \begin{bmatrix} x_{221} \\ x_{222} \\ x_{223} \end{bmatrix} \\ x_{23} \end{bmatrix} \\ \begin{bmatrix} x_{31} \\ x_{32} \end{bmatrix} \end{bmatrix}^T \tag{2}$$

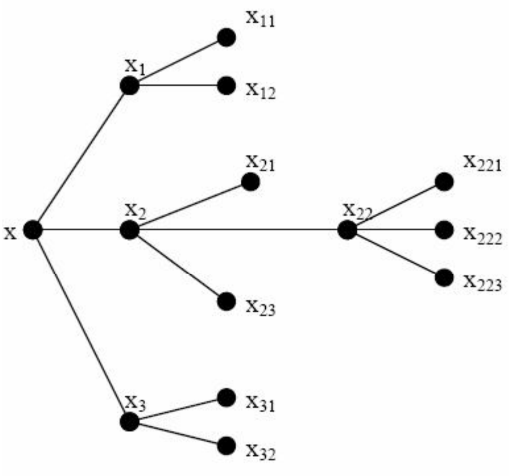


Figure 1. A Fuzzy Signature Structure

Here $[x_{11}\ x_{12}]$ from a sub-group that corresponds to a higher level compound variable of x_1. $[x_{221}\ x_{222}\ x_{223}]$ will then combine together to form x_{22} and $[x_{21}\ [x_{221}\ x_{222}\ x_{223}]\ x_{23}]$ is equivalent on higher level with $[x_{21}\ x_{22}\ x_{23}] = x_2$. Finally, the fuzzy signature structure will become $x = [x_{221}\ x_{222}\ x_{223}]$ in the example.

The relationship between higher and lower level is governed by the set of fuzzy aggregations. The results of the parent signature at each level are computed from their branches with appropriate aggregation of their child signature. Let a_1 be the aggregating associating x_{11} and x_{12} used to derive x_1, thus $x_1 = x_{11}\mathrm{a}_1x_{12}$. By referring to Figure 1, the aggregations for the whole signature structure would be a_1, a_2, a_{22} and a_3. The aggregations a_1, a_2, a_{22} and a_3 are not necessarily identical or different. The simplest case for a_{22} might be the min operation, the most well known t-norm. Let all aggregation be min except a_{22} be the averaging aggregation. We will show the operation based on the following fuzzy signature values for the structure in the example.

Each of these signatures contains information relevant to the particular data point x0; by going higher in the signature structure, less information will be kept. In some operations it is necessary to reduce and aggregate information obtained from another source (some detail variables missing or simply being locally omitted). Such a case occurs when interpolation within a fuzzy signature rule base is done, where the fuzzy signatures flanking an observation are not exactly of the same structure. In this case the maximal common sub-tree must be determined and all signatures must be reduced to that level in order to be able to interpolate between the corresponding branches or roots in some cases [9].

$$x = \begin{bmatrix} \begin{bmatrix} 0.3 \\ 0.4 \end{bmatrix} \\ \begin{bmatrix} 0.2 \\ \begin{bmatrix} 0.6 \\ 0.8 \\ 0.1 \end{bmatrix} \\ 0.9 \end{bmatrix} \\ \begin{bmatrix} 0.1 \\ 0.7 \end{bmatrix} \end{bmatrix}^T \tag{3}$$

After the aggregation operation is performed to the lowest branch of the structure, it will be described on higher level as:

$$x = \begin{bmatrix} 0.3 \\ \begin{bmatrix} 0.2 \\ 0.5 \\ 0.9 \end{bmatrix} \\ 0.1 \end{bmatrix}^T \tag{4}$$

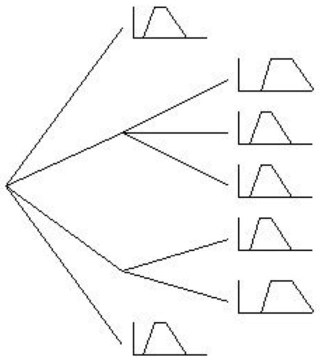
Finally, the fuzzy signature structure will be:

$$x = \begin{bmatrix} 0.3 \\ 0.2 \\ 0.1 \end{bmatrix}^T \tag{5}$$

Thus, complex objects and situations can be described by fuzzy signatures. This is a useful tool for building the codebook in a context dependent reconstructive communication situation.

3.2. Fuzzy Signature Sets

The basic structure of fuzzy signature sets is similar to that of fuzzy signatures, the only difference being that instead of having fuzzy variables on the leaves of the structure, membership functions are present (see Figure 2). The only constraint for the membership functions is that their domain must be the [0,1] interval [10].



$$S = \begin{bmatrix} \mu_1(x) \\ \begin{bmatrix} \mu_{21}(x) \\ \mu_{22}(x) \\ \mu_{23}(x) \end{bmatrix} \\ \begin{bmatrix} \mu_{31}(x) \\ \mu_{32}(x) \end{bmatrix} \\ \mu_4(x) \end{bmatrix}$$

Figure 2. The tree structure and vector form of a fuzzy signature set

3.3. Signature Structure Modification: The Aggregation Operators

The advantages of fuzzy signatures lie in organizing the available data components into a hierarchy. This hierarchy determines the arbitrary structure of our fuzzy signature observations. As some of the components of this arbitrary structure might be missing from the specific observations, some kind of structure modifying operation is essential when comparing these differently structured signatures. Aggregation operations result in a single fuzzy value calculated from a set of other fuzzy values, while satisfying a set of axioms. The most common operators are the maximum, minimum and arithmetic mean operator [11, 12].

Aggregation operators can be used to transform fuzzy signature structures by reducing a sub-tree of variables to their parent node. It is necessary to mention that only whole sub-trees of the structure can be reduced. The fuzzy value (or fuzzy set) assigned to the parent node is calculated by aggregating the fuzzy values (or fuzzy sets) of its children using the aggregation operator of the parent node. This way, the depth of this branch of the structure is reduced by one.

This procedure can only be performed when all elements of the aggregated sub-tree are leaves of the structure and have an assigned value. If one of the elements of the sub-tree branches out into a sub-tree of its own, in order to reduce the whole sub-tree to the original parent node, first the sub-subtree has to be reduced to its parent node (which is the child node of the original sub-tree's parent node) by aggregation. After performing the aggregation, the original sub-tree can also be reduced.

For example, to reduce the sub-tree of node x_i, first the sub-tree of node x_{i2} has to be aggregated. The value obtained can then be used when calculating the aggregate value from the sub-tree of x_i. This recursive procedure is shown in Figure 3, where $@_i$ denotes the aggregation operator of node x_i.

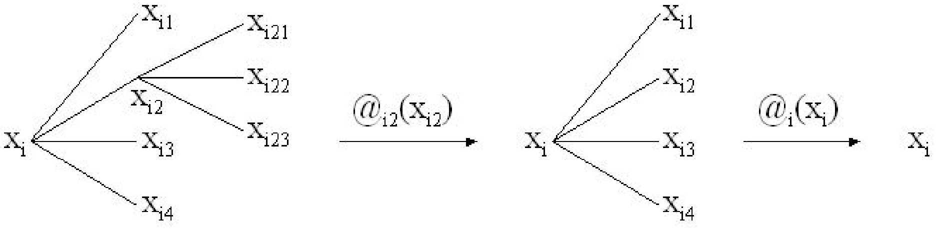


Figure 3. Recursion used to reduce a sub-tree with several layers

Because of these terms, when reducing a signature to a predefined structure it is wise to use a bottom-up method. This means to start the reduction from the leaves of the structure and work your way up one sub-tree at a time towards the intended structure.

According to the definition of fuzzy signatures, aggregation operators define the connection between a component, and its sub-components, therefore the aggregation operators are not necessarily identical for all the nodes of the structure. Finding the relevant aggregation operator for each node is a very important problem of fuzzy signatures, because when comparing two signatures, the obtained results may greatly depend on the aggregation operators used to reduce the signatures to a common structure.

It is also important to mention that when reducing a signature's sub-tree, some information is lost in all cases, because the calculated aggregated value can be the same for many different values and differently structured sub-trees.

In order to reduce fuzzy signature sets to a different structure aggregation has to be generalized to work not only on fuzzy values but on fuzzy sets as well. When aggregating fuzzy sets, the membership values for each element x of [0,1] (the domain of the fuzzy sets on the leaves of the structure) are calculated for all the fuzzy sets which are subject to the aggregation. The original aggregation operator is then used on these membership values to obtain the aggregated membership value belonging to x. Let the fuzzy sets in the sub-tree be A_i. The membership function of the aggregated fuzzy set G is given in 6, where h denotes the aggregation operator.

$$G = h(A_1, A_2, \ldots, A_k) \quad \forall x \in [0,1] \quad \mu_G(x) = h\left\{\mu_{A_1}(x), \mu_{A_2}(x), \ldots, \mu_{A_k}(x)\right\} \qquad (6)$$

The aggregation of two fuzzy sets (A_1 and A_2) is shown in Figure 4. In the example, the aggregation operator is the arithmetic mean operator. The resulting fuzzy set (denoted by G) is marked with a broken line.

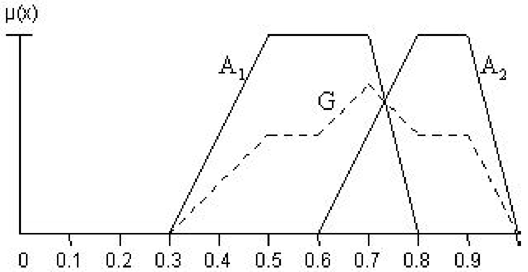


Figure 4. Aggregation of two fuzzy sets with the arithmetic mean operator
Weighted Relevance Aggregation Operator

With the introduction of weights [13] for each node of the fuzzy signature structure additional expert knowledge about the field can be contained within the model. The relevance weight depicts how relevant a node is in its parent's sub-tree. The weights of the nodes are taken from the [0,1] interval, and it is not necessary for the weights of the leaves in a sub-tree to add up to 1.

A method for learning weights was shown in [14]. The most general form of aggregation operators is the Weighted Relevance Aggregation Operator (WRAO) introduced by Mendis et al. in [14]. The values and weights belonging to each child l in the sub-tree are denoted by x_l and w_l respectively. The definition of the WRAO is as follows:

$$@(x_1, x_2, \ldots, x_n; w_1, w_2, \ldots, w_n) = \left\{\frac{1}{n}\sum_{i=1}^{n}(w_l \cdot x_l)^p\right\}^{\frac{1}{p}} \qquad (7)$$

where *p* is the aggregation factor of the above function. The well-known aggregation operators are all special cases of WRAO depending on the value of *p* in 7.

$$\begin{array}{llll} p \to -\infty & \text{WRAO} & \to & \text{minimum} \\ p = -1 & \text{WRAO} & = & \text{harmonic mean} \\ p \to 0 & \text{WRAO} & \to & \text{geometric mean} \\ p = 1 & \text{WRAO} & = & \text{arithmetic mean} \\ p \to \infty & \text{WRAO} & \to & \text{maximum} \end{array}$$

4. Fuzzy Communication of Cooperating Robots

One of the most important parameters of effective cooperation is efficient communication. Because communication itself very expensive, it is much more advisable to build up as large as possible contextual knowledge bases and codebooks in robot controllers in order to shorten their communication process. That is, if it essentially reduces the amount of information that must be transmitted from one to another, than to concentrate all contextual knowledge in one of them and then to export its respective parts whenever they are needed in other robot(s).

It appears to be very important in the cooperation and communication of intelligent robots or physical agents that the information exchange among them is as effective and compressed as possible [15].

4.1. Experimental task and environment

The actual stage of our research we use simulation of our real differential driven autonomous micro-robots (Figure 5). The physical simulation is exact model of our robots in the case of scale, weight, mechanical systems and sensors.

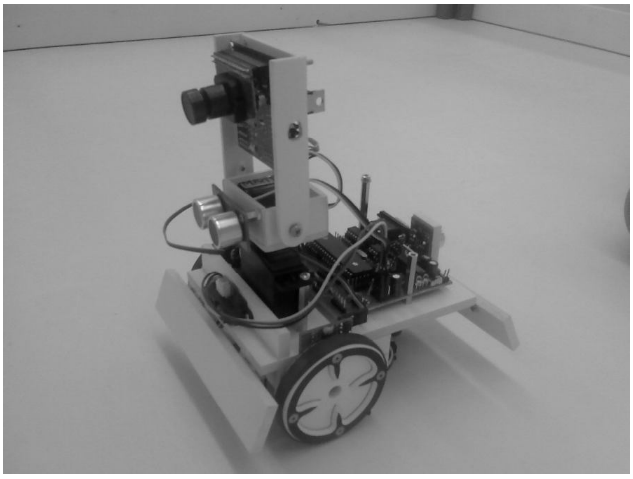


Figure 5. The real box-pushing robot

Three robots operate in a 2 × 2 m square arena. The box to be pushed is a 10 cm high and 20 cm wide and long. The goal region, located near to one corner of the arena. In the experiments described here the goal location is fixed but it can be moved before and during experiments.

Each robot pushes the box with two “whiskers” which have a pair of force sensors. The left and right whiskers each provides an analog force signal which is combined to give information the relative position to the box and via the control loop keep it contact on both sides and thus perpendicular to the box. Let us examine a subset of our overall robot cooperation problem works in practice. There is an arena where five square boxes wait for ordering. Various configurations can be made from them, but here only the “T” form is enabled with any orientation as Figure 6 shows.

The three robots form a group which try to build the actual order of boxes according to the exact instructions given to the R0 (foreman) robot. The other robots have no direct communication links with R0, but they are able to observe the behavior of R0 and all others, and they all posses the same codebook containing the base rules of storage box ordering. The individual boxes can be shifted or rotated, but always two robots are needed for actually moving a box, as they are heavy. If two robots are pushing the box in parallel the box will be shifted according the joint forces of the robots. If the two robots are pushing in opposite directions positioned at the diagonally opposite ends, the box will turn around the center of gravity. If two robots are pushing in parallel, and one is pushing in the opposite direction, the box will not move or rotate, just like when only a single robot pushes. Under these conditions the task can be solved, if all robots are provided with suitable algorithms that enable intention guessing from the actual movements and positions, even though they might be unambiguous. Figure 6 presents an example of how the five boxes can be arranged.

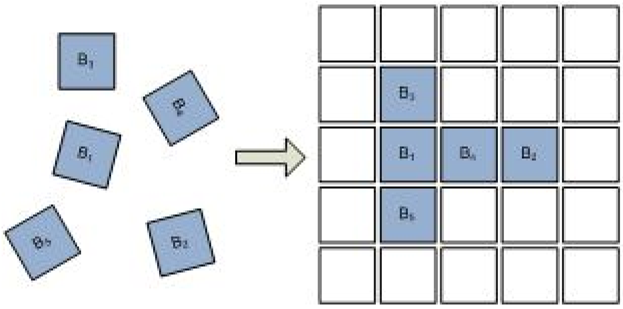
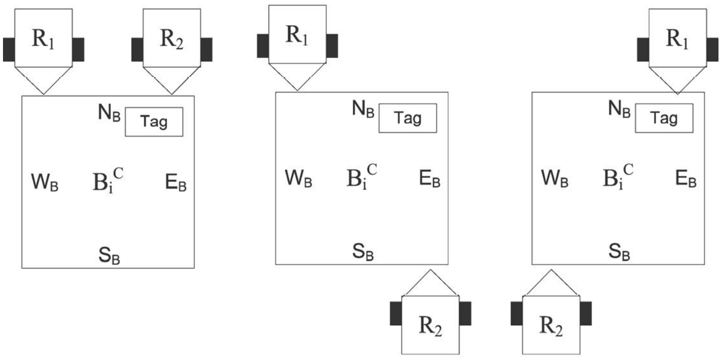


Figure 6. A box arrangement

The box has a B_i sign which means that is the i-th Box. The R_j is the sign of the j-th robot. The R0 is a distinct one, namely it is the robot foreman, the only robot that exactly knows the task on hand. The cooperating combination of robots is denoted by $C^{b}_{i,j,(k)}$ where i; j and k is the number of the robots (k appears only in stopping combinations), and b is the number of the box. There are three essentially different combinations (Figure 7), $C^{i}_{1,2} = P$ is the “pushing or shifting combination”, when two robots (R1 and R2) are side by side at the same side of the table; $C_{1,2}^{i} = RC$ stands for "counterclockwise rotation combination"; and $C_{1,2}^{i} = RW$ denotes "clockwise rotation combination".

Now the rotation problem is suppressed, each box is right direction, because it makes the system simpler and the rotation can be considered as an initial task.



$$\left.\begin{array}{l} P_1^i = [N_B, CC] \\ P_2^i = [N_B, CW] \end{array}\right\} C_{1,2}^i = P \qquad \left.\begin{array}{l} P_1^i = [N_B, CC] \\ P_2^i = [S_B, CC] \end{array}\right\} C_{1,2}^i = RC \qquad \left.\begin{array}{l} P_1^i = [N_B, CW] \\ P_2^i = [S_B, CW] \end{array}\right\} C_{1,2}^i = RW$$

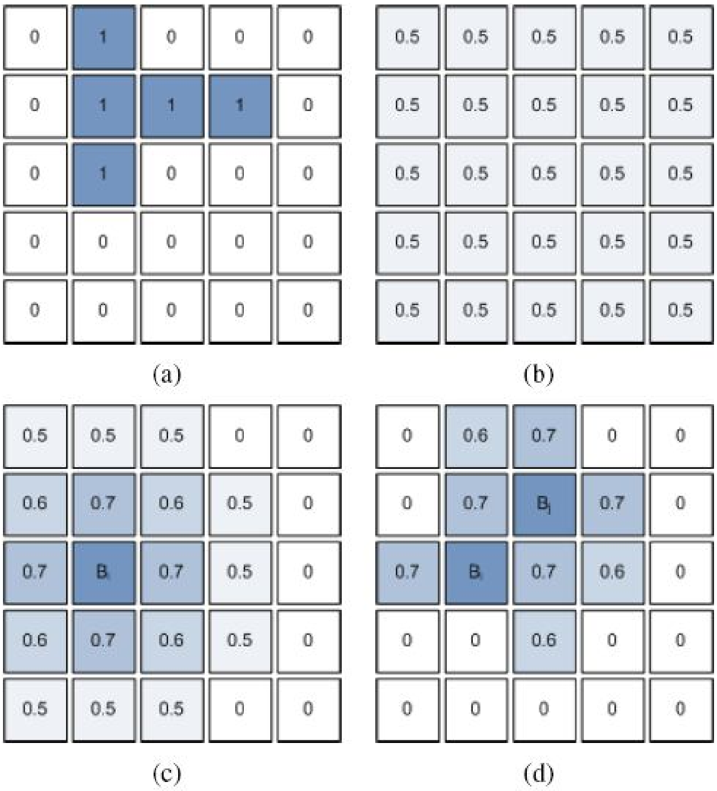
Figure 7. Allowed combinations of two robots for moving a box

4.2. The Fuzzy Signature Sets of Box State Representation

For the action selection or decision making each robot guesses the intentions of others and considers the actual states of the boxes. The robot uses the own codebook and sensory perceptions for this task.

The context dependent codebook is the soul of the onboard intention guessing and decision making system. It has many complex parts which build-up on fuzzy signature sets. Two main branches are the intention guessing and the state observation. The intention guessing means the other robots behaviors and actions are monitored and an intelligent manner processed for action or behavior selection. The state observation writes down the actual state of the whole working environment, the goal-area, the boxes, etc. The effective state observation of a high-scale system is a real complex task and needs strong computational intelligent algorithms.

The presentation the whole system is beyond the range of this paper, so let us see a part of the state observation subsystem. The other parts of our system are similarly formed as the presented. Let us consider the fuzzy signature sets of the box state and manipulation necessity. Each box has the own instance of this fuzzy signature set and the actual robot makes decision about the manipulation of that box by the fuzzy signature sets.



0	1	0	0	0
0	1	1	1	0
0	1	0	0	0
0	0	0	0	0
0	0	0	0	0

(a)

0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5

(b)

0.5	0.5	0.5	0	0
0.6	0.7	0.6	0.5	0
0.7	B_i	0.7	0.5	0
0.6	0.7	0.6	0.5	0
0.5	0.5	0.5	0	0

(c)

0	0.6	0.7	0	0
0	0.7	B_j	0.7	0
0.7	B_i	0.7	0.6	0
0	0	0.6	0	0
0	0	0	0	0

(d)

Figure 8. The arrangement-maps of robots

Here we have to say about two other elements of the observation system, the focal-point and the arrangement map. The focal-point means the area where the robot focuses its perception, as the focal-point is widen the actual inference has less impact on the action selection and robot behavior. There are three main focal-points: the box focal-point (BFP), the robot focal-point (RFP) and the goal-area focal-point (GFP). The focal-point selection is a complex task especially in case of the goal-area focal-point. For instance if one box was pushed in its position by robot foreman (R0) then the position of this box will be the next GFP, but if there are two or more boxes then GFP needs an intelligent selection. The arrangement-map (AM) depicts the fuzzy data of set-ability of a box in each position of goal-area. The actual arrangement of the boxes in goal-area is given to R0 foreman robot, so it has a precise arrangement-map as you can see in Figure 8/a. The Figure 8/b shows the arrangement-map of R1 and R2 robots in the starting phase where not any box in goal-area is. The Figure 8/c and /d show the AM if one and two boxes in the area respectively.

The above mentioned signature records the position, the arrangement, the dynamic and the robots working on the actual box. A possible arbitrary structure of the fuzzy signature representing the multidimensional complex features of the box is shown in

Figure 9. Each node in the tree has the own meaning and has close relations to focal-points and arrangement-maps.

The node x_1 describes the actual position of the box relative for its goal position. The distance (x_{11}) can be described by one of the four fuzzy sets in Figure 10, where in this case the fuzzy set *A* means that the box in position, *B* that it is close to the goal position, *C* means the near and *D* the far distance.

The fuzzy value on node x_{12} describes the rotation of the box from the North-South axis. In our case this node is always zero.

The x_2 node describes the accessibility of the box. The x_{21} fuzzy leaf represents the distance from robot. In this case fuzzy set *A* means the touching distance, fuzzy set *B* the close, fuzzy set *C* represents near and fuzzy set *D* the far distance respectively. The fuzzy value on node x_{22} describes any other feature of accessibility of the actual box, for instance obstacles, etc.

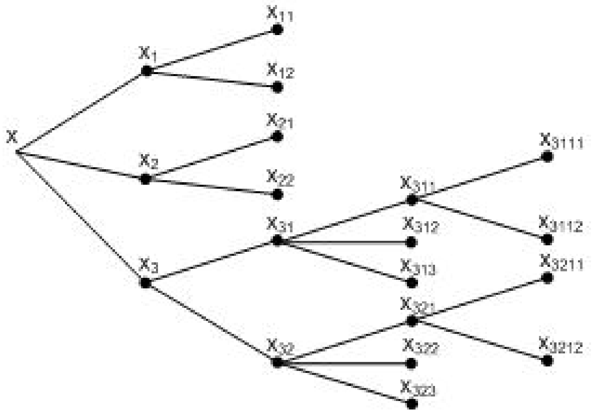


Figure 9. The box state and manipulation fuzzy signature

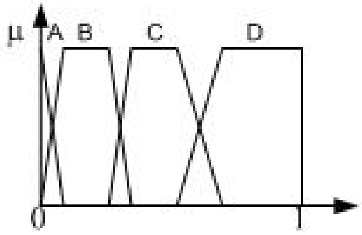


Figure 10. Fuzzy sets used in distance measurement

The node x_3 the most complex sub-tree in this structure. The fuzzy value on node x_3 describes the positions and states of other robots in viewpoint of the actual box. The sub-sub trees describe the number of touching robots, the touching points, cooperative combination and touch time, the distance and course of moving robots near the box. The detailed descriptions of this structure on fuzzy signature sets are beyond the extent of this paper.

An aggregation operator has to be defined for all nodes of the arbitrary structure. The notation $@_i$ will be used to refer to the aggregation operator assigned to node *i*. The inferences on these fuzzy signatures are able to describe the actual states of the box and give a basis for the fuzzy decision process in the robot control. Every robot builds its actual knowledge-base from the fuzzy signature classes and then boxes are assigned individual signatures in each individual robot controller. The leaves of the actual structures are fuzzy sets or membership functions therefore fuzzy signature sets are used in decision making and action selection mechanism.

5. Results

There is built up a simulated arena, where three robots cooperate to arrange some boxes. Every robot has the above described algorithms for intention guessing and action selection. The Figure 11 presents some steps of arranging process. Firstly the robots search the nearest disordered box and take a combination for pushing it. If they reach a pivot point then take a new combination and do it cyclic till order the actual box. This task recurs until all box get to the good place. Numerous scenarios are simulated and result a good collaboration with 90% of acceptability. Some simulation results will be downloaded from our website in the near future.

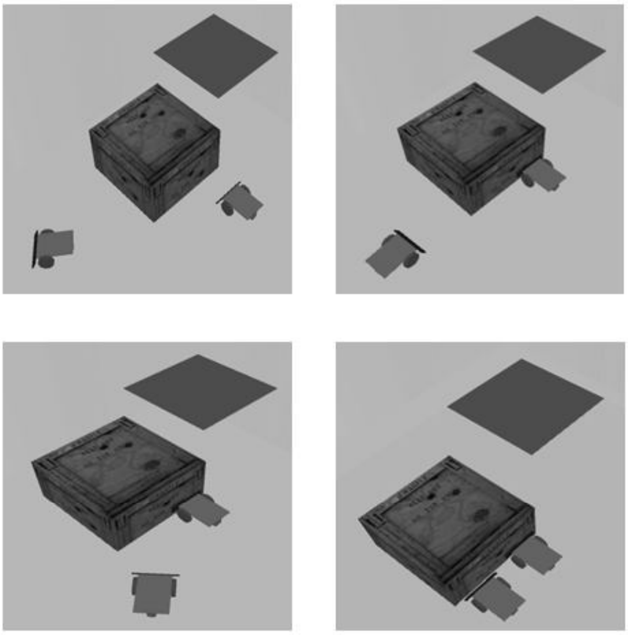


Figure 11. The robots take the starting combination

Conclusions

Fuzzy communication contains vague or imprecise components and it might lack of abundant information. If two entities (man or machine) are communicating by a fuzzy channel, it is necessary that both ends possess the same codebook. The codebook might partly consist of common knowledge but it usually requires a context dependent part that is either learned by the communicating entity or defined by expert knowledge. Possibly it is continuously adapting to the input information. If such a codebook is not available or it contains too imprecise information, the information to be transmitted might be too much distorted and might lead to misunderstanding, misinterpretation and serious damage. If however the quality of the available codebook is satisfactory, the communication will be efficient i.e., the original contents of the message can be reconstructed. At the same time it is cost effective, as fuzzy communication is compressed as compared to traditional "abundant communication". This advantage can be deployed in many areas of engineering, especially where the use of the communication channel is expensive in some sense, or where there is no proper communication channel available at all. We suggest that in distant locations and dangerous environment, further in applications where human friendly user interfaces are important, such as intelligent man-machine communication, the advantages of well designed CDRC communication systems be deeper investigated in the future.

Acknowledgements

The research was supported by HUNOROB project (HU0045,0045/NA/2006-2/ÖP-9), a grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Hungarian National Development Agency, a Széchenyi István University Main Research Direction Grant, and National Scientific Research Fund Grant OTKA K75711.

References

[1] A. Ballagi, L. T. Kóczy, "Fuzzy communication in a cooperative multi-robot task," in Proceedings of the 10th International Carpathian Control Conference, ICCC'2009, Zakopane, Poland, May 2009, pp. 59–62.

[2] A. Ballagi, L. T. Kóczy, T. D. Gedeon, "Local codebook construction for fuzzy communication in cooperation of mobile robots," Acta Technica Jaurinensis, vol. 1, no. 3, pp. 547–560, december 2008.

[3] M. Flaherty, "Abundant communication! the university of montana rural institute."

[4] T. Terano et al., "Reaearch projects at life," 1993/94, laboratory for International Fuzzy Engineering Research(Yokohama),oral presentations.

[5] P. Pedrycz, E. Roventa, "From fuzzy information processing to fuzzy communication channels," Kybernetes, no. 28, pp. 515–526, 1999.

[6] L. A. Zadeh, "Precisiated natural language – toward a radical enlargement of the role of natural languages in information processing, decision and control," in ICONIP'02, Singapore, 2002, keynote speech.

[7] J. A. Goguen, "L-fuzzy sets," J. Math. Anal. Appl, no. 18, pp. 145–174, 1967.

[8] L. T. Kóczy, ed. Gupta, M. M. and Sanchez, E.: Approximate Reasoning in Decision Analysis. Amsterdam: North Holland, 1982, ch. Vectorial I-fuzzy Sets, pp. 151–156.

[9] K. W. Wong, A. Chong, T. D. Gedeon, L. T. Kóczy, T. Vámos, "Hierarchical fuzzy signature structure for complex structured data," in Proceedings of the International Symposium on Computational Intelligence and Intelligent Informatics (ISCIII), Nabeul, 2003, pp. 105–109.

[10] K. Tamás, L. T. Kóczy, "Mamdani-type inference in fuzzy signature based rule bases," in Proceedings of the 8th International Symposium of Hungarian Researchers (CINTI 2007), Budapest, Hungary, November 2007, pp. 513–525.

[11] ——, "Selection from a fuzzy signature database by mamdani-algorithm," in Proceedings of the 6th International Symposium on Applied Machine Intelligence an Informatics (SAMI 2008), Herlany, Slovakia, January 2008, pp. 63–68.

[12] K. Tamas, L. T. Koczy, "Inference in fuzzy signature based models," Acta Technica Jaurinensis, vol. 1, no. 3, pp. 573–594, december 2008.

[13] B. S. U. Mendis, T. D. Gedeon, L. T. Kóczy, "Investigation of aggregation in fuzzy signatures," in Proceedings of the 3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems, Singapore, 2005, cD proc.

[14] B. S. U. Mendis, T. D. Gedeon, L. T. Kóczy, "On the issue of learning weights from observations for fuzzy signatures," in World Automation Congress (WAC), 2006.

[15] L. T. Kóczy, T. D. Gedeon, "Context dependent reconstructive communication," in Proceedings of the International Symposium on Computational Intelligence and Intelligent Informatics, ISCIII 2007, Agadir, Morocco, 2007, pp. 13–19.

Parallel Gene Transfer Operations for the Bacterial Evolutionary Algorithm

M. Hatwagner, A. Horvath

SZE, 9026 Győr, Egyetem tér 1.
Phone: 503 400, fax: 503 400
e–mail: miklos.hatwagner@sze.hu, horvatha@sze.hu

Abstract: Bacterial evolutionary algorithm (BEA) is a special evolutionary method, originally developed to optimize fuzzy system parameters, but it is also useful in many design and engineering problems. In the latter case, applications kill most of their time with the evaluation of the objective function. One possible way to speed up BEA is the use of many CPUs, and evaluate the objective functions of different individuals on them simultaneously. This can be realized e.g. with a cluster of workstations or with an SMP/SMC system. Unfortunately, the BEA in its original form is not suitable for parallelization, because one of the two main operations, the "gene transfer" is sequential. In this article we propose three new alternatives of BEA and present the results of test calculations with 5 test problems, on 1 to 16 processors to find a good parallel version of BEA.

Keywords: Bacterial Evolutionary Algorithm, parallel computing.

1. Introduction

Evolutionary algorithms are optimisation methods inspired by the Darwinian theory of evolution. The four main tends are evolutionary strategies, evolutionary programming, genetic programming and genetic algorithms (GA) [1]. The bacterial evolutionary algorithm (BEA) is a relatively new member of evolutionary algorithms, proposed by Norberto Eiji Nawa and Takeshi Furuhashi [13][14][15]. It is a descendant of the Pseudo-Bacterial Genetic Algorithm (PBGA) and traditional GA [5].

Several researchers have been dealing with the usage of memetic algorithms [6][11][14] to make evolutionary algorithms more efficient. The first implementation of them related to BEA was the Bacterial Memetic Algorithm (BMA) [2]. The simple genetic operators of Microbial Genetic Algorithm (MGA) [7][8] were successfully used in BMA as well to solve the travelling salesman problem [4].

BEA is used primarily to discover or optimize fuzzy system parameters, but it is very useful in many other cases. It can be considered as a global search algorithm, where an exact answer is not always required; thus, a near optimal solution is acceptable. It is able to solve and quasi-optimize complex optimization and design problems having non-linear, high-dimensional, multi-modal, and non-continuous characteristics. This algorithm does not demand the use or the existence of derivatives of the objective functions, such as in the case of gradient-based methods.

BEA always keeps a record of the possible solutions. The solutions are also called bacteria, they form together the population. BEA includes two operations: the so-called "bacterial mutation", and the "gene transfer" operation of the PBGA. Bacterial mutation optimizes the bacteria individually. The recombination operation of the GA was replaced by gene transfer in PBGA and BEA. This operation allows bacteria to directly transfer information to other bacteria of the population (see Figure 1).

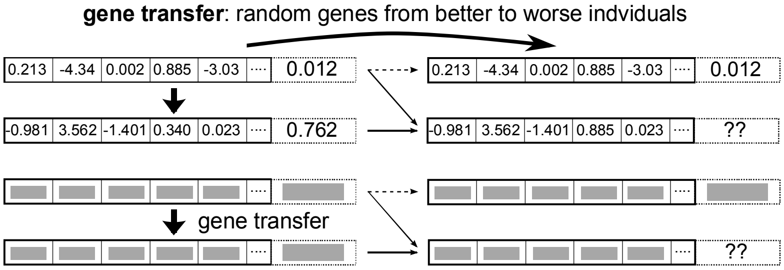


Figure 1. Gene transfer allows each bacterium to directly transfer genetic information into another bacterium.

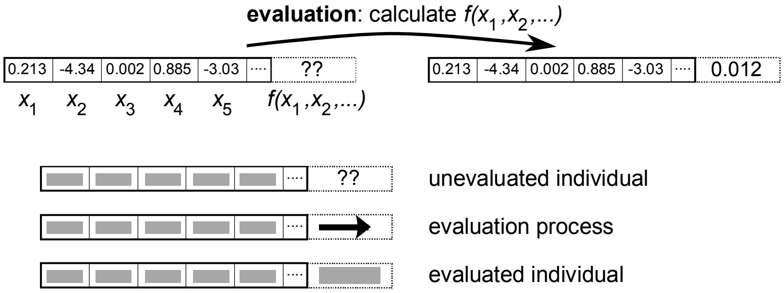


Figure 2. The destination bacterium has to be re-evaluated after the gene transfer.

Using these two operators (mutation and gene transfer) a new population can be created. A series of populations leads to the final population, when some kind of termination condition is fulfilled. The bacterium with the best objective function value will be accepted as the solution of the original problem. The exact details of the algorithm can be found in [13].

A significant number of real applications kill most of their time with the evaluation of the objective functions (see Figure 2), because both of the operations mentioned above use the objective values frequently. In a complex optimization problem evaluation of the objective function may require a complete finite element calculation or solving a system of ordinary differential equations. (See e.g. [9].) In such cases the cost of an objective function evaluation is much higher than the one of starting a new process or communicating with another machine. Thus, if the optimization problem is a complex one, simultaneous evaluation of the objective function for different individuals can save a lot of time.

There is a trivial way of making the bacterial mutation parallel, because all the bacteria can be mutated at the same time. All clones of a specified bacterium can be evaluated simultaneously as well, but only one gene can be modified in each computational cycle, i.e. several genes cannot be modified parallel. If the population consists of N_p bacteria, the number of clones is N_c and every bacterium contains N_g genes, the mutation operation evaluates the objective function $E_o = N_p N_c N_g$ times.

With $N_{cpu} = N_p N_c$ processors it is possible to calculate the objective values in N_g computational rounds, thus the speed-up of the calculation can be theoretically even $S_m = \frac{E_o}{N_g} = N_p N_c$ [10]. (It is assumed in the article, that the time of objective function evaluation is constant and independent of the content of the chromosome.) In most cases N_{cpu} is much bigger then the number of available CPUs in today's systems, so it can be declared, that bacterial mutation in its original form is suitable to run on a multicore/multiprocessor system. (In a typical calculation $N_p \approx 30-100$, $N_c \approx 20-50$, therefore $N_{cpu,\max} \approx 600-5000$.)

The second main operation of the BEA, the gene transfer sorts the bacteria based on their objective values. After then, it divides the population in two halves. The bacteria with better objective values get into the superior half, the others in the inferior half. After then, the algorithm selects randomly the source bacterium from the superior half, and the destination bacterium from the inferior half. Some selected parts of the source bacterium will be transferred to the destination bacterium, so the objective value of the modified bacterium has to be recalculated. The whole process starting from the sorting of the population is repeated several (N_{gt}) times (see Figure 3 and legend in Figure 2).

It is obvious, that the newly evaluated bacterium may belong to any half of the population, according to its objective value and can cause an originally superior individual to become inferior. Thus the successive gene transfers are not independent, so it is impossible to evaluate the objective value of more bacteria at the same time in the original form of BEA. The only way of making the gene transfer operator parallel, is to create an alternative, new version of it. In the next section, the authors propose a variant of the original BEA, which permits the parallel evaluation of new individuals. Furthermore we examine a variant of the MGA, which is similar to BEA but can be made parallel at a limited degree. Note the parallel nature of MGA is mentioned in [4], however here we show an implementation which guaranties the parallel evaluation in the case of arbitrary number of processors. Finally we show a 4th version, a fully parallel MGA-variant.

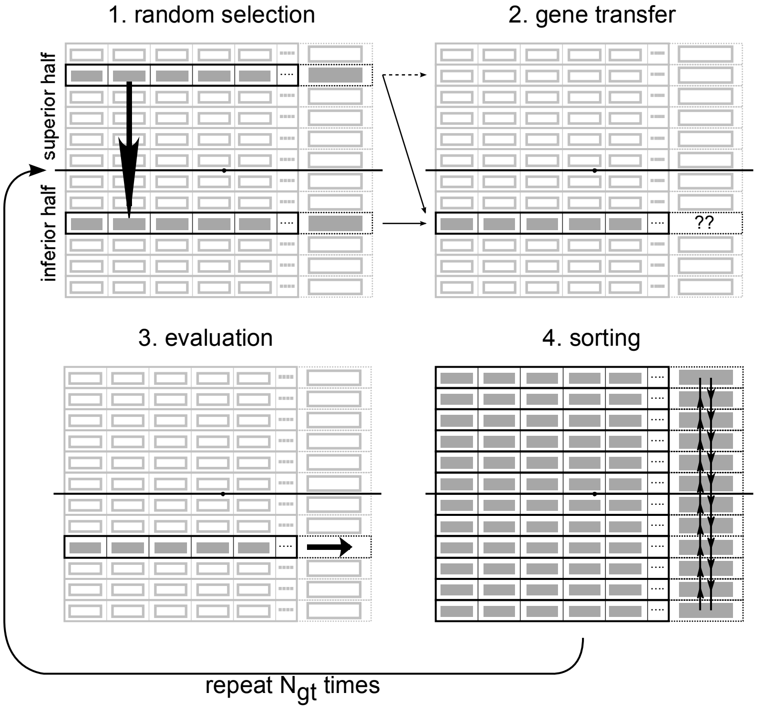


Figure 3. Operation of the original gene transfer

2. New kinds of gene transfer operators

2.1. Original gene transfer with auxiliary population

In this version the sorting of the population on the grounds of the objective values and selection of source and destination bacteria from inferior and superior half are the same as before. The main idea of this version is to keep the population untouched and collect the modified (infected) individuals to an "auxiliary population", and start a parallel evaluation of these individuals only when this auxiliary population is full. After this evaluation, the original and auxiliary populations are sorted together and the best N_p will be the new population, and the worse N_{aux} will be deleted. This has to be repeated until N_{gt} gene transfers are executed.

The size of the auxiliary population (N_{aux}) is a free parameter of the algorithm, but practically it is a multiple of the number of available processors.

The operation can be described with the following C-based pseudo code. The same algorithm is visualized in Figure 4.

```
/*
   gt: the remaining number of gene transfers
   numberOfGTs: the total number of gene transfers per generation
   auxPopSize: size of the auxiliary population
*/
for(gt=numberOfGTs; gt>0; gt-=auxPopSize) {
   sort(population);
   numberOfNewBacteria = gt>auxPopSize?auxPopSize:gt;
   for(auxIndex=0; auxIndex<numberOfNewBacteria; auxIndex++) {
      src = rnd(0, popSize/2);  /* rnd(a, b) returns an integer in the range [a; b) */
      dest = rnd(popSize/2, popSize);
      newBact = population[dest];
      gene = rnd(0, numberOfGenes);
      newBact[gene] = src[gene];  /* transferring a gene from source to destination */
      auxPop[auxIndex] = newBact;
   }
   evaluateObjectiveFnsParallel(auxPop);
   sort(auxPop);
   /* copy the best popSize pieces of bacteria from population and auxPop to population
*/
   merge(population, auxPop, popSize);
}
```

An interesting property of the original gene transfer is that the objective value of the infected destination bacterium can be worse, than the former value. As opposed to this, using auxiliary population only better bacteria can replace old ones. It can be considered as a kind of elitism. However, there is a big danger in using the auxiliary population: it may make the genetic diversity worse. It may happen that the profit of parallel evaluation will be less than the deficit of this effect. Therefore the performance will be tested in several cases to measure the whole behaviour of this new version.

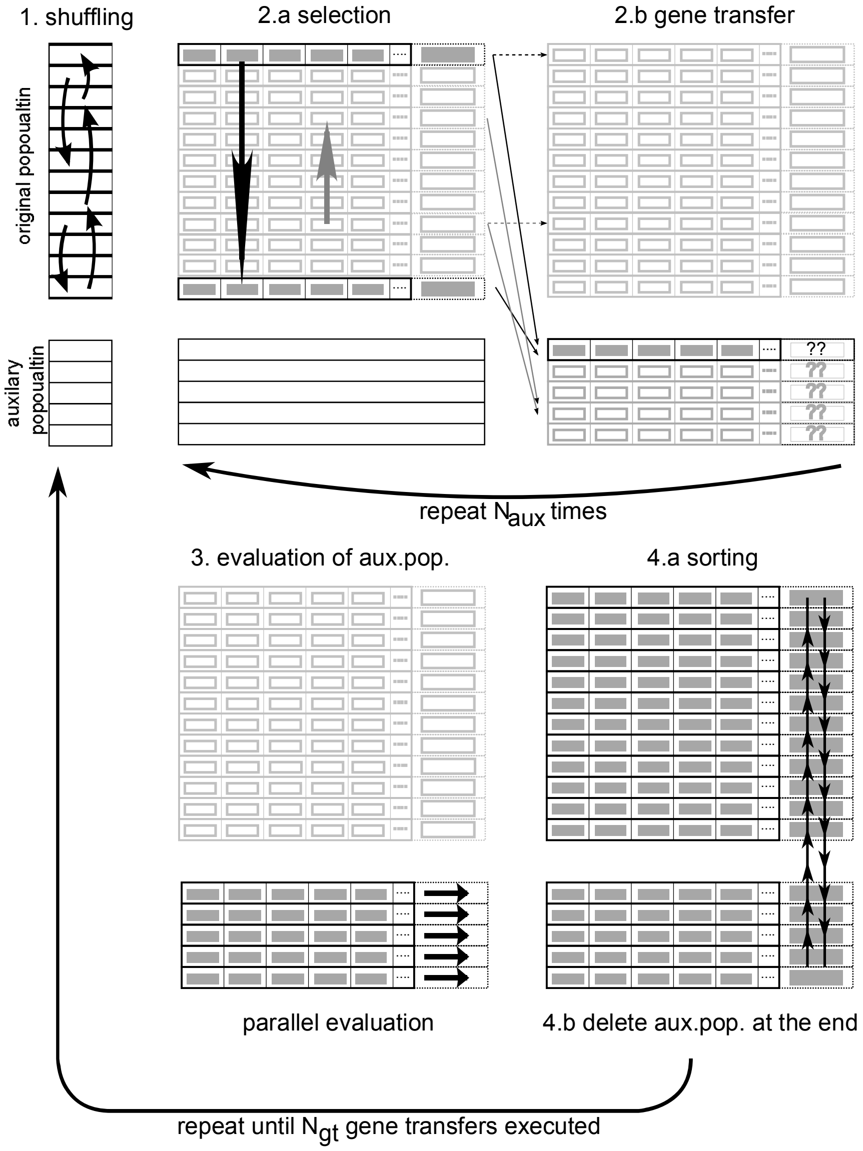


Figure 4. Operation of the original gene transfer with auxiliary population

2.2. Gene transfer inspired by Microbial Genetic Algorithm

The goal of creating the Microbial Genetic Algorithm (MGA) was to strip down the original GA and reduce it to its basics. In one cycle the MGA picks two individuals at random from the population. The one with the better objective function is called winner, the other is the loser. During recombination the winner infects the loser with some portion of its genetic material. The recombination is followed by mutation and the replacement of the loser with its genetically modified chromosome in the population.

The MGA gave the idea of creating a new gene transfer operation (hereafter this operation is signed shortly as MGA in figures and tables), because its selection and recombination can be regarded as a gene transfer operation. It is necessary to create disjoint source-destination pairs for the sake of parallel execution of objective functions. The number of such pairs cannot be more than half of the population size at a time. If more gene transfers are needed, the process has to be repeated.

The following pseudo-code and Figure 5 shows the MGA-based gene transfer method.

```
src = popSize;  /*index of source bacteria */
dest = src-1;  /* index of destination bacteria */
/* numberOfGTs: the total number of gene transfers per generation */
for(gt=0; gt<numberOfGTs; gt++) {
   if(src>=dest) {
      /* evaluate popSize-dest-1 pieces of bacteria from dest+1 */
      evaluateObjectiveFnsParallel(population, dest+1, popSize-dest-1);
      src = 0;
      dest = popSize-1;
      shuffle(population);
   }
   /* move the loosers back (lower objective value is better) */
   if(objectiveValue(population[src]) > objectiveValue(population[dest])) {
      commute(population[src], population[dest]);
   }
   gene = rnd(0, numberOfGenes);  /* rnd(a, b) returns an integer in the range [a; b) */
   /* transferring a gene from source to destination */
   population[dest][gene] = population[src][gene];
   src++;
   dest--;
}
/* evaluate the remainders */
evaluateObjectiveFnsParallel(population, dest+1, popSize-dest-1);
```

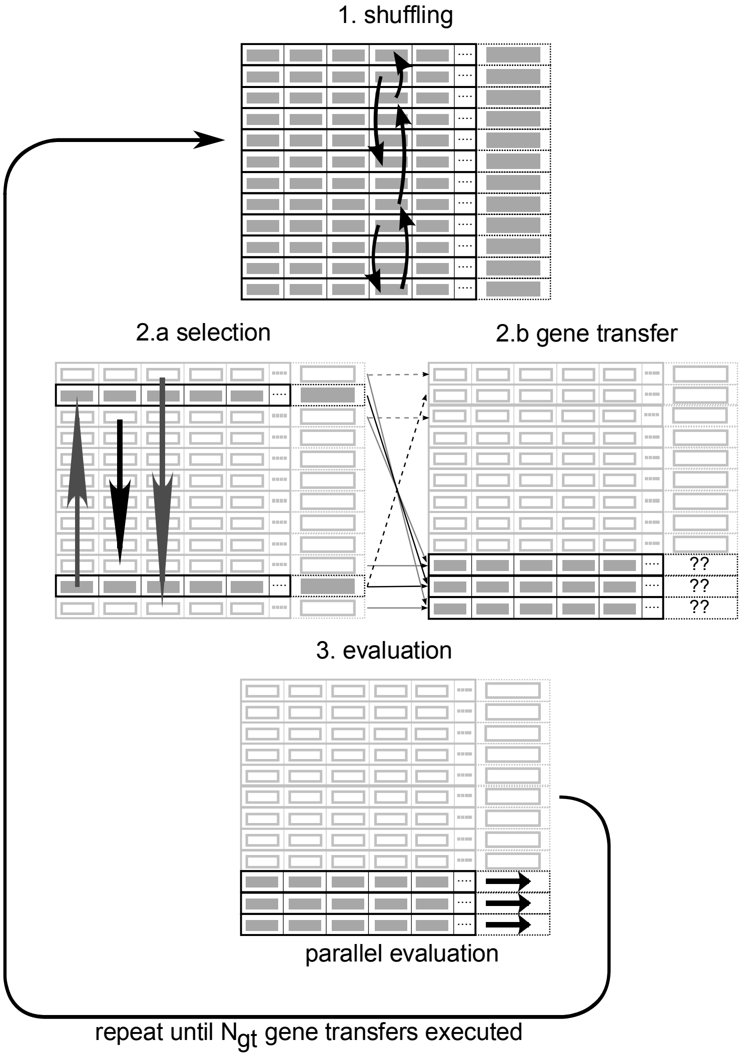


Figure 5. Operation of the gene transfer inspired by Microbial Genetic Algorithm

The “shuffling” step is a simple algorithmic trick: after randomization of positions in the population, any fixed pair-making strategy will produce random, disjoint pairs. We choose an inward advancing coupling strategy: the first pair will contain the 1st and ${N_p}^{th}$ element (from shuffled population), the second pair will contain the 2nd and $N_p - 1^{th}$ element, and so on.

2.3. Gene transfer inspired by MGA with auxiliary population

The MGA inspired gene transfer can be extended with an auxiliary population. In this case the $N_p / 2$ limit for the number of parallel evaluations can be exceeded.

This algorithm picks two bacteria at random. A new bacterium will be created in the auxiliary population, using the previously selected two bacteria's genetic information. The new bacterium inherits the genes of the destination bacteria, except one, which comes from the source. The auxiliary population can be filled with repeated infections, and all the bacteria of the auxiliary population can be evaluated simultaneously. The original and auxiliary populations have to be merged and the N_{aux} least fit bacteria have to be dropped.

This method is elitist, and the number of parallel evaluations is not limited to half of the population size.

The following pseudo-code and Figure 6 shows the MGA-based gene transfer method with auxiliary population.

```
/*
  gt: the remaining number of gene transfers
  numberOfGTs: the total number of gene transfers per generation
  auxPopSize: size of the auxiliary population
*/
for(gt=numberOfGTs; gt>0; gt-=auxPopSize) {
  numberOfNewBacteria = gt>auxPopSize?auxPopSize:gt;
  for(auxIndex=0; auxIndex<numberOfNewBacteria; auxIndex++) {
    src = rnd(0, popSize);  /* rnd(a, b) returns an integer in the range [a; b) */
    dest = rnd(0, popSize);
    /* The bacteria with smaller objective fn. value is better */
    if(objectiveValue(population[src]) > objectiveValue(population[dest])) {
      commute(src, dest);
    }
    newBact = population[dest];
    gene = rnd(0, numberOfGenes);
    newBact[gene] = src[gene];  /* transferring a gene from source to destination */
    auxPop[auxIndex] = newBact;
  }
  evaluateObjectiveFnsParallel(auxPop);
  sort(auxPop);
```

```
        /* copy the best popSize pieces of bacteria from population and auxPop to population
    */
        merge(population, auxPop, popSize);
    }
```

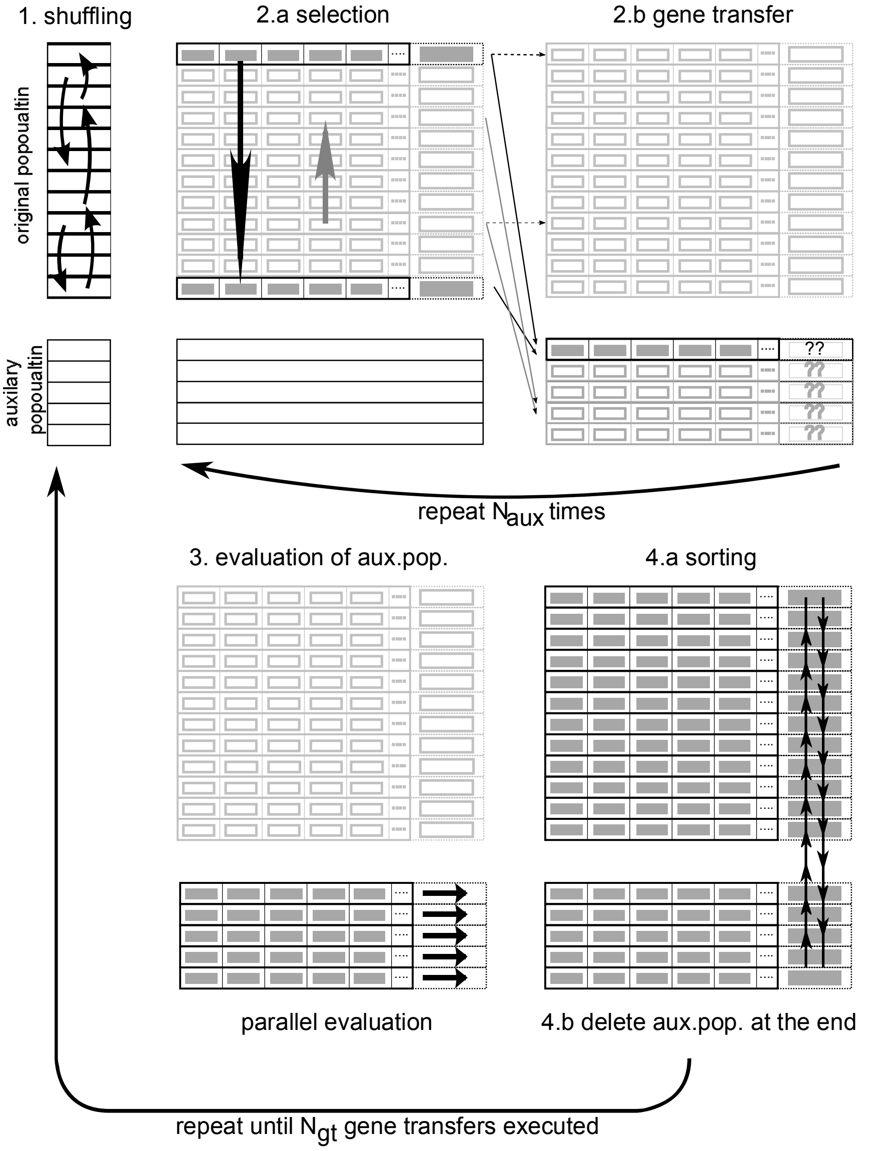


Figure 6. Operation of the gene transfer inspired by MGA with auxiliary population

All the gene transfers above can be improved with a simple modification. Copying more genes from the source bacteria to the destination at once typically increases the speed of convergence, but the chance of premature convergence as well. The analysis of this phenomenon is not subject of this article, but we remark that we used identical settings throughout the execution of the test functions to avoid the side effects.

3. Test functions

The working of the original and the suggested three new gene transfer operators was tested on artificial test functions. These well-known functions [3][12][16] were selected on the basis of their properties. Table 1 summarizes some properties of the functions.

Table 1. Properties of the test functions

	Linearity	**Modality**	**Continuity**	**Plateaux/shoulder**
De Jong's 1st (sphere)	yes	unimodal	yes	no
De Jong's 3rd	yes	multimodal	no	yes
Step	yes	multimodal	no	yes
Rastrigin's	no	multimodal	yes	no
Keane's	no	multimodal	no	no

The equations of the test functions are the following.

$$f_{DeJong1} = \sum_{i=1}^{n} x_i^2, -5.12 \le x_i \le 5.12 \tag{1}$$

$$f_{DeJong3} = \sum_{i=1}^{n} \lfloor x_i \rfloor, -5.12 \le x_i \le 5.12 \tag{2}$$

$$f_{Step} = \sum_{i=1}^{n} \lfloor x_i + 0.5 \rfloor^2, -5.12 \le x_i \le 5.12 \tag{3}$$

$$f_{Rastrigin} = n \cdot 10 + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) \right), \quad -5.12 \le x_i \le 5.12 \tag{4}$$

$$f_{Keane} = \left| \frac{\left(\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i) \right)}{\sqrt{\sum_{i=1}^{n} i x_i}} \right|, \quad \begin{array}{l} \prod_{i=1}^{n} x_i \ge 0.75, \\ \sum_{i=1}^{n} x_i \le 7.5n, \end{array} \quad 0 \le x_i \le 10 \tag{5}$$

4. Test environment of the measurements

The BEA used in the measurements was a custom implementation, which has some interesting properties.

- 20 floating point genes have been used because the aim is to investigate optimization methods of continuous problems arising in engineering.
- The software is able to handle multiple stop conditions, but only the wall clock time limit of the simulation was used. The application terminates only if all the operations of the current population are finished. All the measurements were repeated 20 times, and the average of them is considered a result. The standard deviation of these values was also calculated to determine the statistical uncertainty. Note that the artificial delay in the objective function is necessary, if we want to measure how efficient a BEA-like method in complex problems will be, where the evaluation of individuals take much more time than the communication between computational nodes. However, the standard test functions above are better for testing than complex real-life problems, because their qualitative properties and global maximum is well known from the literature and their CPU-usage is limited, which allows us to perform a high number of test calculations.
- Master-slave style parallelism was used in order to speed up the optimisation process. The master CPU was responsible to execute all the bacterial operators and handle the population. The only job of the slave CPUs were to calculate the values of the objective function of the bacteria. Because the test functions are relative easy to evaluate, a small delay (approx. 0.005 second) was built in the objective function. Without this modification the overhead of the communication between the CPUs would be disproportionately high and could distort our results.
- The communication between the CPUs has been realised with MPICH 1.0, an implementation of the Message Passing Interface.

The software ran on a cluster of workstations containing 1 master + 16 slave computers connected with Ethernet network. Nine of the slave machines were equipped with Intel Pentium IV CPUs running at 2.4 GHz, the other machines were built with more up-to-date Hyper-Threaded Intel Pentium IV CPUs running at 3.0 GHz. The difference in the computing power of the slave machines did not mean a problem, because the delay of the objective function and the small amount of code practically removed all the differences.

5. Test results

5.1. Rastrigin function

Due to the size limits of the paper, it is impossible to display all data in maximum detail. We decided to show most of the results of the test calculations with the Rastrigin function, but from the rest of the functions we give a short summary only.

The global minimum of the Rastrigin function is at $x_i = 0$ for all i values, and here the value of the objective function is 0. (Every other places $f_{Rastrigin}$ has positive value.) Therefore if we show the best objective function value (B) of the population as a function of wall-clock time (T_{wall}), we get curves that asymptotically converge to 0. The faster the curve approaches zero, the better the method is. We can observe this nature in Figure 7, 8 and 9, which shows the results for 1, 4 and 16 processors respectively.

We can conclude that during the optimisation of the Rastrigin function, the original gene transfer performs well if only one CPU is available, and the MGA inspired gene transfer needs the most time to finish. On the other hand, the performance of the original gene transfer is very weak if it is running on more CPUs. All the three new alternatives behave very well, but in general, the ones with auxiliary population end their job faster, however the difference between them is not significant.

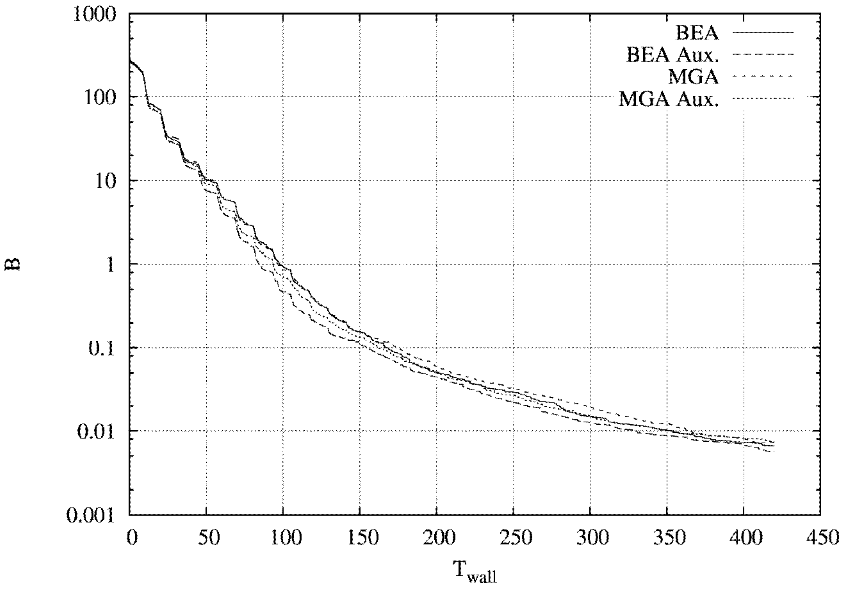


Figure 7. Optimisation of Rastrigin's function, using 1 CPU

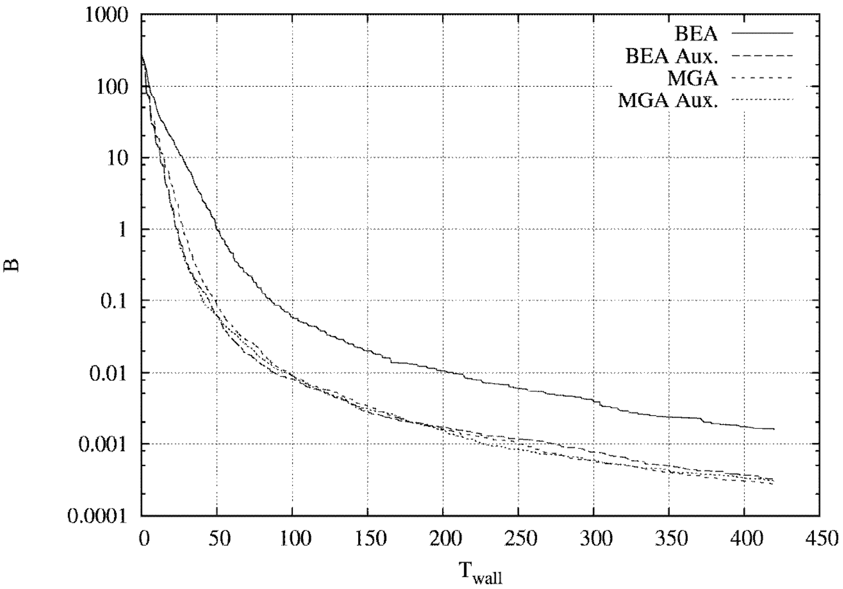


Figure 8. Optimisation of Rastrigin's function, using 4 CPUs

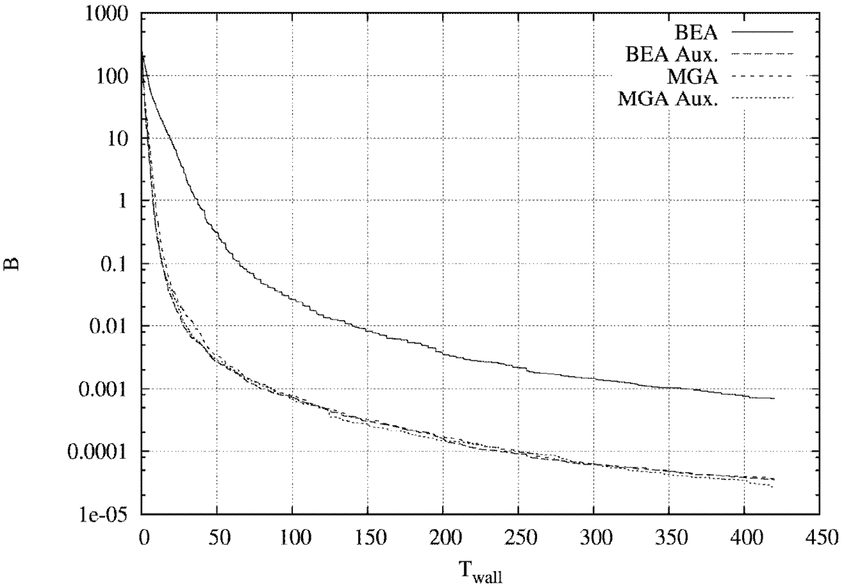


Figure 9. Optimisation of Rastrigin's function, using 16 CPUs

Figure 10 shows how the original BEA scales for $N_{cpu} = 1,2,4,8,16$. However, even if one can observe some improvement at every increase in the number of processors, the scaling is far from ideal. On the other hand, BEA with auxiliary population scales very well, which is demonstrated in Figure 11.

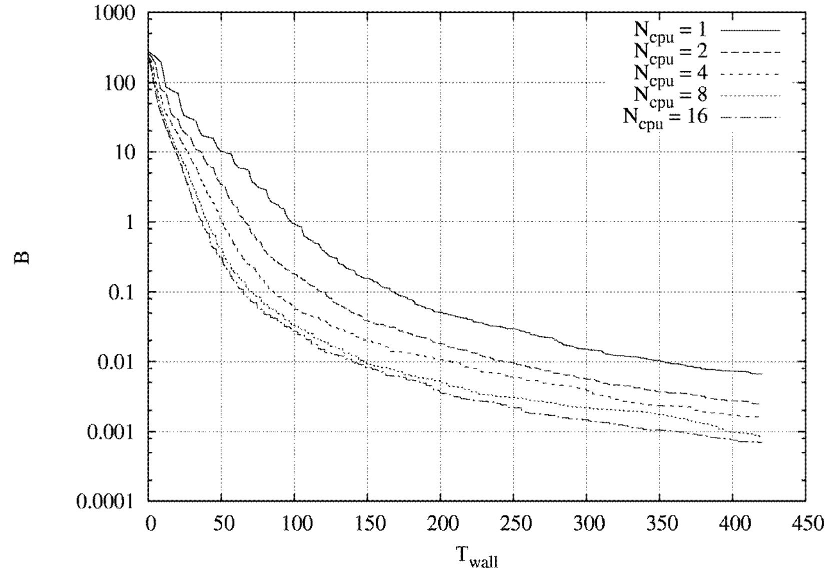


Figure 10. Optimisation of Rastrigin's function, using original gene transfer operation of BEA

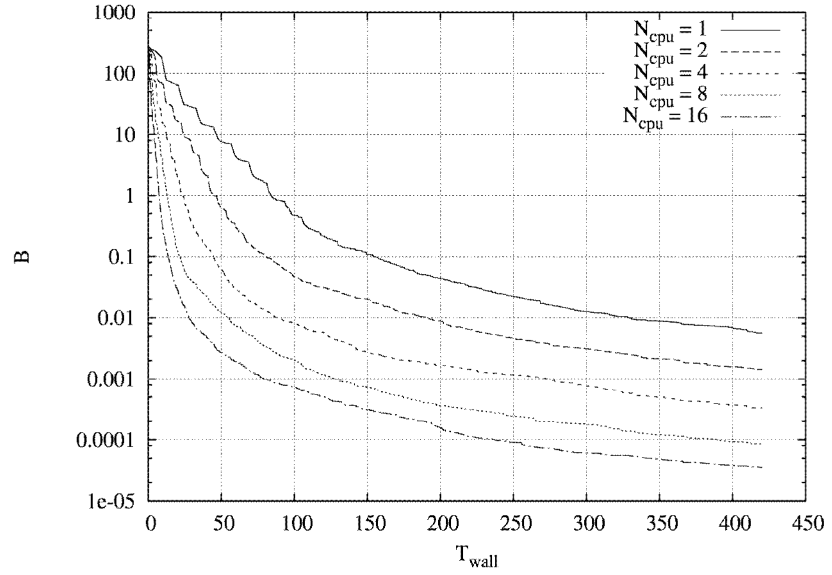


Figure 11. Optimisation of Rastrigin's function, using original gene transfer operation of BEA with auxiliary population

The effectiveness of the proposed methods can be shown in a different way too. We can show the time needed to reach a specific value of the objective function which is near the optimum. We performed calculations for 3 different values (0.05, 0.15 and 0.01). They gave qualitatively the same results, therefore here we only show the T_{wall} needed to reach 0.01. Figure 12 shows this wall-clock time as a number of processors for the 4 examined method.

In an ideal case the measured points in Figure 12 will be on straight lines with slope – 1. This is clearly not fulfilled in the original BEA, but the other methods show this nature approximately.

Note that we visualized the uncertainty of measured values in Figure 12. The small error bars show the standard deviation of the average values of the 20 different test calculations.

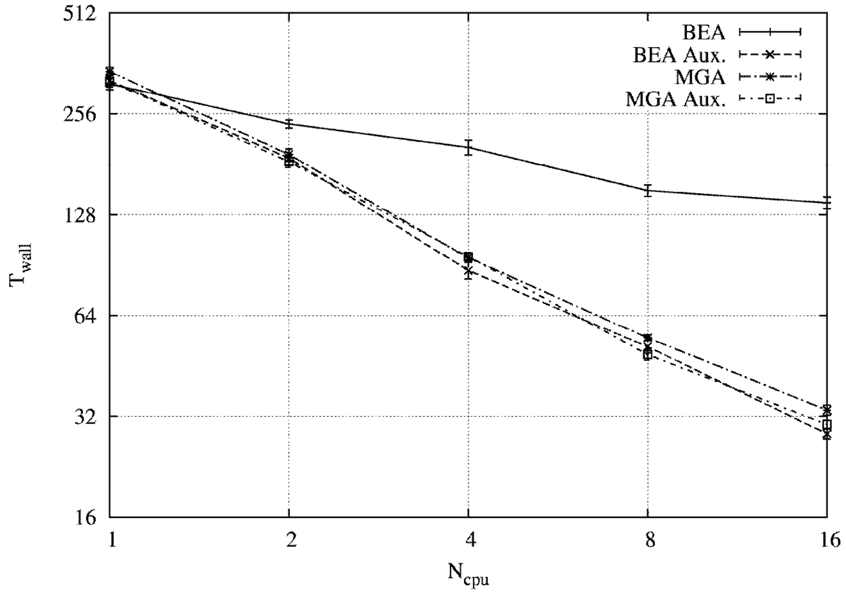


Figure 12. Optimisation of the Rastrigin function with different gene transfer methods and different number of CPUs

It is more precise but less expressive to show the numerical values of test calculations. Table 2 shows the wall-clock time, its standard deviation, the calculated parallel effectiveness and its standard deviation for the same case as in Figure 12. The corresponding tables for 0.015 and 0.05 objective functions value were calculated, but as they are significantly similar and the 0.01 case is so strong, that we do not publish them to keep the extent of paper reasonable.

Table 2. Time and efficiency values of the optimisation of the Rastrigin function. The software stopped when the objective value reached 0.01.

	BEA	**BEA Aux.**	**MGA**	**MGA Aux.**
T*	314.461	321.165	342.870	319.934
σ(T*)	12.814	13.440	10.232	10.748
T_2	238.598	189.014	193.519	184.372
$\sigma(T_2)$	6.601	4.699	7.537	7.159
E_2	0.659	0.850	0.886	0.868
$\sigma(E_2)$	0.032	0.041	0.043	0.045
T_4	203.384	87.132	95.193	95.989
$\sigma(T_4)$	10.219	4.835	2.535	2.825
E_4	0.387	0.921	0.900	0.833
$\sigma(E_4)$	0.025	0.064	0.036	0.037
T_8	151.276	51.559	54.813	48.813
$\sigma(T_8)$	5.977	2.270	1.082	1.875
E_8	0.260	0.779	0.782	0.819
$\sigma(E_8)$	0.015	0.047	0.028	0.042
T_{16}	139.205	28.382	33.388	30.294
$\sigma(T_{16})$	5.575	0.998	1.163	0.919
E_{16}	0.141	0.707	0.642	0.660
$\sigma(E_{16})$	0.008	0.039	0.029	0.030

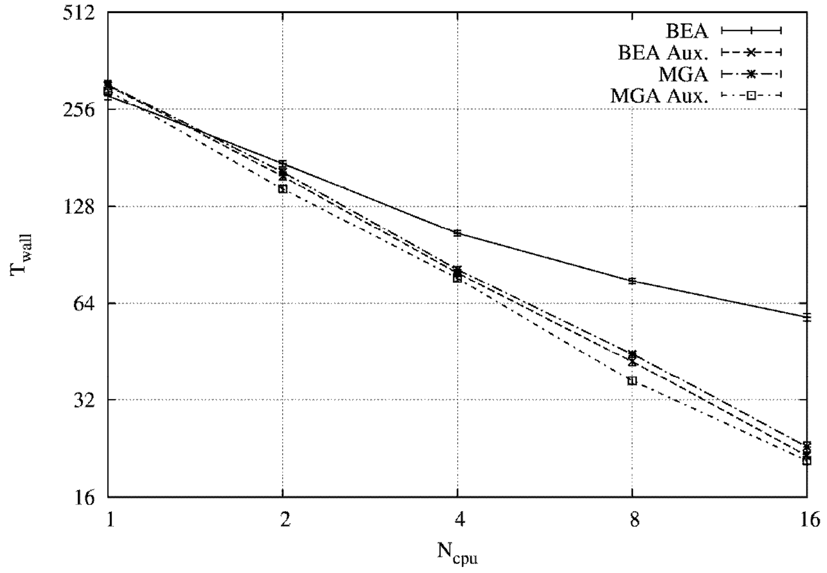


Figure 13. Optimisation of the Keane function with different gene transfer methods and different number of CPUs

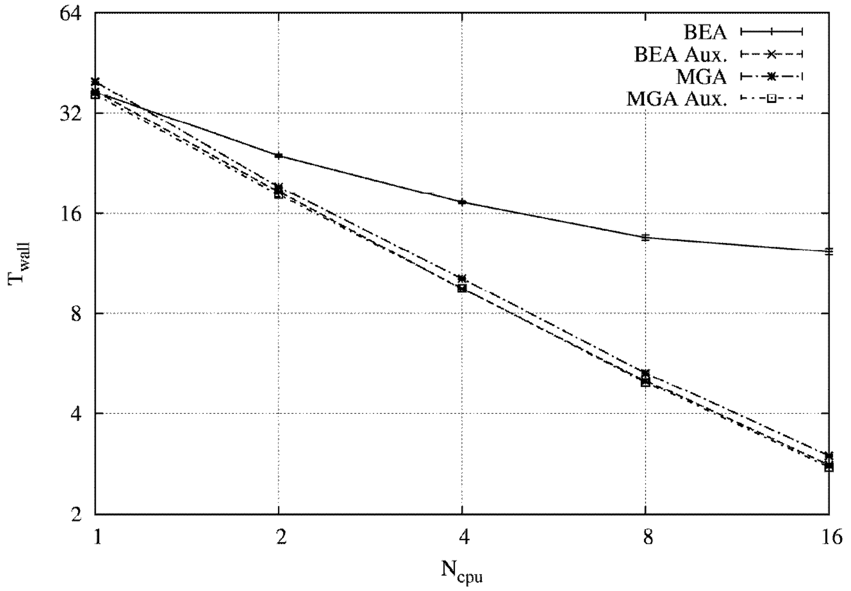


Figure 14. Optimisation of the Step function with different gene transfer methods and different number of CPUs

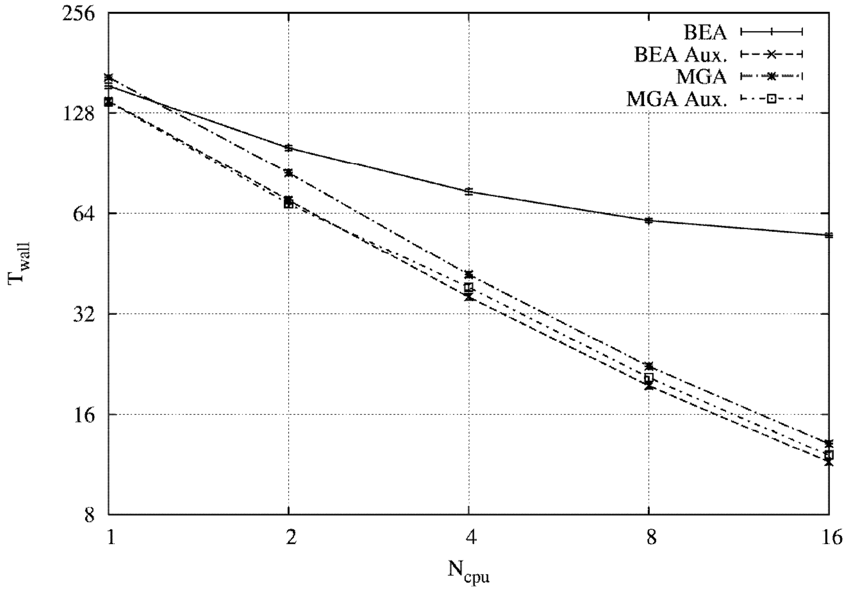


Figure 15. Optimisation of De Jong's 1st function with different gene transfer methods and different number of CPUs

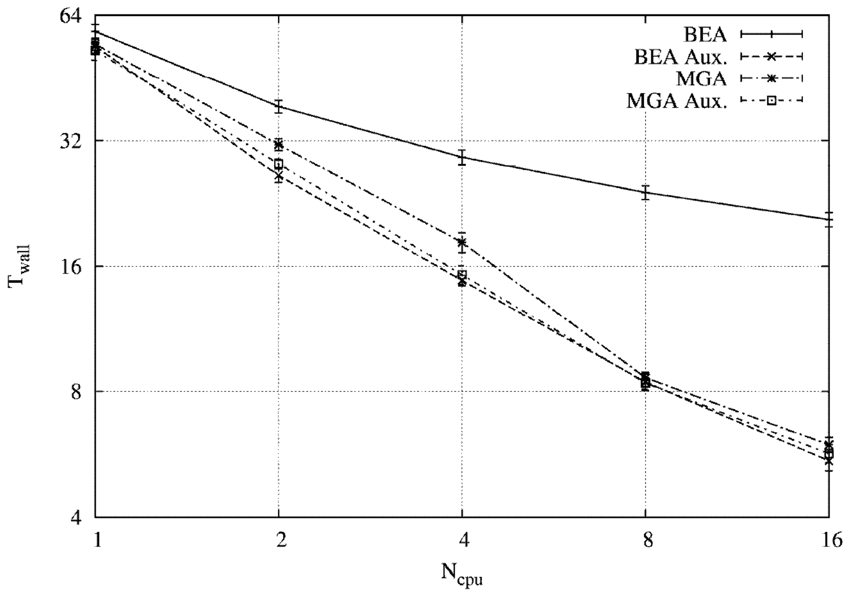


Figure 16. Optimisation of De Jong's 3rd function with different gene transfer methods and different number of CPUs

5.2. Other test functions

Here we present a small number of graphs and tables from the other 4 test functions.

Figure 13, 14, 15 and 16 are the correspondent graphs of Figure 12.

We can draw similar inferences in connection with the optimisation of the Keane function as well. With one CPU, the original gene transfer is competitive, but using more CPUs, its arrears are evident. The winner of the optimisation is the MGA inspired gene transfer with auxiliary population, according to wall clock time and efficiency. The second best method is the original gene transfer with auxiliary population. Note, that the difference between methods is not significant: the error bars of the standard deviation of values are often overlapping.

Table 3. Time and efficiency values of the optimisation of the Keane function. The software stopped when the objective value reached 0.76.

	BEA	BEA Aux.	MGA	MGA Aux.
T*	282.408	304.116	305.870	293.280
T_2	173.776	158.711	163.589	144.986
E_2	0.813	0.958	0.935	1.011
T_4	105.415	79.412	81.359	76.441

	BEA	BEA Aux.	MGA	MGA Aux.
$\mathbf{E_4}$	0.670	0.957	0.940	0.959
$\mathbf{T_8}$	74.974	42.197	44.603	36.697
$\mathbf{E_8}$	0.471	0.901	0.857	0.999
$\mathbf{T_{16}}$	57.980	21.524	22.924	20.803
$\mathbf{E_{16}}$	0.304	0.883	0.834	0.881

Table 3 shows wall-clock time needed to reach the objective function value 0.76 and efficiency values for Keane's test function. For simplicity, we neglected the standard variances of the displayed values. (The relative standard deviations were between 1 and 4% for T_{wall} and 3-5% for efficiency.)

Not surprisingly, the optimisation of the step function has ended with a similar result. The two kinds of gene transfer with auxiliary population performed practically the same, and they are the winner of the optimisation. (The relative standard deviations were 0-2% for T_{wall} and 3-5% for efficiency.) The MGA inspired gene transfer was the slowest of the three new alternative methods, but its high efficiency is remarkable. Its disadvantage decreases by the increase of the number of the CPUs. For much higher number of processors, MGA may be the winner.

Table 4. Time and efficiency values of the optimisation of the step function. The software stopped when the objective value reached 0.

	BEA	BEA Aux.	MGA	MGA Aux.
$\mathbf{T^*}$	37.011	37.223	39.888	36.342
$\mathbf{T_2}$	23.816	18.590	19.234	18.220
$\mathbf{E_2}$	0.777	1.001	1.037	0.997
$\mathbf{T_4}$	17.257	9.485	10.184	9.516
$\mathbf{E_4}$	0.536	0.981	0.979	0.955
$\mathbf{T_8}$	13.538	5.032	5.316	4.981
$\mathbf{E_8}$	0.342	0.925	0.938	0.912
$\mathbf{T_{16}}$	12.272	2.808	2.991	2.768
$\mathbf{E_{16}}$	0.188	0.828	0.833	0.820

In case of optimising De Jong's 1st function, it is worth using gene transfers with auxiliary population even in the case of one processor (see Table 5). MGA inspired gene transfer is the slowest of the three new operations, but its high efficiency helps reducing its disadvantages if high number of processor is available. (The relative standard deviations were 1-3% for T_{wall} and 2-3% for efficiency.)

Table 5. Time and efficiency values of the optimisation of De Jong's 1st function. The software stopped when the objective value reached 0.001.

	BEA	**BEA Aux.**	**MGA**	**MGA Aux.**
T*	154.714	139.269	164.104	138.584
T_2	100.573	69.999	84.420	68.255
E_2	0.769	0.995	0.972	1.015
T_4	74.202	35.933	41.883	38.397
E_4	0.521	0.969	0.980	0.902
T_8	60.917	19.485	22.279	20.637
E_8	0.317	0.893	0.921	0.839
T_{16}	55.086	11.497	13.069	12.041
E_{16}	0.176	0.757	0.785	0.719

With optimising De Jong's 3rd function (Table 6), all new operations are faster than the original gene transfer, and the difference grows by the increase of the number the processors. The two operations with auxiliary population are the best in this comparison. (The relative standard deviations were 3-5% for T_{wall} and 4-7% for efficiency.)

Table 6. Time and efficiency values of the optimisation of De Jong's 3rd function. The software stopped when the objective value reached 119.9.

	BEA	**BEA Aux.**	**MGA**	**MGA Aux.**
T*	58.537	53.980	54.566	52.711
T_2	38.584	26.380	31.309	28.152
E_2	0.759	1.023	0.871	0.936
T_4	29.203	14.786	18.261	15.257
E_4	0.501	0.913	0.747	0.864
T_8	24.018	8.422	8.622	8.382
E_8	0.305	0.801	0.791	0.786
T_{16}	20.701	5.449	5.956	5.669
E_{16}	0.177	0.619	0.573	0.581

6. Conclusion

Based on the extended test calculations with 5 different standard test functions, we can conclude that all the three proposed alternative gene transfer operations are well suited for parallel execution. In general, the auxiliary population increased the speed of convergence, thus its use is recommended. The MGA inspired gene transfer performs better with the auxiliary population, but its simplicity and high efficiency makes its implementation a reasonable decision without auxiliary population, too. In many cases, the usage of some of the new alternative methods is effective even if only one CPU is available.

For complex objective functions (e.g. the ones that arise in engineering optimization problems) the original BEA is a bad choice due to its bad scaling properties. For at most 16 processors the BEA or MGA-based method with auxiliary population appears to be the good choice. (There is no significant difference between them.)

The efficiency values show that for some cases the MGA-like version (without auxiliary population) has the best scaling from the four methods examined here, therefore it is possible, that for much higher number of processors, MGA will be the winner in this case. We will examine this effect in a subsequent paper.

Acknowledgement

The authors' research is supported by the National Development Agency and the European Union within the frame of the project TAMOP 4.2.2-08/1-2008-0021 at the Széchenyi István University entitled "Simulation and Optimization – basic research in numerical mathematics".

References

[1] Bäck, T., Fogel, D. B., Michalewicz, Z.: *Handbook of Evolutionary Computation*, IOP Publishing and Oxford University Press, (1997).

[2] Botzheim, J., Cabrita, C., Kóczy, L. T., Ruano, A. E.: *Fuzzy rule extraction by bacterial memetic algorithms*, Proceedings of the 11th World Congress of International Fuzzy Systems Association, IFSA 2005, Beijing, China, (July 2005), pp. 1563–1568.

[3] De Jong, K. A.: *An analysis of the Behavior of a Class of Genetic Adaptive Systems*, dissertation, University of Michigan, (1975).

[4] Farkas, M., Földesi, P., Botzheim, J., Kóczy, L. T.: *A Comparative Analysis of Different Infection Strategies of Bacterial Memetic Algorithms*, INES 2010 14th International Conference on Intelligent Engineering Systems, Las Palmas of Gran Canaria, Spain, (May 5–7, 2010).

[5] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc., (1989).

[6] Grosan, C., Abraham, A.: *Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews*, Studies in Computational Intelligence, Vol. 75/2007, (2007), pp. 1-17.

[7] Harvey, I.: *The Microbial Genetic Algorithm*, in Proceedings of the Tenth European Conference on Artificial Life, editor G. Kampis et al, Springer LNCS., (2009).

[8] Harvey, I.: *The Microbial Genetic Algorithm*, School of Cognitive and Computing Sciences, University of Sussex, submitted as a Letter to Evolutionary Computation, (1996), unpublished report (http://www.informatics.sussex.ac.uk/users/inmanh/Microbial.pdf)

[9] Horváth, A., Horváth Z.: *Optimal shape design of diesel intake ports with evolutionary algorithm*, Proceedings of 5th European conference on numerical mathematics and advanced applications (ENUMATH 2003), edited by Feistauer, M. et al., Springer Verlag, (2004), pp. 459–470.

[10] Li, K., Pan, Y., Shen, H., Zheng, S. Q.: *A Study of Average-Case Speedup and Scalability of Parallel Computations on Static Networks*, Mathematical and Computer Modelling 29, (1999), pp. 83-94.

[11] Moscato P.: *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms*, Caltech Concurrent Computation Program, Tech. Rep., California, (1989).

[12] Mühlenbein, H., Schomisch, D., Born, J.: *The Parallel Genetic Algorithm as Function Optimizer*, Parallel Computing, Vol. 17, (1991), pp. 619–632.

[13] Nawa, N. E., Furuhashi, T.: *Fuzzy System Parameters Discovery by Bacterial Evolutionary Algorithm*, IEEE Transactions on Fuzzy Systems, Vol. 7, No. 5, (1999), pp. 608-616.

[14] Nawa, N. E., Hashiyama, T., Furuhashi, T., Uchikawa, Y.: *A study on fuzzy rules discovery using pseudo-bacterial genetic algorithm with adaptive operator*, Proceedings of IEEE Int. Conf. on Evolutionary Computation, ICEC'97, (1997).

[15] Nawa, N. E., Furuhashi, T.: *A Study on the Effect of Transfer of Genes for the Bacterial Evolutionary Algorithm*, Second Intenational Conference on Knowledge-Based Intelligent Electronic System, editors Jain, L. C., Jain, R. K., Adelaide, Australia, (21-23 April 1998), pp. 585-590.

[16] Törn, A., Zilinskas, A.: *Global Optimization*, Lecture Notes in Computer Science, No. 350, Springer-Verlag, Berlin, (1989).

Fuzzy Model based Prediction of Ground-Level Ozone Concentration

Z. C. Johanyák[1], J. Kovács[2]

Institute of Information Technologies,
Kecskemét College, GAMF Faculty,
Kecskemét, Hungary
Email: johanyak.csaba@gamf.kefo.hu[1]

Denis Gabor College (DGC),
Budapest, Hungary
Email: kovacsj@gdf.hu[2]

Abstract: Ground-level ozone is a dangerous pollutant for which the prediction of the concentration could be of great importance. In this paper, we present and compare three fuzzy models aiming the forecasting of ground-level ozone concentration. The models apply Takagi-Sugeno, respective LESFRI fuzzy inference techniques and were generated using the ANFIS method of the Matlab's Fuzzy Logic ToolBox, respective the RBE-DSS method of the SFMI toolbox. Although all of the methods proved to be applicable the model using LESFRI ensured the best results with a low number of rules.

Keywords: Ground-level ozone, fuzzy models, LESFRI, ANFIS, RBE-DSS

1. Introduction

The analysis and forecasting of air quality parameters are important topics of atmospheric and environmental research. In many of the applications, data are generated in the form of a time series. Therefore, time series analysis is a major task in forecasting average ozone concentrations, where one tests and predicts known or estimated observations for past times using them as input into the model to see how well the output matches the known observations.

Ground-level ozone (O_3) is one of the air pollutants of most concern in Europe. It is an irritating and reactive component in atmosphere that has negative impacts on human health, climate, vegetation and materials [23].

Ground-level ozone is a highly reactive oxidant and is unique among pollutants because it is not emitted directly into the air [20]. It is a secondary pollutant that results from complex chemical reactions in the atmosphere. In the presence of the sun's ultraviolet radiation (RAD), oxygen (O_2), nitrogen dioxide (NO_2), and volatile organic compounds (VOCs) react in the atmosphere to form ozone and nitric oxide (NO) through the reactions given in (1) and (2)

$$NO_2 + h\nu \rightarrow NO + O\,, \tag{1}$$

$$O_2 + O \rightarrow O_3\,. \tag{2}$$

With regards to the prediction of O_3 concentrations, several studies have been published. Sousa, Martins, Alvim-Ferraz, and Pereira [27] applied multiple linear regression (MLR) and artificial neural networks (ANNs); Ozdemir, Demir, Altay, Albayrak, and Bayat [21] used ANNs; Al-Alawi, Abdul-Wahab, Bakheit [1] combined principal component regression and ANNs; Pires, Martins, Pereira and Alvim-Ferraz [22] developed three different models an MLR based, an ANN based and one based on multi-gene genetic programming (MGP), from which the last one (MGP) ensured the best predictions.

Fuzzy systems have been used successfully for numerous practical applications. Kovács and Kóczy [18] developed a fuzzy rule interpolation (FRI) based model for behaviour-based control structures; Johanyák and Ádámné [9] constructed fuzzy models for the prediction of thermoplastic composites' mechanical properties; Wong and Gedeon [33] as well as Johanyák and Kovács [13] developed FRI based systems for prediction of petrophysical properties. Hládek, Vaščák and Sinčák [5] proposed a hierarchical multi agent control system based on rule based fuzzy system for pursuit-evasion task. Despite their advantages and wide applicability area fuzzy logic based solutions for ozone concentration prediction have not been published previously.

Therefore our research aimed the development and analysis of two types of fuzzy systems one applying a traditional Takagi-Sugeno [28] inference method using a dense rule base and another applying fuzzy rule interpolation (FRI) based reasoning technique using a sparse rule base. The results proved the applicability of the above mentioned methods in this case as well.

The rest of this paper is organized as follows. Section II reviews briefly the applied methods. Section III introduces the experiments the data came from and the results of the modelling. The conclusions are drawn in section IV.

2. Fuzzy Modeling and Inference

A fuzzy rule based system describes usually a nonlinear mapping between inputs and outputs based on fuzzy set concept. One can assign to set A a characteristic function $\boldsymbol{x_A: X \rightarrow \{0,1\}}$, which can take only the 0 or 1 (crisp) numerical values in case of the classical set concept (3) and values from a continuous interval (usually [0,1]) in case of the fuzzy concept [34].

$$x_A = \begin{cases} 1, & if \quad x \in A \\ 0, & otherwise \end{cases} \tag{3}$$

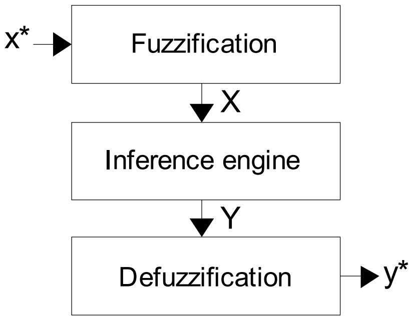


Figure 1. Block diagram of functioning of a fuzzy rule based system

In fuzzy logic the mapping of crisp inputs $\boldsymbol{x^*}$ to crisp outputs $\boldsymbol{y^*}$ generally is solved in three steps, which can be seen on Figure 1. where $\boldsymbol{x^*=(x_1^*, x_2^*,\ldots, x_n^*)}$ is the input, $\boldsymbol{y^*=(y_1^*, y_2^*,\ldots, y_m^*)}$ is the output, $\boldsymbol{n}$ is the number of input dimensions, and $\boldsymbol{m}$ is the number of output dimensions.

Depending on the number of input and output linguistic variables (dimensions) one can define four groups of fuzzy systems, i.e. multiple-input multiple-output (MIMO), multiple-input single-output (MISO), single-input multiple-output (SIMO), single-input single-output (SISO). In the case of ozone concentrations' forecasting we used MISO models. In the following subsections we review shortly the methods and tools we used for the generation of the three fuzzy models aiming the better prediction of ozone concentration.

2.1. Takagi-Sugeno type fuzzy inference

The mapping of inputs to outputs in a fuzzy system is determined by a set of "IF–THEN" rules of form

$$If\ X\ is\ A^i\ then\ Y\ is\ B^i\ ; i=1,\ldots,R, \tag{4}$$

where in case of a MISO system $\boldsymbol{X=(x_1, x_2, \ldots, x_n)}$ consists of a set of input variables, $\boldsymbol{Y}$ is the output variable, and $\boldsymbol{R}$ is the number of rules [31]. The fuzzy sets $\boldsymbol{A^i}$ = $(A_1^i, A_2^i,\ldots, A_n^i)$ and $\boldsymbol{B^i}$ are the antecedent and consequent parts of the fuzzy rules.

The Takagi-Sugeno type fuzzy system [28] also called "functional fuzzy system", uses a function $\boldsymbol{g^i}$ instead of a linguistic term

$$If\ X\ is\ A^i\ then\ y^i\ is\ g^i\ ; i=1,\ldots,R, \tag{5}$$

where the consequents $\boldsymbol{g^i=f(X)}$. When the values $\boldsymbol{g^i}$ are constants the system is called zero order Takagi-Sugeno system. The crisp output of the fuzzy system is determined by

$$y = \frac{\sum_{i=1}^{R} w_i g_i}{\sum_{i=1}^{R} w_i} . \tag{6}$$

where $\boldsymbol{w_i}$ is the firing strength of the i^{th} rule. Li Xin Wang [32] proved that any continuous function can be approximated by zero order Takagi-Sugeno systems.

2.2. ANFIS, Adaptive-Network-Fuzzy Inference System

The Matlab's ANFIS software generates a Takagi-Sugeno type fuzzy system from sample data using an adaptive neural network [6]. An adaptive network can be considered in some sense as the generalization of neural networks and fuzzy systems [6][7]. The typical structure of an adaptive network is shown in Figure 2. The network consists of nodes connected by directed edges. The typical adaptive network does not contain any feedback and it is organized in layers. The inputs and outputs of the adaptive network are denoted by $\boldsymbol{x_i}$ and $\boldsymbol{O_i^L}$. The number of layers is $\boldsymbol{L}$. The number of nodes in the $\boldsymbol{k}$-th layer is denoted by **#(*k*)**. Figure 3. shows a simple example of an adaptive network.

2.3. LESFRI

In many cases the dense rule base (e.g. Figure 4.) demanded by the classical compositional fuzzy inference techniques contains a large number of rules that increases exponentially with the number of input dimensions which fact also increases the computational complexity and the storage demand.

This problem led to the development of fuzzy systems that are able to produce the output relaying only on a minimal set of rules. Thus it is not necessary to ensure a full coverage of the antecedent space by rules and a sparse rule base with low complexity can be applied (see Figure 5) .

The development of Fuzzy Rule Interpolation (FRI) based Inference Techniques (FRITs) gives new methodology on the field for practical applications due to the reduced complexity and storage space demand as well as due to its ability to handle cases when there are no rules that would describe the expected output for all the possible inputs.

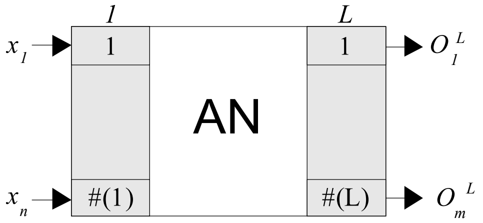


Figure 2. The layer structure of an adaptive network [19]

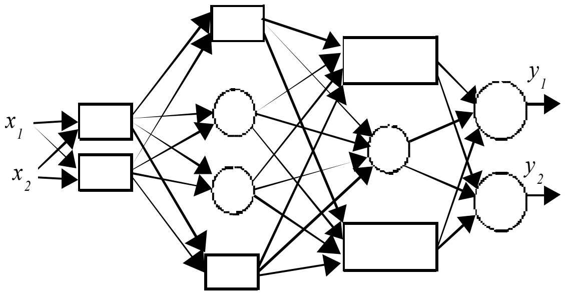


Figure 3. Simple example for an adaptive network [19]

FRITs can be divided into two groups depending on whether they are producing the estimated conclusion directly or they are interpolating an intermediate rule first.

Relevant members of the first group are among others the linear rule interpolation (KH method) [15] proposed by Kóczy and Hirota, which is the first developed one, the MACI (Tikk and Baranyi) [29], the FIVE [17] introduced by Kovács and Kóczy as well as the interpolation method developed by Kovács [16] that extended the fuzzy interpolation to the general metric spaces.

The methods belonging to the second group follow the concepts laid down by the generalized methodology (GM) defined by Baranyi et al. in [2]. Typical members of this group are e.g. the technique family proposed by Baranyi et al. in [2], the ST method suggested by Yan, Mizumoto and Qiao [30], the transformation based technique published by Chen and Ko [4] as well as the techniques LESFRI [11], FRIPOC [12] and VEIN [14] developed by Johanyák and Kovács.

We chose LESFRI (LEast Squares based Fuzzy Rule Interpolation) [11] for the task of FRI based fuzzy inference. It was applied owing to the good practical experiences (e.g. [9]) in course of previous applications. In its first step LESFRI interpolates a new rule into the position of the observation. The task is solved in three phases. Firstly, the antecedent membership functions are calculated using the FEAT-LS (Fuzzy sEt interpolATion based on method of Least Squares) fuzzy set interpolation method. Next, one determines the position (reference points) of the consequent linguistic terms of the new rule using an adapted version of the Shepard interpolation [25]. Thirdly, the shapes of the consequent sets are calculated using the same set interpolation technique (FEAT-LS) as in the first phase.

LESFRI determines the conclusion in its second step using the single rule reasoning method SURE-LS (Single rUle REasoning based on the method of Least Squares) that calculates the necessary modifications of the new rule's consequent sets based on the dissimilarities between the rule antecedent and observation sets.

2.3.1. FEAT-LS

The FEAT-LS method aims the determination of a new linguistic term in a fuzzy partition based on a supposed regularity between the known sets of the partition. First all linguistic terms are shifted horizontally into the interpolation point and next, one calculates the shape of the new set from the overlapped membership functions (A^i Figure 5 right side).

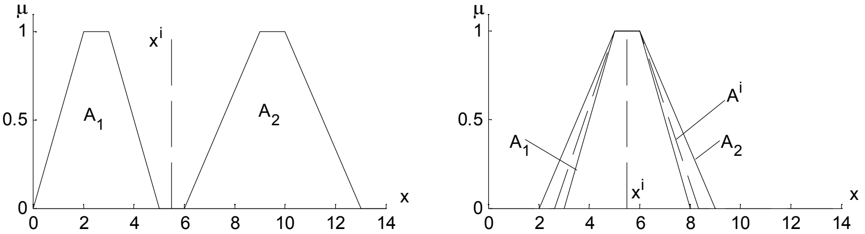


Figure 5. Original partition and interpolation point (x^i)

FEAT-LS targets the preservation of the characteristic shape type of the partition (e.g. trapezoidal on Figure 5) therefore it applies the method of the weighted least squares for the identification of the new set's parameters. The weighting is related to the original distance between the sets and the interpolation point. The calculations are done α-cut wise using only the α-levels corresponding to the characteristic points of the partition's default shape type.

2.3.2. SURE-LS

The revision method SURE-LS (Single rUle REasoning based on the method of Least Squares) is a special fuzzy inference technique that takes into consideration only one rule for the determination of the conclusion. The method is applicable when its antecedent sets are in the same position as the observation sets in each antecedent dimension and the heights (maximal membership value) of all involved fuzzy sets are the same.

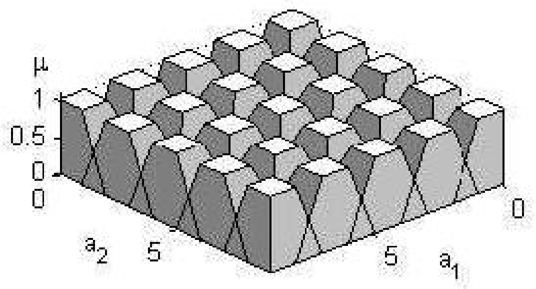


Figure 4. Antecendent space of a dense rule base

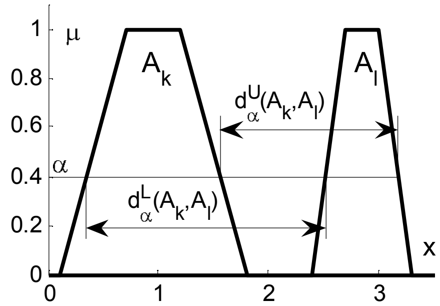


Figure 6. Lower ($d_\alpha^L(A_k, A_l)$) and upper ($d_\alpha^U(A_k, A_l)$) fuzzy distance at the α-level

SURE-LS calculates the conclusion by modifying the consequent sets of the rule. This modification is related to the similarity between the antecedent linguistic terms and the observation sets, which is measured independently in each input dimension by the means of their fuzzy distance (see Figure 6) and is aggregated by calculating the average distance.

2.4. RBE-DSS

In course of the rule base generation one can follow two different approaches. The first one divides the task in two separate steps, i.e. the structure definition and the parameter identification (e.g. Precup, Doboli and Preitl [24]; or Botzheim, Hámori and Kóczy [3], or Škrjanc, Blažič and Agamennoni [26]).

The second approach works incrementally by simultaneously modifying the structure and the parameters, i.e. introducing or eventually eliminating rules and tuning the parameters of the membership functions (e.g. Johanyák and Kovács [10]).

The Rule Base Extension with Default Set Shapes (RBE-DSS) [10] starts with an empty rule base and a set of training data points given in form of coherent input and output values. First the starting rule base is defined by determining the first two rules. They aim the description of the minimum and maximum output. One seeks the two extreme output values and a representative data point for each of them. If several data points correspond to an extreme value, one should select the one that is closer to an endpoint of the input domain.

Next, a tuning algorithm starts aiming the identification of the parameters of the initial fuzzy sets. This algorithm uses an iterative approach adjusting each parameter in several steps separately. The system is evaluated in each iteration step for different parameter values against a training data set and the parameter values ensuring the best performance index are kept for the next iteration.

If the decreasing velocity of the performance index of the system is too slow, i.e. it falls below a specified threshold after two consecutive iterations a new rule is generated.

It is because the system tuning reached a local or global minimum of the performance index and the performance cannot ameliorate further by the applied parameter identification algorithm. The new rule introduces additional tuning possibilities.

In order to create the new rule, one seeks for the calculated data point, which is the most differing one from its corresponding training point. The input and output values of this training point will be the reference points of the antecedent and consequent sets of the new rule.

3. Experimental data and modeling results

The air pollution data were collected in an urban site of Northern Portugal with traffic influences situated in Oporto [22]. The site is situated on the left edge of the Douro River, at an altitude of 90 m approximately. The study period was two weeks of July 2004, where high O_3 concentrations were measured and there was no missing data.

In course of the experiments 10 characteristics were measured: the hourly average concentrations (in $\mu g/m^3$) of carbon monoxide (CO), nitrogen oxides (NO, NO_2 and NOx) and O_3; hourly averages of air temperature (T), solar radiation (RAD), relative humidity (RH) and wind speed (WS); the day of week (DW; the O_3 behavior is different on weekdays and on weekend). All environmental and meteorological inputs corresponded to the same hour of the previous day.

Based on the results published in [22] we took into consideration in course of the modeling only the most important factors that are T, RH, $O_{3,}$ NO_2, NO. We formed two groups of the experimental data: one containing 259 measurements for system training purposes and one with 84 measurements for testing purposes. The test data were selected randomly from the original sample.

The quality of a fuzzy model is measured using a performance index that aggregates the distances between the measured and calculated output points. One can choose from several possible performance indices available in the literature (e.g. in [25]). We used the root mean square of the error (RMSE) as performance index owing to its good comprehensibility and comparability to the range of the output linguistic variable.

3.1. Modeling results using ANFIS and Takagi-Sugeno inference

We created two fuzzy models using the ANFIS software. The first one (labeled as O3_Anfis_3S_Trimf_Corr.fis) was a zero ordered Takagi-Sugeno model having triangle shaped membership functions and three fuzzy sets in each dimension. We used the hybrid training algorithm with three epochs. Figure 7. and Figure 8. present the measured and calculated output points in case of the training respective test data sets.

The second fuzzy system (labeled as O3_Anfis_5S_Trimf_Corr.fis) was a zero ordered Takagi-Sugeno model having triangle shaped membership functions and five fuzzy sets in each dimension. We used the hybrid training algorithm with five epochs. Figure 9. and Figure 10. present the measured and calculated output points in case of the train respective test data sets.

The performance of the systems was measured using the root mean square of the error (RMSE). The numerical results are summarized in Table I. In case of both systems one can identify clearly a slightly overfitting of the models to the training data.

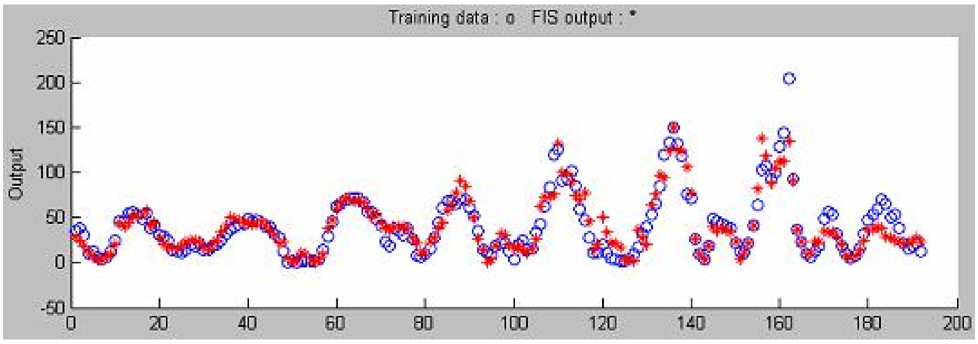


Figure 7. Measured and calculated output points in case of the first fuzzy system and the training data set

Table 1. System performance (RMSE) in case of the training and testing data

	Training	Test	Number of rules
O3_Anfis_3S_Trimf__Corr.fis	10.5101	95.0337	243
O3_Anfis_5S_Trimf_Corr.fis	4.4400	105.9679	3125
O3_2R_Reduced_01_640_00705.fis	8.0007	14.8703	66

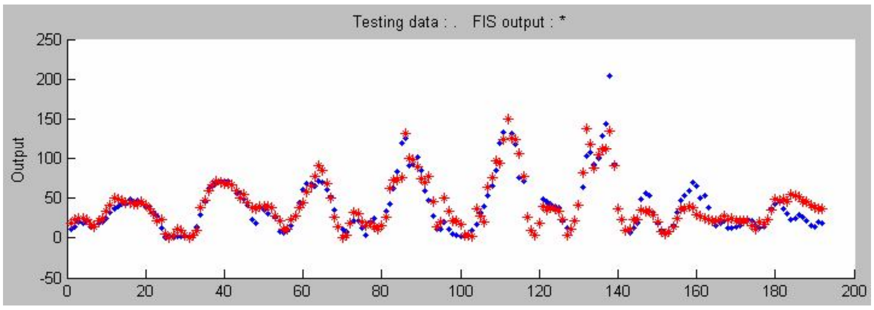


Figure 8. Measured and calculated output points in case of the first fuzzy system and the testing data set

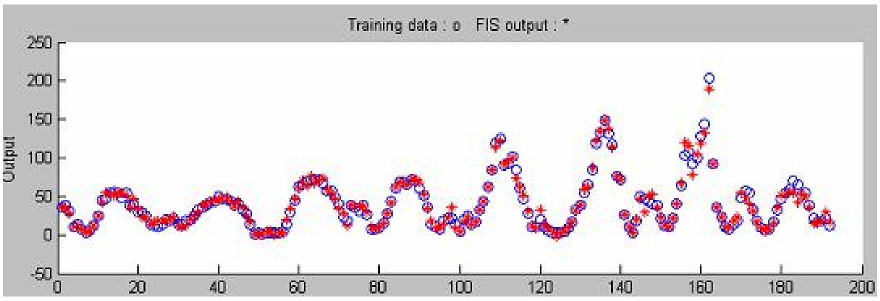


Figure 9. Measured and calculated output points in case of the second fuzzy system and the training data set

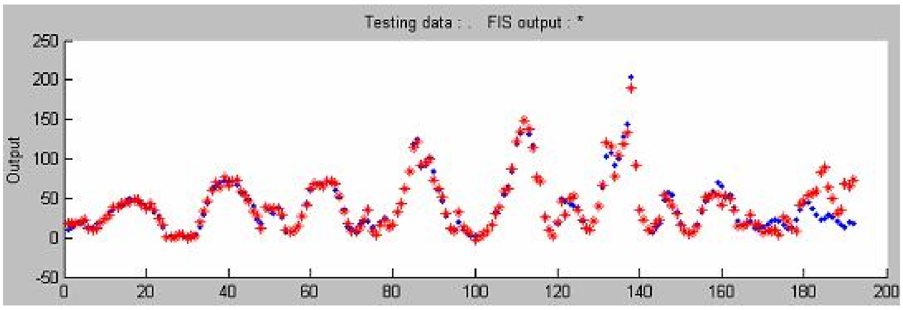


Figure 10. Measured and calculated output points in case of the second fuzzy system and the testing data set

3.2. Modeling results using RBE-DSS and LESFRI

We also created a fuzzy model using the SFMI toolbox [8]. The selected model identification method was RBE-DSS and we used LESFRI for fuzzy inference in the resulting sparse rule base. The system performance (RMSE) in case of the training data set was between the results obtained in case of the two ANFIS created systems (see Table I). On the other hand, there was a much smaller overfitting, i.e. this system presented the best performance in case of the test data. Besides, the number of rules necessary for the description of the relation between the input and output variables was the smallest in the case of the third fuzzy system. Figure 11. and Figure 12. present the measured and calculated output points in case of the train respective test data sets. The numerical results are summarized in Table I.

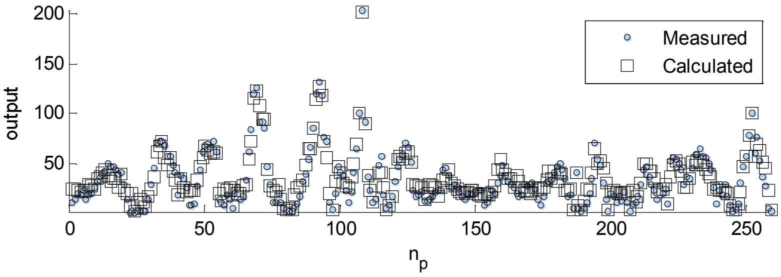


Figure 11. Measured and calculated output points in case of the third fuzzy system and the training data set

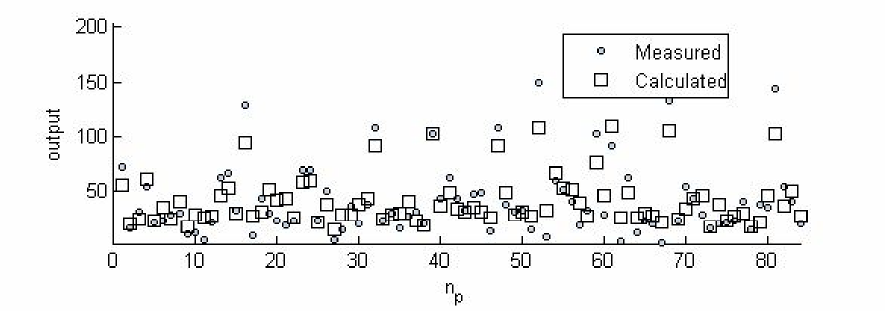


Figure 12. Measured and calculated output points in case of the third fuzzy system and the testing data set

4. Conclusion

This paper presented the application of two different fuzzy rule base generation approaches in order to model the relation between five environmental characteristics and ground level ozone concentration. The aim of our research was the creation of fuzzy models that can be used in practice for the prediction of the ozone level.

In our case the fuzzy system applying a sparse rule base and inference based on fuzzy rule interpolation ensured the best results taking into consideration both the training and testing data samples. This solution ensured slightly better performance than the previously applied approaches published e.g. in [22].

Acknowledgement:

The authors wish to thank J.C.M. Pires for air pollution database. This research was supported by Kecskemét College GAMF Faculty grant no: 1KU31, and the Hungarian National Scientific Research Fund Grant OTKA K77809.

References

[1] Al-Alawi, S. M. , Abdul-Wahab, S. A., Bakheit, C. S.: *Combining principal component regression and artificial neural networks for more accurate predictions of ground-level ozone.* in Environmental Modelling & Software 23(4), 396-403. 2008.

[2] Baranyi, P. , Kóczy, L. T., Gedeon, T. D.: *A Generalized Concept for Fuzzy Rule Interpolation*. In IEEE Transaction On Fuzzy Systems, ISSN 1063-6706, Vol. 12, No. 6, 2004. pp 820-837.

[3] Botzheim, J. , Hámori, B., Kóczy, L.T.: *Extracting trapezoidal membership functions of a fuzzy rule system by bacterial algorithm*, in Proceedings of the International Conference 7th Fuzzy Days, Dortmund, Germany, 2001, Lecture Notes in Computer Science, Springer-Verlag, 2206, pp. 218-227.

[4] Chen, S. M., Ko, Y.K.: *Fuzzy Interpolative Reasoning for Sparse Fuzzy Rule-Based Systems Based on α-cuts and Transformations techniques*, IEEE Transactions on Fuzzy Systems, Vol. 16, No. 6, pp. 1626-1648, 2008.

[5] Hládek, D., Vaščák, J., Sinčák, P.: *Hierarchical fuzzy inference system for robotic pursuit evasion task*, in: SAMI 2008, 6th International Symposium on Applied Machine Intelligence and Informatics, January 21-22, 2008, Herľany, Slovakia, pp. 273-277, 2008.

[6] Jang, Sh. R. , *ANFIS: Adaptive-Network-Based Fuzzy Inference System.* ,IEEE Transactions on systems, Man, and Cybernetics, 1993, Vol. 23, No. 3, pp.665-685.

[7] Jang, Sh. R., Sun, C. T. , *Neuro-fuzzy modelling and control.*, Proceedings of the IEEE, 1995, Vol. 83, No. 3, pp. 378-406.

[8] Johanyák, Z. C. *Sparse Fuzzy Model Identification Matlab Toolbox - RuleMaker Toolbox*, IEEE 6th International Conference on Computational Cybernetics, November 27-29, 2008, Stara Lesná, Slovakia, pp. 69-74.

[9] Johanyák, Z. C., Ádámné, A. M.: *Mechanical Properties Prediction of Thermoplastic Composites using Fuzzy Models*, SCIENTIFIC BULLETIN of "Politehnica" University of Timisoara, ROMANIA, Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE, Vol: 54(68) No: 4/ 2009, pp. 185-190.

[10] Johanyák, Z. C., Kovács, S.: *Sparse Fuzzy System Generation by Rule Base Extension* in Proceedings of the 11th IEEE International Conference of Intelligent Engineering Systems (IEEE INES 2007), Budapest, Hungary, 2007, pp. 99-104.

[11] Johanyák, Z. C., Kovács, S.: *Fuzzy Rule Interpolation by the Least Squares Method*, 7th International Symposium of Hungarian Researchers on Computational Intelligence (HUCI 2006), November 24-25, 2006 Budapest, ISBN 963 7154 54 X, pp. 495-506.

[12] Johanyák, Z. C., Kovács, S.: *Fuzzy Rule Interpolation Based on Polar Cuts,* Computational Intelligence, Theory and Applications, Springer Berlin Heidelberg, 2006, pp. 499-511.

[13] Johanyák, Z. C., Kovács, S.: *Polar-cut Based Fuzzy Model for Petrophysical Properties Prediction*, SCIENTIFIC BULLETIN of "Politehnica" University of

Timisoara, ROMANIA, Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE, Vol: 57(67) No: 24/ 2008, pp. 195-200, 2008.

[14] Johanyák, Z. C., Kovács, S.: *Vague Environment-based Two-step Fuzzy Rule Interpolation Method*, 5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics (SAMI 2007), January 25-26, 2007 Poprad, Slovakia, pp. 189-200.

[15] Kóczy, L. T., Hirota, K.: *Approximate reasoning by linear rule interpolation and general approximation.* International Journal of Approximative Reasoning, 9:197–225, 1993.

[16] Kovács, L.: *Rule approximation in metric spaces,* Proceedings of 8th IEEE International Symposium on Applied Machine Intelligence and Informatics SAMI 2010, Herl'any, Slovakia, pp. 49-52.

[17] Kovács, S.: *Extending the Fuzzy Rule Interpolation "FIVE" by Fuzzy Observation,* Advances in Soft Computing, Computational Intelligence, Theory and Applications, Bernd Reusch (Ed.), S pringer Germany, 2006, pp. 485-497.

[18] Kovács, S. , Kóczy, L. T.: *Application of Interpolation-based Fuzzy Logic Reasoning in Behaviour-based Control Structures*, Proceedings of the FUZZIEEE, IEEE International Conference on Fuzzy Systems, 25-29 July, Budapest, Hungary, pp.6, 2004.

[19] Lantos, B.: *Fuzzy systems and genetic Algorithms*, Műegyetem Kiadó 2002, ISBN 963 420 706 5

[20] Mahapatra, A.: *Prediction of ground-level ozone concentration maxima over New Delhi*, Environ Monit Assess, DOI 10.1007/s10661-009-1223-z, 27 October 2009.

[21] Ozdemir, H., Demir, G., Altay, G., Albayrak, S., Bayat, C.: *Prediction of Tropospheric Ozone Concentration by Employing Artificial Neural Networks* Environmental Engineering Science 25(9), 1249-1254. 2008.

[22] Pires, J. C. M. , Martins, F. G., Pereira, M. C., Alvim-Ferraz, M. C. M.: *Prediction of ground-level ozone concentrations through statistical models* 2009.

[23] Pires, J. C. M. , Sousa, S. I .V., Pereira, M. C., Alvim-Ferraz, M. C. M., Martins, F. G.: *Management of air quality monitoring using principal component and cluster analysis – Part II: CO, NO2 and O3*, Atmospheric Environment 42(6), 1261-1274. 2008a

[24] Precup, R. E. , Doboli, S., Preitl, S.: *Stability analysis and development of a class of fuzzy systems,* in Engineering Applications of Artificial Intelligence, vol. 13, no. 3, June 2000, pp. 237-247.

[25] Precup, R. E., Preitl, S.: *Optimisation criteria in development of fuzzy controllers with dynamics,* Engineering Applications of Artificial Intelligence, vol. 17, no. 6, Sep. 2004, pp. 661-674.

[26] Shepard, D.*: A two dimensional interpolation function for irregularly spaced data,* in Proceedings of the 23rd ACM International Conference, New York, USA, 1968, pp. 517-524.

[27] Škrjanc, I. , Blažič, S., Agamennoni, O. E.: *Identification of dynamical systems with a robust interval fuzzy model,* in Automatica, vol. 41, no. 2, Feb. 2005, pp. 327-332.

[28] Sousa, S. I. V., Martins, F. G., Alvim-Ferraz, M. C. M., Pereira, M. C.: *Multiple linear regression and artificial neural networks based on principal components to predict ozone concentrations*, Environmental Modelling & Software 22(1), 97-103. 2007.

[29] Takagi, T., Sugeno, M.: *Fuzzy identification of systems and its applications to modeling and control,* IEEE Trans. on SMC, 15:116-132, 1985.

[30] Tikk D., Baranyi, P.: *Comprehensive analysis of a new fuzzy rule interpolation method,* In IEEE Trans. Fuzzy Syst., vol. 8, pp. 281-296, June 2000.

[31] Yan, S., Mizumoto M., Qiao, W. Z.: *An Improvement to Kóczy and Hirota's Interpolative Reasoning in Sparse Fuzzy Rule Bases,* In International Journal of Approximate Reasoning, 1996, Vol. 15, pp. 185-201.

[32] Yiqiu, L. , Cobourn, W. G.: *Fuzzy* , ScienceDirect, Atmospheric Environment 41 (2007) 3502–3513.

[33] Wang, L. X.: *Adaptive fuzzy systems and control.*, Prentice Hall, 1994.

[34] Wong, K. W., Gedeon, T. D.: *Petrophysical Properties Prediction Using Self-generating Fuzzy Rules Inference System with Modified Alpha-cut Based Fuzzy Interpolation*, Proceedings of The Seventh International Conference of Neural Information Processing ICONIP 2000, Korea, pp. 1088 – 1092, 2000.

[35] Zadeh, L. A.: *Outline of a new approach to the analysis of complex systems and decision processes,* IEEE Trans. on SMC, 3:28-44, 1973.

Temporary Properties of RS Fuzzy Flip – Flops

M. Klimo, J. Boroň

University of Žilina
Univerzitná 1, 01026 Žilina
Slovakia
E-mail: {martin.klimo, juraj.boron}@fri.uniza.sk

Abstract. The RS flip-flop is a basic element of the computer random access memory. The same structure provides new properties by using fuzzy operations instead of Boolean ones. Applying probabilistic fuzzy operators, we have found that bi-stability of RS flip-flop is changed into mono-stability, i.e. the memory achieves the ability of spontaneous forgetting. When two fuzzy flip-flops are interconnected in a proper way, Hebbian learning occurs. This means that if inputs occur simultaneously, the outputs are turned up for much longer period, comparing to pulses which are not overlapped. This property can be applied to RS flip-flops grid network. In this case, the excitation of flip-flop outputs depends on input pulses (their magnitudes, durations and positions in the time scale) as well as on a connection topology of the grid.

1. Introduction

At the Department of Infocom Networks at the University of Zilina, voice processing and transmission is studied for longer time period. One task in this area is also the voice recognition problem. As the basis for this task we have decided to build up a network of non linear elements. This system structure consists of flip-flops which base function is defined with probability algebra. The main property of flip-flops is their ability of forgetting. At first the probability algebra is introduced. Then the option of implementing forgetting function into the R-S flip flop known in Boolean algebra is presented. We concentrate on the effect of repetitiveness next to forgetting and we expand the main structure of R-S flip flop by other inputs, which can influence the process of forgetting.

2. Probabilistic algebra characteristics

Classical Neural Networks neuron model is represented by nonlinear function of linear combination of inputs, where coefficients of linear combination represent the synaptic weights. The purpose of teaching stage is to find these coefficients, by minimizing the error function of the solution for specific task. Our intent is to study neural networks without synaptic weights in which the process of teaching will change the structure of network, not synaptic weights. Matrix of the system is than represented only by the connections between the neurons. State variables are outputs of the single neurons. This of course did not exclude the use of synaptic weights (real numbers in the system matrix). It can be used in the future. It appears to be convenient to use only the condition of two state set of Boole algebra for establishing continuous inputs, states and outputs as the first step. This approach makes it possible to simply reduce networks with the probabilistic algebra to networks with the Boole algebra. That is why enclosed unit interval $a,b \in \langle 0,1 \rangle \subset \Re$, where $\Re$ is the set of real numbers was chosen for the definition scope of input values, states and outputs and as the main structure probabilistic algebra was chosen. The most often used logical functions are listed in table Table. 1, where $A,B \in \{0,1\}$.

Table 1 logical functions in probabilistic algebra

A	B	A'	$A \wedge B$	$A \vee B$	$A \to B$	$A \equiv B$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

Remark: negation: A' , conjunction: $\wedge$, disjunction: $\vee$, implication: $\to$, equivalence: $\equiv$.

For better understanding, we show one of the logical functions (equivalence), with using of the probabilistic algebra in the figure below.

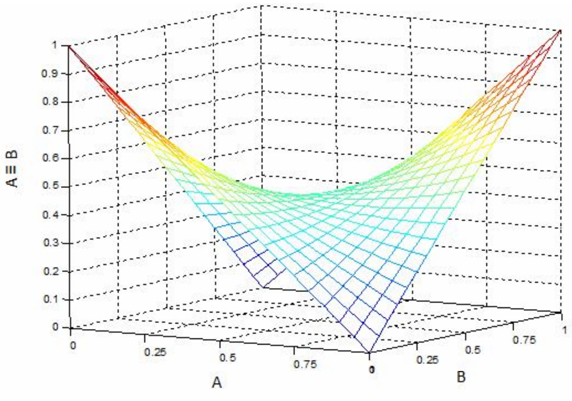
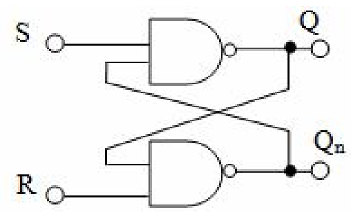


Figure 1 equivalence $A \equiv B \approx a.b + (1-a).(1-b).(1-a.b)$ *in probabilistic algebra*

3. Flip-flops with the probabilistic algebra

Flip-flops have the property of remembering the previous states, i.e. their output depends not only on current inputs but also on one or more previous states of circuit. The main flip-flop from which the other ones are derived, is binary circuit of type RS, which have two inputs marked by symbols R (reset) and S (set) and outputs Q, Qn (Qn is negation of Q). The truth table and the scheme of the circuit are shown below.



Input S	Input R	Output Q_i
0	0	forbidden state
1	0	1
0	1	0
1	1	Q_{i-1}

Figure 2 R-S flip flop and its truth table

Modeling of neural networks with flip-flops and non-linear mathematic is also described in [1], [2].

The behavior of RS fuzzy flip-flop with use of probabilistic algebra is analogical to binary RS flip-flop as is shown in figure Figure 3. It shows values of the output in a time after bringing the flipping impulse on the input R in the time t = 50.

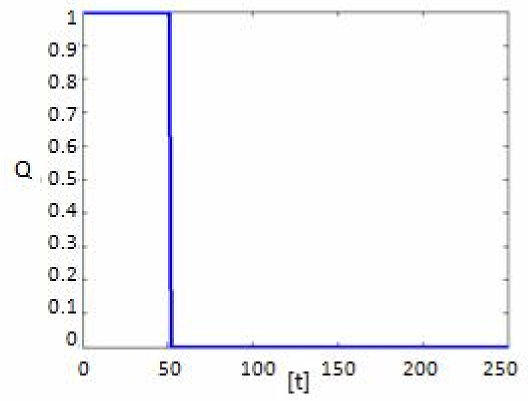


Figure 3 Output Q of R-S flip-flop with use of probabilistic algebra

By adding more inputs, logical units, etc. to the main flip-flop, more flip-flops with different properties can be assembled.

3.1. R-S flip flop with memory

In the circuits below, we will use invert units after inputs, in order to vivid the activity of the circuit, i.e. desire changes are induced by the greater input values. By adding next input, a memory, to the basic circuit and by adding other logical units, we have achieved, that the output of such a circuit will flip as in an R-S circuit but after some time it returns to the previous value. The diagram of circuit with memory is shown in the figure below.

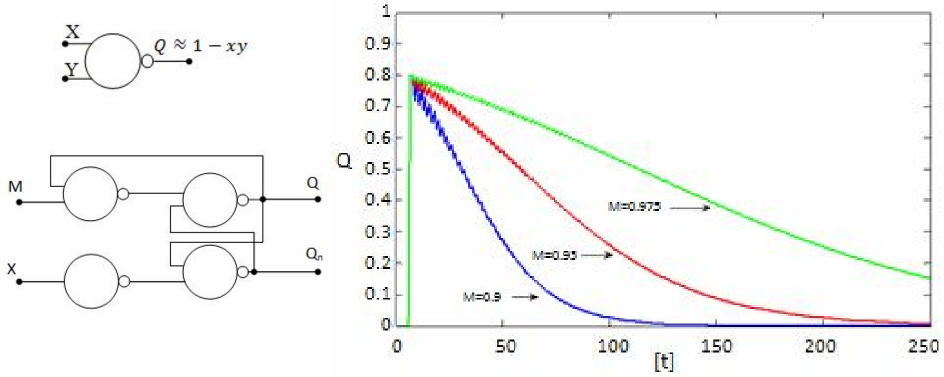


Figure 4 R-S flip flop with memory and its output in time for different value of input M (memory)

In the figure Figure 4 we can also see the output of the circuit in time. The input value X is set to 0,5. As shown in figure, by increasing (decreasing) input *M* (memory), the output of the circuit returns to the previous value slower (faster).

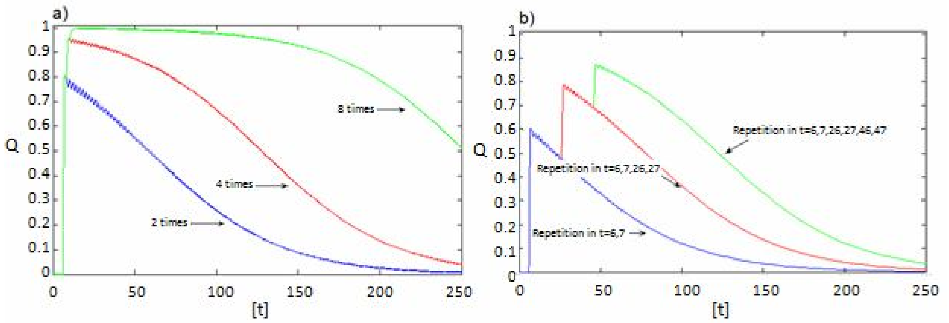


Figure 5 Output of a circuit with a) dependency on exciting time b) dependency on repeating of input in time

In the Figure 5 is shown that not only the input *M* can influence the output but also the time of excitement of input X or repeating the input X in time.

3.2. Flip-flop circuit with memory and limit

The behavior of flip-flop circuit above shows that for the inputs and state values near to zero, circuit gets to the relatively stable states, which are not possible to change by input value. That is why next input, which limit's values to the entered input value (precisely, to the inverted value of input L), was added. This changes the sensitivity of the circuit toward inputs.

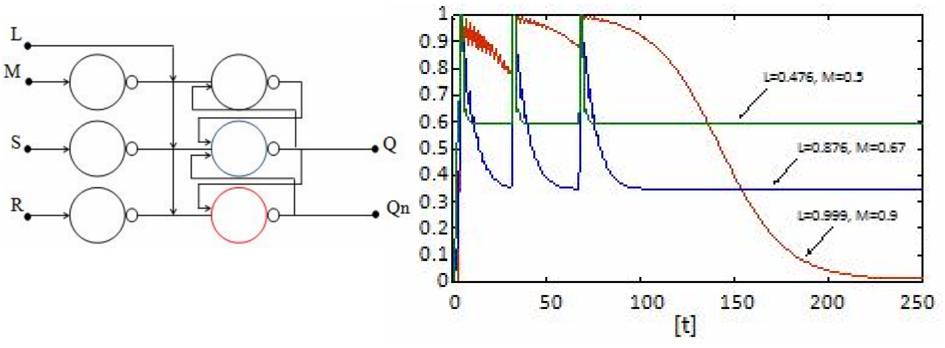


Figure 6 Output Qn of the flip-flop circuit with memory and limit, with different input values of L (limit)

The figure above shows the behavior of such circuit by setting different values of the limit.

3.3. Flip-flop circuit with reward and penalty

We plan to use above mentioned probabilistic flip-flops as elementary units of networks in the time domain. This means that parts of network, which are flipped in the purpose of a network, should be flipped considerably longer time (they should be memorized) according to the parts of network flipped in contradiction of network purpose and should be forgotten. Such strengthening or weakening will not be addressed on individual flip-flop circuits but overall for the whole network. That is why next inputs R (reward) and P (penalty) was added.

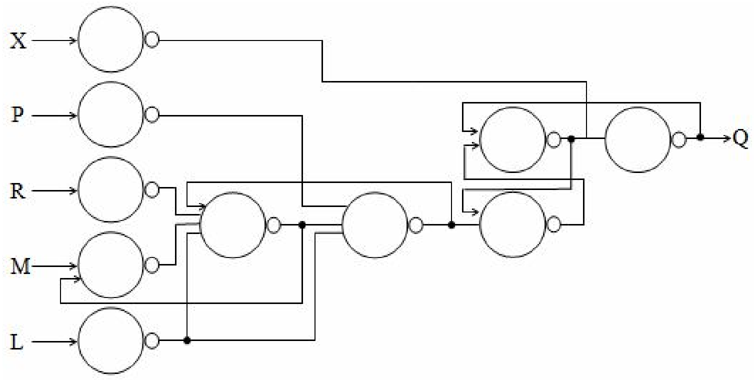


Figure 7 flip-flop circuit with reward and penalty

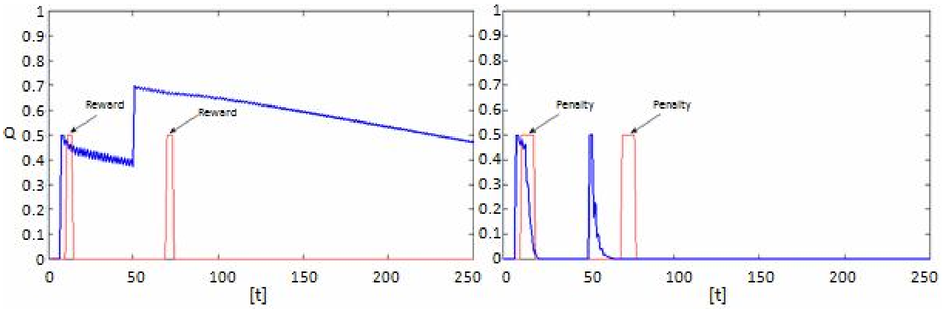


Figure 8 Output Q of flip-flop circuit with reward and penalty

In the above figure we can see the output of the circuit, first with rewarding the circuit and next with penalization of circuit.

Conclusions

Current neural networks, including Kohonen`s networks, use properties of feed-back nonlinear systems relatively little. This paper shows that by the combination of flip-flop circuit as the main memory unit in digital systems and function of forgetting (obtained by the use of probability algebra) is possible to achieve time coding in the term of remembering the duration of element state. As an example a R-S flip-flop was chosen, but also other types (e.g. J-K) can be taken into account. The results of experiments with such structure show, that it is possible to achieve some desired behavior, such as expansion of states with higher values of input, longer duration of input or repeating. It is also possible to influence the duration of state by another input (memory), what allows Hebbian learning. Inputs for reward and penalty allows to affect states of more flip-flop circuits contemporary, what imitates tools of teacher for remembering and forgetting support. We assume and plan to verify the teaching of such networks in contrast with neural networks without change of synaptic weights in further research. The very process of teaching, in this case, will mean to lead nonlinear system (without modification of system "*matrix*") to such state, in which system for given inputs gives desired outputs. The process of finding a suitable matrix of system could then be separated from learning process and can be moved to the sphere of evolutionary development of populations of such networks. Our effort is to test mentioned ideas in a role of speech recognition.

References

[1] Dubois D. M.: Hyperincursive McCulloh and Pitts Neurons for Designing a Computing Flip-Flop Memory, American Institute of Physics, AIP Conference Proceedings, 1999.

[2] Lovassy R., Kóczy L. T., Gál L.: Fuzzy Flip-Flop Based Neural Network as a Function Approximator, CIMSA 2008 – IEEE International Conference on

Computational Intelligence for Measurement Systems And Applications, Istanbul – Turkey , 14-16 July 2008

[3] Klimo M., Boroň J.: Dynamické vlastnosti pravdepodobnostných fuzzy klopných obvodov (Dynamic Behavior of Probabilistic Fuzzy Flip-Flops, in Slovak) , In: ITAT 2009 – Information Technologies – Applications and Theory: Conference on Theory and Practice of Information Technologies - Proceedings, Kralova Studna, Slovakia, September 2009, pp. 103-104, ISBN 978-80-970179-1-0

[4] Klimo M., Boroň J.: Temporary Properties of RS Fuzzy Flip-Flops, In: Conference on Fuzzy Set Theory and Applications FSTA 2010, Armed Forces of General M.R. Štefánik, Lipt. Mikuláš, 2010, pp. 79, ISBN 978-80-8040-391-1

Fuzzy based Prequalification Methods for EoSHDSL Technology

F. Lilik, J. Botzheim

Department of Automation, Széchenyi University
H-9026 Győr, Egyetem tér 1., Hungary
e-mail: lilik.ferenc@telekom.hu, botzheim@sze.hu

Abstract: The aim of the study was to create a prequalification system which can prognosticate the available maximal data transmission speed on a copper wire pair over EoSHDSL without installing a real EoSHDS connection. In order to find the relationship between line parameters and bit rates it was necessary to make comparative measurements.

The widest spectrum of the line parameters was analyzed, because at the beginning of the work the influence of parameters on the issue could not be determined.

Analyzing the results of the measurements, it clarified that the connection between line parameters and available data rate is not strict. In lots of cases different data rates belonged to similar line parameter values or the same data rate was set by different line parameters. Moreover, the groups of line parameters belonging to the same bit rate were overlapped. To handle this uncertainty fuzzy systems are most appropriate, so fuzzy logic was chosen for solving the problem.

During the work around 140000 measured data were collected. Being in possession of measured results an initiative rule base was set up without using any algorithm. This rule base did not give acceptable results. In order to higher the precision a second rule base was created using bacterial memetic algorithm.

The method was tested on real telecommunication networks. All the tests were successful.

Keywords: telecommunication, EoSHDSL, prequalification system, fuzzy system, rule base, memetic algorithms

1. Introduction

The prequalification of quality is demanded in all fields of telecommunication. Industrial companies, banks, financial organizations look for highly safe services in good quality. Observance of preliminary agreements between them and telecom companies is compulsory also in view of business aspects, so it is very important to know the technical possibilities beforehand.

In order to get the needed information different prequalification systems are used by telecom suppliers to figure the available level of service. These systems are able to prognosticate quite efficiently but they are expensive.

This article would like to give a short review of a work aimed at working out a computer aided prequalification system which could be more inexpensive than others. However the results may be widely generalized the study was based on a specific transport medium (symmetrical twisted pair) and on a specific data transmission technology (EoSHDSL). The reason for the choice was that the copper based twisted pair is the mostly available transmission medium, so thus it provides good possibilities for making of numerous measurements. It is important to mention that – according to the principles of prequalification – only twisted pairs in working order were selected which were in good condition. Chosen technology (EoSHDSL) has a good perspective because its huge permeation is expected in local networks that will function for years.

In this paper EoSHDSL data transmission technology and transmission parameters of symmetrical twisted pairs will be briefly introduced, selection of the mathematical model fitting with the problem and its use will be shown. The control tests will be detailed as well. At the end of the article the results of the work and the further possibilities of the suggested method's utilization will be presented.

2. Telecommunication parameters that have influence on data transmission speed

2.1. Ethernet over SHDSL technology

The EoSHDSL technology establishes connections between Ethernet endpoints using SHDSL transmission on symmetrical copper wire pairs. This method has opportunities for bounding more wire pairs in order to reach higher data transmission speed. In this case the whole data transmission rate (DTRW) is calculated as the sum of each lines' bit rate (DTRL):

$$DTRW = \sum_{i=1}^{n} DTRL_i \,, \tag{1}$$

where n is the number of used lines, and $n<9$.

Since SHDSL technology (described in ITU-T G.991.2. [1][2]) is used on every pairs, also the searches are needed to make from this point of view.

Figure 1 shows an application model for a typical SHDSL system [1]. In such an application STU-R (SHDSL Transceiver Unit at the Remote end) is connected to one or more user terminals through the S/T reference points. These terminals can be data transmission equipments or telecommunication devices. The connection between STU-R and STU-C (SHDSL Transceiver Unit at the Central Office) is digital local loop (DLL) ending on U-R and U-C reference points. This local loop may contain one or more SHDSL signal regenerator units (SRU). STU-C is connected to a Central Office network through the V reference point.

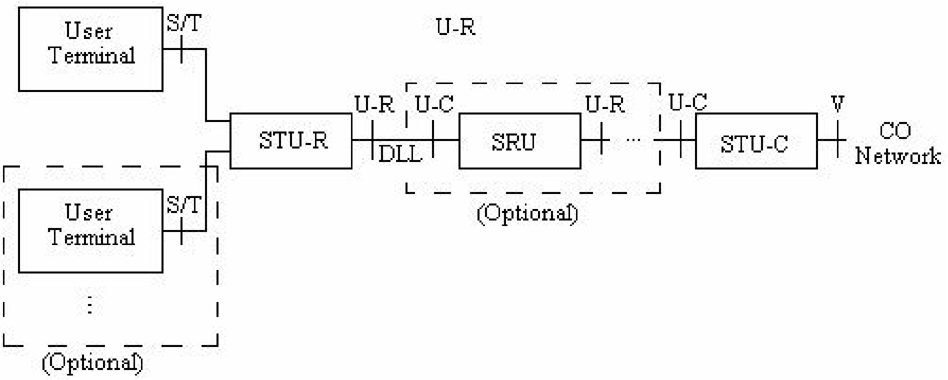


Figure 1. Typical SHDSL application model

With respect to prequalification the part of line between U-R and U-C reference points have to be examined in which in this case the usage of SRU is not allowed. If the distance between STU-R and STU-C requires using SRU-s then each DLL is needed to be prequalified separately. In this case the speed of the whole line (*DTRL*) is equal to the bit rate (BR) of the worst segment (*DTRS*), as it is seen on formula (2):

$$BR_{DTRL} = \min[BR_{DTRS\,i}]_{i=1}^{n}, \tag{2}$$

where *n* is the number of segments.

EoSHDSL uses 32 constellation Trellis Coded Pulse Amplitude Modulation on the DLL. It is seen on its Power Spectral Density diagram (Figure 2) that the upper limit of the frequency is under 1400 kHz.

All parameters influencing the quality of connection are set out in the ITU-T recommendation number G.991.2. Three of them are connected with the copper wire pair used for transmission. These are Loop Attenuation, Signal-to-Noise Ratio margin (SNR margin) and Return Loss.

Loop Attenuation defect is declared by the STU if the value of the loop attenuation is higher than the preliminary configured threshold. G.991.2 recommendation defines this value as it is seen on formula (3) [1]:

$$LA_{SHDSL}(H) = \frac{2}{f_{sym}} \left\{ \int_{0}^{\frac{f_{sym}}{2}} 10 \times \log_{10}\left[\sum_{0}^{1} S(f - nf_{sym})\right] df - \int_{0}^{\frac{f_{sym}}{2}} 10 \times \log_{10}\left[\sum_{0}^{1} S(f - nf_{sym}) \left| H(f - nf)^{2} \right| \right] df \right\}, \tag{3}$$

where f_{SYM} is the symbol rate, *1/H(f)* is the insertion loss of the loop, and *S(f)*is the nominal transmit PSD.

It can be seen that the value of the Loop Attenuation is influenced by only one parameter that is connected to copper wire pair beside the configured ones. This parameter is the insertion loss.

Signal-to-Noise Margin defect is declared if the SNR margin is declined under the configured value. "SNR margin is defined as the maximum dB increase in equalized noise or the maximum dB decrease in equalized signal that a system can tolerate and maintain a BER of 10^{-7}."[1]

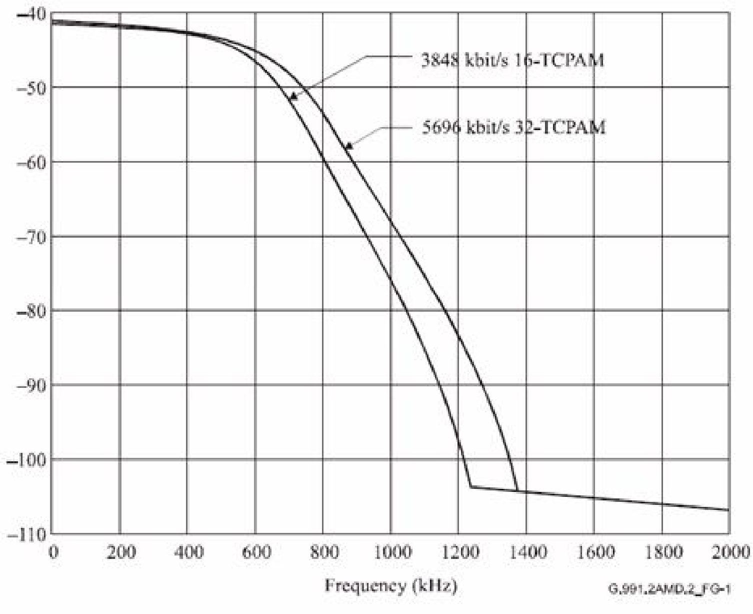


Figure 2. Nominal PSD masks for 0-DB power backoff [2]

The level of signal is set up during the synchronization process, so assuming a constant signal level the change of SNR (Signal-to-Noise Ratio) margin apparently depends only on the actual change of the line noise.

Return Loss is representative for termination impedance distortion. It shows how good the used termination impedances are. In case of measurements performed with an impedance analyzer, the return loss is defined as formula (4) [1]:

$$\mathrm{ReturnLoss(f)} = 20\log\left|\frac{Z_{\mathrm{TEST}}(\mathrm{f}) + Z_{\mathrm{REF}}}{Z_{\mathrm{TEST}}(\mathrm{f}) - Z_{\mathrm{REF}}}\right|, \tag{4}$$

where $Z_{TEST}(f)$ is the measured complex impedance at frequency f at the loop interface of the tested device and Z_{REF} is the reference impedance (135 Ω).

2.2. Line parameters that have influence on data transmission speed

Influencing parameters mentioned by G.991.2 recommendation can not be measured on empty wire pairs [4]. What measurable on these lines and influenced on parameters mentioned before are Line Noise, Insertion Loss and Return Loss of the line terminated with measuring instrument [3].

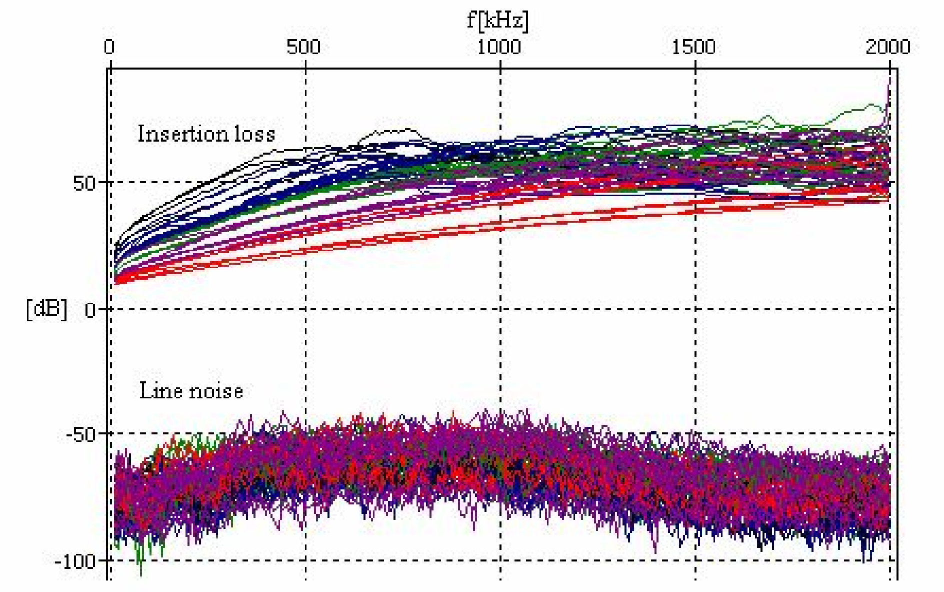


Figure 3. Line Noise (lower) and Insertion Loss (upper)

Line Noise has an effect on Signal-to-Noise Ratio and Insertion Loss affects Loop Attenuation.

The correlation supporting the prequalification must be found between the change of data transmission speed and the change of parameters listed above.

Examination of the Line Noise (results are shown in Figure 3) unequivocally confirmed that due to the circumstances existing on examined lines, Line Noise does not take part in forming differences between the speed groups (that are signed by different colours on the graph).

The role of **Return Loss** is very similar to the role of Line Noise. It can be seen on the graphs (Figure 4) that there is no exact correlation between data speed and Return Loss.

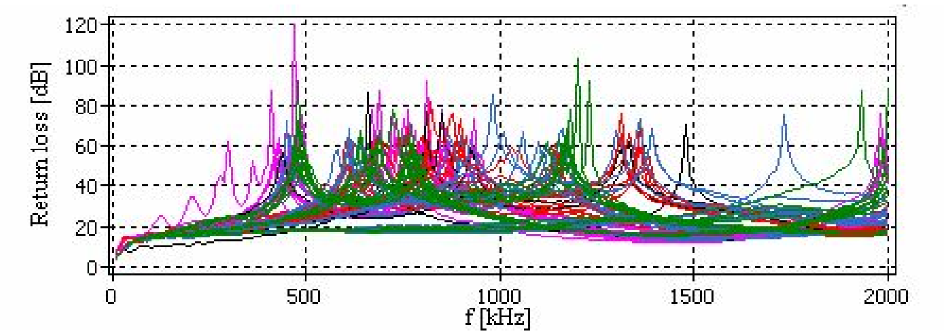


Figure 4. Return loss

However, the role of **Insertion Loss** is obtrusive. The graphs of Insertion Loss in Figure 3 can be grouped easily by the evolving data transmission speed.

2.3. Input parameters for prequalification method

According to subsections 2.1. and 2.2. it can be fixed that values of Insertion Loss measured on certain wire pairs up to frequency 1500 kHz are minimally satisfactory input parameters for such a prequalification system that qualify empty copper wire pairs in order to predict probable EoSHDSL data transmission speed. Later, the results confirmed that it is not necessary to know all values of the frequency spectrum, it is enough to have Insertion Loss values on some properly chosen frequencies.

3. Fuzzy Logic as the effective computational intelligence tool for handling impacts of influencing parameters

3.1. The choice of a mathematical tool

Looking over the input data it can be seen that they are not consequent and they have no univocal correspondence with data transmission speed. A measurement result as an input can be connected to different bit rates as outputs and more different input values can be belonged to the same output speed value. Moreover, the number of input values is large. (More than 140000 measured values were processed only in this study.)

Solving problems like this is very difficult using traditional mathematical methods. Fuzzy logic seems to be the most appropriate way [5][6][7][10][11][12].

3.2. Principles of building the rule base

The rule base was built based on measured values of line parameters that influence the EoSHDSL bit rate.

In order to evaluate the results, the lines were rated into five speed groups. These five groups represent the outputs of the rule base. Output groups essentially mimic the offers of telecommunication providers, guaranteeing the lowest and allowing to reach the highest values. Speed groups are shown in Table 1.

Table 1. Data transmission speed groups used in the solution

DTS group	Minimum [kbit/s]	Maximum [kbit/s]
G1	520	840
G2	904	1544
G3	1608	2440
G4	2504	3144
G5	3208	5448

It could be possible to raise the number of the DTS groups by placing more measures. In this case the steps of the rule base outputs could be finer and the steps of 64 kbit/s of the technology may be reached.

The input field is described by continuous functions on the 1,5 MHz wide range of the frequency. For the solution sampling this band and evaluating the samples are sufficient. In order to keep the computing costs low and heighten the accuracy, the Insertion Loss values at six points of frequency are compared with the fuzzy sets being on the antecedent side of the rule base, so the rule base is based on six variables. These six points (except the first in Table 2) divide the interval into parts of the same length. With this method the input containing continuous functions was reduced to a six dimensional space.

Table 2. Discrete frequency values where Insertion Loss is measured for input

Measuring points	100 kHz	500 kHz	750 kHz	1000 kHz	1250 kHz	1500 kHz

Building the rule base, the frequency values are shown in Table 2 were considered. The rules evaluate a six dimensional observation vectors and result in one of the five DTS groups. The dimensions of the observation vectors are the Insertion Loss values measured on the chosen points.

3.3. The first rule base and its efficiency

First a “hand made” rule base was implemented according to subsection 3.2. without using any algorithm. This rule base contained five rules with six dimensional antecedents.

Rules were created as easily as it was possible. According to the measurement results the maximum (M+), the minimum (M-) and the average (A) values were defined for each DTS group on every frequency points (Table 2). Using these values, the lower value of the support was M-, the upper value was M+ and the core point was A for each DTS groups and each frequency values. In this mode triangular shaped fuzzy sets were created.

Figure 5 and Figure 6 give examples from the rule base (note that axis X is reversely directed). As long as Figure 6 shows the complete antecedent side of the fifth rule, the fuzzy sets of Figure 5 do not belong to the same rule, but it displays the third dimension of each rule.

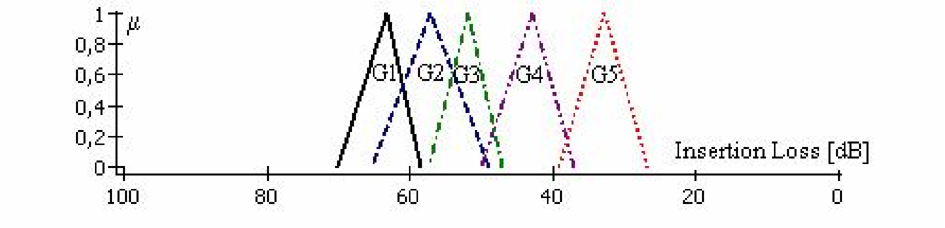


Figure 5. "Hand made" fuzzy sets on 750 kHz

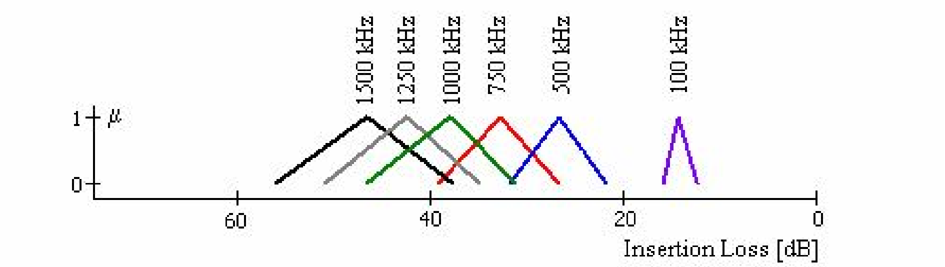


Figure 6. The antecedents of the fifth rule

Tests were run on 20 lines of real telecommunication networks with this rule base and the results were compared with measured DTS values. It turned out that results were equal with measured data only in 63%. The rule base provided incorrect results in 37%. This fact claimed to retune the rule base.

3.4. Final rule base generated by algorithm

The moderate success of the rule base described in the previous subsection is not enough to use it in industrial expert systems. A new rule base was needed to create. Optimization of the rules demanded so much work that an automatic method was applied. Bacterial memetic algorithm [8] was applied on training patterns. This technique is a nature inspired, population based, global optimization algorithm. It combines the bacterial evolutionary algorithm with a local search technique. The bacterial evolutionary algorithm performs global search, however, the solution found by the algorithm converges pretty slowly. The local search technique is a gradient based algorithm, which can find a more accurate solution, however, this method often find only the local optimum, nearest to the initial point of state space. Combining the evolutionary and gradient based approaches might be useful in improving the performance of the basic evolutionary algorithm, which may find the global optimum with sufficient precision in this combined way. Combinations of evolutionary and local-search methods are usually referred to as memetic algorithms [9].

The parameter setting for the bacterial memetic algorithm can be seen in Table 3.

Table 3. Parameters for Bacterial Memetic Algorithm

Number of rules	10
Size of the population	20
Number of clones	8
Number of generations	3000
Number of gene transfers	12

This rule base contains 10 rules with six dimensional antecedent and consists of trapezoidal fuzzy sets. The fuzzy sets are normal as it can be seen in Figure 7.

For fuzzy inference the Mamdani reasoning method is used [6] where t-norm is the minimum and t-conorm is the maximum operator. The crisp output value is calculated by the Center of Gravity defuzzification technique.

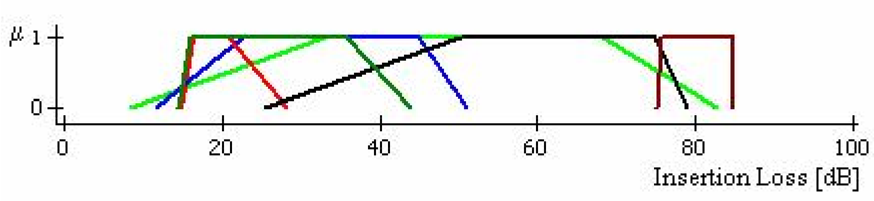


Figure 7. Example of antecedents from the final rule base

Success percentage of the tests (on real telecom networks) of this rule base was 100%. This accuracy is due to correct choice of number of output groups to available training patterns. Having more patterns also the number of DTS groups could rise.

4. Conclusions

In this study a solution was searched for a recent telecommunication problem. Inspecting the physical parameters of symmetric copper wire pairs which compose the base of the telecommunication local networks, the proper set of input parameters was defined. Two rule bases were created for solving the problem, but as it turned out, the complexity of the task demanded to use the second rule base, which was made by bacterial memetic algorithm. Finally, tests were made for defining the accuracy of the method.

Observations of working out this method showed that the complexity of such telecommunication problems justifies the utilization of fuzzy logic and needs the use of rule base identification algorithm.

The implemented system was tested on industrial telecommunication networks. The experiences verified that the suggested method is operable.

Improving the resolution of the method is possible. In order to build a system with more detailed output it is necessary to have much more preliminary line measurement and to create new rules based on the new results.

However this method was worked out for EoSHDSL systems, it is able to handle other telecommunication problems as well. Certainly, in this case it is crucial to observe those systems, to find the proper input parameters and to create new rule base according to the observations.

References

[1] Single-pair high-speed digital subscriber line (SHDSL) transceivers ITU-T G.991.2 (02/2001) Series G: Transmission systems and media, digital systems and networks, Digital sections and digital line system – Access networks

[2] Single-pair high-speed digital subscriber line (SHDSL) transceivers amendment 2 ITU-T G.991.2 amendment 2 (02/2005) Series G: Transmission systems and media, digital systems and networks, Digital sections and digital line system – Access networks

[3] *Helyi távbeszélő hálózati kábelek mérése és szerelése (in Hungarian)*, Közlekedési Dokumentációs Vállalat, Budapest, 1982.

[4] Elek, A.: *Távközlő kábelek és vezetékek (in Hungarian)*, Magyar Posta Könyvkiadó, Budapest, 1989.

[5] Zadeh, L. A.: *Fuzzy sets*, Inf. Control, Vol. 8, pp. 338-353, 1965.

[6] Mamdani, E. H., Assilian, S.: *An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller*, International Journal of Man-Machine Studies, Vol. 7, pp. 1-13, 1975

[7] Klir, G. J., Yuan, B.: *Fuzzy Sets and Fuzzy Logic. Theory and Applications.* Prentice Hall, Upper Saddle River, New Jersey, 1995.

[8] Botzheim, J., Cabrita, C., Kóczy, L. T., Ruano, A. E.: *Fuzzy rule extraction by bacterial memetic algorithms*, International Journal of Intelligent Systems, Vol. 24, No. 3, pp. 312-339, 2009.

[9] Moscato, P.: *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms.* Caltech Concurrent Computation Program, Tech. Rep., California, 1989.

[10] Johanyák, Z. C., Ádámné, M. A.: *Fuzzy Modeling of the Relation between Components of Thermoplastic Composites and their Mechanical Properties*, Proceedings of the 5th International Symposium on Applied Computational Intelligence and Informatics (SACI 2009), May 28-29, 2009, Timisoara, Romania, pp. 481-486

[11] Johanyák, Z. C.: *Survey on Five Fuzzy Inference-Based Student Evaluation Methods*, in I. J. Rudas et al. (Eds.):Studies in Computational Intelligence, 2010, Volume 313, Computational Intelligence in Engineering, pp. 219-228

[12] L. T. Kóczy, D. Tikk: *Fuzzy rendszerek* (in Hungarian), Budapest: Typotex Kiadó, 2000.

Virtual Collaboration Arena, Platform for Research, Development and Education

Péter Galambos [1,2], István Marcell Fülöp [1,3] and Péter Baranyi [1,3]

[1] Computer and Automation Research Institute, Hungarian Academy of Sciences
[2] Department of Manufacturing Science and Technology,
Budapest University of Technology and Economics
[3] Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics
Budapest, HUNGARY

Abstract: The VirCA system (Virtual Collaboration Arena) is a component based, 3D interactive virtual environment for ambient supervision of robots and other hardware or software devices. This software framework provides a basis for modular robotic system development with a kind of augmented collaboration in 3D virtual world. The paper presents the conceptual background of VirCA, the build-up and operation of the system as well as the possible fields of application. The experiences raised in the course of a VirCA training and the possible improvement trends are also going to be reviewed.

Keywords: *3D Internet, Augmented reality, Distributed intelligence, Middleware, Networked robots*

1. Introduction

The main concept of VirCA can be expressed with the underlying technologies: it is considered as the combination of 3D computer graphics, distributed intelligent systems [1] and modular framework. This makes VirCA a component based, 3D interactive virtual environment for collaborative operation of robots and other hardware or software devices. In VirCA, three key technologies are integrated using popular open source implementations and standards:

- 3D Computer graphics - Ogre3D [2]
- Distributed systems - CORBA and ICE [3]
- Modular framework - RT-Middleware [4]

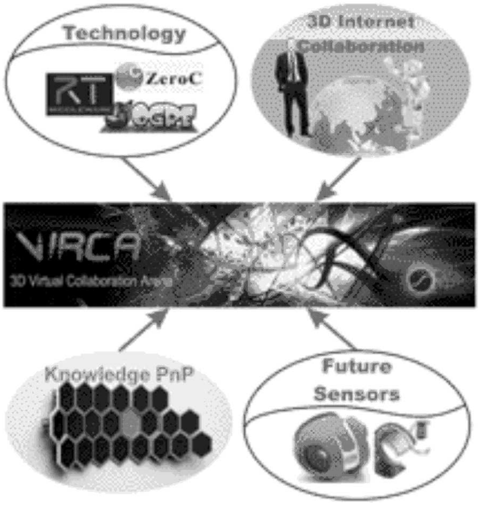


Figure 1: Illustration of the VirCA concept

Component-based robotics and distributed software environments are popular research topics in recent years. Sziebig et al. reviewed the existing middleware technologies focusing on distributed video image processing for robotic applications [5]. A detailed discussion on state of the art architectural solutions for distributed robotics was published by Amoretti and Reggiani in [6]. VirCA is free to use for academic purposes and available online at the www.virca.hu website. VirCA is developed by the Cognitive Informatics Research Group of MTA SZTAKI[1]. This paper introduces the concept behind the VirCA framework as a part of an ongoing academic research project. Here, we do not deal with the deep technical details of VirCA. More details about the architecture of VirCA can be found in [7]. Fig. 1. illustrates the VirCA concept.

2. Motivation

There are two main reasons why we have started to develop the VirCA system. The first goal is to speed up the academic research → R&D → Industry pathway. If the researchers could wrap their softwares into a standard form using custom or generic interfaces, system integrators would try and evaluate the novel algorithms without deep understanding of the solution just by placing a given component into their test application. In this way, a fast transfer gateway could be established between the scientific research, education and everyday engineering. As a second aim, we would like to introduce a surface extending the potential of conventional paper and text based scientific publication with focus on advanced

[1] MTA SZTAKI is the official abbreviation of Computer and Automation Research Institute, Hungarian Academy of Sciences (www.sztaki.hu)

robotics and related disciplines. Our goal is to make VirCA to be a common publication platform where the novel results can be published, compared and applied in partly virtual environments. With such a common platform, the otherwise hardly comparable results of e.g. machine vision, SLAM, collision avoidance etc. become much easier to evaluate by inserting the functional component (e.g. an object recognizer) into a benchmark scenario. In both featured application area, generic quasi-standard interfaces of components have to be defined to make a widespread applied platform. We think, that the RT-Middleware community has reached the critical mass of users, hence could be the basis of the common platform.

3. Pillars of the VirCA Paradigm

In this section, the principal properties of the VirCA framework is discussed.

3.1. 3D Internet

In recent time, the fast development of the visualization systems allowed the virtual reality applications to be appeared in a wide range of use. The first step was the introduction of 3D rendering in computer graphics that the laymen could meet primarily in computer games. The second step was the appearance of 3D graphics accelerator cards. Here we could see a beneficent interaction between the development of the technology and its utilization in applications: High performance 3D accelerator cards became affordable for everyday usage in very short time. Today development in computer visualization is the spread of ”real” 3D displays. These displays generate real stereoscopic sensation that enhances the 3D experience. Real 3D technologies can be differentiated into passive and active stereo and autostereoscopic systems.

In the case of passive stereo systems the left and right images - which create the stereoscopic sense - arrive to the user at the same time, where some passive tool sorts out the appropriate image for the proper eye. In the color filtering method, which has been applied in the cinemas for a long while, the left and the right images are in separate color range, while the glasses worn by the user contain color filters: they pass through only the color range which belongs to the appropriate eye. The polarization method also needs to be mentioned which operates on a discipline similar to the color filtering method, but in that case the complete original color range is available.

In active stereo systems the left and right images arrive in separate channels to the user (temporal separation), where some active tool let trough the image to the appropriate eye. In case of the so-called ”shutter glass” method the user’s glasses contain liquid crystals. These are controlled electrically in such a way that by turns the left and the right one passes through the light. In this case the two images arrive interlaced in time from the display and the glasses are synchronized to it. It means, if the left image is visible on the display the left side of the glasses passes through and the right side conversely.

Such systems already appear today in households. The 3D Vision system of nVidia - graphics card, display, glasses - can be bought for a reasonable price. In case of this system the display is an LCD monitor of 120 Hz which displays the left and right image with 60-60 Hz. The so-called autostereoscopic systems without any glasses should be mentioned as well, where on a special display many images are showed at once: due to the placement of the pixels and the optical grid in front of the display, from different places different images are visible. In this way different images arrive to the left and right eyes of the user.

Based on the before mentioned 3D visualization technologies, VirCA provides a platform where users can build, share and manipulate 3D content, and collaboratively interact with real-time processes in a 3D context, while the participating hardware and software devices can be spatially and/or logically distributed and connected together via IP network. For example, engineers and researchers from different countries can put a mobile robot and an obstacle avoidance software module together to test and tune it in a semi-virtual manner. They do not have to move their devices to the same place and take effort in the system integration. Or even, an architect can discuss the design of the future house with a customer, or a robot expert can train a technician how to program an industrial robot. The VirCA 3D virtual reality component can work together with the cutting edge stereoscopic 3D display technologies including the immersive 3D CAVE systems [8].

3.2. Augmented Collaboration

The 3D content and processes in VirCA can be synchronized with the real world, which allows the combination of reality and virtuality in the collaboration arena. This allows VirCA users to virtually interact not only with other users, but also with existing, remotely operated hardware and software such as robots, sensors or software pieces of control algorithms. This type of semi-virtual interaction allows the users to build distributed systems consisting real and virtual parts at the same time.

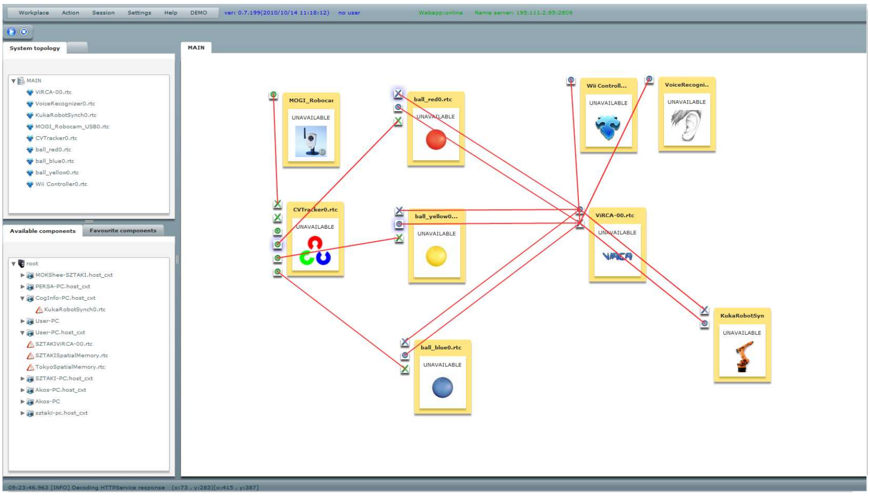
3.3. Knowledge Plug and Play

In modern information systems modularity and compatibility are important aspects. It means, that the system does not consist of a monolithic unit, but of many modules which can be treated as separate units. This structure has economic advantages: On a new demand - if the modules are functionally well delimited -, the complete system has not to be changed, but only the module which is responsible for the appropriate functionality. So in case of a modular system, the single modules can be replaced separately without changing other modules. Of course, this can be only realized if the modules are ”compatible” with each other which means that they operate according to given standards providing standard interfaces to be connected to each other. In other words, they realize the standards which serve as the base of the cooperation.

RT-Middleware (Robot Technology Middleware) is a standard system in robotics which is supported by the Japanese government. It aims the modular build-up of industrial systems. With the spread of this standard it can come true that different parts of an industrial system can be ordered from different manufacturers by the end-user thus the decrease of the prices can be expected.

RT-Middleware uses standard channels so-called ports for connecting different components. Ports realize one-way channels: they can be either providers (outputs) or consumers (inputs). Ports can be data ports through which simple date flows flow or so-called service ports through which complex functions can be accessed (remote procedure call). Ports of the components can be connected if their descriptions are identical and their directions are opposite.

RT-Middleware uses the CORBA standard for data transfer and with the extension developed in MTA SZTAKI - ”Turbo RT-Middleware” - it can use the ICE standard as well. RT-Middleware based systems must contain a CORBA naming service. The components register to the name server telling their own address. With a system editor the registered components can be queried and connected graphically to each other (Fig. 2). The connected components contact each other using the access information stored by the name server.



```
09:23:46.963 [INFO] Decoding HTTPService response   (x:73 , y:283)[x:415 , y:387]
```

Figure 2: An application in the system editor

As VirCA provides a uniform, RT-Middleware based framework the components from different sources can be easily organized into a working systems using a browser-based graphical programming environment. There are already existing VirCA components including the 3D interactive virtual environment, communication and interaction techniques, support for widely used hardware and software tools from robots to algorithms and many other useful services. This allows the plug and play exploitation of the knowledge that is embedded into the community developed components and simplifies the creation of new state of the art solutions. In our vision, RT-Middleware components and VirCA-enabled software pieces (Cyber Devices) implementing standard generic or even fully customized interfaces could be collected in a community website. System integrators could use these functional building blocks for free or even on commercial basis similarly to the popular online application stores. In this way the knowledge that is integrated into the downloadable components can be easily added to any VirCA based system in a plug and play manner.

3.4. Beyond cutting edge technology

Since the virtual environment in VirCA is generated by computers, all information has a corresponding internal representation in the system. Therefore, in this environment everything can be measured or detected, and various kinds of information can be extracted via virtual sensors, which need not even exist in the real world today. Such combination of real and virtual sensors can be very powerful. Using this design philosophy, VirCA is capable of accommodating virtual experiments using both existent and non-existent future technologies for the development of new algorithms for various sensor and actuator setups. As a practical example, a mobile platform can be equipped with virtual sensors what real counterpart would be too costly for experimental purposes (e.g. laser range finder, depth camera etc.). In this manner, the controller software can be built and tested in the same way either using real or virtual devices. In case of a robot navigation algorithm the real mobile robot is moving in a room while its virtual manifestation doing the same in the virtual environment and collecting sensor data from the virtual world using virtual sensors. The navigation software uses the virtual sensor data similarly as real world sensor input would have been processed and command the mobile platform according to the implemented algorithm.

4. Application

4.1. Architeture

The VirCA system developed in MTA SZTAKI has a modular build-up. Its central part is the virtual reality handler component together with the 3D visualization. This component operates as a database which registers the objects and the related events in the virtual reality on the one hand, and provides connection to the user in the form of 3D visualization and user interaction on the other hand. VirCA uses the Ogre3D graphical engine for 3D visualization, and the Bullet physics engine for the physical simulation of the motion of the objects. The different components of VirCA are connected together with RT-Middleware and the ICE communication engine is used for the data transfer through internet.

To the virtual reality handler component arbitrary further components can be connected, which realize some kind of functionality. Among them one type is the ”cyber device” which represents either a real or a pure virtual device in the virtual reality. In the case of a real device the cyber device realizes the connection between the ”true” reality and the virtual. In the case of a pure virtual device there is no real hardware behind the cyber device, but only a program. The connection between the cyber device and the virtual reality is two-way: on the one hand the cyber device can manipulate the virtual room through its representation and on the other hand it can receive user commands from the virtual room.

4.2. Fields of use

One important attribute of the VirCA system is the integration of information. It means that using ”real” 3D visualization, all three dimensions are available for the representation of the information. In case of a complex process, different information do not arrive in separate channels to the user - e.g. in separate displays -, but according to the reality, they can be displayed on a spatial representation. This can largely increase the efficiency of human-machine communication. In the interactive virtual environment the user can control different devices through a standard interface. For this control, arbitrary input devices are available which can pass a given information to the system in the appropriate way, e.g. a cyber device can be selected by moving the 3D pointer with the mouse, or the cyber device can be addressed with the help of the voice recognizer by saying its name (Fig. 3).

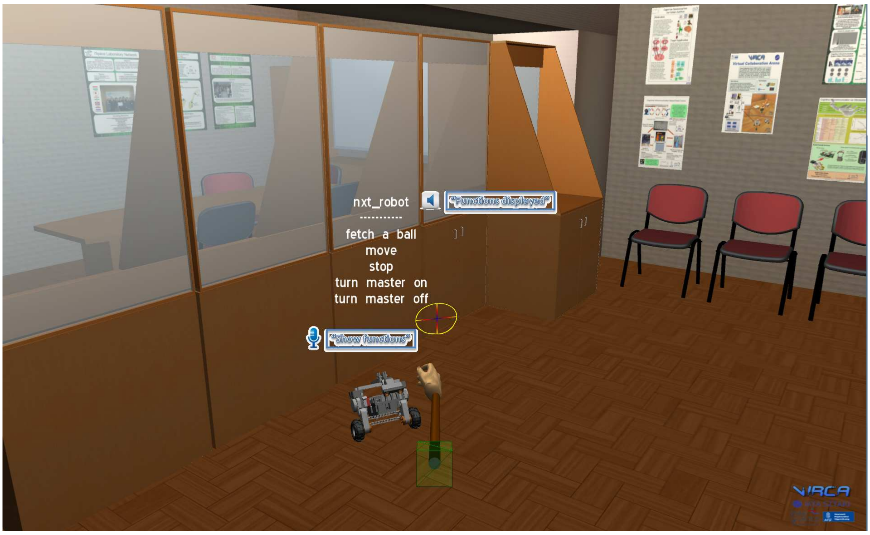


Figure 3: Commanding a cyber device in the virtual reality

One other important attribute of the VirCA system is the distributed system approach. It means that the devices represented in the virtual reality can be controlled in the same way without reference to their location in the reality. VirCA conceals the complex network connections, so the user can simply select the required device as if it would be available on the local machine.

With the help of the VirCA system virtual test cases consisting of many different systems can be created. It means that the cooperation of devices far from each other in the room can be tested without the need of delivering them to the same place. In case of large industrial robots it would be very expensive. It differs from the simulation of the devices in that way as in this case the representations of the real devices are in the virtual reality, whose motions follow the motions of the real devices. So on the one hand, with the development of simple tracker systems the development of expensive simulators can be avoided while on the other hand, the virtual representation tracked in this way corresponds better with the real device than the simulated. Before the purchase of an expensive device it can be tested whether it suits the cooperation demands of the application.

The VirCA system can be applied as a virtual information room between different devices. The status of the virtual reality can be refreshed with different kinds of sensor and tracker devices corresponding to the true reality, so that other devices can manipulate in it. So the single devices do not need to have all the abilities they use. Instead, they can use the abilities of other devices through the virtual reality, e.g. a mobile robot could have the ability of self-localization based on virtual reality refreshed by an other sensor. Moreover, in the virtual reality such sensors can be simulated as well. So complex systems can be tested without the existence of real constituent devices that are required for their usage. It facilitates the separate, independent development and testing of the devices and in case of purchase the trial of different devices. In the case of a mobile robot it can be decided easily, what kind of sensor should be selected for the realization of self-localization. In the room not only sensor and tracker devices, but only-in-program realized intelligence can be ”placed” as well. So this way, the knowledge ”plug’n’play” can be realized, i.e. the intelligence in the virtual reality can be used by other devices.

4.3. Component development

The usage of the VirCA system starts with the execution of the installer which can be downloaded from the VirCA website. It installs the virtual reality core component and other required programs. The VirCA system can be used with other components in such a way, that first the CORBA name service should be started to which the component registers in. Then the editor can be used to build or load arbitrary system of components (Fig. 2).

To the VirCA system a new cyber device can be developed basically in three ways. Either an own RT-Middleware component can be written completely, or it can be inherited from the so-called ”cyber device base” component, or the cyber device API library can be used. The first choice is the hardest but has the most facilities, while the last is the easiest but only with limited capabilities.

5. Experiences in education

The usage of the VirCA system and the development of applications for it were taught between 8-10. September 2010 at the Faculty of Informatics at the University of Debrecen related to a laboratory course. Participant students could listen presentations about the build-up, usage and programming of the VirCA system. After the presentations they could use the system on the computers of the laboratory. VirCA system, the software development toolkit and other programs required for the development had been installed in the laboratory in advance. The theoretical and pragmatical lessons were divided into five parts. In the first lesson the students studied the concept, sense and build-up of the VirCA system and its setup and handling. In the second lesson they could get familiar with the 3D content generation using free softwares and they could equip an own virtual world in the system with the generated content. In the third lesson they could come to know the structure and application of the RT-Middleware system together with the efficient usage of the RT-Middleware components of the VirCA system. In the fourth lesson the students were introduced to the possible ways of application development. At the final part of the course the gained knowledge was tested through the development of own applications: here we developed a fully functioning application step-by-step starting from an ”empty” cyber device going trough various VirCA features. In the fifth lesson the development of the application was carried on with the opportunity of discussing questions and personal ideas. As conclusion, it can be stated that the course was successful, the students could fulfil the planned tasks and succeed to learn the way of developing applications to the VirCA system. On the basis of the experience of the course we prepared some compact tutorials that are published on the webpage of VirCA.

6. Direction of development

One possible way of developing VirCA is extending the possibilities of collaboration between intelligent systems by the usage of the so-called ”semantic” information. It means, that not only data is stored in the system but in an appropriate, standard format, its meaning is represented as well. In this way, it would be semantically self-describing. Such a standard format is the ontology which is used to represent the concepts and their relations of a domain. If intelligent systems collaborate, the certain systems share their functionality in the VirCA that can be applied by other systems. Applying such shared functionality, a functional collaboration can be realized. But to apply the functionality of a certain system, in the traditional case, it needs to be known, which functionality the system exactly realizes, and how this functionality can be accessed through the interface of the system. This information can be published e.g. in a natural language description by the system developers and the developers of other systems need to interpret this description. Contrarily, in case of semantic information the system describes its functionality according to the given standard representation e.g. ontology, which representation can be interpreted by other systems using the standard representation. In this manner, the efficiency of the collaboration can be extended because the configuration of the independent systems for the given task can be generated automatically by intelligent inference based on the semantic description. The VirCA system cannot handle semantic information originally. For this task, the system needs to be extended with components capable of semantic communication. Every conventional agent can be wrapped into a semantic device in such a way that the wrapper knows the functionality of the given device. The wrapper shares its semantic representation and then mediates between the interface of the semantic communication and the original component. Beyond that, a component receives some determined user requests, represents them semantically and then performs the inference on the basis of the semantic information shared by the devices whether an appropriate configuration exists. If does, it sets it up through semantic communication.

7. Summary

VirCA (Virtual Collaboration Arena) is an ongoing development of a software framework mainly for distributed intelligent systems, integrating three key technologies: augmented 3D virtual reality, distributed systems and modular framework. In this paper the motivation and the conceptual pillars of VirCA are discussed.

Acknowledgment

The research was supported by HUNOROB project (HU0045, 0045/NA/2006-2/ÖP-9), a grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Hungarian National Development Agency.

References

[1] H. Hashimoto, “Intelligent space: Interaction and intelligence,” *Artificial Life and Robotics*, vol. 7, pp. 79–85, 2003.

[2] G. Junker, *Pro OGRE 3D Programming*, 1st ed. Apress, Sep. 2006.

[3] S. Khan, K. Qureshi, and H. Rashid, “Performance comparison of ICE, HORB, CORBA and dot NET remoting middleware technologies,” *International Journal of Computer Applications*, vol. 3, no. 11, pp. 15–18, 2010.

[4] N. Ando, T. Suehiro, K. Kitagaki, and T. Kotoku, “RT(Robot Technology)-Component and its standardization - towards component based networked robot systems development,” in *2006 SICE-ICASE International Joint Conference*, Convention Center-BEXCO, Busan, Korea, 2006, pp. 2633–2638.

[5] G. Sziebig, A. Gaudia, P. Korondi, and N. Ando, “Video image processing system for rt-middleware,” in *7th International Symposium of Hungarian Researchers on Computational Intelligence (HUCI’06)*, Budapest, Hungary, 2006, pp. 461–472.

[6] M. Amoretti and M. Reggiani, "Architectural paradigms for robotics applications," *Advanced Engineering Informatics*, vol. 24, no. 1, pp. 4–13, 2010.

[7] A. Vamos, I. Fülöp, B. Reskó, and P. Baranyi, "Collaboration in virtual reality of intelligent agents," *Acta Electrotechnica et Informatica*, vol. 10, no. 2, pp. 21–27, 2010.

[8] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, "Surround-screen projection-based virtual reality: the design and implementation of the CAVE," in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. Anaheim, CA: ACM, 1993, pp. 135–142.

Fuzzy Rule Base Extraction by Bacterial Type Algorithms using selected T-norms

L. Gál[1,2], L. T. Kóczy[1,3]

[1] SZE, 9026 Győr, Egyetem tér 1.
Phone: 96 503 400, fax: 96 503 400
[2] NyME-SEK, 9700 Szombathely, Károlyi G. tér 4.
Phone: 94 504-300, fax: 94 504-404
[3] BME, 1111 Budapest, Műegyetem rkp. 3-9.
Phone: 1 463-1111, fax: 1 463-1110
e-mail: laci.gal@gmail.com , koczy@sze.hu

Abstract: In our previous paper we proposed the Trigonometric t-norm and t-conorm and we found that it showed good properties (among various types of t-norms and t-conorms) in Fuzzy Flip-Flop based Neural Networks.

In our previous research we have done a number of efforts to improve the efficiency of the *bacterial memetic type algorithms* in the field of the *fuzzy rule base identification*. The efficiency of the bacterial type algorithms were tested by Mamdani type inference system.

Now, we have examined how the *different t-norms* used instead of the widely used *min* operator affect the learning capabilities of the system applied, the speed of the convergence in case of various training algorithms. We implemented and tested some promising t-norms (*Algebraic*, *Hamacher* (p=0) and *Trigonometric*) for the Mamdani type inference system in order to reduce computational effort and processing time. This research is focused on *non-parametric t-norms*.

Keywords: T-norms, Mamdani inference system, bacterial type algorithms, Bacterial Memetic Algorithm

1. Introduction

Since Zadeh defined the triplet of standard fuzzy complement, union and intersection in his seminal paper [13], a large variety of fuzzy set operators have been proposed. Several classes of functions have been introduced whose members satisfy the axiomatic skeleton for fuzzy sets.

Recently we investigated the function approximation properties of Fuzzy Flip-Flop based Neural Networks built on various types of t-norms and t-conorms. We were looking for a *new t-norm pair* in order to improve the capabilities of these types of neural networks. We proposed the *Trigonometric t-norm and t-conorm* [3]. In that application, among the various types of fuzzy flip-flops [9] this new t-norm pair based J-K fuzzy flip-flop performed best, or at least at the same level as the best ones [3].

In our previous research we did a number of efforts to improve the efficiency of the *bacterial memetic type algorithms* in the field of the *fuzzy rule base identification* (**FRBI**) (IBMA, BMAM, MBMA, 3 Step BMA) [4, 5, 6]. On this area the efficiency of the bacterial type algorithms were tested by *Mamdani type inference system* with *trapezoidal shaped fuzzy membership functions*, *min fuzzy operator* and *COG defuzzification method*, as these are widely used and general enough. The application of the BMA type algorithms for FRBI is possible through the method proposed by Botzheim et al. [1, 2], which determined the derivatives needed for applying the *Levenberg-Marquardt method* (***LM***) [10].

During our past FRBI-related researches we examined that which of the various bacterial algorithms show better results concerning the size of the rule base, the function approximation capabilities of the identified rule base and system, and the speed of convergence of the algorithm applied in case of certain input-output data sets. The input-output data sets we used were derived from more or less complicated test functions and from data samples of real world applications, such as

One variable functions: sine wave section, two sine waves combined with different amplitude and frequency, pH-problem, Nickel-Metal Hydride accumulator charge characteristic [3];

Two variable functions: combination of various trigonometric functions, Inverse Coordinate Transform problem;

Six variable function: a six dimensional non-polynomial function (introduced in [11].

As we mentioned before, the model used can be characterized as a *Mamdani type inference system*, with *trapezoidal shaped membership functions*, *min fuzzy operator* and *COG defuzzification*. The most interesting algorithms from the viewpoint of the examinations made were the *Bacterial Evolutionary Algorithm* (***BEA***) [11], the *Improved Bacterial Memetic Algorithm* (***IBMA***) [4] and the *Modified Bacterial Memetic Algorithm* (***MBMA***) [5].

Now, we examined how the *different t-norms* used instead of the *min fuzzy operator* affect the *learning capabilities* of the system applied and the *speed of the convergence* in case of various training algorithms. We also investigated how accurately the system thus constructed could reproduce the output values from the known inputs. Beside the *min* and the new *Trigonometric t-norms* we used other *non-parametric t-norms* that seemed to be useful in our previous research [3]. This way we applied the following t-norms in our examinations as fuzzy operators: *minimum*, *algebraic*, *Hamacher* (p=0 in the parametric version), *Trigonometric*.

In order to use these in BMA type algorithms we had to determine all the partial *derivatives* needed for the *Jacobian matrix* used by the *Levenberg-Marquardt method.* This work was based on the method (elaborated for the min fuzzy operator) [2], the derivatives were modified for the corresponding t-norms, respectively.

The outline of this paper is as follows. After the Introduction, Section 2 deals with the fuzzy rule base identification process. In Section 3 we present various bacterial type evolutionary algorithms and calculate the derivatives for the Jacobian computation.. Various t-norms and co-norms are given in Section 4. Section 5 shows how we apply other t-norms in FRBI. Finally, Section 6 provides some *simulation results* of *fuzzy rule base identification processes* using *different t-norms* in a *Mamdani type inference system* and *three different bacterial type evolutionary algorithms*, followed by a brief Conclusion and References.

2. Fuzzy rule base identification process

When we use *Mamdani type inference systems* the system utilizes the fuzzy rule base to transform the input into output data. The fuzzy rule base can be built up from rules specified by human experts, or – if no such rules are available – can be formed by automatic fuzzy rule base identification based on input-output data obtained from certain systems.

Methods developed earlier by us [4, 5, 2] are able to generate quasi-optimal rule bases based upon such input-output data. The method we used for identifying fuzzy rule bases can be characterized as follows. The input data can be obtained from a certain system in operation, can be some commonly used benchmark data, or can be produced by some kind of test function. We dealt with 1, 2 and 6-dimensional input data. The *inference system* applied deployed *with trapezoidal shaped fuzzy membership functions*, in case of more than one input variables we utilized the *minimum t-norm* as *aggregation operator*, and the *COG method* was used for *defuzzification*.

In order to obtain *quasi-optimal fuzzy rule bases* the first step is to generate a population of random fuzzy rules, then evaluate the rule bases. The evaluation of the rule bases built upon the examining of the differences between the (input and) output data and the output data produced by the inference system (***MSE*** – *Mean Squared Error*). Then we begin to improve the rule base by some method that modifies the rule base towards having a smaller error. These methods can be global or local search methods, as well as a combination of both of them the so-called memetic algorithms. We applied several bacterial type evolutionary algorithms (***PBGA*** – *Pseudo Bacterial Genetic Algorithm* [12], **BEA**, ***BMA*** – *Bacterial Memetic Algorithm*, **IBMA**, ***BMAM*** – *Bacterial Memetic Algorithm with Modified Operator Execution Order* [4] and **MBMA**; more details in Section 3) for that purpose because they proved to be rather successful in the area of the *fuzzy rule base identification*. In case of bacterial memetic algorithms we applied the *Levenberg-Marquardt method* as the local searcher part as this has a very impressive speed of convergence. During the process of fine tuning of the rule bases (the training) we can analyze the speed of the convergence in case of various training algorithms, and we check how accurately the input-output data samples are reproduced by the system using the identified fuzzy rule base.

In case of Mamdani type inference system the relative importance of the jth fuzzy variable in the ith rule is:

$$\mu_{ij}(x_j) = \frac{x_j - a_{ij}}{b_{ij} - a_{ij}} N_{i,j,1}(x_j) + N_{i,j,2}(x_j) + \frac{d_{ij} - x_j}{d_{ij} - c_{ij}} N_{i,j,3}(x_j) \tag{1}$$

where a_{ij}, b_{ij}, c_{ij} *and* d_{ij} are the breakpoints of the trapezoidal shaped fuzzy membership functions, $a_{ij} \le b_{ij} \le c_{ij} \le d_{ij}$ must hold, and

$$N_{i,j,1}(x_j) = \begin{cases} 1, & if \ \ x_j \in [a_{ij}, b_{ij}] \\ 0, & othervise \end{cases}, \ N_{i,j,2}(x_j) = \begin{cases} 1, & if \ \ x_j \in [b_{ij}, c_{ij}] \\ 0, & othervise \end{cases}, \ N_{i,j,3}(x_j) = \begin{cases} 1, & if \ \ x_j \in [c_{ij}, d_{ij}] \\ 0, & othervise \end{cases}. \tag{2}$$

The activation degree of the ith rule (if the t-norm is the minimum):

$$w_i = \min_{j=1}^{n} \mu_{ij}(x_j) \tag{3}$$

where n is the number of input dimensions. Then the *COG method* is applied as the *defuzzification* method [1].

In this paper we discuss the impact of the learning capabilities of such a system, where other t-norms are used instead of the minimum t-norm as aggregation operator.

In the cases of *non-memetic* bacterial algorithms (PBGA, BEA) it is enough to replace the minimum t-norm to another t-norm in the inference system. However, if the algorithm applied is one of the *bacterial memetic algorithms* (e.g. IBMA, MBMA) then the *derivatives* needed by the application of the *Levenberg-Marquardt method* have to be redetermined as well.

3. Bacterial type evolutionary algorithms

The original *genetic algorithm* (***GA***) was developed by Holland [7] and was based on the process of evolution of biological organisms. These processes can be easily applied in optimization problems where one individual corresponds to one solution of the problem.

Nawa, Hashiyama, Furuhashi and Uchikawa proposed a novel type of evolutionary algorithm called *Pseudo-Bacterial Genetic Algorithm* (***PBGA***) for fuzzy rule base extraction in 1997 [12]. The *Pseudo-Bacterial Genetic Algorithm* is essentially a special kind of g*enetic algorithm.* Its core contains a new genetic operation called *bacterial mutation*, which is inspired by the biological bacterial cell model, so this method mimics the microbial evolution phenomenon. Its basic idea is to improve the *parts* of *chromosomes* contained in each *bacterium*.

For the *bacterial algorithm*, the first step is to determine how the problem can be encoded in a *bacterium* (*chromosome*). Our task is to find the optimal *fuzzy rule base* for a pattern set. Thus, the parameters of the *fuzzy rules* must be encoded in the *bacterium*. In the general case the parameters of the rules are the *breakpoints* of the trapezoids, thus, a *bacterium* will contain these *breakpoints*.

The next step is to *optimize the parameters*. Therefore a procedure is working on changing the parameters, testing the model obtained by this way and selecting the best models (*bacterial mutation*). In the course of testing the input-output data used for training are compared to the input and the output of the model (SSE, *MSE*, BIC). The smaller is the error, the better is the performance of the model. The *inference system* used for the model calculations can be any of the various types of *fuzzy inference systems*.

After introducing the *PBGA* further *bacterial type evolutionary algorithms* (based on the *bacterial mutation operation*) were proposed. The *BEA* is an enhancement over the PBGA (by the *gene transfer operation)*. The *BMA* is a *memetic algorithm*, which combines the BEA and the LM method for the global and local search task. The *IBMA* improves the efficiently of the BMA specifically for FRBI applications with trapezoidal shaped fuzzy membership functions. The *BMAM* is the further development of the BMA, where higher convergence speed can be achieved by the efficiently use of the LM method. The *MBMA* combines the steps of the IBMA and the BMAM, taking advantage of their beneficial properties when using *trapezoidal shaped fuzzy membership functions* in FRBI.

For our tests the *BEA*, the *IBMA* and the *MBMA* was selected from the above methods, as these are properly representative of the 3 different groups of *bacterial type evolutionary algorithms.*

3.1. Bacterial Evolutionary Algorithm (BEA)

BEA is based on the PGBA [11]. It introduces a new genetic operation, called *gene transfer operation*. This new operation establishes *relationships* between *bacteria*. BEA is not a memetic algorithm, as it doesn't contains a local search part (in our cases the Levenberg-Marquardt method), therefore, there is no need to determine certain derivatives in order to apply it. This way, it can be easily adapted for various applications. Its main operation is the *bacterial mutation*, and the supplementary operation is the *gene transfer operation*. Its simple to implement, the algorithm's complexity is low, but the speed of convergence and the obtainable accuracy is similarly low. Figure 1 shows the flowchart of the BEA.

In our cases – namely, *using different t-norms in the inference system* and applying BEA for FRBI – it is enough to modify the aggregation operation to another one in the inference system, which is extremely simple to implement.

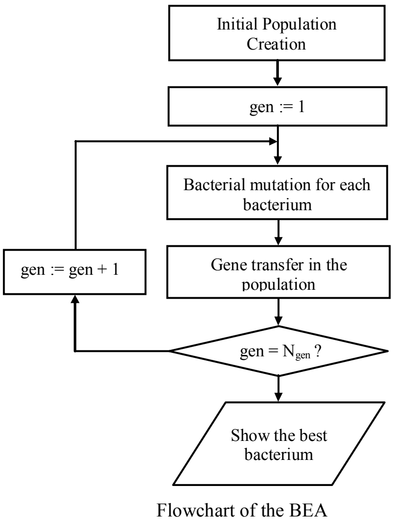

Initial Population Creation
gen := 1
Bacterial mutation for each bacterium
gen := gen + 1
Gene transfer in the population
gen = N_{gen} ?
Show the best bacterium
Flowchart of the BEA

Figure 1. Flowchart of the BEA

3.2. Improved Bacterial Memetic Algorithm (IBMA)

IBMA is practical modification of the BMA, which combines the BEA and the LM method [4]. The *evolutionary part* is able to *find the global optimum region*, but is not suitable to find the local minimum in practice. The *gradient based part* is able to reach the *local optimum*, but is very sensitive to the initial position in the search space and is unable to avoid the local optimum. Combining global and local search is expected to be beneficial.

The main operations of the IBMA are the *bacterial mutation* followed by the *Levenberg-Marquardt method*, and the supplementary operation is the *gene transfer operation*. It is almost identical to the BMA except the *knot order violation handling method* in the Levenberg-Marquardt part (method *swap* for *handling knot order violation*). Figure 2 shows the flowchart of the IBMA.

Applying the BMA or IBMA for FRBI requires to determine certain derivatives (Jacobian matrix). In case of using the minimum t-norm in the Mamdani inference system the derivatives can be computed as these are determined (by J. Botzheim et al.) [1].

In our cases – *utilizing different t-norms* – the derivatives had to be *redetermined*, so the adaptation needs much more effort.

3.2.1. Derivatives for the Jacobian computation

The key in applying the LM method is how to get the LM update vector. To calculate the update vector the Jacobian matrix $\underline{J}$ with respect to the parameters in the rules has to be computed. In case of *minimum t-norm aggregation operation* and trapezoidal shaped fuzzy membership function based Mamdani inference system with COG defuzzification method this will be done as shown below, in a pattern by pattern (*pt*) basis. [2]

$$\underline{\underline{J}}[k]=\left[\frac{\partial y(\underline{x}^{(pt)})[k]}{\partial \underline{p}[k]}\right] \tag{4}$$

where *k* is the iteration variable. This can be written as follows:

$$\underline{\underline{J}}=\left[\frac{\partial y\left(\underline{x}^{(pt)}\right)}{\partial a_{11}}\frac{\partial y\left(\underline{x}^{(pt)}\right)}{\partial b_{11}}\cdots\frac{\partial y\left(\underline{x}^{(pt)}\right)}{\partial a_{12}}\cdots\frac{\partial y\left(\underline{x}^{(pt)}\right)}{\partial d_{1}}\cdots\frac{\partial y\left(\underline{x}^{(pt)}\right)}{\partial d_{R}}\right], \tag{5}$$

where

$$\begin{aligned}\frac{\partial y\left(\underline{x}^{(pt)}\right)}{\partial a_{ij}}&=\frac{\partial y}{\partial w_i}\frac{\partial w_i}{\partial \mu_{ij}}\frac{\partial \mu_{ij}}{\partial a_{ij}}\\ \frac{\partial y\left(\underline{x}^{(pt)}\right)}{\partial b_{ij}}&=\frac{\partial y}{\partial w_i}\frac{\partial w_i}{\partial \mu_{ij}}\frac{\partial \mu_{ij}}{\partial b_{ij}}\\ \frac{\partial y\left(\underline{x}^{(pt)}\right)}{\partial c_{ij}}&=\frac{\partial y}{\partial w_i}\frac{\partial w_i}{\partial \mu_{ij}}\frac{\partial \mu_{ij}}{\partial c_{ij}}\\ \frac{\partial y\left(\underline{x}^{(pt)}\right)}{\partial d_{ij}}&=\frac{\partial y}{\partial w_i}\frac{\partial w_i}{\partial \mu_{ij}}\frac{\partial \mu_{ij}}{\partial d_{ij}}\end{aligned} \tag{6}$$

w_i denotes the activation degree of the ith rule (the t-norm is the minimum):

$$w_i=\min_{j=1}^{n}\mu_{ij}\left(x_j\right) \tag{7}$$

where *n* is the number of the input dimensions and μ_{ij} denotes the relative importance of the jth fuzzy variable in the ith rule.

$$\mu_{ij}(x_j)=\frac{x_j-a_{ij}}{b_{ij}-a_{ij}}N_{i,j,1}(x_j)+N_{i,j,2}(x_j)+\frac{d_{ij}-x_j}{d_{ij}-c_{ij}}N_{i,j,3}(x_j) \tag{8}$$

$$\begin{aligned}N_{i,j,1}(x_j)&=\begin{cases}1, & if\ \ x_j\in\left[a_{ij},b_{ij}\right]\\ 0, & othervise\end{cases}\\ N_{i,j,2}(x_j)&=\begin{cases}1, & if\ \ x_j\in\left[b_{ij},c_{ij}\right]\\ 0, & othervise\end{cases}\\ N_{i,j,3}(x_j)&=\begin{cases}1, & if\ \ x_j\in\left[c_{ij},d_{ij}\right]\\ 0, & othervise\end{cases}\end{aligned} \tag{9}$$

The ith output is being cut in the height w_i, and with the *Center of Gravity* (COG) *defuzzification method* the output is calculated:

$$y(\underline{x}) = \frac{\sum_{i=1}^{N_{FuzzRules}} \int_{y\in \mathrm{supp}\mu_i(y)} y\mu_i(y)dy}{\sum_{i=1}^{N_{FuzzRules}} \int_{y\in \mathrm{supp}\mu_i(y)} \mu_i(y)dy} \tag{10}$$

If this defuzzification method is used, the integrals can be easily computed:

$$y(x) = \frac{1}{3}\frac{\sum_{i=1}^{N_{FuzzRules}}(C_i + D_i + E_i)}{\sum_{i=1}^{N_{FuzzRules}} 2w_i(d_i - a_i) + w_i^2(a_i - b_i + c_i - d_i)} \tag{11}$$

$$C_i = 3w_i(d_i^2 - a_i^2)(1 - w_i)$$
$$D_i = 2w_i^2(c_i d_i - a_i b_i)$$
$$E_i = w_i^3(c_i - d_i + a_i - b_i)(c_i - d_i - a_i + b_i)$$

It can be seen from (7) that w_i depends on the membership functions, and each membership function depends only on four parameters (breakpoints). So, the derivatives of w_i will be:

$$\frac{\partial w_i}{\partial \mu_{ij}} = \begin{cases} 1, & if \quad \mu_{ij} = \min_{k=1}^{n} \mu_{ik} \\ 0, & \quad othervise \end{cases} \tag{12}$$

$$\frac{\partial \mu_{ij}}{\partial a_{ij}} = \frac{x_j^{(p)} - b_{ij}}{\left(b_{ij} - a_{ij}\right)^2} N_{i,j,1}\left(x_j^{(p)}\right) \quad , \quad \frac{\partial \mu_{ij}}{\partial b_{ij}} = \frac{a_{ij} - x_j^{(p)}}{\left(b_{ij} - a_{ij}\right)^2} N_{i,j,1}\left(x_j^{(p)}\right) \tag{13}$$

$$\frac{\partial \mu_{ij}}{\partial c_{ij}} = \frac{d_{ij} - x_j^{(p)}}{\left(d_{ij} - c_{ij}\right)^2} N_{i,j,3}\left(x_j^{(p)}\right) \quad , \quad \frac{\partial \mu_{ij}}{\partial d_{ij}} = \frac{x_j^{(p)} - c_{ij}}{\left(d_{ij} - c_{ij}\right)^2} N_{i,j,3}\left(x_j^{(p)}\right) \tag{14}$$

$\frac{\partial y}{\partial w_i}$ and the derivatives of the output membership functions' parameters can be also computed from (11).

After calculating the Jacobian matrix the LM update vector can be calculated as follows:

$$\underline{s} = -\begin{bmatrix} \underline{\underline{J}} \\ \sqrt{\lambda}\,\underline{\underline{I}} \end{bmatrix}^{+} \begin{bmatrix} \underline{y}(\underline{p}) - \underline{t} \\ \underline{0} \end{bmatrix} \tag{15}$$

where the operator $^{+}$ denotes the Moore-Penrose pseudoinverse.

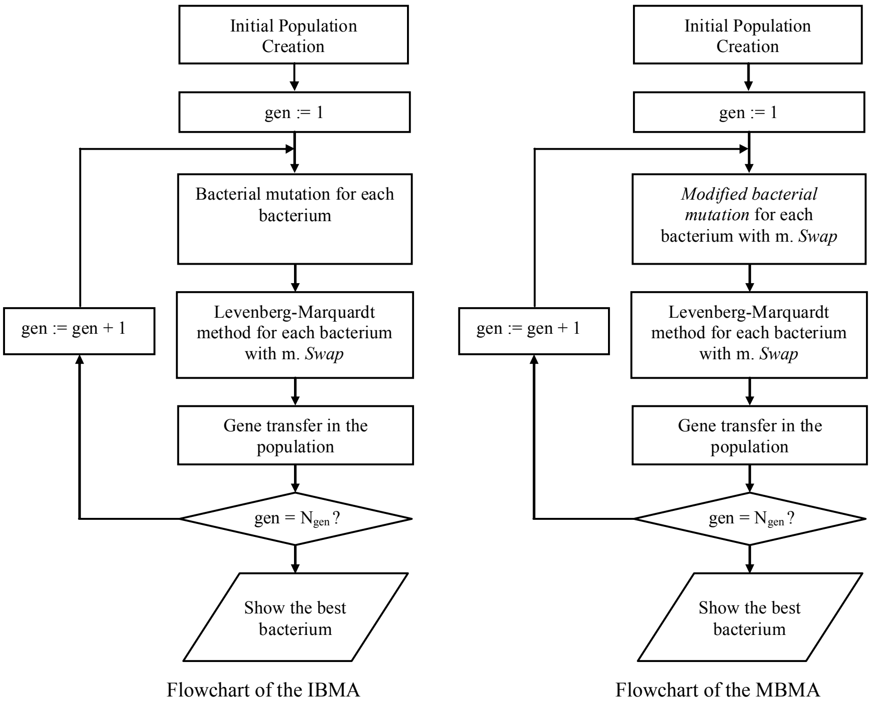

Initial Population Creation
gen := 1
Bacterial mutation for each bacterium
Levenberg-Marquardt method for each bacterium with m. Swap
gen := gen + 1
Gene transfer in the population
gen = N$_{gen}$?
Show the best bacterium
Flowchart of the IBMA
Initial Population Creation
gen := 1
Modified bacterial mutation for each bacterium with m. Swap
Levenberg-Marquardt method for each bacterium with m. Swap
gen := gen + 1
Gene transfer in the population
gen = N$_{gen}$?
Show the best bacterium
Flowchart of the MBMA

Figure 2. Flowcharts of IBMA and MBMA

3.3. Modified Bacterial Memetic Algorithm (MBMA)

MBMA is a very recent approach (2008) [5]. It differs from the IBMA in the *bacterial mutation* step, as the MBMA utilizes a *modified version of the bacterial mutation operation*. In this *modified bacterial mutation* a number of Levenberg-Marquardt steps are executed *after each bacterial mutational step*. This way higher convergence speed and lower MSE values can be achieved. The main steps are: *bacterial mutational steps + LM iterations*, *LM iterations*, *gene transfer*. Figure 2 shows the flowchart of the MBMA.

As this algorithm is working with the same operation set as the IBMA there is nothing else need to implement for utilizing the other t-norms, if the modifications for the IBMA are done.

4. Various t-norms an co-norms

Klement, Mesiar and Pap enumerate and give the basic definitions and properties of the most general fuzzy operations in [8] including also graphical illustrations and comparisons. Table 1 shows some selected t-norms and t-conorms. Each class contains parameterized as well as non-parameterized operators. All t-norms could be used as aggregation operator in the Mamdani type inference system. We implemented the *min*, *Algebraic*, *Hamacher* (p=0) (= *Dombi* (α=1)) types from the t-norms in Table 1 and a new t-norm (*Trigonometric*) for FRBI.

Parameters w, α, v and s lie within the open interval $(0,\infty)$, $d \in [0,1]$, $s \neq 1$ and $p \neq 0$.

Table 1. T-norms and t-conorms

Fuzzy operation	t-norm; *i* (*a*, *b*)	co-norm; *u* (*a*, *b*)
Standard (min-max)	$\min(a,b)$	$\max(a,b)$
Algebraic	ab	$a+b-ab$
Drastic	a when b=1, b when a=1, 0 otherwise	a when b=0, b when a=0, 1 otherwise
Łukasiewicz	$\max(0,a+b-1)$	$\min(1,a+b)$
Yager	$1-\min\left[1,((1-a)^w+(1-b)^w)^{1/w}\right]$	$\min\left[1,(a^w+b^w)^{1/w}\right]$
Dombi	$\frac{1}{1+\left[(1/a-1)^\alpha+(1/b-1)^\alpha\right]^{1/\alpha}}$	$\frac{1}{1+\left[(1/a-1)^{-\alpha}+(1/b-1)^{-\alpha}\right]^{-1/\alpha}}$
Hamacher	$\frac{ab}{v+(1-v)(a+b-ab)}$	$\frac{a+b-(2-v)ab}{1-(1-v)ab}$
Frank	$\log_s\left[1+\frac{(s^a-1)(s^b-1)}{s-1}\right]$	$1-\log_s\left[1+\frac{(s^{1-a}-1)(s^{1-b}-1)}{s-1}\right]$
Dubois-Prade	$\frac{ab}{\max(a,b,d)}$	$\frac{a+b-ab-\min(a,b,1-d)}{\max(1-a,1-b,d)}$
Schweizer-Sklar	$\max(0,a^p+b^p-1)^{1/p}$	$1-\max(0,(1-a)^p+(1-b)^p-1)^{1/p}$

4.1. The Trigonometric t-norm and co-norm

The fuzzy literature offers a large variety of triangular operators; researchers still propose again and again new fuzzy operations to be used in a given field. Obviously, the performance of fuzzy systems depends on the choice of different triangular operators. Despite the variety of available fuzzy set operators, however, the classic triplet of complement, intersection and union still bear particular significance, especially in the practical applications. The big challenge for fuzzy researchers is to fit the fuzzy sets into the context of applications.

In [3] a new triangular t-norm and t-conorm was presented. The novel fuzzy operations combined with the standard negation are applied in a practical problem, namely, they are proposed as suitable triangular norms for defining a fuzzy flip-flop based neuron.

The proposed connectives consist of simple combinations of trigonometric functions. Naturally, the new t-norm and t-conorm satisfy the axiomatic skeleton for fuzzy sets and are continuous. The basic motivation for constructing new norms was to have fuzzy flip-flops with sigmoid transfer characteristics in some particular cases.

For fuzzy intersection and union we have defined the following:

The Trigonometric t-norm is specified by the function

$i_T : [0,1] \times [0,1] \to [0,1]$ defined as:

$$i_T(x,y) = \frac{2}{\pi} \cdot \arcsin\left(\sin\left(x\frac{\pi}{2}\right) \cdot \sin\left(y\frac{\pi}{2}\right)\right) \tag{16}$$

The Trigonometric t-conorm is specified by the function

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

$$u_T(x,y) = \frac{2}{\pi} \cdot \arccos\left(\cos\left(x\frac{\pi}{2}\right) \cdot \cos\left(y\frac{\pi}{2}\right)\right) \tag{17}$$

The characters of the well known *standard*, *algebraic*, *Hamacher*, *Dombi*, *Yager* and of the novel *Trigonometric t-norms* for selected parameter values (p=0, *w* = 2 and α = 2) are illustrated in Figure 3. It can be seen, that the *Yager t-norm* produced a horizontal plate at the beginning of the interval.

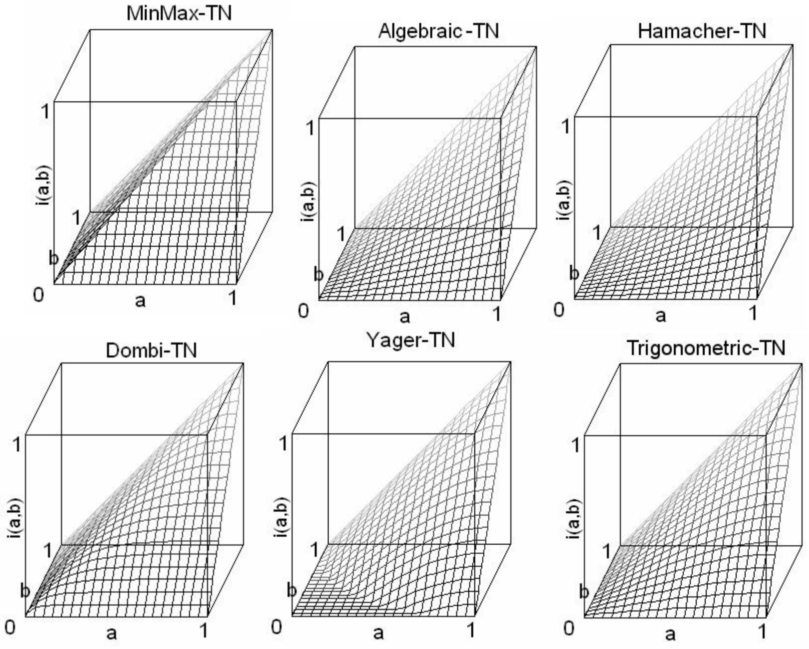


Figure 3. Characteristics of some selected t-norms

5. Applying other t-norms in FRBI

Beside the *minimum t-norm* we decided to implement some promising t-norms for the Mamdani type inference system and for the FRBI process. We avoided using parametric t-norms (much more complicated derivatives and computation complexity) and t-norms with min or max operations (because the horizontal plates in the graph should be avoided as those result 0 valued derivatives).

We have selected the following t-norms for comparison in the area of FRBI:

- Standard $\min(a,b)$ (18)
- Algebraic $a \cdot b$ (19)
- Hamacher (p=0) (Dombi (α=1)) $\dfrac{a \cdot b}{a+b-a \cdot b}$ (20)
- Trigonometric $\dfrac{2}{\pi} \cdot \arcsin\left(\sin\left(x\dfrac{\pi}{2}\right) \cdot \sin\left(y\dfrac{\pi}{2}\right)\right)$ (21)

When using the Levenberg-Marquardt method in FRBI process (BMA, IBMA, BMAM, MBMA), you need the derivatives of the multiple argument aggregation expressions.

5.1. Derivatives for the Jacobian computation

The derivatives can be calculated in the same way as for the minimum t-norm based ones (3.2.1), except the parts concerned by the t-norm used (3, 6, 7, *12*). These parts can be calculated by the following.

5.1.1. Algebraic

$$i_{A\lg}(a,b) = a \cdot b \tag{22}$$

$$w_i = \mu_{11} \cdot \mu_{12} \cdot \mu_{13} = \prod_{j=1}^{n} \mu_{ij} = \mathrm{P}_i \tag{23}$$

$$\frac{\partial w_i}{\partial \mu_{ij}} = \frac{\mathrm{P}_i}{\mu_{ij}} = \frac{1}{\mu_{ij}} \cdot w_i \tag{24}$$

if $w_i > 0$.

5.1.2. Hamacher (p=0) *(=Dombi (α=1))*

The Hamacher (p=0) operator is the Dombi operator if α=0.

$$i_{Ham}(a,b) = \frac{a \cdot b}{a+b-a \cdot b} \tag{25}$$

$$w_i = \frac{\mu_{11} \cdot \mu_{12} \cdot \mu_{13}}{\mu_{11} \cdot \mu_{12} + \mu_{11} \cdot \mu_{13} + \mu_{12} \cdot \mu_{13} - 2\mu_{11} \cdot \mu_{12} \cdot \mu_{13}} \tag{26}$$

$$w_i = \frac{\mu_{11} \cdot \mu_{12} \cdot \mu_{13} \cdot \mu_{14}}{\mu_{11} \cdot \mu_{12} \cdot \mu_{13} + \mu_{11} \cdot \mu_{12} \cdot \mu_{14} + \mu_{11} \cdot \mu_{13} \cdot \mu_{14} + \mu_{12} \cdot \mu_{13} \cdot \mu_{14} - 3 \cdot \mu_{11} \cdot \mu_{12} \cdot \mu_{13} \cdot \mu_{14}} \tag{27}$$

$$w_i = \frac{1}{1-n+\sum_{j=1}^{n}\frac{1}{\mu_{ij}}} = \frac{1}{S_r} \tag{28}$$

$$\frac{\partial w_i}{\partial \mu_{ij}} = \frac{1}{\mu_{ij}^{\;2}\cdot\left(1-n+\sum_{j=1}^{n}\frac{1}{\mu_{ij}}\right)^2} = \frac{1}{\mu_{ij}^{\;2}\cdot S_r^{\;2}} = \frac{1}{\mu_{ij}^{\;2}}\cdot w_i^{\;2} \tag{29}$$

if $w_i > 0$.

5.1.3. Trigonometric

$$i_{Trig}(a,b) = \frac{2}{\pi}\arcsin\left[\sin\left(a\cdot\frac{\pi}{2}\right)\cdot\sin\left(b\cdot\frac{\pi}{2}\right)\right] \tag{30}$$

$$w_i = \frac{2}{\pi}\arcsin\left[\sin\left(\mu_{11}\cdot\frac{\pi}{2}\right)\cdot\sin\left(\mu_{12}\cdot\frac{\pi}{2}\right)\cdot\sin\left(\mu_{13}\cdot\frac{\pi}{2}\right)\right] \tag{31}$$

$$w_i = \frac{2}{\pi}\arcsin\left[\prod_{i=1}^{n}\sin\left(\mu_{ij}\cdot\frac{\pi}{2}\right)\right] = \frac{2}{\pi}\arcsin[\mathrm{P}_i] \tag{32}$$

$$\mathrm{P}_i = \prod_{i=1}^{n}\sin\left(\mu_{ij}\cdot\frac{\pi}{2}\right) \tag{33}$$

$$\frac{\partial w_i}{\partial \mu_{ij}} = \frac{\cos\left(\mu_{ij}\cdot\frac{\pi}{2}\right)\cdot\frac{\mathrm{P}_i}{\sin\left(\mu_{ij}\cdot\frac{\pi}{2}\right)}}{\sqrt{1-\mathrm{P}_i^2}} \tag{34}$$

$$\frac{\partial w_i}{\partial \mu_{ij}} = \frac{\cos\left(\mu_{ij}\cdot\frac{\pi}{2}\right)}{\sin\left(\mu_{ij}\cdot\frac{\pi}{2}\right)}\cdot\frac{\mathrm{P}_i}{\sqrt{1-\mathrm{P}_i^2}} = \cot\left(\mu_{ij}\cdot\frac{\pi}{2}\right)\cdot\frac{\mathrm{P}_i}{\sqrt{1-\mathrm{P}_i^2}} \tag{35}$$

if $0 < w_i < 1$.

6. Simulations

We have run extensive simulations in order to examine how the *different t-norms affected the learning capabilities* of the system applied, the *speed of the convergence* in case of using different training algorithms.

With 1-dimensional input there is no sense to examine the effect on using different t-norms, as these are only required in the Mamdani inference machine for more than one input. Therefore, the 1-dimensional ones from our standard test functions were omitted; the tests were performed with two and six variable test functions. These were as follows:

6.1. Test functions, t-norms tested, training algorithms used

a, Test function with 2 input variables (2iv)

$$\begin{gathered} y = f_2(\underline{x}) = \sin^5(0.5 \cdot x_1) \cdot \cos(0.7 \cdot x_2) \\ x_1 \in [0...3\pi], x_2 \in [0...3\pi] \end{gathered} \tag{36}$$

200 samples
T-norms: Minimum, Algebraic, Hamacher (p=0), Trigonometric
Training algorithms: BEA, IBMA, MBMA

b, Test function with 6 input variables (6iv)

$$\begin{gathered} y = f_3(\underline{x}) = x_1 + x_2^{0.5} + x_3 \cdot x_4 + 2 \cdot e^{2\cdot(x_5 - x_6)} \\ x_1 \in [1...5], x_2 \in [1...5], x_3 \in [0...4], \\ x_4 \in [0...0.6], x_5 \in [0...1], x_6 \in [0...1.2] \end{gathered} \tag{37}$$

500 samples
T-norms: Minimum, Algebraic, Hamacher (p=0), Trigonometric
Training algorithms: BEA, IBMA, MBMA

Common parameters:

Fuzzy rules: 5
Population size: 7
Clones: 7
Gene transfers per generation: 3
MSE: 20 runs average
BEA: 100.000 evaluations/run
IBMA, MBMA: 10.000 evaluations/run

6.2. Results

Figures 4-6 and Figures 7-9 present the graphs of the simulations in case of the test function with 2 and 6 input variables, respectively. We compared the behavior of the systems trained with BEA, IBMA and MBMA algorithms. The graphs show the relation between the number of model performance evaluations (horizontal axis) and the model mean or median of the train MSE values (vertical axis).

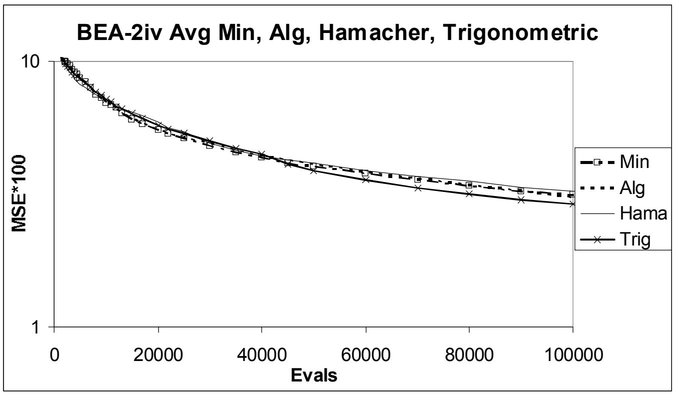


Figure 4. 2iv, BEA

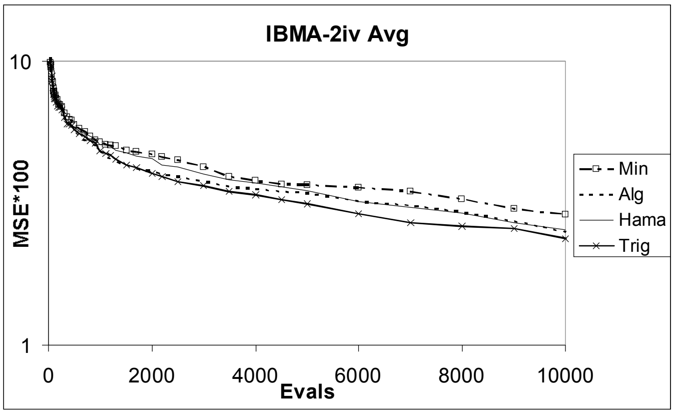


Figure 5. 2iv, IBMA

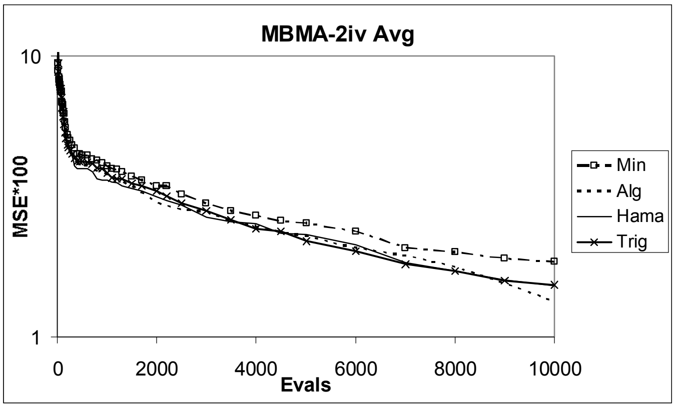


Figure 6. 2iv, MBMA

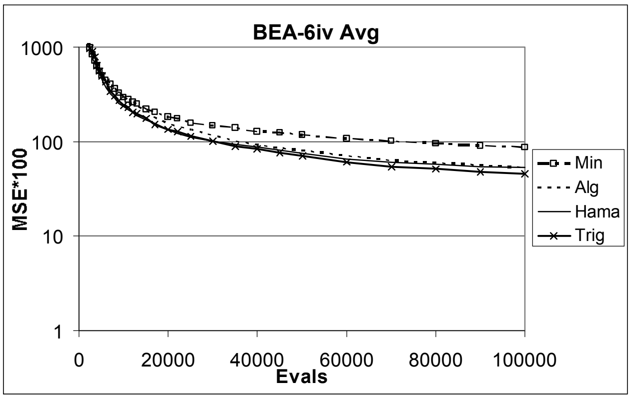


Figure 7. 6iv, BEA

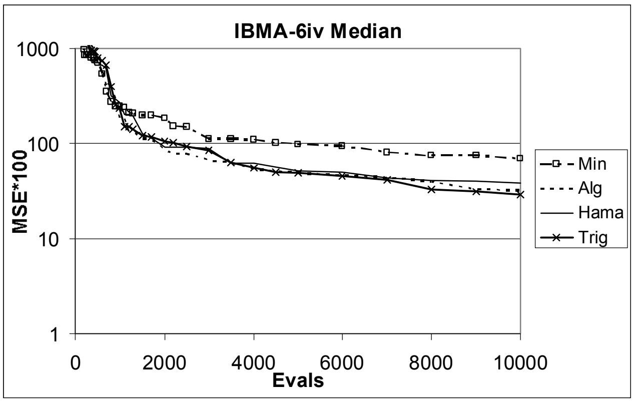


Figure 8. 6iv, IBMA

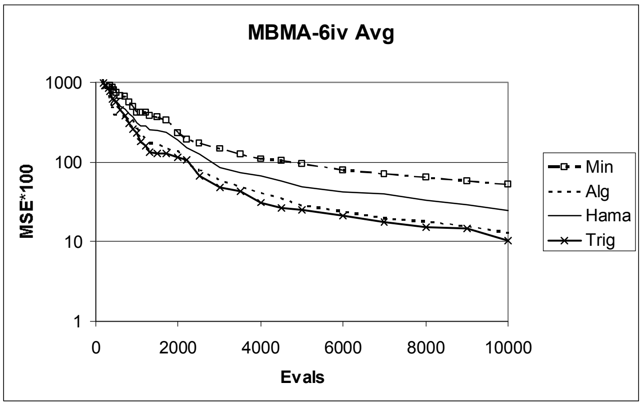


Figure 9. 6vi, MBMA.

Comparing the simulation results it is showed that the system based on Trigonometric and algebraic t-norms provide the best results, respectively. Furthermore, in the last case (Figure 9) it was enough to run about *one fourth* of the 10.000 evaluations (2.500) with the *Trigonometric t-norm* to achieve the same auspicious error values as the final error with the *minimum t-norm*.

Table 2 summarizes the simulation results by indicating the average and median of the train MSE values of the various systems trained with BEA, IBMA and MBMA algorithms, case of test function with 2 and 6 input variables.

In the table the worst MSE results are highlighted in *italics*, while the best MSE values are highlighted by *bold* characters. The *improvements* related to the *minimum aggregation operation* based values are also calculated in the best cases. For example, –80% means five times better result. In the last column we show the total number of *model performance evaluations* (Evals) for one simulation (one run), and additionally below them the approximately number of evaluations needed with the *Trigonometric t-norm* to achieve the same error as with the minimum t-norm (~Evals). For example, 10000 / 2500 means that one fourth of training cycles (and time) is required to achieve the same result.

*Table 2. Simulation results, MSE*100 values*

T-norm	Minimum		Algebraic		Hamacher		Trigonometric		Evals
	Avg.	Med.	avg.	med.	avg.	Med.	avg.	Med.	~Evals
2iv, BEA Improvement	*3.148*	*3.01*	3.064	3.07	*3.24*	*3.38*	**2.88** –8%	**2.9** –4%	100000 81000
2iv, IBMA	*2.886*	*2.661*	2.509	2.527	2.546	2.597	**2.378** –18%	**2.235** –16%	10000 6000
2iv, MBMA	*1.844*	*1.809*	**1.338** -27%	**1.38** -24%	1.519	1.515	1.528 –17%	**1.374** –24%	10000 7100
6iv, BEA	*86.42*	*86.98*	53.12	50.21	53.34	53.58	**45.32** –48%	**40.09** –54%	100000 36000
6iv, IBMA	*76.32*	*69.68*	33.33	32.66	44.99	38.56	**29.2** –62%	**28.87** –59%	10000 3500
6iv, MBMA	*52.85*	*50.82*	13.02	16.04	24.45	23.77	**10.29** –81%	**7.158** –86%	10000 2500

7. Conclusions

In our previous researches we have investigated the potential enhancements of the *Bacterial Memetic Algorithm used for automatic fuzzy rule base identification*. We have proposed some new algorithms in this field (IBMA, BMAM, MBMA and 3 Step BMA). Furthermore, we proposed a new triangular norm and conorm pair, called *Trigonometric t-norm a t-conorm*, which is proved to be rather successful in fuzzy flip-flop based neural networks.

In this work we have examined how the *different t-norms* used instead of the *min fuzzy operator* affect the *learning capabilities* of the system applied, the *speed of the convergence* in case of different training algorithms, and *how accurately* input-output data samples could be reproduced by using the fuzzy rule base obtained from the *fuzzy rule base identification process*. We have extensively tested the promising *non-parametric t-norms*, included the new *Trigonometric t-norm*. This way we applied the following t-norms in our examinations as fuzzy aggregation operator: *minimum*, *algebraic*, *Hamacher (p=0) (Dombi (α=0))*, *Trigonometric*, and tested them by three different training algorithms, namely BEA, IBMA and MBMA.

In order to use the other t-norms in the inference system trained by a memetic algorithm (IBMA, MBMA) we had to redetermine a number of derivatives in the Jacobian matrix used for the Levenberg-Marquardt method.

We expected that using other t-norms as aggregation operator in the Mamdani type inference system result in improved learning capabilities – higher speed of convergence and lower overall MSE values.

The simulations showed that it was reasonable to investigate the use of the other t-norms in the inference system, because in almost all cases the convergence speed and the accuracy of the fuzzy rule bases had been improved (except in the case of the BEA with 2 input variables test function, where there were no appreciable differences – 8% for the Trigonometric t-norm). The overall errors have been decreased by 18% (simple test function) up to *80%* (more complex test function), when using the Trigonometric t-norm instead of the conventional minimum operation.

While this research was focused on non-parametric t-norms, further effort to investigate the behavior of the system based on various parametric t-norms well known from the literature seems to be necessary.

Acknowledgement:

Research supported by the National Scientific Research Fund Grant OTKA K75711, further by Széchenyi István University Main Research Direction Grant 2010 and Scientific Committee, University of West Hungary, Campus Savaria.

References

[1] Botzheim, J., Cabrita, C., Kóczy, L. T., Ruano, A. E.: *Estimating Fuzzy Membership Functions Parameters by the Levenberg-Marquardt Algorithm*, FUZZ-IEEE 2004, Budapest, Hungary pp. 1667-1672.

[2] Botzheim, J., Cabrita, C., Kóczy, L. T., Ruano, A. E.: *Fuzzy Rule Extraction by Bacterial Memetic Algorithm*, IFSA 2005, Beijing, China, 2005, pp.1563-1568.

[3] Gál, L., Lovassy, R., Kóczy, L. T.: *Function Approximation Performance of Fuzzy Neural Networks Based on Frequently Used Fuzzy Operations and a Pair of New Trigonometric Norms*, WCCI 2010 IEEE World Congress on Computational Intelligence, FUZZ-IEEE, July, 18-23, 2010, Barcelona, Spain, pp. 1514-1521.

[4] Gál, L., Botzheim, J., Kóczy, L. T.: *Improvements to the Bacterial Memetic Algorithm used for Fuzzy Rule Base Extraction*, Computational Intelligence for Measurement Systems and Applications, CIMSA 2008, Istanbul, Turkey, 2008, pp. 38-43.

[5] Gál, L., Botzheim, J., Kóczy, L. T.: *Modified Bacterial Memetic Algorithm used for Fuzzy Rule Base Extraction*, 5th International Conference on Soft Computing as Transdisciplinary Science and Technology, CSTST 2008, Paris, France, 2008.

[6] Gál, L., Kóczy, L. T., Lovassy, R.: *Three Step Bacterial Memetic Algorithm*, Proc. of 14th IEEE International Conference on Intelligent Engineering Systems INES 2010, Las Palmas of Gran Canaria, Spain, pp. 31-36.

[7] Holland, J. H.: *Adaptation in Nature and Artificial Systems: An Introductory Analysis with Applications to Biology*, Control, and Artificial Intelligence, The MIT Press, Cambridge, MA, 1992.

[8] Klement, E. P., Mesiar, R., Pap, E.: *Triangular Norms*, Series: Trends in Logic , vol.8, 2000.

[9] Koshak, E., Noore, A., Lovassy, R.: *Intelligent Reconfigurable Universal Fuzzy Flip-Flop*, IEICE Electronics Express, Vol. 7, No. 15, 2010, pp. 1119-1124.

[10] Marquardt, D. *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*, SIAM J. Appl. Math., 11, 1963, pp. 431-441.

[11] Nawa, N. E., Furuhashi, T.: *Fuzzy Systems Parameters Discovery by Bacterial Evolutionary Algorithms*, IEEE Transactions on Fuzzy Systems 7, 1999, pp. 608-616.

[12] Nawa, N. E., Hashiyama, T., Furuhashi, T., Uchikawa, Y.: *A study on fuzzy rules discovery using pseudo-bacterial genetic algorithm with adaptive operator*, Proceedings of IEEE Int. Conf. on Evolutionary Computation, ICEC'97, 1997.

[13] Zadeh, L. A.: *Fuzzy Sets*, Information and Control 8, 1965, pp. 338-353.