INTERNATIONAL JOURNAL

of

COMPUTERS, COMMUNICATIONS & CONTROL

With Emphasis on the Integration of Three Technologies

IJCCC

Year: 2010 Volume: V Number: 3 (September)

Agora University Editing House

CCC Publications

www.journal.univagora.ro

Ştefan I. Nitchi
Department of Economic Informatics
Babes Bolyai University, Cluj-Napoca, Romania
St. T. Mihali, Nr. 58-60, 400591, Cluj-Napoca
nitchi@econ.ubbcluj.ro

Shimon Y. Nof
School of Industrial Engineering
Purdue University
Grissom Hall, West Lafayette, IN 47907, U.S.A.
nof@purdue.edu

Stephan Olariu
Department of Computer Science
Old Dominion University
Norfolk, VA 23529-0162, U.S.A.
olariu@cs.odu.edu

Horea Oros
Department of Mathematics and Computer Science
University of Oradea, Romania
St. Universitatii No. 1, 410087, Oradea, Romania
horos@uoradea.ro

Gheorghe Păun
Institute of Mathematics
of the Romanian Academy
Bucharest, PO Box 1-764, 70700, Romania
gpaun@us.es

Mario de J. Pérez Jiménez
Dept. of CS and Artificial Intelligence
University of Seville
Sevilla, Avda. Reina Mercedes s/n, 41012, Spain
marper@us.es

Dana Petcu
Computer Science Department
Western University of Timisoara
V.Parvan 4, 300223 Timisoara, Romania
petcu@info.uvt.ro

Radu Popescu-Zeletin
Fraunhofer Institute for Open
Communication Systems
Technical University Berlin, Germany
rpz@cs.tu-berlin.de

Imre J. Rudas
Institute of Intelligent Engineering Systems
Budapest Tech
Budapest, Bécsi út 96/B, H-1034, Hungary
rudas@bmf.hu

Athanasios D. Styliadis
Alexander Institute of Technology
Agiou Panteleimona 24, 551 33
Thessaloniki, Greece
styl@it.teithe.gr

Gheorghe Tecuci
Learning Agents Center
George Mason University
University Drive 4440, Fairfax VA 22030-4444, U.S.A.
tecuci@gmu.edu

Horia-Nicolai Teodorescu
Faculty of Electronics and Telecommunications
Technical University "Gh. Asachi" Iasi
Iasi, Bd. Carol I 11, 700506, Romania
hteodor@etc.tuiasi.ro

Dan Tufiş
Research Institute for Artificial Intelligence
of the Romanian Academy
Bucharest, "13 Septembrie" 13, 050711, Romania
tufis@racai.ro

Lotfi A. Zadeh
Department of Computer Science and Engineering
University of California
Berkeley, CA 94720-1776, U.S.A.
zadeh@cs.berkeley.edu

Short Description of IJCCC

Title of journal: International Journal of Computers, Communications and Control
Acronym: IJCCC
International Standard Serial Number: ISSN 1841-9836, E-ISSN 1841-9844
Publisher: CCC Publications - Agora University
Starting year of IJCCC: 2006
Founders of IJCCC: I. Dzitac, F.G. Filip and M.J. Manolescu
Editorial Office:
R&D Agora Ltd. / S.C. Cercetare Dezvoltare Agora S.R.L.
Piaţa Tineretului 8, Oradea, jud. Bihor, Romania, Zip Code 410526
Tel./ Fax: +40 359101032
E-mail: ijccc@univagora.ro, rd.agora@univagora.ro, ccc.journal@gmail.com
Website: www.journal.univagora.ro
Number of issues/year: IJCCC has 4 issues/odd year (March, June, September, December) and 5 issues/even year (March, September, June, November, December). Every even year IJCCC will publish a supplementary issue with selected papers from the International Conference on Computers, Communications and Control.
Coverage:

- Beginning with Vol. 1 (2006), Supplementary issue: S, IJCCC is covered by Thomson Reuters - SCI Expanded and is indexed in ISI Web of Science.
- Journal Citation Reports/Science Edition 2009:
  - Impact factor = 0.373
  - Immediacy index = 0.205
- Beginning with Vol. 2 (2007), No.1, IJCCC is covered in EBSCO.
- Beginning with Vol. 3 (2008), No.1, IJCCC, is covered in SCOPUS.

Scope: IJCCC is directed to the international communities of scientific researchers in universities, research units and industry. IJCCC publishes original and recent scientific contributions in the following fields: Computing & Computational Mathematics; Information Technology & Communications; Computer-based Control.
Unique features distinguishing IJCCC: To differentiate from other similar journals, the editorial policy of IJCCC encourages especially the publishing of scientific papers that focus on the convergence of the 3 "C" (Computing, Communication, Control).
Policy: The articles submitted to IJCCC must be original and previously unpublished in other journals. The submissions will be revised independently by at least two reviewers and will be published only after completion of the editorial workflow.

Contents

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 3, pp. 268-279

Adding Lifetime to Objects and Membranes in P Systems

B. Aman, G. Ciobanu

Bogdan Aman, Gabriel Ciobanu
Romanian Academy, Institute of Computer Science
and A.I.Cuza University of Iaşi, Romania
E-mail: baman@iit.tuiasi.ro, gabriel@info.uaic.ro

Abstract: Membrane systems are computing devices inspired from the cell functioning. A feature of membrane systems is the fact that objects and membranes are persistent. In fact, this is not quite true in the real world: cells and intracellular proteins have a well-defined lifetime. Inspired from these biological facts, we define a model of membrane systems in which each membrane and each object has attached a lifetime. Some results show that this model is at least as powerful as the usual one.

1 Introduction to Membrane Computing

Membrane systems are essentially parallel and nondeterministic computing models inspired by the compartments of eukaryotic cells and their biochemical reactions. The structure of the cell is represented by a set of hierarchically embedded regions, each one delimited by a surrounding boundary (called membrane), and all of them contained inside an external special membrane called *skin*. The molecular species (ions, proteins, etc.) floating inside cellular compartments are represented by multisets of objects described by means of symbols or strings over a given alphabet. The objects can be modified or communicated between adjacent compartments. Chemical reactions are represented by evolution rules which operate on the objects, as well as on the compartmentalized structure (by dissolving, dividing, creating, or moving membranes).

A membrane system can perform computations in the following way: starting from an initial configuration which is defined by the multiset of objects initially placed inside the membranes, the system evolves by applying the evolution rules of each membrane in a nondeterministic and maximally parallel manner. A rule is applicable when all the objects which appear in its left hand side are available in the region where the rule is placed. The maximally parallel way of using the rules means that in each step, in each region of the system, a multiset of rules is chosen which is maximal and applicable, namely a multiset of rules such that no further rule can be added to the multiset. A halting configuration is reached when no rule is applicable. The result is represented by the number of objects from a specified membrane.

Several variants of membrane systems are inspired by different aspects of living cells (symport and antiport-based communication through membranes, catalytic objects, membrane charge, etc.). Their computing power and efficiency have been investigated using the approaches of formal languages, grammars, register machines and complexity theory. Membrane systems (also called P systems) are presented together with many variants and examples in [21]. Several applications of these systems are presented in [12]. An updated bibliography can be found at the P systems web page [22].

For an alphabet $V = \{a_1, \ldots, a_n\}$, we denote by V^* the set of all strings over V; λ denotes the empty string and $V^+ = V^* \backslash \{\lambda\}$. A multiset over V is represented by a string over V (together with all its permutations), and each string precisely identifies a multiset.

A *language (over V)* is any subset of V^*; a language is denoted usually by L. Given a language L, we define the set $Ps(L) = \{\Psi_V(x) \mid x \in L\}$ called the *Parikh image of L*. If FL is a family of languages, then $PsFL$ denotes the family of Parikh images of languages in FL.

Definition 1. A P system of degree $n \geq 1$ is a construct

$$\Pi = (V, T, C, H, \mu, w_1, \ldots, w_n, (R_1, \rho_1), \ldots, (R_n, \rho_n), i_O)$$

where:

1. V is an alphabet of symbols; its elements are called objects;
2. $T \subseteq V$ is the terminal (or output) alphabet;
3. $C \subseteq V$, $C \cap T = \emptyset$ is the alphabet of catalysts;
4. H is a set of membrane labels;
5. $\mu \subseteq H \times H$ is a tree that describes the membrane structure, such that $(i, j) \in \mu$ denotes that the membrane labelled by j is contained in the membrane labelled by i;
6. $w_i \in V^*$, for each $1 \leq i \leq n$, is a multiset of objects initially assigned to membrane i;
7. R_i, for all $1 \leq i \leq n$, is a finite set of evolution rules that is associated with membrane i; an evolution rule is a multiset rewriting rule of the form $u \rightarrow v$, with $u \in V^+$, either $v = v'$ or $v = v'\delta$, $v' \in ((V \times \{here, out\}) \cup (V \times \{in_j \mid 1 \leq j \leq n\}))^*$, and δ a special symbol not appearing in V;
8. ρ_i, for all $1 \leq i \leq n$, is a partial order relationship defined over the rules in R_i specifying a priority relation between these rules;
9. i_O is the label of an elementary membrane of μ that identifies the output region.

Therefore, a P systems of degree $n \geq 1$ consists of a membrane structure μ containing $n \geq 1$ membranes where each membrane i gets assigned a finite multiset of objects w_i and a finite set of evolution rules R_i. An evolution rule is a multiset rewriting rule which consumes a multiset of objects from V and produces a multiset of pairs (a,t), with $a \in V$ and $t \in \{here, out\} \cup \{in_j \mid 1 \leq j \leq n\}$ a *target* specifying where to move the objects after the application of the rule. As well as this, an evolution rule can produce the special object δ to specify that, after the application of the rule, the membrane containing the special object δ has to be dissolved. After dissolving a membrane, all objects and membranes previously present in it become elements of the immediately upper membrane, while the rules of the dissolved membrane are removed.

We use *P systems without lifetimes* instead of *P systems* in order to make a clear distinction from the *P systems with lifetimes* which are introduced in what follows.

2 P Systems with Lifetimes

The evolution of complicated real systems frequently involves various interdependence among components. Some mathematical models of such systems combine both discrete and continuous evolutions on multiple time scales with many orders of magnitude. For example, in nature the molecular operations of a living cell can be thought of as such a dynamical system. The molecular operations happen on time scales ranging from 10^{-15} to 10^4 seconds, and proceed in ways which are dependent on populations of molecules ranging in size from as few as approximately 10^1 to approximately as many as 10^{20}. Molecular biologists have used formalisms developed in computer science (e.g. hybrid Petri nets) to get simplified models of portions of these transcription and gene regulation processes. According to molecular cell biology [18]:

(i) "the life span of intracellular proteins varies from as short as a few minutes for mitotic cycles, which help regulate passage through mitosis, to as long as the age of an organism for proteins in the lens of the eye."

(ii) "Most cells in multicellular organisms ... carry out a specific set of functions over periods of days to months or even the lifetime of the organism (nerve cells, for example)."

It is obvious that lifetimes play an important role in the biological evolution. We use an example from the immune system.

Example 1 ([18]). T-cell precursors arriving in the thymus from the bone marrow spend up to a week differentiating there before they enter a phase of intense proliferation. In a young adult mouse the thymus contains around 10^8 to 2×10^8 thymocytes. About 5×10^7 new cells are generated each day; however, only about 10^6 to 2×10^6 (roughly $2-4\%$) of these will leave the thymus each day as mature T cells. Despite the disparity between the numbers of T cells generated daily in the thymus and the number leaving, the thymus does not continue to grow in size or cell number. This is because approximately 98% of the thymocytes which develop in the thymus also die within the thymus.

Inspired from these biological facts, we add lifetimes to objects and membranes. We use a global clock to simulate the passage of time in a membrane system.

Definition 2. A P system with lifetimes of degree $n \geq 1$ is a construct

$$\Pi = (V_t, T, C, H_t, \mu_t, w_1, \ldots, w_n, (R_1, \rho_1), \ldots, (R_n, \rho_n), i_O)$$

where:

1. $V_t = V \times (\mathbb{N} \cup \infty)$ is a set of pairs of the form (a, t_a), where $a \in V$ is an object (as in Definition 1) and $t_a \in (\mathbb{N} \cup \infty)$ is the lifetime of the object a;
2. T, C are as in Definition 1;
3. $H_t = H \times (\mathbb{N} \cup \infty)$ is a set of set of pairs of the form (h, t_h), where $a \in H$ is a membrane label (as in Definition 1) and $t_h \in (\mathbb{N} \cup \infty)$ is the lifetime of the membrane h;
4. $\mu_t \subseteq H_t \times H_t$ is a tree that describes the membrane structure, such that $((i,t_i),(j,t_j)) \in \mu_t$ denotes that the membrane labelled by j, with the lifetime j_t, is contained in the membrane labelled by i, with the lifetime i_t;
5. $w_i \subseteq (V_t)^*$ is a multiset of pairs from V_t assigned initially to membrane i;
6. R_i, for all $1 \leq i \leq n$, is a finite set of evolution rules that is associated with membrane i of the following forms:
   (a) $u \to v$, with $u \in V_t^+$, either $v = v'$ or $v = v'\delta$, $v' \in ((V_t \times \{here, out\}) \cup (V_t \times \{in_j \mid 1 \leq j \leq n\}))^*$; δ is a special symbol not appearing in V;
   (b) $(a,t) \to (a, t-1)$, for all $a \in V$ and $t > 0$
   If an object a is not involved in a rule of type (a) and it has a lifetime $t > 0$, then its lifetime is decreased.
   (c) $(a, 0) \to \lambda$, for all $a \in V$
   If an object a has the lifetime 0 then the object is replaced with the empty multiset λ, thus simulating the degradation of proteins.
   (d) $[\,]_{(i,t)} \to [\,]_{(i,t-1)}$, for all $1 \leq i \leq n$
   In each evolution step the lifetime of each membrane of the membrane structure is decreased with one.
   (e) $[\,]_{(i,0)} \to [\delta]_{(i,0)}$, for all $1 \leq i \leq n$
   If the lifetime of a membrane reaches 0 the membrane is dissolved.
7. ρ_i, for all $1 \leq i \leq n$, is a partial order relationship defined over the rules in R_i specifying a priority relation between these rules;

8. i_O is the label of an elementary membrane of μ_t that identifies the output region.

These rules are applied according to the following principles:

1. All the rules are applied in parallel: in a step, the rules are applied to all objects and to all membranes; an object can be used only by one rule, non-deterministically chosen (there is no priority among rules), but any object which can evolve by a rule of any form, should evolve.

2. If a membrane is dissolved, then all the objects in its region are left free in the region immediately above it. Because all rules are associated with membranes, the rules of a dissolved membrane are no longer available at the next step.

3. The skin membrane has the lifetime equal to ∞, so it can never be dissolved.

4. If a membrane or object has the lifetime equal to ∞, when applying the rules simulating the passage of time we use the equality $\infty - 1 = \infty$.

The computation stops when the membrane system contains only objects and membranes that have the lifetime equal to ∞.

Example 2. The concentration of a molecule can be adjusted quickly only if the lifetime of the molecule is short [1]. It is natural to think of signaling systems in terms of the changes produced when a signal is delivered. But it is just as important to consider what happens when a signal is withdrawn. During development transient signals often produce lasting effects: they can trigger a change that persists indefinitely, through cell memory mechanisms. But in most cases, especially in adult tissues, when a signal ceases, the response fades. The signal acts on a system of molecules that is undergoing continual turnover, and when the signal is shut off, the replacement of the old molecules by new ones wipes out the traces of its action. It follows that the speed of reaction to shutting off the signal depends on the rate of turnover of the molecules that the signal affects. It may not be as obvious that this turnover rate also determines the promptness of the response when the signal is turned on.

The Figure 1 shows the predicted relative rates of change in the intracellular concentrations of molecules with differing turnover times when their rates of synthesis are increased suddenly by a factor of 10. The concentrations of those molecules that are normally being rapidly degraded in the cell (red lines) change quickly, whereas the concentrations of those that are normally being slowly degraded (green lines) change proportionally more slowly. The numbers (in blue) on the right-hand side are the half-lives assumed for each of the different molecules.

Consider, for example, two intracellular molecules X and Y, both of which are normally maintained at a concentration of 1000 molecules per cell. Molecule X has a slow turnover rate: it is synthesized and degraded at a rate of 10 molecules per second, so that each molecule has an average lifetime in the cell of 100 seconds. Molecule Y turns over 10 times as quickly: it is synthesized and degraded at a rate of 100 molecules per second, with each molecule having an average lifetime of 10 seconds. If a signal acting on the cell boosts the rates of synthesis of both X and Y tenfold without any change in the molecular lifetimes, at the end of 1 second the concentration of Y will have increased by nearly 900 molecules per cell $(10 \times 100 - 100)$ while the concentration of X will have increased by only 90 molecules per cell. In fact, after its synthesis rate has been either increased or decreased abruptly, the time required for a molecule to shift halfway from its old to its new equilibrium concentration is equal to its normal half-life - that is, it is equal to the time that would be required for its concentration to fall by half if all synthesis were stopped (Figure 1).

The same principles apply to proteins as well as to small molecules and to molecules in the extracellular space as well as to those in cells. Many intracellular proteins that are rapidly degraded have short half-lives, some surviving less than 10 minutes; in most cases these are proteins with key regulatory roles,

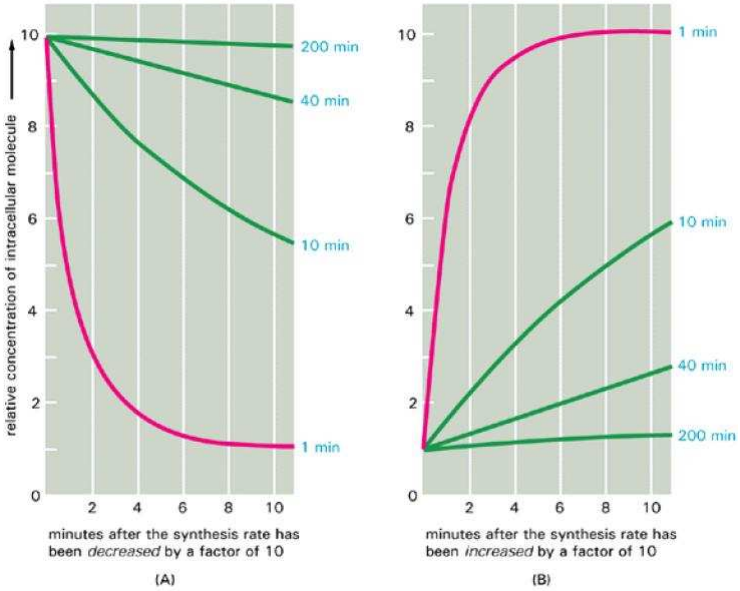


Figure 1: The importance of rapid turnover [1]

whose concentrations are rapidly regulated in the cell by changes in their rates of synthesis. Likewise, any covalent modifications of proteins that occur as part of a rapid signaling process - most commonly the addition of a phosphate group to an amino acid side chain - must be continuously removed at a rapid rate to make such signaling possible.

Example 3. The scenario presented in Example 2 can be modelled using P systems with lifetimes. Consider the membrane configuration

$$[(X_s,\infty)(X,0)^{10}\dots(X,99)^{10}(Y_s,\infty)(Y,0)^{100}\dots(Y,9)^{100}]_{(cell,\infty)}$$

describing the structure of the cell when in equilibrium. X_s and Y_s represent the molecules from which X and Y are synthesized, while $(X,t_1)^{10}$ and $(Y,t_2)^{100}$ represent the fact that there are 10 molecules X which have the lifetime equal to t_1 and 100 molecules Y which have the lifetime equal to t_2. The rules describing the evolution of this system are:

1. $(X,0)\to\lambda$

   A molecule X which is at the end of its lifetime is degraded.

2. $(Y,0)\to\lambda$

   A molecule Y which is at the end of its lifetime is degraded.

3. $(X_s,\infty)\to(X_s,\infty)(X,99)^{10}$

   Each second 10 new X molecules are synthesized.

4. $(Y_s,\infty)\to(Y_s,\infty)(Y,9)^{100}$

   Each second 100 new Y molecules are synthesized.

5. $(X,t)\to(X,t-1), t>0$

   The lifetime of a molecule X which is not involved in any reaction is decreased.

6. $(Y,t)\to(Y,t-1), t>0$

   The lifetime of a molecule Y which is not involved in any reaction is decreased.

After applying rules in a maximal manner after each second we reach the initial configuration $[(X_s,\infty)(X,0)^{10}\dots(X,99)^{10}(Y_s,\infty)(Y,0)^{100}\dots(Y,9)^{100}]_{(cell,\infty)}$.

If two signals enter the cell (we model the signals by the pairs (c_X,t_c) and (c_Y,t_c)) then we consider two new rules:

7. $(c_X,t_c)(X_s,\infty)\rightarrow(c_X,t_c-1)(X_s,\infty)(X,99)^{100}$

   If an object c_X is present in the cell then the rate of synthesis of X is increases 10 times.

8. $(c_Y,t_c)(Y_s,\infty)\rightarrow(c_Y,t_c-1)(Y_s,\infty)(Y,9)^{1000}$

   If an object c_Y is present in the cell then the rate of synthesis of Y is increases 10 times.

When adding this rules we also add some priorities between rules, namely $7>5$ and $8>6$. After one second the concentration of Y increases by nearly 900 molecules per cell $(10\times100-100)$ while the concentration of X increases by only 90 molecules per cell.

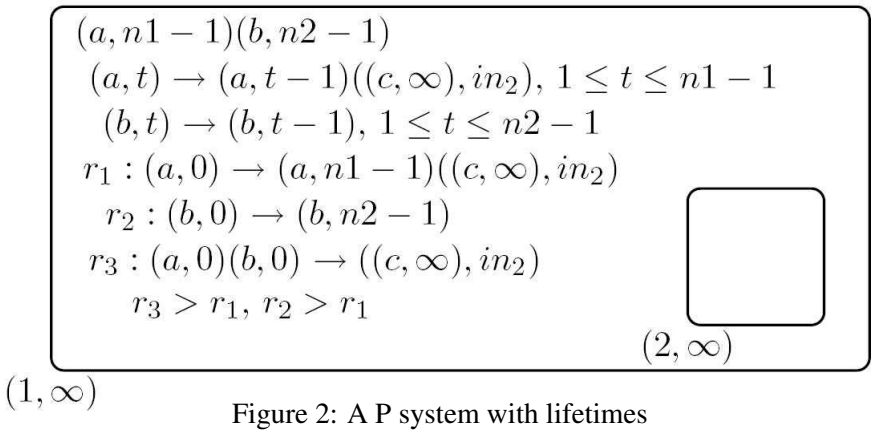


Figure 2: A P system with lifetimes

In Figure 2 an example of a P system with lifetimes is shown. Graphically, the boxes represent the membranes and their nesting reflects the hierarchy. Membrane 1 represent the skin membrane. Both membranes have a lifetime equal to ∞ meaning that no dissolving rule is not necessary. Inside membrane 1 we have the initial multiset of pairs, the evolution rules and some priorities between them. In the evolution rule we omit the subscript *here* for objects, in the products, that remain in the same membrane.

The defined P system with lifetimes computes the least common multiple of n_1 and n_2, namely $lcm(n_1,n_2)$. The idea is to put a pair $(a,n1-1)$ and a pair $(b,n2-1)$ at the beginning of the computation, and to produce a pair (c,∞) (which is send in membrane 2) each time the lifetime of the object a is decreased. The first time the objects a and b appear together with the lifetime 0 is exactly after $lcm(n_1,n_2)$ time units. At this moment in membrane 2 are $lcm(n_1,n_2)$ objects c, which represent the output of the system.

3 Systems with and without Lifetimes

The following results describe certain relationships between P systems with lifetimes and P systems without lifetimes, and between similar P systems with lifetimes.

Proposition 4. For every P system without lifetimes there exists a P system with lifetimes providing the same output by performing an equal number of steps.

Proof: [Proof (Sketch)] It is easy to state that the class of P systems with lifetimes includes the class of P systems without lifetimes, since we can assign ∞ to all lifetimes appearing in the membrane structure and evolution rules. □

A somehow surprising result is that P systems with lifetimes can be simulated by P systems without lifetimes.

Proposition 5. For every P system with lifetimes there exists a P system without lifetimes providing the same output by performing an equal number of steps.

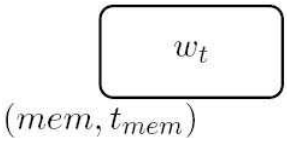
Proof: We use the notation $rhs(r)$ to denote the multisets of pairs which appear in the right hand side of a rule r. This notation is extended naturally to multisets of rules: given a multiset of rules R, the right hand side of the multiset $rhs(R)$ is obtained by adding the right hand sides of the rules in the multiset, considered with their multiplicities.

Each object $a \in V$ from a P system with lifetimes has a maximum lifetime (we denote it by $lifetime(a)$), which can be calculated as follows:

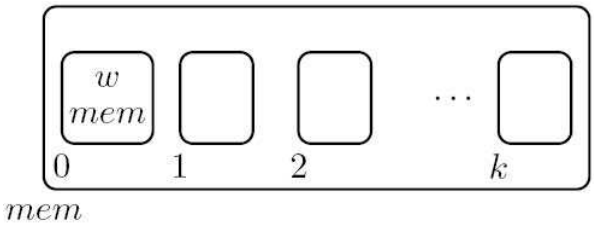
$lifetime(a) = max(\{t \mid (a,t) \in w_i, 1 \leq i \leq n\} \cup \{t \mid (a,t) \in rhs(R_i), 1 \leq i \leq n\})$

In what follows we present the steps which are required to build a P system without lifetimes starting from a P system with lifetimes, such that both provide the same output.

1. A membrane structure from a P system with lifetimes



is translated into a membrane structure of a P system without lifetimes in the following way



The lifetimes of elements from a P system with lifetimes are simulated using the membranes 0, ..., k in the corresponding P system without lifetimes as we show at the next steps of the translation. The value of k is the maximum from the finite lifetime of objects and surrounding membrane *mem*, namely $k = max(\{lifetime(a) \mid a \in V, lifetime(a) \neq \infty\} \cup \{t_{mem} \mid t_{mem} \neq \infty\})$. If an object or the surrounding membrane has the lifetime equal to ∞ in the P system with lifetimes then we do not need to count the passage of time, namely to use the membranes 0, ..., k in the P system without lifetimes for the corresponding object or membrane. The object *mem* placed inside the membrane labelled 0 is used to simulate the passage of time for the membrane. The initial multiset of objects w_t from membrane *mem* in the P system with lifetimes, in translated into the multiset w which is added into membrane 0 inside membrane *mem* in the corresponding P system without lifetimes since all objects from the initial multiset are just starting their life.

2. The rules $(a,t) \rightarrow (a,t-1)$, for all $a \in V$, $t > 0$ and $t \neq \infty$ from the P system with lifetimes can be simulated in the P system without lifetimes using the following rules:
   (a) $a \rightarrow (ao_{ai}, out)$ placed inside membrane i, for all $0 \leq i \leq lifetime(a) - 1$
   The object o_{ai} is used to keep track of the units of time that have past from the lifetime of the object a

(b) $ao_{aj} \to (a, in_{j+1})$, placed inside membrane *mem*, for all $0 \leq j \leq lifetime(a) - 1$ and $a \in V$

This rules together with the previous ones simulate the passage of a unit of time from the lifetime of object a in the P system with lifetimes, by moving object a from a membrane j to a membrane $j+1$ for $0 \leq j \leq lifetime(a) - 1$ in the P system without lifetimes.

3. The rules $(a, 0) \to \lambda$, for all $a \in V$ from the P system with lifetimes can be simulated in the P system without lifetimes using the following rules:

(a) $a \to \lambda$ placed inside membrane $lifetime(a)$

If the object a reaches the membrane labelled with $lifetime(a)$ in the P system without lifetimes, it means that the lifetime of object a is 0 in the corresponding P system with lifetimes, so it is replaced by λ.

4. The rules $u_t \to v_t$ from the P system with lifetimes can be simulated in the P system without lifetimes using the following rules:

(a) $uo_{uj} \to (v, in_0)$

The multiset $o_u j$ contains objects of the form o_{aj}, where $a \in u$ and $0 \leq j \leq lifetime(a) - 1$. When replacing the multiset u with the multiset v, we also remove the objects o_{aj} that keep track of the life of the objects appearing in u since we do not need them anymore. We send the newly obtained multiset v in membrane 0 since all objects from this multiset are just starting their life.

5. The rules $[\,]_{(mem,t)} \to [\,]_{(mem,t-1)}$, for all $t > 0$ and $t \neq \infty$ from the P system with lifetimes can be simulated in the P system without lifetimes using the following rules:

(a) $mem \to (mem\, o_{mem\, i}, out)$ placed inside membrane i, for all $0 \leq i \leq lifetime(mem) - 1$

The object $o_{mem\, i}$ is used to keep track of the units of time that have past from the lifetime of the membrane *mem*;

(b) $mem\, o_{mem\, j} \to (mem, in_{j+1})$, placed inside membrane *mem*, for all $0 \leq j \leq lifetime(mem) - 1$

This rules together with the previous ones simulate the passage of a unit of time from the lifetime of membrane *mem* in the P system with lifetimes, by moving object *mem* from a membrane j to a membrane $j+1$ for $0 \leq j \leq lifetime(mem) - 1$ in the P system without lifetimes.

6. A rule $[\,]_{(mem,0)} \to [\delta]_{(mem,0)}$ from the P system with lifetimes can be simulated in the P system without lifetimes using the following rules:

(a) $mem \to (o_\delta, out)$ placed inside membrane $lifetime(mem) - 1$

If the object *mem* reaches the membrane labelled with $lifetime(mem) - 1$ in the P system without lifetimes, it means that the lifetime of membrane *mem* is 0 in the corresponding P system with lifetimes, so it needs to be dissolved.

(b) $o_\delta \to \delta(\delta, in_1) \ldots (\delta, in_k)$

Once this object is created by the previous rule, an object δ is created inside membrane *mem* and inside each membrane j, $0 \leq j \leq k$, is send an object δ. This means that all these membranes are dissolved, all the rules are deleted, and all objects are send in the parent membrane. The dissolving of the membranes take place after applying all other possible rules. At the moment of the dissolution the only existing objects are found in membrane *mem*. For each object a there exists an object o_{aj} that keeps track of the life of a, thus being able to continue the increment the life of a in the parent membrane.

The output membrane from the P system with lifetimes, is translated in the output membrane from the P system without lifetimes. After performing the same number of evolution steps in both systems, the output membranes contain the same multisets of objects. □

We are now able to prove the computational power of P systems with lifetimes. We denote by $\mathbb{N}tP_m(coo,tar)$ the family of sets of natural numbers generated by P systems with lifetimes of degree at most $m \geq 1$, using cooperative rules, and communication of objects through membranes. We also denote by $\mathbb{N}RE$ the family of all sets of natural numbers generated by arbitrary grammars.

Proposition 6. $\mathbb{N}tP_m(coo,tar) = \mathbb{N}RE$, for all $m \geq 1$.

Proof: [Proof (Sketch)] Since the outcome of each P system with lifetimes can be obtained by an P system without lifetimes, we cannot get more than the computability power of P systems. Therefore, according to Theorem 3.3.3 from [21], we have that the family $\mathbb{N}tP_m$of sets of natural numbers generated by P systems with lifetimes is the same as the family $\mathbb{N}RE$ of sets of natural number generated by arbitrary grammars. □

Remark 3.1. Consider the membrane system $\Pi = [[\,]_{(2,\infty)}(a,1)(a,1)]_{(1,\infty)}$ with the set of rules $R_1 = \{(a,1) \to (a,1)((c,\infty),in_2)^3; (a,1) \to (a,0); (a,0) \to \lambda\}$. Since the membranes have the lifetime ∞ it is not necessary to consider rules for decreasing lifetimes to membranes.

If we rewrite this as $\Pi' = [[\,]_{(2,\infty)}(a,1)(a,1)(d,t)]_{(1,\infty)}$, with $R_1 = \{(a,1) \to (a,1)((c,\infty),in_2)^3; (a,1) \to (a,0); (a,0) \to \lambda; (d,i) \to (d,i-1); (d,0) \to \lambda\}$ we have that after t units of time the membrane system Π has the same evolution as the membrane system Π'.

In automata theory the problem of optimizing a finite-state machine, meaning to find the machine with the minimum number of states that performs the same function, was addressed by the theorem of Myhill and Nerode; a fast algorithm doing this is the Hopcroft minimization algorithm [16].

Using a similar approach we want to optimize a given P system with lifetimes. This can be realized with the passage of time, namely all objects and membranes which are not used in rewriting rules and have a finite lifetime are eliminated.

Proposition 7. Let Π and Π' be two P systems with lifetimes such that:

1. $\Pi = (V_t, T, C, H_t, \mu_t, w_1, \ldots, w_n, (R_1, \rho_1), \ldots, (R_n, \rho_n), i_O)$
2. $\Pi' = (V_t, T, C, H_t, \mu_t, w_1', \ldots, w_n', (R_1, \rho_1), (R_1', \rho_1'), \ldots, (R_n', \rho_n'), i_O')$
3. $i_O' = i_O$

   The output membrane is the same for the two membrane systems.
4. $w_i \subseteq w_i$, for all $1 \leq i \leq n$

   The initial multiset from membrane i of Π' contains the same objects as the initial multiset from membrane i of Π and some other objects together with their initial lifetimes.
5. $R_i' = R_i \cup \{(a,t) \to (a,t-1), (a,0) \to \lambda \mid a \in w_i' \backslash w_i\}$

   The set of rules R_i' contains all the rules of R_i and some additional rules to simulate the passage of time for all the objects appearing in $w_i' \backslash w_i$.
6. $\rho_i' = \rho_i$, for all $1 \leq i \leq n$

   The priority orders are the same for the two membrane systems.

Then the P systems with lifetimes Π and Π' have the same membrane structure and evolution after $max\{t \mid (a,t) \in w_i' \backslash w_i, 1 \leq i \leq n\}$ units of time.

Proof: [Proof (Sketch)] After $max\{t \mid (a,t) \in w_i' \backslash w_i, 1 \leq i \leq n\}$ units of time all objects which appeared in $w_i \backslash w_i'$ in the description of the membrane system Π' are consumed. In this case we have that the contents of the membranes of Π is the same as the contents of the membranes of Π', and the applicable rules for the two membrane systems are only the rules of R_i, $1 \leq i \leq n$. □

4 Related Work and Conclusion

We introduce a new class of P Systems, namely the P Systems with lifetimes. Lifetimes are assigned to each membrane and to each object. This new feature is inspired from biology where cells and intracellular proteins have a well defined lifetime. In order to simulate the passage of time, we use rules of the form $(a,t) \rightarrow (a,t-1)$ for objects, and $[\,]_{(i,t)} \rightarrow [\,]_{(i,t-1)}$ for membranes. If the lifetime of an object reaches 0 then the object is consumed by applying a rule of the form $(a,0) \rightarrow \lambda$, while if the lifetime of a membrane i reaches 0 then the membrane is marked for dissolution by applying a rule of the form $[\,]_{(i,0)} \rightarrow [\delta]_{(i,0)}$. After dissolving a membrane, all objects and membranes previously contained in it become elements of the immediately upper membrane.

We do not obtain a more powerful formalism by adding lifetimes to objects and to membranes into a P system. According to Proposition 4, Proposition 5 and Proposition 6, P systems with lifetimes and P systems without lifetimes have the same computational power. However the P systems with lifetimes are able to describe more naturally some biological phenomena involving timing, as in Example 3.

A similar idea appears in the framework of spiking P systems: considering a duration of life for spikes, but not for cells [15]. If a spike is not used in a number of steps larger than its lifetime, then it is removed.

There are also some papers using time in the context of membrane computing in a different manner than in this paper. In [8] a timed P system is introduced by associating to each rule a natural number representing the time of its execution. Then a P system which always produces the same result, independently from the execution times of the rules, is called a time-independent P systems. The notion of time-independent P systems tries to capture the class of systems which are robust against the environment influences over the execution time of the rules of the system. Other types of time-free systems are considered in [7, 9].

Time can also be used to "control" the computation, for instance by appropriate changes in the execution times of the rules during a computation, and this possibility has been considered in [11]. Moreover, timed P automata have been proposed and investigated in [5], where ideas from timed automata have been incorporated into timed P systems.

Frequency P systems has been introduced and investigated in [19]. In frequency P systems each membrane is clocked independently from the others, and each membrane operates at a certain frequency which could change during the execution. Dynamics of such systems have been investigated.

If one supposes the existence of two scales of time (an external time of the user, and an internal time of the device), then it is possible to implement accelerated computing devices which can have more computational power than Turing machines. This approach has been used to construct accelerated P systems where acceleration is obtained by either decreasing the size of the reactors or by speeding-up the communication channels [6].

In [10, 17] the time of occurrence of certain events is used to compute numbers. If specific events (such as the use of certain rules, the entering/exit of certain objects into/from the system) can be freely chosen, then it is easy to obtain computational completeness results. However, if the length (number of steps) are considered as result of the computation, non-universal systems can be obtained. Time is considered in [17, 20] as the result of the computation by using special "observable" configurations taken in regular sets (with the time elapsed between such configurations considered as output).

The authors of the current paper have also considered time to "control" the computation in two other formalisms: mobile ambients [2–4] and distributed π-calculus [13, 14]. Timers define timeouts for

various resources, making them available only for a determined period of time. The passage of time is given by a discrete global time progress function.

Acknowledgments

This work was partially supported by CNCSIS research grants IDEI 402/2007 and CNMP PARTENERIATE D1/1052/2007.

Bibliography

[1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter. *Molecular Biology of the Cell - Fifth Edition*. Garland Science, Taylor & Francis Group, 2008.

[2] B. Aman, G.Ciobanu. Timers and Proximities for Mobile Ambients. *Lecture Notes in Computer Science*, vol.4649, 33–43, 2007.

[3] B. Aman, G.Ciobanu. Mobile Ambients with Timers and Types. *Lecture Notes in Computer Science*, vol.4711, 50–63, 2007.

[4] B. Aman, G.Ciobanu. Timed Mobile Ambients for Network Protocols. *Lecture Notes in Computer Science*, vol.5048, 234–250, 2008.

[5] R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, L. Tesei. Timed P Automata. *Electronic Notes in Theoretical Computer Science*, vol.227, 21–36, 2009.

[6] C.S. Calude, Gh. Păun. Bio-Steps Beyond Turing. *Biosystems*, vol.77, 175–194, 2004.

[7] M. Cavaliere, V. Deufemia. Further Results on Time-Free P Systems. *International Journal of Foundations of Computer Science*, vol.17, 69–89, 2006.

[8] M. Cavaliere, D. Sburlan. Time-Independent P Systems. *Lecture Notes in Computer Science*, vol.3365, 239–258, 2005.

[9] M. Cavaliere, D. Sburlan. Time and Synchronization in Membrane Systems. *Fundamenta Informaticae*, vol.64, 65–77, 2005.

[10] M. Cavaliere, R. Freund, A.Leitsch, Gh. Păun. Event-Related Outputs of Computations in P Systems. *Journal of Automata, Languages and Combinatorics*, vol.11, 263–278, 2006.

[11] M. Cavaliere, C. Zandron. Time-Driven Computations in P Systems. Proceedings of Fourth Brainstorming Week on Membrane Computing, 133–143, 2006.

[12] G. Ciobanu, Gh. Păun, M.J. Pérez-Jiménez (Eds.). *Applications of Membrane Computing*, Springer, Natural Computing Series, 2006.

[13] G. Ciobanu, C. Prisacariu. Timers for Distributed Systems. *Electronic Notes in Theoretical Computer Science*, vol.164(3), 81–99, 2006.

[14] G. Ciobanu, C. Prisacariu. Coordination by Timers for Channel-Based Anonymous Communications. *Electronic Notes in Theoretical Computer Science*, vol.175(2), 3–17, 2007.

[15] R. Freund, M. Ionescu, M. Oswald. Extended spiking neural P systems with decaying spikes and/or total spiking. *International Journal of Foundations of Computer Science*, vol.19, 1223–1234, 2008.

[16] J. E. Hopcroft. An nlogn Algorithm for Minimizing the States in a Finite Automaton. *The Theory of Machines and Computations*, Academic Press, 189–196, 1971.

[17] O.H. Ibarra, A. Păun. Computing Time in Computing with Cells. *Lecture Notes in Computer Science*, vol.3892, 112–128, 2006.

[18] H. Lodish, A. Berk, P. Matsudaira, C. Kaiser, M. Krieger, M. Scott, L. Zipursky, J. Darnell. *Molecular Cell Biology - Sixth Edition*. Freeman, 2008.

[19] D. Molteni, C. Ferretti, G. Mauri. Frequency Membrane Systems. *Computing and Informatics*, vol.27(3), 467–479, 2008.

[20] H. Nagda, A. Păun, A. Rodríguez-Patón. P Systems with Symport/Antiport and Time. *Lecture Notes in Computer Science*, vol.4361, 463–476, 2006.

[21] Gh. Păun. *Membrane Computing. An Introduction*. Springer, 2002.

[22] Web page of the P systems: http://ppage.psystems.eu.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 3, pp. 280-291

Extreme Data Mining: Inference from Small Datasets

R. Andonie

Răzvan Andonie
Computer Science Department
Central Washington University, Ellensburg, USA
and
Department of Electronics and Computers
Transylvania University of Braşov, Romania
E-mail: andonie@cwu.edu

Abstract: Neural networks have been applied successfully in many fields. However, satisfactory results can only be found under large sample conditions. When it comes to small training sets, the performance may not be so good, or the learning task can even not be accomplished. This deficiency limits the applications of neural network severely. The main reason why small datasets cannot provide enough information is that there exist gaps between samples, even the domain of samples cannot be ensured. Several computational intelligence techniques have been proposed to overcome the limits of learning from small datasets.
We have the following goals: **i.** To discuss the meaning of "small" in the context of inferring from small datasets. **ii.** To overview computational intelligence solutions for this problem. **iii.** To illustrate the introduced concepts with a real-life application.

1 Introduction

Small dataset conditions exist in many applications, such as disease diagnosis, fault diagnosis or deficiency detection in biology and biotechnology, mechanics, flexible manufacturing system scheduling, drug design, and short-term load forecasting (an activity conducted on a daily basis by electrical utilities). In this section, we describe a computational chemistry problem, review a class of neural networks to be used, and summarize our previous work in this area.

1.1 A Real-World Problem: Assist Drug Discovery

Current treatments for HIV/AIDS consist of co-administering a protease inhibitor and two reverse transcriptase inhibitors (usually referred to as combination therapy). This therapy is effective in reducing viremia to very low levels; however, in 30-50% of patients it is ineffective due to resistance development often caused by viral mutations. Due to resistance and poor bioavailability [1] profiles, as well as toxicity associated with these therapies, there is an urgent need for more efficient design of drugs.

We focus on inhibitors to the HIV-1 protease enzyme, using the IC_{50} as the target value. A detailed description of the problem, from a computational chemistry point of view, can be found in our papers [1–3]. The IC_{50} value represents the concentration of a compound that is required to reduce enzyme activity by 50%. A low IC_{50} value indicates good inhibitory activity. The available dataset consists of 196 compounds with experimentally determined IC_{50} values. Twenty of these molecules are used as an external test set after the training is completed. The remaining 176 molecules are used for training and cross-validation. Our practical goal is to predict the (unknown) IC_{50} values for 26 novel compounds which are candidates for HIV-1 protease inhibitors. We use two IC_{50} prediction accuracy measures: the RMSE (Root Mean Squared Error) and the Symmetric Mean Absolute Percentage Error (sMAPE).

[1] Bioavailability is the rate at which the drug reaches the systemic circulation.

The easiest way to represent a molecule is by a vector of features (molecular descriptors) which may be both topological indices and physico-chemical properties. The resulting features may be numerous and inter-correlated. Using the complete set of descriptors may lead to overfitting, if it is too large compared to the size of the training set. We select 35 molecular descriptors based on their contribution to molecular entity.

Although biological activity data has been obtained for many more chemical structures at various pharmaceutical companies and academic laboratories, they are not available in the public domain. Actually, most classical studies for a specific enzyme system have been performed on small datasets, due to limited experimentally determined biological activity values in the public domain. The dimensionality (the number of physico-chemical features) characterizing these molecules is relatively high. Our dataset shares these undesired characteristics: it is small, with relatively many features, and highly overlapping.

1.2 Prerequisites: FAMR for IC_{50} prediction

The FAMR is a Fuzzy ARTMAP (FAM) incremental learning system used for classification, probability estimation, and function approximation. We review the basic FAMR notation. Details can be found in [4].

A FAM consists of a pair of fuzzy ART modules, ART_a and ART_b, connected by an inter-ART module called Mapfield. The fuzzy ART_a module contains the input layer, F_1^a, and the competitive layer, F_2^a [5]. A preprocessing layer, F_0^a, is also added before F_1^a. The ART modules create stable recognition categories in response to arbitrary sequences of input patterns. The ART_a and ART_b vigilance parameters, ρ_a and ρ_b, control the matching mechanism inside the modules.

During learning, the Mapfield weights are updated: the strength of the weight projecting from the selected ART_a category to the correct ART_b category is increased, while the strengths of the weights to other ART_b categories are decreased. A Mapfield vigilance parameter ρ_{ab} calibrates the degree of predictive mismatch necessary to trigger the search for a different ART_a category. If the weight projecting from the active ART_a category through the Mapfield to the active ART_b category is smaller than ρ_{ab} (vigilance test), then the system responds to the unexpected outcome through the so-called *match tracking*. This triggers an ART_a search for a new input category. After choosing an ART_a category whose prediction of the correct ART_b category is strong enough, match tracking is disengaged, and the network is said to be in a resonance state. In this case, Mapfield learns by updating the weights w_{jk}^{ab} of associations between each j-th ART_a category and each k-th ART_b category.

The FAMR uses the following iterative updating scheme:

$$w_{jk}^{ab(new)} = \begin{cases} w_{jk}^{ab(old)} & \text{if } j \neq J \\ w_{JK}^{ab(old)} + \frac{q_t}{Q_J^{new}} \left(1 - w_{JK}^{ab(old)}\right) & \\ w_{Jk}^{ab(old)} \left(1 - \frac{q_t}{Q_J^{new}}\right) & \text{if } k \neq K \end{cases} \tag{1}$$

where q_t is the relevance assigned to the tth input pattern ($t = 1, 2, \ldots$) and $Q_J^{new} = Q_J^{old} + q_t$. The *relevance* q_t is a real positive finite number directly proportional to the importance of the experiment considered at step t. This w_{jk}^{ab} approximation is a correct biased estimator of the posterior probability $P(k|j)$, the probability of selecting the k-th ART_b category after having selected the j-th ART_a.

FAM (and FAMR) networks map subsets of $\mathbb{R}^n$ to $\mathbb{R}^m$ and can be used for function approximation. The FAM has been proven to be a universal function approximator [6]. We use the FAMR to predict functions that are known only at a certain number of points. More specifically, we predict IC_{50} values.

1.3 Our previous work

The present paper is based on a sequence of results, each describing new computational intelligence tools for biological activity (IC_{50}) prediction. In [7], we investigated the use of a fuzzy neural network

(FNN) for (IC_{50}) prediction. In [1] and [2], we improved this model by adding a two-stage Genetic Algorithm (GA) optimizer: the first for selecting the best subset of features and the the second for optimizing the FNN parameters. We will refer to this GA-optimized FNN as FS-GA-FNN.

In [8] we also focused on the IC_{50} prediction task, using the FAMR model. During the learning phase, each sample pair is assigned a relevance factor proportional to the importance of that pair. The prediction method consists of two stages. First, GA-optimization incorporating cross-validation is used to modify the training dataset. This modification consists of finding the best relevances for the data, according to some fitness criterion. The fitness criterion measures the FAMR IC_{50} prediction accuracy for a given training/validation dataset with given relevances. In stage two, the final FAMR is obtained by training it using the dataset with optimized relevances. In other words, stage one improves the generalization capability of the FAMR which will be obtained in stage two. We will refer to this model with GA-optimized relevances as GA-FAMR.

We compared the GA-FAMR and the Ordered FAMR (a FAMR algorithm which optimizes the order of training data presentation) in [9]. Both methods compensate for insufficient training data by additional optimizations. A trade-off between computational overhead and generalization capability is obtained.

Recently, we performed rule extraction from the trained FAMR model [10]. We post-processed the set of generated rules in order to improve generalization. We eliminated overfitting by heuristic generalization of rules and by adding new rules. This method proved to be efficient for small training sets.

The present paper results from several invited talks [9,11,12]. In Section 2, we discuss the capability of neural network to infer from rare samples. Section 3 describes two methods for neural training on small datasets. After presenting and discussing experimental results in Section 4, we conclude with our final remarks (Section 5).

2 Neural Networks Trained on Small Datasets

We aim to discuss the difficulties of inferring a Neural Network (NN) from small, or non-representative, training sets. We will look closer at the overfitting and generalization aspects of the network. But first, we need to define formally what we understand by "small training set".

2.1 What is "small"?

In many multivariable classification or regression (e.g., estimation or forecasting) problems we have a training set $T_p = (x_i, t_i)$ of p pairs of input/output vector $\mathbf{x} \in \Re^{\mathbf{n}}$ and scalar target t, and the unfortunate circumstance that T_p is small. The VC (Vapnik-Chervonenkis) dimension is a measure of the capacity of a classificator, defined as the cardinality of the largest set of points that the algorithm can shatter. According to Vapnik:

> "For estimating functions with VC dimension h, we consider the size p of data to be small if the ratio p/h is small (say $p/h < 20$)" [13].

The main reason why small datasets cannot provide enough information is that there exist gaps between samples, even the domain of samples cannot be ensured. For a small training set, even a simple neural network can have a complexity (e.g., number of connections/parameters) that is comparable to, or exceeds, the training size p. In such a case, we may expect to fit T_p very well. However, we can also expect poor generalization to new data identically distributed as the data in T_p. In effect, the VC dimension is too large relative to the size of the training set.

A completely different definition for “small” sets comes from algorithmic information theory. The Kolmogorov complexity of an object such as a string is a measure of the computational resources needed

to specify the object. More formally, the complexity of a string is the length of the string's shortest description in some fixed universal description language. It can be shown that the Kolmogorov complexity of any string cannot be too much larger than the length of the string itself. A string is considered to be "random" if the length of the shortest problem that generates the string is the same as that of the string itself. Strings whose Kolmogorov complexity is small relative to the string's size are considered to have small information content [14]. Kolmogorov's complexity has been studied in the context of inductive inference [15, 16]. It is an open problem how to relate the Kolmogorov complexity of a training set and the generalization capability of the inferred NN.

We will use a simplified definition: A training set is small if p and n are comparable. In accordance with this definition, the training set for our chemistry problem is small.

There is no universally optimal solution to the problem of inferring from small datasets. We only can state some very general principles one can follow. For instance, one principle would be to extract from the training data the maximum useful information available. If not done thoroughly, this may lead to overfitting, and/or to a time-prohibitive training process. A principle for controlling the generalization capability of a NN is to design a network with much fewer connections than the size of the training set.

To overcome the limits of learning from small datasets, several general techniques have been proposed [17–26]: generate artificial training samples, feature selection, and parameter fine-tuning of the inferred model.

A special learning method designed for small training sets is *adaptive learning with domain range expansion*. In this case, additional information is used to dynamically improve training. Such an approach is, for instance, the Central Location Tracking method [25, 26]. This algorithm attempts to explore the predictive information through the generation of trend value of each datum. The extra information extracted from the data trend stabilizes the learning task and improves the derived knowledge from the occurrence of the latest data. The domain range is expanded to obtain the probable change of the small training data behavior.

The choice of specific technique is domain dependent. In computational chemistry, only feature selection and parameter fine-tuning have been used [27–29]. It is very difficult to generate artificial samples because, most probably, they will not physically exist.

2.2 Overfitting vs. generalization

Inference is based on a strong assumption: using a *representative* training set of samples to infer a model. In this case, we select a subset of the population, perform a statistical analysis on this sample, and use these results as an approximation to the desired statistical characteristics of the population as a whole. The more representative the sample, the larger our confidence that the statistical results obtained by using this sample are indeed a good approximation to the desired population statistics. We gauge the representativeness of a sample by how well its statistical characteristics reflect the statistical characteristics of the entire population. Many standard techniques may be used to select a representative sample set [30]. However, if we do not use expert knowledge, selecting the most representative training set from a given dataset was proved to be computationally difficult (NP-hard) [31]. The problem is actually more difficult, since in most applications the complete dataset is unknown or too large to be analyzed. Therefore, we have to rely on a more or less representative training set.

Another problem may arise from the training process itself. Especially in cases where learning was performed too long or where training the training samples are rare, the inferred model may adjust to very specific random features of the training data, that have no causal relation to the target function. In this process of *overfitting*, the performance on the training examples still increases while the performance on unseen data becomes worse (the generalization performance is poor).

In NN learning, overfitting generally occurs when excessive number of neurons is generated; the network overestimates the complexity of the problem and it cost more resources to train and implement.

There are three major strategies to avoid overfitting:

1. **Before learning**. Before being used, training samples are pre-processed, or new training samples are artificially created. A widely used before learning technique is to artificially extend the training set by introducing new training samples with additive noise [32–34]. It helps to enhance the generalization performance, speed up the training algorithm, and reduce the possibility of local minima entrapment [33–36].

2. **After learning**. The network is trained (with possible overfitting) and processed afterwards. Such techniques include *pruning*, *weight sharing*, *weight decay*, *ensemble neural networks*, and *complexity regularization* [35, 37, 38]. Pruning is the process of eliminating nodes and connections from the trained network. The reduced size network has to be sometimes retrained. NN pruning algorithms have practically developed for all major NN architectures [39].

2.3 How to detect overfitting

Beside preventing overfitting, a major question is how to detect it. It is desirable to have a measure that can quantify underfitting or overfitting of a network on a given learning problem. We do have again two general strategies: before and after learning.

The most common after learning technique is to perform learning/validation iteratively and optimize the learning/validation generalization error by adjusting the parameters and/or architecture of the network. Several constructive/destructive algorithms were adopted to incrementally increase or decrease the parameter to be optimized [40]. During the constructive/destructive process, cross-validation is commonly used to check the network quality and the design parameter is chosen using early stopping [41]. The training data is usually divided into two independent sets: a training set and a validation or testing set. Only the training set participates in the NN learning, and the validation set is used to compute a validation error, which approximates the generalization error. The inferred NN performance during training and validation is measured, respectively, by training error E_{train} and validation error E_{valid} presented. Once the validation performance stops improving as the target parameter continues to increase, it is possible that the training has begun to fit the noise in the training data, and overfitting occurs. Therefore, the stopping criterion is set so that, when E_{valid} starts to increase, or equivalently, when E_{train} and E_{valid} start to diverge, it is assumed that the optimal value of the target parameter has been reached [36]. Cross-validation + early stopping are the common techniques used in finding optimal network structure up to date. An alternative to cross-validation is bootstrapping.

More flexible stopping criteria based on early stopping were proposed by Prechelt [41]. It helped the users to choose stopping criterion in a systematic and automatic way, based on efficiency, effectiveness, or robustness. Liu *et al.* have introduced an algorithm which, on a given NN is able to recognize the occurrence of overfitting by examining the training error without using a validation set [36]. The algorithm also shows where the recycling of the training samples can be safely stopped so that the optimal structure of the NN is found. A signal-to-noise-ratio figure (SNRF) is defined to measure the goodness-of-fit using the training error. Based on the SNRF measurement, an optimized approximation algorithm is proposed to avoid overfitting in function approximation.

An open problem is how to detect before learning the generalization capability, without even knowing the NN to be used. In this case, one should be able to determine the generalization capability of a given training set before using it! For instance, we should determine if a training set is sufficiently smooth and covers sufficiently well the input space in order to produce a reasonably good approximation of an unknown function. Such a regression problem depends on the quality of available samples. Can we determine if the training set is good enough for being used? Can we do this independent of the NN model?

3 Two Efficient Methods

We will illustrate the concepts introduced in Section 2 with two FAMR methods which work well with small training sets. Since the methods have been previously described in [9], we will only review them here.

3.1 The GA-FAMR

The relevances attached to the input data are considered as adaptive parameters to be optimized by a GA.

The GA-FAMR operates on an initial population of relevance vectors. Each relevance vector has a single relevance associated with a specific training datum in accordance with the FAMR. Because the relevance of specific data is not known beforehand, this population must be optimized using the following GA:

Step One. Initialize a population of Pop_{size} chromosomes. Each chromosome is composed of N genes, where N equals the size of the training dataset. Each gene is a real value in the range (0, 10), defining the relevance of one of the training molecules.

Step Two. For each chromosome, train and validate the FAMR using cross-validation. Compute the fitness value of each chromosome: $Fit = 1/sMAPE$.

Step Three. Establish the next generation.

1. Find Fit_{low}, which is the smallest fitness value in the population.
2. Subtract Fit_{low} from the fitness value of each chromosome.
3. Sum the fitness values of all chromosomes to calculate the total fitness, Fit_{all}, of the population.
4. Divide each chromosome's fitness value by Fit_{all}.
5. Generate Pop_{size} new chromosomes to replace the current population. Each new chromosome is created by one of two methods: breeding or elitism.
   (a) BREEDING:
      i. For each child, two parents are selected according to the concept of the survival of the fittest.
      ii. Each parent is selected by first generating a random number, $0 < s < 1$.
      iii. Iterate through the chromosomes in the population. If $Fit \geq s$, the chromosome is selected. Else, subtract Fit from s, and continue to the next chromosome. The probability that a chromosome will be selected for reproduction at any given time is given by: $(Fit - Fit_{low})/(Fit_{all} - Fit_{low} * Pop_{size})$.
      iv. When two parents have been selected for each child, perform crossover to generate the new chromosome. For each child, one of two crossover methods is chosen with equal probability:
         A. For each gene, copy the genetic material from one or the other parent; the parent copied for each gene is selected randomly.
         B. Average the genes of the two parents. Because the effect of switching specific bits in a real value can be extremely unpredictable, it may be more effective to average two real values.
      v. Before the new child is introduced into the next generation, there is a 0.25 probability that it will undergo mutation in one of its genes, by randomly generating a new real value.

(b) ELITISM: At all times, eight global best chromosomes are retained as a possible source of members of the new generation. There is a 1/500 probability that a new chromosome is generated by selecting one of these elite, rather than by crossover of two members of the current population.

3.2 Ordered FAMR

For optimizing the FAM training data ordering, Dagher *et al.* [42, 43] and Tan *et al.* [44] have introduced efficient procedures. Essentially, the training data is preprocessed to identify a fixed order of pattern presentation. We refer to this procedure as the ordering algorithm. When the training input patterns are presented to the FAMR according to this fixed order, we obtain a FAMR with improved generalization capability.

Preprocessing consists of clustering input data. Each cluster center will be a molecule in the training set. The ordering of the training data is determined by the order in which the cluster centers are obtained. It is noteworthy that this clustering is different than the formation of ART_a categories, which is also a clustering of the same input dataset.

The ordering algorithm is controlled by a pre-defined parameter, n_{clust}, which is the number of input data clusters, and consists of the following three stages:

1. Determine the first pattern to be presented. This pattern corresponds to the first cluster center of the training data.
2. Determine the next $n_{clust}-1$ patterns to be presented. These patterns correspond to the next $n_{clust}-1$ cluster centers of the training data, and are identified through the Max-Min clustering algorithm [45].
3. Determine the order of the remaining patterns. These patterns are chosen according to the minimum Euclidean distance criterion from the n_{clust} centers defined in Stages 1 and 2.

Stage 1. We start with an M-dimensional input pattern $\mathbf{a} = (a_1, \cdots, a_M)$ and obtain 2M-dimensional input pattern $\mathbf{A} = (a_1, \cdots, a_M, 1-a_1, \cdots, 1-a_M)$ by complement coding [5].

Input pattern $\mathbf{a}$, which maximizes the sum in eq (2), is selected as the first pattern to be presented. This pattern is also treated as the first cluster center of the training patterns.

$$\sum_{i=1}^{M} |a_{M+i} - a_i| \tag{2}$$

Stage 2. The next $n_{clust}-1$ input patterns are identified for presentation during network training. These patterns represent the next cluster centers of the training patterns. They are determined consecutively using the Max-Min clustering algorithm. In this stage, the Euclidean distances between the remaining input patterns and the existing cluster centers $\mathbf{a}^1, \cdots, \mathbf{a}^k$ ($k \leq n_{clust}$) are computed. The minimum Euclidean distance between each remaining input pattern $\mathbf{a}$ and the existing cluster centers is identified: $d^{\mathbf{a}}_{min} = min\ dist(\mathbf{a}, \mathbf{a}^j)$ ($1 \leq j \leq k$). The input pattern which maximizes $d^{\mathbf{a}}_{min}$ is selected as the next cluster center.

Stage 3. The presentation order of the remaining input patterns is determined by finding the minimum Euclidean distances between these patterns and the n_{clust} cluster centers. The whole procedure of Stage 3 is repeated until the order of all input patterns for the network training phase has been identified.

The value of n_{clust} not only influences the input data ordering, but also has a major impact on the number of ART_a categories created. Thus, n_{clust} controls the generalization capability of the network.

Successive optimization of relevances and ordering is not a good strategy. The two optimizations can possibly cancel each other out, since they may influence each other. Therefore, we do not optimize both

Table 1: Prediction performance analysis on the training set [9].

	FS-GA-FNN	**Standard FAMR**	**GA-FAMR**	**Ordered FAMR**
sMAPE	89.28	89.99	77.65	86.04
RMSE	1132.12	1401.94	1332.53	1366.04

Table 2: Prediction performance analysis on the test set [9].

	FS-GA-FNN	**Standard FAMR**	**GA-FAMR**	**Ordered FAMR**
sMAPE	111.91	99.01	105.17	84.51
RMSE	506.08	43.45	56.99	25.49

relevances and ordering for the same network. We will refer in the following to the Ordered FAMR - a FAMR with equal (not optimized relevances) and optimized training data ordering.

4 Experimental Results

In our experiments, all networks were trained with the same set of 176 molecules, using twenty-fold cross-validation. Thus, we improve the generalization performance on this small training set by introducing some computational overhead. In all experiments we used on-line (incremental) learning: the training set is processed only once.

When estimating the quality of a prediction model, the prediction accuracy obtained both for training data and new data is important. One is interested not only in how accurately the model approximates the learning data, but also how the model generalizes on new data. The test set, which is not used for training, consists of twenty molecules. This set is from a different group of molecules than the one used for training, making prediction more difficult.

We investigate the GA-FAMR and the Ordered FAMR. The results are compared to the standard FAMR model (with no optimizations and equal relevances), and to the FS-GA-FNN.

The parameters of the network are determined experimentally, and are fixed for all FAMR models considered. The ρ_a and ρ_b parameters control the number of generated FAMR categories. It is important to limit the number of categories to prevent overfitting. Maintaining constant FAMR parameters for all tested models simplifies comparison. For the standard FAMR, the number of ART_a categories is 13 and the number of ART_b categories is 8. The experimentally optimized number of ART_a categories is close to the number of scaffold subtypes, which is a significant match.

The statistical results for the test sets are in Tables 1 and 2. As expected, the optimized FAMR models adjust better than the standard FAMR to the training data (Table 1). Of the three FAMR models, the GA-FAMR adjusts best to the training data.

Does the GA-FAMR overfit? We may find the answer by analyzing the prediction performance for test data. From Table 2 we conclude that the Ordered FAMR improves the standard FAMR over the test set. The GA-FAMR appears to overfit the training data and has therefore a less performant generalization.

Overall, from Tables 1 and 2, we conclude that the Ordered-FAMR performs better than the other models.

For low IC_{50} values, all three FAMR models exhibit a similar prediction pattern and they clearly overpredict the target values of the test molecules, which is good in our particular application.

We have predicted the IC_{50} value of 26 novel potential inhibitors using all four models (see [3]). The FS-GA-FNN and the FAMR are two radically different neural paradigms. The training datasets are the same, but the number of descriptors is different: FAMR uses 35, while FS-GA-FNN uses a feature selected subset of 22 descriptors. For some of the novel molecules, all methods predicted very low IC_{50}

Table 3: GA-FAMR prediction performance analysis on the training and test sets for different number of GA generations [9].

	25 generations	**50 generations**	**100 generations**
Training sMAPE	87.78	86.48	84.66
Training RMSE	1389.54	1381.47	1360.18
Test sMAPE	95.56	101.42	106.20
Test RMSE	41.03	48.80	63.09

values. Since radically different methods indicate high inhibitory activity, these are the molecules we consider as excellent candidates for organic synthesis and further drug discovery.

It is interesting to analyze the way the GA optimization performs for different numbers of generations (Fig. 3). With an increasing number of generations, the network adjusts better to the training set, but it also reduces its generalization capability with respect to the test set. Thus, the number of generations controls overfitting. In our experiments (Tables 1 and 2), we have used 2000 generations and this explains the relatively poor generalization obtained. We could use less generations and thus improve generalization with the cost of adjusting less to the training data. To determine the optimal number of generations and establish the best trade-off between generalization and overfitting, we may use an early stopping technique, or Liu's *et al.* algorithm [36].

The generalization capability of the Ordered FAMR is good, but depends on an appropriate selection of the n_{clust} parameter, which is a weakness of this algorithm. The GA-FAMR is also a good choice, but early stopping should be used to avoid overfitting. The computational overhead of the two algorithms is insignificant when compared to the value of the results. A computationally intensive solution is acceptable because drug synthesis requires years of time and great expense. Therefore, obtaining an accurate prediction is more important than execution time.

5 Conclusions

We have discussed and illustrated how to infer from small datasets. We do not have a nice mathematical solution to the general problem of learning from small datasets. But why? Here is our answer: If the VC dimension is too large relative to the size of the training set and we do not have any information about the quality of our training data and how representative it is, then the problem is ill-posed. We only can state the general rule of thumb: From the available samples, extract maximum information, without overfitting. There is no free lunch and we have to balance overfitting and generalization.

Both presented techniques work well, but we may have a significant computational overhead, which can make our solution non-scalable. The paradox is that, in our computational chemistry problem, we do not need scalability, since we do not have enough data anyway! These methods may be used for similar application, whenever we have to infer from small training sets. This does not mean that we prefer small training sets, but that we have to adapt our methods to what is available.

Bibliography

[1] R. Andonie, L. Fabry-Asztalos, S. Abdul-Wahid, C. Collar, and N. Salim, "An integrated soft computing approach for predicting biological activity of potential HIV-1 protease inhibitors," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2006)*, Vancouver, BC, Canada, July 16-21 2006, pp. 7495–7502.

[2] L. Fabry-Asztalos, R. Andonie, C. Collar, S. Abdul-Wahid, and N. Salim, "A genetic algorithm optimized fuzzy neural network analysis of the affinity of inhibitors for HIV-1 protease," *Bioorganic and Medicinal Chemistry*, vol. 16, pp. 2903–2911, 2008.

[3] R. Andonie, L. Fabry-Asztalos, C. B. Abdul-Wahid, S. Abdul-Wahid, G. I. Barker, and L. C. Magill, "Fuzzy ARTMAP prediction of biological activities for potential HIV-1 protease inhibitors using a small molecular dataset," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 99, no. PrePrints, 2009.

[4] R. Andonie and L. Sasu, "Fuzzy ARTMAP with input relevances," *IEEE Transactions on Neural Networks*, vol. 17, pp. 929–941, 2006.

[5] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 698–713, 1992.

[6] S. Verzi, G. Heileman, M. Georgiopoulos, and G. Anagnostopoulos, "Universal approximation with fuzzy art and fuzzy ARTMAP," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN '03)*, vol. 3, Portland, Oregon, 20-24 July 2003, pp. 1987–1992.

[7] R. Andonie, L. Fabry-Asztalos, C. Collar, S. Abdul-Wahid, and N. Salim, "Neuro-fuzzy prediction of biological activity and rule extraction for HIV-1 protease inhibitors," in *Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB'05)*, 2005, pp. 113–120.

[8] R. Andonie, L. Fabry-Asztalos, L. Magill, and S. Abdul-Wahid, "A new Fuzzy ARTMAP approach for predicting biological activity of potential HIV-1 protease inhibitors," in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2007)*, I. C. S. Press, Ed., San Jose, CA, 2007, pp. 56–61.

[9] R. Andonie, "Inference from small training sets - a computational intelligence perspective," University of Ulster, Jordanstown, Nothern Ireland, United Kingdom, invited talk, June 2008.

[10] R. Andonie, L. Fabry-Asztalos, B. Crivat, S. Abdul-Wahid, and B. Abdul-Wahid, "Fuzzy ARTMAP rule extraction in computational chemistry," in *IJCNN'09: Proceedings of the 2009 International Joint Conference on Neural Networks*. IEEE, 2009, pp. 2961–2967.

[11] R. Andonie, "Extreme data mining: Inference from small datasets," National University of Ireland, Maynooth, Ireland, invited talk, June 2008.

[12] ——, "How to learn from small training sets," Dalle Molle Institute for Artificial Intelligence (IDSIA), Manno-Lugano, Switzerland, invited talk, September 2009.

[13] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 2000.

[14] J. L. Balcázar and R. V. Book, "Sets with small generalized Kolmogorov complexity," *Acta Inf.*, vol. 23, no. 6, pp. 679–688, 1986.

[15] A. Ambainis, "Application of Kolmogorov complexity to inductive inference with limited memory," in *ALT '95: Proceedings of the 6th International Conference on Algorithmic Learning Theory*. London, UK: Springer-Verlag, 1995, pp. 313–318.

[16] A. Ambainis, K. Apsitis, C. Calude, R. Freivalds, M. Karpinski, T. Larfeldt, I. Sala, and J. Smotrovs, "Effects of Kolmogorov complexity present in inductive inference as well," in *ALT '97: Proceedings of the 8th International Conference on Algorithmic Learning Theory*. London, UK: Springer-Verlag, 1997, pp. 244–259.

[17] J.-L. Yuan and T. Fine, "Neural-network design for small training sets of high dimension," *IEEE Tnansactions on Neural Networks*, vol. 9, pp. 266–280, 1998.

[18] J.-L. Yuan, "Bootstrapping nonparametric feature selection algorithms for mining small data sets," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 1999, pp. 2526 – 2529.

[19] C. Huang and C. Moraga, "A diffusion-neural-network for learning from small samples," *International Journal of Approximate Reasoning*, vol. 35, pp. 137–161, 2004.

[20] R. Mao, H. Zhu, L. Zhang, and A. Chen, "A new method to assist small data set neural network learning," in *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)*, 2006, pp. 17–22.

[21] D.-C. Li, C.-S. Wu, T. T.-I., and L. Y.-S., "Using mega-trend-diffusion and artificial samples in small data set learning for early flexible manufacturing system scheduling knowledge," *Computers and Operations Research*, vol. 34, pp. 966–982, 2007.

[22] D.-C. Li, C.-W. Yeh, T.-I. Tsai, Y.-H. Fang, and S. Hu, "Acquiring knowledge with limited experience," *Expert Systems*, vol. 24, pp. 162–170, 2007.

[23] D.-C. Li, C.-S. Wu, T.-I. Tsai, and F. M. Chang, "Using mega-fuzzification and data trend estimation in small data set learning for early FMS scheduling knowledge," *Comput. Oper. Res.*, vol. 33, no. 6, pp. 1857–1869, 2006.

[24] T.-I. Tsai and D.-C. Li, "Approximate modeling for high order non-linear functions using small sample sets," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 564–569, 2008.

[25] D.-C. Li and C.-W. Yeh, "A non-parametric learning algorithm for small manufacturing data sets," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 391–398, 2008.

[26] D.-C. Li and C.-W. Liu, "A neural network weight determination model designed uniquely for small data set learning," *Expert Syst. Appl.*, vol. 36, no. 6, pp. 9853–9858, 2009.

[27] I. V. Tetko, A. I. Luik, and G. I. Poda, "Application of neural networks in structure-activity relationships of a small number of molecules," *J. Med. Chem.*, vol. 36, pp. 811–814, 1993.

[28] D. Hecht and G. Fogel, "High-throughput ligand screening via preclustering and evolved neural networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, pp. 476–484, 2007.

[29] M. Cheung, S. Johnson, D. Hecht, and G. Fogel, "Quantitative structure-property relationships for drug solubility prediction using evolved neural networks," in *Proceedings of the IEEE World Congress on Computational Intelligence*, 2008, pp. 688–693.

[30] H. Lohr, *Sampling: Design and Analysis*. Duxbury Press, 1999.

[31] J. Gamez, F. Modave, and O. Kosheleva, "Selecting the most representative sample is NP-hard: Need for expert (fuzzy) knowledge," in *Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence). IEEE International Conference on*, June 2008, pp. 1069–1074.

[32] L. Holmstrom and P. Koistinen, "Using additive noise in backpropagation training," *IEEE Transactions on Neural Networks*, vol. 3, pp. 24–38, 1992.

[33] C. Wang and J. C. Principe, "Training neural networks with additive noise in the desired signal," *IEEE Transactions on Neural Networks*, vol. 10, pp. 1511–1517, 1995.

[34] K. Wang, J. Yang, G. Shi, and Q. Wang, "An expanded training set based validation method to avoid overfitting for neural network classifier," *International Conference on Natural Computation*, vol. 3, pp. 83–87, 2008.

[35] G. N. Karystinos and D. A. Pados, "On overfitting, generalization, and randomly expanded training sets," *IEEE Transactions on Neural Networks*, vol. 5, pp. 1050–1057, 2000.

[36] Y. Liu, J. A. Starzyk, and Z. Zhu, "Optimized approximation algorithm in neural networks without overfitting," *IEEE Transactions on Neural Networks*, vol. 19, no. 6, pp. 983–995, 2008.

[37] S. Bos and E. Chug, "Using weight decay to optimize the generalization ability of a perceptron," in *Proceedings of the 1996 International Conference on Neural Networks*. IEEE, 1996, pp. 241–246.

[38] K. Mahdaviani, H. Mazyar, S. Majidi, and M. H. Saraee, "A method to resolve the overfitting problem in recurrent neural networks for prediction of complex systems' behavior," in *IJCNN'08: Proceedings of the 2008 International Joint Conference on Neural Networks*, 2008, pp. 3723–3728.

[39] R. Reed, "Pruning algorithms - a survey," *IEEE Transactions on Neural Networks*, vol. 4, pp. 740–747, 1993.

[40] T.-Y. Kwok and D.-Y. Yeung, "Constructive algorithms for structure learning in feedforward neural networks for regression problems," *IEEE Transactions on Neural Networks*, vol. 8, pp. 630–645, 1997.

[41] L. Prechelt, "Automatic early stopping using cross validation: Quantifying the criteria," *Neural Networks*, vol. 11, pp. 761–767, 1998.

[42] I. Dagher, M. Georgiopoulos, G. Heileman, and G. Bebis, "Ordered Fuzzy ARTMAP: a Fuzzy ARTMAP algorithm with a fixed order of pattern presentation," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 1998), IEEE World Congress on Computational Intelligence*, Anchorage, Alaska, 1998, pp. 1717–1722.

[43] I. Dagher, M. Georgiopoulos, G. L. Heileman, and G. Bebis, "An ordering algorithm for pattern presentation in Fuzzy ARTMAP that tends to improve generalization performance," *IEEE Transactions on Neural Networks*, vol. 10, pp. 768–778, 1999.

[44] S. Tan, M. Rao, and C. P. Lim, "A hybrid neural network classifier combining ordered Fuzzy ARTMAP and the dynamic decay adjustment algorithm," *Soft Computing*, vol. 12, pp. 765–775, 2008.

[45] J. Tou and R. Gonzales, *Pattern recognition principles*. Reading, MA: Addison-Wesley, 1976.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 3, pp. 292-300

A Novel Model for Adaptive Control Systems A State Machine Approach

F. Valles-Barajas

Fernando Valles-Barajas
Universidad Regiomontana
Information Technology Department
15 de Mayo 567 pte., C.P. 64000 colonia centro, Monterrey, Nuevo León, México
E-mail: fernando.valles@acm.org, fernando.valles@ieee.org

Abstract: In this paper a new model, based on state machines, of adaptive control systems is presented. Due to its high level of expressiveness, UML was chosen as the modeling language. In particular the paper presents a model of an indirect adaptive control system. This model can be used to document and to have a better understanding of adaptive control systems.
Keywords: adaptive control systems, state machines, Personal Software Process (PSP), software design

1 Introduction

Adaptive control is one of the research areas of control engineering that deals with time-varying systems [9]. To control a process using this technique, the first step is to obtain a model of the process based on measurements of the input and output of the process (these measurements are stored in a vector called measurement or observation vector $\phi(k)$); this step is called identification and is done by using a parameter adaptation algorithm (PAA) like the recursive least squares algorithm. Once the parameters of the process are obtained, the controller is designed using these parameters.

The Personal Software Process[1] is a modern methodology that helps the engineer to ensure quality software products [7]. The design phase of this methodology requires that the software engineer builds four models. Every of these models represents a different view of a system; for example there model that represents the states and the transitions between these states of a particular entity using state machines, which represent the internal dynamic view of a system.
State machines are used in software engineering to analyze the behavior of complex systems, for example the rational unified process (RUP)[2] uses state machines to model a use case, an operation or an object [2].

In this paper the concepts of state machines will be applied to model an adaptive control system (ACS). There are several notations to represent state machines (see for example [5], [6], [7]), in this paper the notation of the UML will be used. The reason for this, is the maturity of this modeling language and the success shown by UML to model complex systems.

Motivation and contribution of the paper:

1. the graphical model obtained with the state machine will help the control engineer to have a better understanding of the ACS control law.

2. this model could be used by a software engineer as a base to implement the software of the ACS.

[1] © Software Engineering Institute
[2] © IBM

3. by using this new model, a better understanding by the software engineer of an adaptive control system will be obtained.

4. this model can be used as documentation of the control system.

5. a better communication between the software engineer and the control engineer will be obtained by using the proposed model.

Conventions used in the paper: In all the paper, for the purpose of clarity, all the parts of the state machines (states, transitions, events, guard conditions and actions) are printed in *italic font*. The key concepts of the state machines will be written in **bold font**.

Related works: In [4] an adaptive strategy based on state machines is presented.
The **events** that are considerd in the strategy are: *threshold crossings*, *commands from the operator of the system* and *the occurrence of several patterns in the process signals* (control error $e(k)$, process input $u(k)$ and process output $y(k)$) among others. The occurrence of one of these events may provoke that the adaptive strategy changes from one state to another.
The **states** modeled in this strategy are: *initial* state, *open loop* state, *closed loop* state and *final* state. The *open* and *closed loop* states are **composite states**, which are states composed of other states [12].
In that paper the author shows a successful example of his proposal.

The paper of [13] shows an application of state machines in fault-adaptive control systems. As stated in that paper this kind of system can deal with time-varying systems and also with systems that present faults; the performance of the control system must remain unaffected in spite of these two problems. To deal with these problems that paper proposes to add to the control system a supervisory controller module and a reconfiguration management module. The relevant point in the paper is that the supervisor controller module is represented as a finite state machine (FSM).
That paper contains an example of the application of the roll control of a simulated airplane.

Organization of the paper: Section 2 presents the necessary concepts of the adaptive control theory to understand the proposed model. In section 3 an explanation of the building blocks for state machines is given. Section 4 contains the proposed model for an adaptive control system using state machines. The last section gives some concluding remarks.

2 Adaptive control systems

Fig. 1 shows the configuration of an indirect adaptive control system. In this kind of adaptive control, the model of the process $G_p(z^{-1})$ is obtained based on a set of input-output measurements ($u(k)$, $y(k)$) and then the controller is designed with this model. The parameter adaptation algorithm (PAA) block is responsible for obtaining this model. The controller design block designs the controller $G_c(z^{-1})$ based on the model obtained by the PAA and on the desired performance specified by the operator of the system. The adaptive control system must control the process in spite of the disturbances $d_1(k)$, $d_2(k)$, the noise $n(k)$ and the parametric variations of the process.

3 State machines

State machines are graphical models that allow the specification of the states of one entity, the legal transitions between these states, the events that can occur in the life of the entity and how the entity manages these events [10]. The transitions from one state to another can be fired by an event. Some

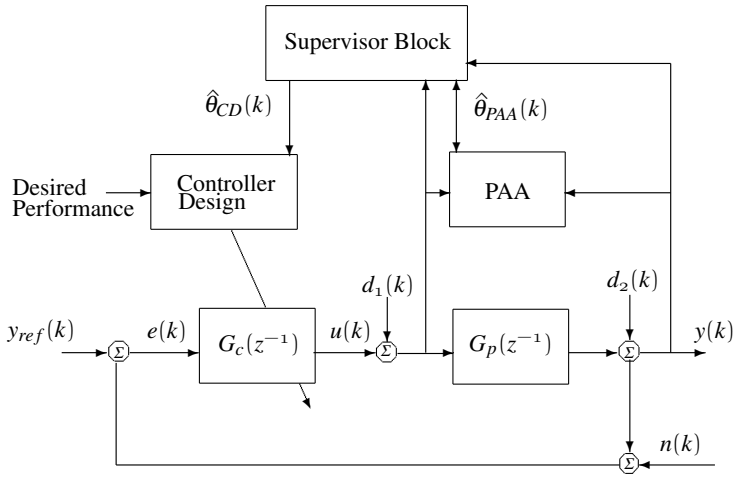


Figure 1: Indirect adaptive control system equipped with a supervisor block

condition (called guard condition) can be specified as a requirement to the occurrence of the transition.

To model an entity using a state machine the entity is isolated from the world and the communication with the rest of the world is specified by detecting events and responding to them. In UML state machines are represented using state diagrams [12].

A state machine can also be used to model use cases, which are useful to specify the functional requirements of a system. A state machine that describes the external events and its sequence in a use case is a kind of state diagram of use cases [10]. The external events happen between the actors, which are any external entity that interacts with the system, and the system [8]. Events that pass between objects residing in the system are referred to as internal events. Some authors like [10] recommend to use state machines to model the external and time events and the responses to them instead of using state machines to model the internal events of objects (instances of a class).

A state machine is drawn inside a rectangle with the name label in the top left-hand side (see fig. 2), though the frame may be omitted if the context is clear [1]. If the context is an instance of a class, all the classes related with it are candidate targets for actions and for inclusion in guard conditions [3].
A state is drawn as a rounded rectangle with two compartments: one for the state name and the other for the internal transitions and the internal behavior (see fig. 2). The second compartment may define entry and exit actions, activities, internal transitions and deferred events.
There must be only one initial state[3], but there may be any number of final states, including no final state at all. Initial states are pseudostates, they do not have all the properties of a real state. Final states are real states; they posses all the properties of a real state; this means that while the entity is in a final state it can perform some activity and can response to some events. The difference between final states and the rest of the real states is that they do not have transitions out. An initial state is drawn using a solid black circle and a final state is drawn using a shallow black dot (see fig. 2).
A state machine can have an exit point (drawn as a circle with a cross, as in fig. 2) that indicates the occurrence of an exception [1].

[3]It is important to mention that [1] considers that a state machine may have more than one initial state only if each of these states are labeled with the event that created the entity.

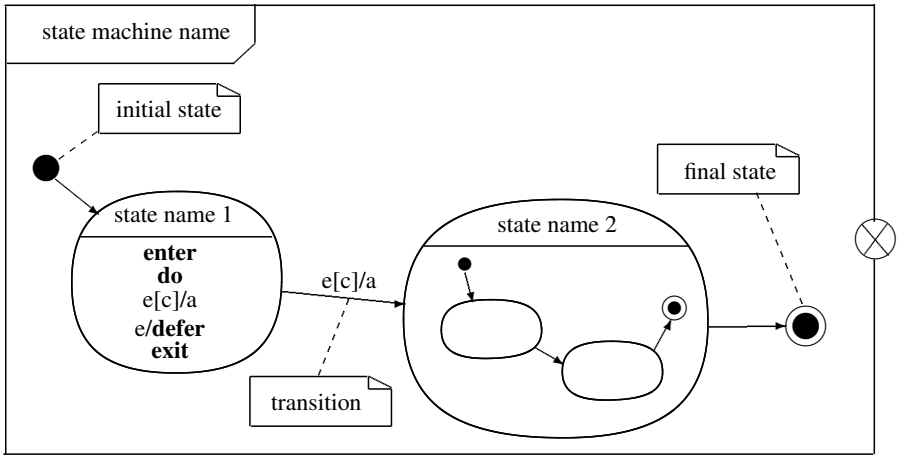


Figure 2: A state machine example

State machine can model parallel operations; this is important in control systems if we consider the situation when it is necessary that the system responses to commands from the operator and at the same time operates the plant.

This section has presented some basic concepts of state machines. The next section contains the model of an adaptive control system based on state machines. Also in the next section advanced concepts of state machines will be illustrated using the proposed model.

4 The model

This section explains how an adaptive control system can be modeled by a state machine. The explanation is divided in three section, which correspond to every of the building blocks of a state machine: the states, the events and the transitions. The actions that occur while a control system is in some state will be explained in the section of the states and the actions executed when a transition occurs will be explained in the section of the transitions.

4.1 States

A state describes the condition of an object [11]. This condition for a control system can be expressed in any of the following ways:

- in the terms of the control system attributes:
  - the values of the attributes that define the control system's structure. For example the values of the poles of the close loop transfer function (see the polynomial $A_m(z^{-1})$ in eq. 1) define if the system is in *stable* or *unstable* state.

$$H_{cl}(z^{-1}) = \frac{z^{-d+1}B_m(z^{-1})}{A_m(z^{-1})} \tag{1}$$

The control system will be in the *stable* state if all the roots of the transfer function denominator of eq. 1 are inside the unit circle, as it is stated in the eq.

$$1 + a_{m1}z^{-1} + \cdots + a_{mn}z^{-n} = 0 \Rightarrow |z| < 1 \tag{2}$$

Otherwise the state of the system will be in the *unstable* state.

– the value of the attributes that define the object's relationships. In this point it is important to mention that the entire control system is represented with a class called ACS. When a fault occurs in the system a fault detection and isolation (FDI) object of the class FDI will be created to manage this fault. A FDI reference variable will be defined inside the ACS class, so when the fault occurs an instance of the class FDI will be created inside a method of the class ACS. If there is no fault in the system, the reference variable of FDI will have the value of null and the system will be in the *no fault* state, but when a fault appears the reference variable will point to the FDI object that will handle the fault and the system will be in the *fault* state.

- in terms of a behavior that the object is engaged in:

  – a period of time during which a control system performs some outgoing activity. For example, one of the activities that a controller must do to achieve robustness against the external agents to the system (disturbances, noise, faults) is to shape the sensitivity functions. This state will be called *shaping the sensitivity functions*. The output sensitivity function $\mathcal{S}$ is the transfer function between the output disturbance $d_2(k)$ and the plant output $y(k)$; this transfer function is usually named as sensitivity function and is given by:

$$\begin{aligned} S_{yd_2}(z^{-1}) &= \frac{A(z^{-1})S(z^{-1})}{A(z^{-1})S(z^{-1})+z^{-d}B(z^{-1})R(z^{-1})} \\ &= \frac{A(z^{-1})S(z^{-1})}{P(z^{-1})} \end{aligned} \tag{3}$$

  The noise sensitivity function $\mathcal{T}$ is the transfer function between the measurement noise $n(k)$ and the plant output $y(k)$ and is given by:

$$\begin{aligned} S_{yn}(z^{-1}) &= \frac{-z^{-d}B(z^{-1})R(z^{-1})}{A(z^{-1})S(z^{-1})+z^{-d}B(z^{-1})R(z^{-1})} \\ &= \frac{-z^{-d}B(z^{-1})R(z^{-1})}{P(z^{-1})} \end{aligned} \tag{4}$$

  An equation that relates these two transfer functions is

$$S_{yd_2}(z^{-1}) - S_{yn}(z^{-1}) = \mathcal{S} + \mathcal{T} = 1 \tag{5}$$

  – a period of time during which an object waits for some event or events to occur. Suppose that the ACS is equipped with a fault detection and isolation module to manage the faults that can appear in the system. The FDI system can be modeled as an object. One of the states of the FDI is the *waiting for a fault* state.

The response of a control system to an event depends on its state. For example if the system is in *closed loop* state and an the event *change reference* $y_{ref}(z^{-1})$ occurs the system will take $y(k)$ close to $y_{ref}(k)$ but if the system is in *closed loop* state and the same event occurs the system will do nothing.

4.2 Events

An event is any occurrence that provokes the reaction of an object, this reaction can be the transition from one state to another, the execution of one action or do nothing. It is important to mention that although events starting with UML 2.0 are called triggers; in this paper we will use the term events.
There are four types of events: signals events, time events, call events and change events. They will be explained in the following sections.

Signal events

A signal is an asynchronous event; it is not necessary that a sender waits for the answer of a receiver when a sender sends a signal to a receiver. Signals can also be triggered by an operation; this kind of signals are called exceptions. Signals are a one-way asynchronous communication between active objects.

An exception is the most common kind of signals that specifies the abnormal behavior of an operation [8]. Exceptions are modeled in UML as stereotyped classifiers. The parameters of the exceptions contain information of the abnormal behavior, and these are modeled as attributes in the exceptions. Because exceptions are modeled as classifiers, a hierarchy of exceptions can be drawn. Fig. 3 shows some of the exceptions that can be generated in an adaptive control system. These exceptions form a hierarchy.

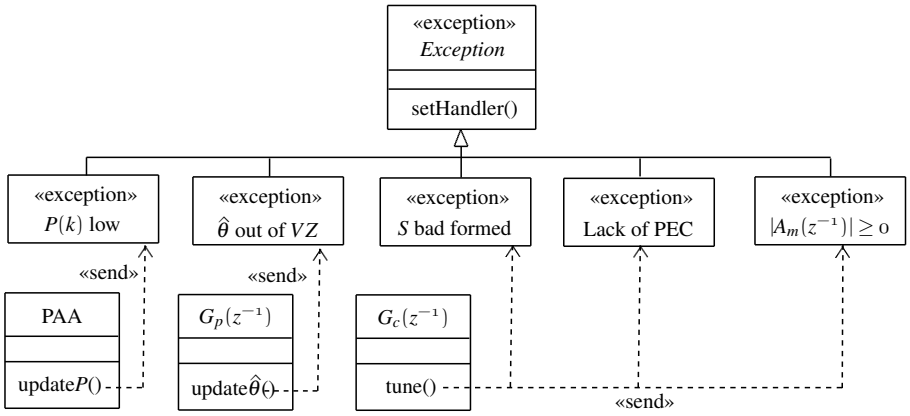


Figure 3: Modeling the exceptions of an adaptive control system

The root of this hierarchy is the abstract class "Exception" which was specified by typing the name of the class in italic font. The "Exception" class has a method called setHanlder(); this method is useful to specify the entity that is going to manage the exception, once it occurs. In this proposal, the supervisor is the entity that handles all the exceptions of an ACS.
Fig. 3 shows some of the exceptions that can be thrown in the methods of the classes PAA, $G_p(z^{-1})$ and $G_c(z^{-1})$. When $G_c(z^{-1})$ is tuned, the resulting controller may provoke that the closed loop system changes to the *unstable* state (the poles of $A_m(z^{-1})$ are outside the unit circle; see eq. 1). To specify that an operation can trigger an exception the exception can be drawn as a stereotyped class and then a send dependency can be drawn between the operation that can trigger the signal and the signal. See for example that in fig. 3 a send dependency between the operation update$\widehat{\theta}$() and the signal "$\widehat{\theta}$ is out of VZ" has been drawn to indicate that the operation update$\widehat{\theta}$() can trigger the exception "$\widehat{\theta}$ is out of VZ" (VZ is the valid zone which is a zone where the parameters of $G_p(z^{-1})$ are allowed. This can be obtained from previous knowledge or by association with the physical parameters of the process). Also the resulting controller may be poorly robust; this can be inspected by looking at the sensitive function $\mathbb{S}$ (see eq. 3) which is a measure of how the disturbance $d_2(k)$ affects the output of the process $y(k)$.

One of the advantages of the configuration of the fig. 3 is that polymorphism can be used to handle the exceptions; the handler of the exceptions can be specified in such a way that it handles all the exceptions of the ACS. Also when we specify a transition, we can specify that any of the exceptions of the ACS can trigger a transition; in this case the transition is polymorphic and can be triggered by the exception "Exception" or by any of its specializations, for example the "Lack of PEC" exception [3].

Signals are used to establish a communication between two objects in an asynchronous way. The reception of a signal is an event for the receiver. In this paper $d_1(k)$, $d_2(k)$, $n(k)$ and the bursting phenomenon will be modeled as signals as fig. 4 indicates. Is is important to mention that because $d_1(k)$, $d_2(k)$ and $n(k)$ are unknown, they must be estimated.
The entity that generates these signals is the environment and the receiver is the adaptive control system. Fig. 4 shows that signals are modeled as stereotyped classifiers and because of this a hierarchy of signals

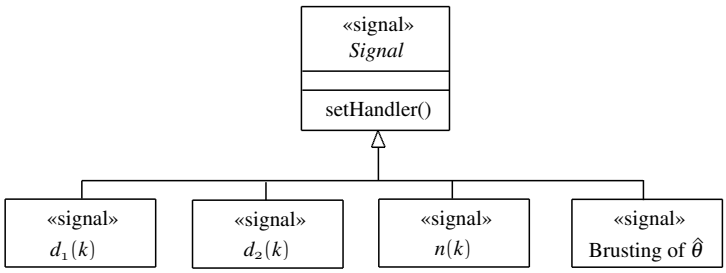


Figure 4: Modeling of the signals for an adaptive control system

can be established. Signals can also have parameters in the form of attributes, these are useful to store information related with the signals.

When we define signals we are making the opposite procedure for exceptions. When signals are defined the entities that handle the signals are defined and when the exceptions are defined the abnormal behavior that can generate an operation is defined [3].

In the UML, you model the signals that an object may receive by naming them in an extra compartment of the class, as shown in fig. 5 where the signals that the supervisor block can handle are defined. Signals do not return values to the caller [8].

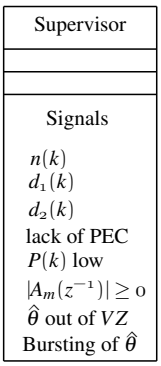


Figure 5: Signals that the supervisor can handle

Time events

A time event evaluates the passage of time as a trigger. It is assumed that the object has some mechanism to monitor the passage of time. Time can be specified either in absolute mode (time of day) or relative mode (time elapsed since a given event). Time can be considered as an event from the environment. The general form of a time event is:

after(exp)[guard condition]/ action.

An example of a time event is:

after(5 minutes)[|P(k)| low]/reset P(k)

where 5 minutes is the expression that is evaluated, [|P(k)| low] is called guard condition and reset $P(k)$ is the action that is executed when the expression and the guard condition are true.

Call events

In the UML, you model the call events that an object may receive as operations on the class of the object. For example, in fig. 3 the operation of the class $G_c(z^{-1})$ was modeled as a call event.

Signal events and call events involve at least two objects: the object that sends the signal or invokes the operation and the object to which the event is directed.

Change events

A change event occurs when a boolean expression becomes true. When an event is declared as a change event, the system always monitors that boolean expression. A change event represents a continuous and potential nonlocal computation (action at a distance, because the value or values tested may be distant). Another alternative for a change event is to model this as a guard condition in the transition. The advantage of a guard condition is that it is evaluated only when a transition occurs.
The general form for a change event is:

when(expression) / operation()

where operation() is executed when the expression is true. The persistent excitation condition can be verified by using the following change event:

$$\text{when}\left(lim_{k_1\to\infty}\frac{1}{k_1}\sum_{k=1}^{k_1}\phi(k)\phi^T(k)\ \leq 0\right)/\ \text{freezeAdaptation}()$$

5 Conclusions

In this paper a new model for adaptive control systems has been presented. This model is based on state machines and is useful to document and to analyse control systems. The states of an adaptive control system, its transitions and the events that may provoke the transition from one state to another were defined in the proposed model. With this model the software implementation of an adaptive control system, made by a software engineer, will be easier.

Bibliography

[1] S. Bennett, J. Skelton, and K. Lunn. *Schaum's Outline of UML*. McGraw-Hill, USA, 2005.

[2] S. Bergstrom and L. Raberg. *Adopting the Rational Unified Process: Success with the RUP*. Addison Wesley, USA, 2004.

[3] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley Professional, 2^{nd} edition, May 19, 2005.

[4] D. Drechsel. An adaptive control system modeled as finite state machine. *Proceedings of XVI Annual Convention and Exhibition of the IEEE In India*, pages 124 – 128, 1990.

[5] M. Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 3^{rd} edition, 2003.

[6] D. Harel. Statecharts: a visual formalism for complex systems. *Science of Computer Programming*, pages 231–274, 1987.

[7] W. S. Humphrey. *PSP^{sm} : A Self-Improvement Process for Software Engineers (SEI Series in Software Engineering)*. Addison-Wesley Professional, USA, 2005.

[8] IBM. *Object Oriented Analysis and Design using UML*. USA, 2004.

[9] P. Ioannou and B. Fidan. *Adaptive Control Tutorial*. Society for Industrial and Applied Mathematics, 2006.

[10] C. Larman. *Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Prentice Hall, USA, 3^{rd} edition, 2004.

[11] T. Pender. *UML Bible*. Wiley, 2003.

[12] J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley Professional, 2^{nd} edition, 2004.

[13] G. Simon, T. Kovácsházy, G. Péceli, T. Szemethy, G. Karsai, and A. Lédeczi. Implementation of reconfiguration management in fault-adaptive control systems. *IEEE Instrumentation and Measurement Technology Conference*, 2002. Anchorage, AK, USA.

Fernando Valles-Barajas obtained a graduate degree in Computer Science at Center for Research and Graduate Programs of La Laguna Institute of Technology (1991). He received an MS in Control Engineering (1997) and a PhD in Artificial Intelligence (2001) from Monterrey Institute of Technology (ITESM) campus Monterrey. He was a research assistant at Mechatronics department of ITESM campus Monterrey (1997-2001). He received certification as a PSP Developer from the Software Engineering Institute of Carnegie Mellon University (2008). He is member of the IEEE and ACM. His research interests include topics in Software Engineering and Control Engineering. Currently, he is full-time professor in the Information Technology Department at Universidad Regiomontana, Monterrey, Nuevo León, México. He also teaches modules at both BS and MS levels in Computer Science and Software Engineering.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 3, pp. 301-313

SRoL - Web-based Resources for Languages and Language Technology e-Learning

S.M. Feraru, H.N. Teodorescu, M.D. Zbancioc

Silvia Monica Feraru
Institute for Computer Science of the Romanian Academy, Iaşi, Romania
E-mail: mferaru@etti.tuiasi.ro

Horia-Nicolai Teodorescu
Institute for Computer Science of the Romanian Academy, Iaşi, and
Technical University "Gheorghe Asachi" of Iaşi, Romania
E-mail: hteodor@etti.tuiasi.ro

Marius Dan Zbancioc
Institute for Computer Science of the Romanian Academy, Iaşi, and
Technical University "Gheorghe Asachi" of Iaşi, Romania
E-mail: zmarius@etti.tuiasi.ro

Abstract: The SRoL Web-based spoken language repository and tool collection includes thousands of voice recordings grouped on sections like "Basic sounds of the Romanian language", "Emotional voices", "Specific language processes", "Pathological voices", "Comparison of natural and synthetic speech", "Gnathophonics and gnathosonics". The recordings are annotated and documented according to proprietary methodology and protocols. Moreover, we included on the site extended documentation on the Romanian language, on speech technology, and on tools, produced by the SRoL team, for voice analysis. The resources are a part of the CLARIN European Network for Language Resources. The resources and tools are useful in virtual learning for phonetics of the Romanian language, speech technology, and medical subjects related to voice. We report on several applications in language learning and voice technology classes. Here, we emphasize the utilization of the SRoL resources in education for medicine and speech rehabilitation.
Keywords: spoken language resources, voice education, gnathosony, gnathophony, education, speech rehabilitation.

1 Introduction

In a world where the Web and Internet communication is pervasive, the computer is more than a study topic for everyone, it is a ubiquitous tool. Computers serve for more than doing computations, they are now one of the most used means of communication and interaction - the very basis of any educational system. As a consequence, computer-based education is an obvious choice whenever a distance separates the learner and the learning person. In a general sense, computer-based education and virtual education based on Internet is today an undeniable fact of life in every academic campus [28], [29]. While computers and the network are the means, the spoken language represents the prevalent support of communication in the teaching-learning process. Hence, the natural need to address e-learning and virtual learning of languages, phonetics, voice pathology, and other aspects related to voice and spoken language.

In view of the above, we built during a timeframe of about five years a web site that offers the possibility of teaching and learning various aspects on the Romanian language, based on an annotated corpus freely accessible on the Internet. The corpus is complemented with in-depth phonetic and linguistic analyses, moreover with specific tools accessible by users from everywhere through the

web [16], [17], [18], [19], [25]. This instrument has a high level of dimensionality and aims to cover numerous aspects of the language that are not typical features in language corpora. This makes this "corpus-tool" an unique instrument of its kind existing today in the domain [22].

During the recent years, we developed an emotional speech database which can help in education and re-education of speech, in diagnosis and treatment, and in learning a language aided by computer; examples of related published results are [5], [16], [22].

Voice and language e-education is a topic addressed by many research and educational groups. Solomon [13] studied the possibilities and issues of learning with and about computers in schools or in other learning environments. The Eric Education Resources Page shows the importance of computer assisted education of speech and voice [24]. On the other side, web-based educational resources and training have received attention during the last decade. Ake Olofsson [10] offers a simple method of compensation for word decoding problems, by using a computer which pronounces the words which can not be read. Olofsson developed a program for the IBM-PC/AT and a Scandinavian multilingual text-to-speech unit that children can use to read a textfile on the monitor and request using a mouse the pronunciation of any word from that text [10].

The computer-assisted learning language software helps the interaction between student and computer by speech, by sound effects, by animation, and by video. On the other hand, the interaction is restricted typically to the mouse and keyboard. An active interaction, through spoken language enhances the educational computer-based tools [1]. In computer-assisted language learning, speech recognition offers the possibilities to have an active participation by oral reading and conversation. The CALL system reported in [1] includes recordings spelled by native speakers. The user has the possibility to compare the quality of her pronunciation with model recordings.

In another direction of research, Warschaue [23] observes the uses of online communications for language teaching. He determined that the interest in this domain grows day by day. He proposed a conceptual framework for understanding the role of the interaction assisted by computer [23]. Lundberg considers the computer a tool of remediation in the education of students with reading disabilities as dyslexic students which can benefit by computer training in correct reading and spelling the words [9].

A speech database is a collection of files with sounds, structured according to its own purpose. The SRoL resource (corpus) is located at the address (www.etc.tuiasi.ro/sibm/romanian_spoken_language/index.htm). The initiator conceived SRoL as an Internet-based "dictionary of sounds and words" for the Romanian language supplemented with specific manifestations of voice (including pathologies) and various tools. The SRoL database includes files with vowels, consonants, diphthongs, sentences with emotional states, linguistic particularities for the Romanian language, dialectal voices, and gnathosonic and gnathophonic sounds. It is the first Internet based annotated database of emotional speech for the Romanian language and contains more than 1500 recordings in different coding formats (.wav, .ogg, .txt, 22 kHz sampling rate, 24 bit or 16 bit precision). The phonetic recordings in SRoL, which refer to an annotated emotional speech corpus (database), are registered to ORDA.

2 The SRoL resources and the SRoL web site

The SRoL corpus evolved from a small research and educational speech database around 1995 (see Annex 1). It currently includes several sections, all freely available on the web. The main sections are:

i) Standard pronunciation of vowels, diphthongs, words and short sentences in Romanian; the recordings in this section are appropriate for learning correct pronunciation in Romanian, moreover for statistical research on the Romanian phonetics;

ii) Special syntactic constructs (linguistic peculiarities), like double subject and apposition; this section is research-oriented;

iii) Emotional voices;

iv) Analytic comparison between the synthetic and natural speech [27];

v) Dialectal utterances;

vi) A small archive of gnathosonic/gnathophonic sounds (included in the general "Archive of Sounds").

Beyond the main sections, the SRoL site includes an introductory section on the phonetics of the Romanian language, descriptions of the recording protocols and descriptions of the methodology, analysis tools (free software), extended research documentation, a video application, references, and a list of potentially useful links. The SRoL team developed instruments for signal processing regarding the extraction of patterns from voice signals, and the computing of the fundamental frequency (pitch) traces, respectively the traces of formants $F1$, $F2$, $F3$. The site offers, beside executables programs, descriptions for each of these tools. Those descriptions are intended for a "general use", offering elementary explanations and relevant references for a better understanding [4], [22].

In this paper, we provide details about applications of the SRoL corpus, available to the address http://www.etc.tuiasi.ro/sibm/romanian_spoken_language/index.htm.

3 SRoL as support for learning the Romanian language

One of the goals of the SRoL web site is to provide a free Romanian database for students and researchers, for linguists, for teachers, in view of teaching, learning and analysis the Romanian language sounds. The database includes the pronunciation corpus and related documentation. The database contains among others, sections with:

- recordings of syllables and words pronounced in various contexts, like accentuated word, interrogative sentences, exclamations, various emotions conveyed by the speaker, etc. This part of the database is aimed as a source for concatenative synthesizers and as benchmark for the voice recognition systems (isolated words), based on statistical models of language and speech, as [26];

- files of sounds, syllables and words pronounced by persons with various pathologies; this section may be useful in medical and phonological researches;

- files with professional voices ("perfect" pronunciations), as well as non-professional voices, the "voices of the people in the street". For the moment, we concentrate on voices from the Iaşi region (East Romania) and middle area of Moldova.

Learning and teaching languages require well documented audio-visual tools that exemplify and fully explain spelling for a large variety of voices and contextual and emotional states. While former methods, like tape recordings and audio disks have been helpful, the multimedia Internet-based tools offer tremendously increased capabilities. SRoL represents such a tool for the Romanian language. Not only it is the first for the Romanian language, but its multidimensionality makes it somewhat unique and novel in concept for language learning and teaching in general.

As an example of use, consider the case of a foreign student who wants to improve her Romanian pronunciation by comparing the prosody of her voice with the prosody of native speakers. The student utters a sentence (from those included in the site), then opens WASP^{TM} or another similar tool and displays the energy and fundamental frequency in her voice. She then compares these prosodic features to the ones of native speakers and tries to improve her prosody until she produces correct prosodic patterns. Also, the student can compare formant values and try improving the formants of the vowels she pronounces.

This instrument is useful for learning to improve speech communication, moreover for human-computer speech interaction, for security, for medical applications, for video-games and interactive TV, for teachers, in the study of the Romanian language, etc.

4 Applications in medical education and re-education of speech

Application fields like language learning, professional voice education, and voice rehabilitation and re-education for medical conditions have different requirements, moreover are based on different methods. On the other side, education in medicine (ORL, phoniatrics, dentistry) and in logopedy are other fields of potential applications of speech resources. Further, voice analysis for diagnosis is a domain that has seen significant progresses in recent decades. Voice education is needed whenever a voice pathology including some neurologic and psychiatric disorders, or pathology of the vocal tract occurs. Several groups have addressed the voice re-education topic [9], [10].

4.1 SRoL resources for minor voice pathology

Till now, we included in SRoL words pronounced by persons with minor pathologies, as trembling voice. We have demonstrated in our research that splitting the signal in frequency bands that correspond to the peaks of the $F0$-$F1$ formants and respectively to the peaks of $F2$-$F3$ formants helps improving the discrimination process in a significant way. The use of fractal dimensions in assessing the jitter or shimmer in voice produce mixed results [21]. Adding other fractal dimension, the rate of recognition of the tremor segments in voice improves, but it still low [21]. The voice pathology section of the database is useful in medical and phonological researches. Also for medical education use, the site comprises a gnathosonic and gnathophonic corpus.

4.2 SRoL resources for gnathosony

The gnathosonic analysis refers to the analysis of sounds produced during occlusion, due to the closing of the mandible over the maxillary at some stage in masticatory-like movements. Watt (cited in [7], [8], [11], [12]) has initiated the analysis of these sounds with application to diagnosis of the state of the stomato-gnathic apparatus during the 1960s and 1970s. The method has seen some interest, but it is not yet a current method in clinical practice.

The shape of the envelope of an occlusal sound is determined by the number of occlusal contacts and by the dynamics of the terminal part of the occlusion, namely by the dynamics of the sliding of the teeth, from the first contact until the equilibrium position in occlusion. A characterization of the waveform should take into account the need to correlate the sound with the medically relevant processes of contact and sliding. A limit in the occlusal sound analysis has been the complexity and the variability of shapes of the sound wave. The envelope of a single contact sound is characterized by the rise and fall times, value of the maximum, duration of the maximum, and total duration. The rise and fall curves follow exponentially laws, whose constants are of interest in the classification of the occlusal dynamics. For gnathosonic purposes, the sound signal $s(t)$ generated by occlusion (teeth impact when closing the mouth like for mastication) and discretized as $s[n]$ is first filtered by an elementary high pass, differential filter, $s[n+1] \leftarrow s[n+1]-s[n]$. Then, the signal is filtered with a nonlinear filter introduced in [14]. The filter first extracts the rough envelopes, averages them, applies to them median filters, sums the two resulting envelopes, and then apply to the sum an averaging filter [14]:

$$u_{inf}[n] = min_{k=-6,..,0} s[n+k],\ u_{sup}[n] = max_{k=-6,..,0} s[n+k]$$

$$v_{inf}[n] = \frac{1}{7} \cdot \sum_{k=-6}^{0} u_{inf}[n+k],\ v_{sup}[n] = \frac{1}{7} \cdot \sum_{k=-6}^{0} u_{sup}[n+k].$$

The next stage in the filtering is constituted by the median filtering on a moving window, as

$$z_{inf} = median_{k=-6,..,0}\{v_{inf}[n+k]\},\ z_{sup} = median_{k=-6,..,0}\{v_{sup}[n+k]\}.$$

and the two envelopes are summed – actually, summed in the sense

$$y[n] = [|z_{inf}| + z_{sup}]/2;$$

$$e[n] = \frac{1}{p+1} \cdot \sum_{k=-p}^{0} y[n+k].$$

We used a window of width 6 ($p = 5$) for the last averaging. The widths of the windows in the above operations depend on the signal sampling frequency used in the recording process. The envelope of the signal is determined by taking the maximal respectively minimal value in a moving window, according to a procedure similar to the one explained for the filtering process. The envelope, e(t), is itself low-pass filtered and then used for determining the occlusal sound parameters. The heuristic procedure applied to determine the duration of the occlusal sounds by forming "binary" impulses during the valid occlusal sound is:

if $(e[n] > c_1)$ *and* $B(e[n-14], ..., e[n+14]) > c_2$ then $h[n] = 0.1$,
else $h[n] = 0$,

where B is a binary function (taking only 0 and 1 values) defined by

$$B = [max(e[n-14], ..., e[n-1]) > c_3] \;\&\; [max(e[n+1], ..., e[n+14]) > c_3].$$

The constants were chosen semi-empirically, as a function of the amplitude of the signal, $c_{1..3} \sim A_s$ where A_s is the average amplitude of the signal after filtering (actually, we used the average amplitude of the sum of the envelopes), and the window width, 14, is determined by tests. We used the values $c_1 = 0.001$, $c_2 = 0.01$, $c_3 = 0.005$, which correspond to the average signal $A = 0.027$, determined as explained. For a normalized amplitude A, $A = 1$, the constants are about $c_1 = 0.05$, $c_2 = 0.4$, $c_3 = 0.1$. The detection procedure can be further improved by reducing the false positives by imposing that the skewness of the impulse is larger than $+0.5$; typical values for the skewness are larger than 0.7, showing that the rise of the impulse is significantly faster than the decreasing part.

5 Research support in gnathophonics and gnathosonics

In previous researches, we identified several ways the pathology of the stomato-gnathic system influences the speech:

i) The lack of the frontal dentition, namely of the upper teeth, may dramatically change the spectrum of the fricative consonants.

ii) The lack of the upper teeth may significantly modify the spectrum of the dento-alveolar sounds *t*, *d*, *n*, and *l*. (Notice that these sounds are rather alveolar in English, while in some other languages, like Spanish and Romanian, they may be dental. Therefore, the influence of the dentition on phonation is language-dependent.)

iii) The limited mobility and the pain in the temporo-mandibular joint (TMJ) impedes the production of fast transient vowels, especially in the diphthongs where the second vowel is pronounced with a largely opened mouth, like *oa*, *ea*, *ua*.

iv) The uncertainty in uttering due to a forcing in the TMJ, or to a poor neuro-muscular control may produce a tremor of the voice (fast amplitude changes, errors in the attacks, i.e. error in transitory regimes etc.).

v) The neurological pathology of the buccal cavity may impede on the accuracy of the pronunciation, including deficient starting of the words.

vi) Defective mobile prostheses may produce extra sounds, especially when the mouth is fast opened for pronunciation, moreover, it may produce clicks before the utterances.

vii) Prostheses of the upper teeth that do not provide for a physiological "*V*" shaped space between the teeth impede on the pronunciation of the fricatives, for example *f*.

viii) Especially the fricative consonants and the labial vowels are affected by the state of the dental furniture.

The *s* consonant uniformly occupies a large spectrum for a healthy dental apparatus, while it has a multi-band spectrum when the upper front teeth are missing or have deficiencies. The pronunciation of *s* and *v* may become close to that of *f*. For subjects with mobile prostheses, we noticed an uncertainty in the starting of the uttering.

The difference ratio in amplitude spectra is a parameter defined as:

$$\Delta S = \sum_k \frac{|S_1(f_k) - S_2(f_k)|}{S_1(f_k) + S_2(f_k)}$$

where $f[k]$ is the k-th frequency in the FFT (Fast Fourier Transform) power spectrum of the two sounds and $S_{1,2}$ are the average power spectra of the two sounds. For two similar sounds uttered by the same speaker, a difference larger than 50% means that the sounds are clearly distinguishable, while a difference smaller than 10% means that the sounds are indistinguishable. For example, if the average spectra for two sustained utterances of f and v have a ΔS index of 40%, they will be distinguished by a listener, while if $\Delta S = 15\%$, they will be confused. We proposed the sustained consonant differential analysis as a method to further assess the impairment of speech production due to dentition. For this test, two similarly produced sounds are generated in a sustained mode and their spectra contrasted. For example, the sounds f and v are both at least partly fricative (v can be a semi-vowel, only partly fricative) that may be poorly produced due to imperfect dentition or neurological control. We conclude this section by stressing that gnathophonic testing should become a standard test for the dentist in the near future. The knowledge in the field is only emerging today, and fully developed, commercial tools are yet lacking, but the importance of the domain can not be refuted [7], [8], [11], [12]. The proposed tests are non-invasive, objective, and purely instrumental, hence their importance in the evaluation of the health state of the buccal system. These methods can easily be extended to remote, web-based diagnosis.

In figure 1, we exemplify a gnathophonic (a) and gnathosonic (b) recording sounds (for the speaker 19743m). In figure 1(a), we exemplified recordings of the Romanian words "*vata*", "*fata*", "*var*", intended to obviate similarities and differences in the pronunciations (Fourier spectra) of the consonants f and v, in the same context (beginning of the word, same $\_CVC$ structure, with the same vowels and consonants, and _ denotes the beginning of the word). This is one of the specific choices of words proposed by the second author to determine when dentition defects produce confusion in the $f-v$ uttered sounds. By analyzing such recordings available at SRoL, students can learn how to differentiate the normal and pathological states.

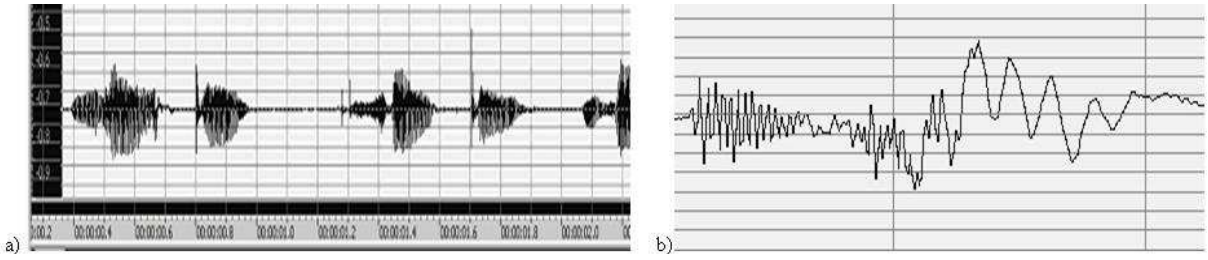


Figure 1: Gnathophonic (a) and gnathosonic (b) recording with details, tool GoldWave^{TM}

6 Applications in teaching the voice signal technology classes

Signal technology classes are taught around the world, especially for the master degrees in computer science and electrical engineering, moreover in some departments of linguistics and in a few medical centers. Some universities and education institutions developed their own databases and tools for spech processing. For examples, the Center for Spoken Language Understanding (CSLU) offers available language database from speech area and hearing science. These resources are important for analyzing the speech, for diagnosing and treating speech and language problems, for training students and so on. The tools and the corpora are distributed to over 2000 sites in 65 countries [2]. In education these tools help students learn about speech, learn a new language, learn through interactive media systems, or to become accustomed to hearing the normal and abnormal voice signal.

The second author currently uses the SRoL corpus in teaching and laboratory activities in the class "Speech Technology" given for the master degree in "Computational Linguistics" at the Faculty of Computer Science, "Al.I. Cuza" University of Iaşi. Details on the use in Voice Technology classes of some topics from SRoL are described in [4]. At the international EUROLAN 2007 summer school, the second author used the SRoL site to present "Traces of emotion, intentions and meaning in spoken Romanian" (http://eurolan.info.uaic.ro/html/profs/HNTeodorescu.html). The second author taught the specific methodology aspects, results obtained on the characterization of emotions in speech, possibilities of recognition of emotions and intentions in speech, and the relationship between specific meanings and the prosody in specific constructions in the Romanian language. The lesson exemplified applications of analysis of the speech emotional prosody to social, psycho-social, educational, and psycho-medical topics.

7 Software tools: pitch (F_0) extractor

The extraction of the fundamental frequency F_0 values combines four different methods: i) autocorrelation method (analysis in time domain) ii) the Average Magnitude Difference Function method, AMDF (analysis in time domain) iii) the Harmonic Product Spectrum method, HPS (based on spectral analysis) iv) the cepstral method (an analysis in que-frequency domain) - also applied for the higher formants searching.

The autocorrelation method is a classical method for pitch detection in the time domain. The method is based on the quasi-periodicity property of the voice signal and generates a local maximum that corresponds to the signal period. In the case of AMDF method, the local minimal values are detected and these values provide the necessary information to compute the fundamental period T_0,

$$C_k = \frac{1}{N} \cdot \sum_{n=0}^{N} x_n \cdot x_{n+k},\ k = \overline{0,W}$$

$$D_k = \frac{1}{N} \cdot \sum_{n=1}^{N} (x_n - x_{n+k}),\ k \in \overline{1,W}$$

Here, C_k is the self-correlation, D_k is the difference function coefficient for a delay k, x_n is the n-th sample of the signal, N is the number of correlation coeficients, W is the width of the analysis window.

The HPS method (Harmonic Product Spectrum) is based on the propriety that the spectrum of a periodic signal with fundamental frequency F_0 has maximal spectral values at the multiples of this frequency $2F_0$, $3F_0$, $4F_0$, ... (the harmonics of fundamental). When the signals are rescaled with the factors 1/2, 1/3, 1/4,... after the decimation operation, by the multiplication of the resulted signals (which all have a spectral maximum in fundamental frequency F_0), the other maximal value from spectrum are strongly attenuated.

$$H_n^k = H_{k \cdot n}^0 \ (decimation) \, or \, H_n^k = \frac{1}{k} \sum_{i=0}^{k-1} H_{k \cdot n+i}^0$$

The cepstral method relies on the separation of the spectrum of the sound generator, H_g (which provide the information regarding the fundamental frequency), from the spectrum of the vocal signal filter, H_f (which describe the resonating cavities model). In the cepstral formula, the multiplication operation between the excitatory signal and the transfer function spectrums is transformed using logarithms into an addition operation:

$$H(\omega) = FFT(s) = H_g(\omega) \cdot H_f(\omega)$$

$$cepstrum = IFFT(log|FFT(s)|^2) = IFFT(log|H_g(\omega) \cdot H_f(\omega)|^2)$$

$$cepstrum = IFFT(log|H_g(\omega)|) + IFFT(log|H_f(\omega)|^2)$$

where *FFT* is the Fast Fourier Transform, and *IFFT* is the inverse *FFT*.

The results of the F_0 extraction methods are compared in a decisional block, and a selection algorithm is used if there are significant differences. Another algorithm compares a current value with a number of neighboring values in order to select the nearest one, moreover compares the current values with mean values of F_0.

The error correction of the F_0 extractors is performed through three methods:

- comparing the "neighbors": use the results provided by the same F_0 extractor and if a difference between two consecutive values greater than a specified threshold value (usually 10-20%) is detected, the corresponding samples are considered errors;

- if the difference in absolute value between the current value of F_0 and the average of fundamental frequency is greater than twice the standard deviation, then we consider those values as erroneous;

- if the current value of F_0 is below 60% or over 150% of the average values of F_0, then we consider that the corresponding value is incorrect.

The threshold values were empirically determined and the final correction is accomplished by applying all the three correction methods described. The decision block receives the F_0 values provided by the detection methods (AMDF differences method, autocorrelation method, HPS method, and cepstral method). To achieve the best possible pitch detection, the output values are weighted according on the performance of each F_0 extractor. We assign smaller weights to the methods with a higher probability of providing incorrect outputs. The false detections of the fundamental frequency often consist in selecting the first subarmonic, or the first harmonic of F_0. When these "false" detection are not repaired by the correction module, we have two options:

- comparing the outputs of different F_0 detection methods for the same window of analysis;

- comparing the outputs with a number of previous final results provided by the decision block.

8 Discussion

Our team has a long standing experience with using novel technologies in teaching, lasting for three decades [3], [7], [15], [20]. We applied that experience to the SRoL e-teaching and e-learning resource.

The SRoL resource is a vast annotated corpus of speech files complemented by tutorials, papers and additional files, moreover with tools for speech processing. If used by an experimented student or teacher, it may become a powerful tool for instruction and learning the Romanian language pronunciation, speech technology, and voice pathology and re-education. The SRoL sound voice resource is useful in many domains, including phonology, applied computer science, and medicine. Students and researchers may use this freely accessible site for learning the pronunciation of Romanian language, for

making comparative study between Romanian and other languages, for development of synthetic voice systems, for other linguistic, phonetic, socio-linguistic or medical applications.

This database is structured corresponding to precise criteria, documented and annotated according to a well defined methodology. The site has more then 1500 recordings of syllable, word, and sentence with various tonalities and pronounced with various emotional states. The database contains recordings of professional and normal voices, from the North-East region of Romania, without dialectal accent.

The SRoL resources have been recognized by several bodies, beyond the scientific publications that included our papers on SRoL. CLARIN European Network of Language Resources accepted SRoL as a member; ORDA (the Romanian Office for Authorship Rights) registered the original recordings, and the SRoL received a gold medal and media attention at the INVENTICA 2009 fair for inventions and creativity. Also, the website of Embassy of France in Romania briefly described in its Bulletin the SRoL site and its use in education (http://www.bulletins-electroniques.com/actualites/58811.htm). The Technical University "Gheorghe Asachi" of Iaşi intends to use SRoL in helping foreign students enrolled at this university to learn the correct Romanian pronunciation.

We hope the SRoL resources will be used in all the universities in Romania by foreign students who learn the Romanian language, moreover in other academic media and as an online tool by foreign students and teachers. We welcome any request for help and educational advice from all those who wish to use SRoL and the language-related web resources in virtual e-teaching and for e-learning.

9 Conclusions and future work

The SRoL speech annotated corpus constitutes the first extensive educational and research web speech corpus for the Romanian language. We believe it also constitutes a speech repository unique in many respects, including the first international language and sound resources for gnathophony and gnathosony, the first resources for comparative study of appositions and double subject constructions, moreover specific features as the rigorous methodology of documenting the records we used.

The objectives for the next two years are to increase the speech data base by about 1000 annotated recordings and to significantly extend the medical-oriented section of the resources. Also, we intend to add more tools for speech processing, including statistical tools on the GRID.

Acknowledgements

The authors have been partly supported by the Romanian Academy, moreover the second author has been partly supported by a grant of the Ministry of Education and Science of Romania, during 2005-2006.

NOTICES

1. A partial version [6] of this paper was presented in the ICVL 2009 conference and received the INTEL Special Award for Education (2009).

2. The authors contributions: the gnathophonic and gnathosonic research was been performed by the second author who also wrote the corresponding section of the paper (Sections 2, 4, 5, 6, and 8, and contributed to writing the other sections); the first author helped with further recordings and with their inclusion on the web page.

Bibliography

[1] K. Cameron, Computer Assisted Language Learning (CALL) Media, Design, and Applications, *Taylor & Francis*, ISBN: 902651543X, http://www.google.com/books?id=dO_ sNQlWhrsC & printsec=frontcover & dq=related, ISBN0940753030 & hl=ro & source=gbs_ similarbooks_s & cad=1.

[2] R.A. Cole, Tools for Research and Education in Speech Science, *Proc. Int. Conf. for Physics Students*, 1999, www.cslu.ogi.edu/toolkit/pubs/pdf/cole_ICPS_99.pdf.

[3] F. De Coulon, E. Forte, D. Mlynek, H.N. Teodorescu, St. Suceveanu, Subject State Analysis by Computer in CAE, *Proc. Int. Conf. on Intelligent Technologies in Human-Related Sciences*, Leon, Spain. Vol .2, pp. 243-250, 1996.

[4] D. Cristea, H.N. Teodorescu, D.I. Tufis, Student Projects in Language and Speech Processing, *4th Conf. on Language Resources and Evaluation, Lisbon, PortugalWorkshop on Language Resources: Integration and Development in E-learning and in Teaching Computational Linguistics*, pp. 17-22, 2004, http://nats-www.informatik.uni-hamburg.de/view/Main/AcceptedPapers.

[5] M. Feraru, H.N. Teodorescu, The Emotional Speech Section of the Romanian Spoken Language Archive, *Conf. on Intelligent Systems and Technologies, Proc. 5th European*, Iaşi, Romania, ISBN 978973730497, 2008.

[6] M.S. Feraru, H.N. Teodorescu, SRoL - Web-based Resources and Tools used for Language and Language Technology e-Learning, *Virtual Learning - Virtual Reality, Proc. 4th International Conference on Virtual Learning, ICVL 2009*, Bucharest University Press, ISSN: 1844-8933, Section Models & Methodologies, pp. 119-127, 2009.

[7] W. Hedzelek, T. Hornowski, Gnathosonic Study of Occlusion in Patients Wearing Complete Dentures, *Eur J Prosthodont Restor Dent.*, Vol. 5, No. 3, pp. 119-23, 1997.

[8] W. Hedzelek, T. Hornowski, The Analysis of Frequency of Occlusal Sounds in Patients with Periodontal Diseases and Gnathic Dysfunction, *J Oral Rehabil.*, Vol. 25, No. 2, pp. 139-45, 1998.

[9] I. Lundberg, The Computer as a Tool of Remediation in the Education of Students with Reading Disabilities: A Theory-Based Approach, Learning Disability Quarterly, *Technology for Persons with Learning Disabilities*, Vol. 18, No. 2, pp. 89-99, 1995 http://www.jstor.org/pss/1511197.

[10] A. Olofsson, Synthetic Speech and Computer Aided Reading for Reading Disabled Children, *Journal: Reading and Writing*, Vol. 4, No. 2, pp. 165-178, ISSN: 09224777, 1992 (http://www.springerlink.com/content/j521536n135x2864/).

[11] J.F. Prinz, Computer Aided Gnathosonic Analysis: Distinguishing Between Single and Multiple Tooth Impact Sounds, *J Oral Rehabil.*, Vol. 27, No. 8, pp. 682-689, 2000.

[12] J.F.Prinz, K.W. Ng, Characterization of Sounds Emanating from the Human Temporomandibular Joints, *Arch Oral Biol.* Vol. 41, No. 7, pp. 631-639, 1996.

[13] C. Solomon, Computer Environments for Children - A Reflection of Theories of Learning and Education, 1988 www.google.com/books?id=EonPZ9A81kkC&printsec= frontcover & hl=ro & source=gbs_v2_summary_r& cad=0.

[14] H.N. Teodorescu, Occlusal Sound Analysis Revisted, *Proc. 3rd Int. Conf. MEDSIP 2006, Advances in Medical, Signal and Information Processing*, ISBN: 0863416586, Glasgow, UK, 17-19 July 2006.

[15] H.N. Teodorescu, Computer Semiotics: Understanding Meanings and Parallel Languages (Refereed invited paper) T. Yamakawa, G. Matsumoto (Eds.), *Proc. Int. Conf. IIZUKA'98, World Scientific Publ.*, pp. 279-283, 1998.

[16] H.N. Teodorescu, M. Feraru, Classification in Gnathophonics - Preliminary Results, *The Second Symposium on Electrical and Electronics Engineering*, Galati University Press, pp. 525-530, ISBN 1842-8046, 2008.

[17] H.N. Teodorescu, M. Feraru, Micro-corpus de Sunete Gnatosonice si Gnatofonice, Pistol, Cristea, Tufis (Eds.) *Resurse lingvistice si instrumente pentru prelucrarea limbii romane*, Ed. Universitatii "Al.I. Cuza" Iaşi, ISBN 978-973-703-297-3, pp. 21-30, 2007.

[18] H.N. Teodorescu, M. Feraru, D. Trandabat, Studies on the Prosody of the Romanian Language: The Emotional Prosody and the Prosody of Double-Subject Sentences, C. Burileanu, H-N. Teodorescu, (Eds.) *Advances in Spoken Language Technology*, The Publishing House of the Romanian Academy, Bucharest, Romania, ISBN 978-973-27-1516-1, pp. 171-182, 2007b.

[19] H.N. Teodorescu, M. Zbancioc, E. Mihailescu, Speech Technology and Bio-Medical Engineering Teaching Based on the Web-A new Tool and Case Study, *Int. Conf. on Interactive Computed Aided Learning*, Villach, Austria, 2006.

[20] H.N. Teodorescu, A. Kandel, B. Paschall, Teaching Modern Chapters in Automata Theory and Formal Languages, (abstract in booklet of the Symposium.) *Symp. 21 Century Teaching Technologies, Univ. South Florida*, Tampa, USA 2000.

[21] H.N. Teodorescu, R. Ganea, M. Feraru, A. Burlui, Assement of Voice Quality Based on Nonlinear Dynamic Analysis, *Proc. of The 15th Int. Conf. on Control Syst. & Computer Sci.*, Bucharest, Romania, pp. 536-542, ISBN 9738449898, 2005.

[22] H.N. Teodorescu, D. Tandabat, M. Feraru, M. Zbancioc, R. Luca, A corpus of the Sounds in the Romanian Spoken Language for Language-Related Education In: C.P. Pascual (Ed.), *Revisiting Language Learning Resources, Cambridge Scholars Pub. (CSP)*,UK, Ch. 6, ISBN 1847181562, pp. 73-89, 2007.

[23] M. Warschaue, Computer-Mediated Collaborative Learning: Theory and Practice, *The Modern Language Journal*, Vol. 81, No. 4, Special Issue - Interaction, Collaboration, and Cooperation - Learning Languages and Preparing Language Teachers (Winter, 1997), pp. 470-481, http://www.jstor.org/pss/328890.

[24] B.W. Wise, R.K. Olson, Computer Speech and the Remediation of Reading and Spelling Problems, *J. Special Education Technology*, Vol. 12, No. 3, pp. 207-220, 1994.

[25] M. Zbancioc, Tools for the Archive of the Romanian Language Sounds Project, *4th European Conf. on Intelligent Systems and Technologies*, Iaşi, Romania, ISBN 973-730-265-6, 2006.

[26] Kenko Ota, Emmanuel Dulfos, Philippe Vanheeghe, Masuzo Yanagida, Bayesian Inference for Speech Density Estimation by the Dirichlet Process Mixture, *, Studies in Informatics and Control Journal*, Bucharest, Romania, ISSN 1220-1776, Vol. 16, No. 3, 2007.

[27] Florin Grigoras, Horia-Nicolai Teodorescu, Vasile Apopei, Nonlinear Analysis and Synthesis Of Speech, *Studies in Informatics and Control Journal*, Bucharest, Romania, ISSN 1220-1776, Vol. 7, No. 1, 1998.

[28] Tom Page, Gisli Thorsteinsson, Andrei Niculescu, Management of Knowledge in a Problem Based Learning Environment, *Studies in Informatics and Control Journal - With Emphasis on Useful Applications of Advanced Technology*, Bucharest, Romania, Vol. 18, No. 1, 2009.

[29] Antonios Andreatos, International Journal of Computers,Virtual Communities and their Importance for Informal Learning Communications and Control, *International Journal of Computers, Communications and Control - IJCCC*, Romania, ISSN 1841-9836, Vol. II, No. 1,pp.39-47, 2007.

Annex 1. Development stages of SRoL

The presently named SRoL corpus started around 1995 as a small, research and educational database including examples of recordings with vowels and a few typical words in Romanian, moreover a few recordings of pathological voices. It was correlated to the class of Image and Speech Processing given by the second author in the "Gheorghe Asachi" Technical University of Iaşi, Romania. Former students (who are now professors in several Romanian universities) contributed to that incipient voice database (credit for recordings and other help for that database deserve the now professors Radu Ciorap and Irinel Pletea, among others). The database was further developed for educational purposes in relation to the the class of Speech Technology given by the second author in the Faculty of Computer Science of "Al. I. Cuza" University in Iaşi.

The third stage of development started in 2004, when the second author decided to significantly enlarge and move the speech database on the web, partly with the help of two grants that helped forming a team in the Institute for Computer Science of the Romanian Academy and in the "Gheorghe Asachi" Technical University of Iaşi. The first author joined the team, at that time as a fresh Ph.D. Student. Since the second author initiated five years ago the Project "The Sounds of Romanian Language" (SRoL), the team increased to 8 researches. The SRoL Web-based spoken language repository and tool collection as it is today was developed during several years by the collaboration of groups from the Institute for Computer Science of the Romanian Academy, CERFS Excellence Center in "Gheorghe Asachi" Technical University of Iaşi and by staff of the discipline of Language Technology, Computer Science Faculty, "Al.I. Cuza" University.

Annex 2. Typical shapes of gnathosonic signals

The sketches below stand for the envelopes of typical gnathosonic signals, corresponding to normal, merged double contact, and isolated double contact signals. The sound is easily categorized by automatic means.

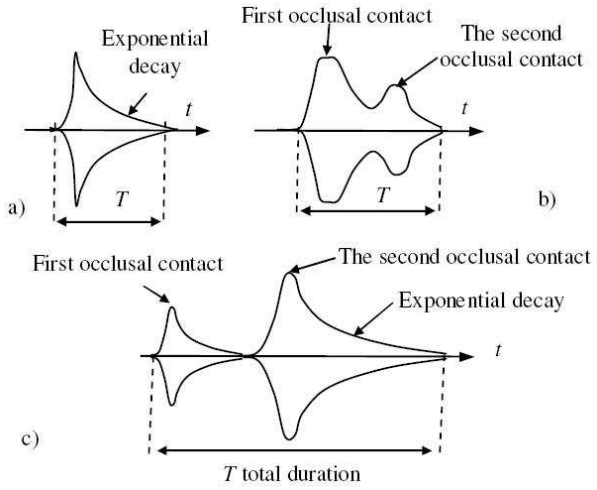


Figure 2: Typical envelopes of occlusal sounds (from [14])

Silvia Monica Feraru (November 21, 1977) received a MSc. degree in BioMedical Engineering (2004) and PhD in Electronics (2009) from "Gheorghe Asachi" Technical University of Iaşi. Now she is research assistant at the Institute for Computer Science of the Romanian Academy, Iaşi branch. She received the Special Awards Intel Education 2009 at The International Conference on Virtual Learning, ICVL 2009. Her current research interests include vocal signal processing, cognitive processes, and various aspects of artificial intelligence. She has (co-)authored more than 21 conference, journal or bookchapter papers.

Horia-Nicolai Teodorescu (November 14, 1951). MS in Electronics, "POLITEHNICA" University, Bucharest, 1975, Ph.D. in Applied Physics - Electronics, under the supervision of the late Prof. Emil Luca, at the Technical University of Iaşi, 1981. Currently, he is a professor at the "Gheorghe Asachi" Technical University of Iaşi and the director of the Institute for Computer Science of the Romanian Academy, Iaşi. He is a correspondent member of the Romanian Academy. Has authored or co-authored about 300 journal and conference papers, holds 24 national and international patents and has received numerous national and international awards and prizes. He is a Senior Member, IEEE.

Marius-Dan Zbancioc (August 15, 1975) teaching assistant at the "Gheorghe Asachi" Technical University of Iaşi and researcher at the Institute of Computer Science of the Romanian Academy, Iaşi branch. His current research interests include signal processing, expert systems, fuzzy systems and several aspects of artificial intelligence. He has (co-)authored 3 books and 39 papers.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 3, pp. 314-324

Image Segmentation using Euler Graphs

T.N. Janakiraman, P.V.S.S.R. Chandra Mouli

T.N. Janakiraman
Department of Mathematics
National Institute of Technology, Trichy, India.
E-mail: janaki@nitt.edu

P.V.S.S.R. Chandra Mouli
School of Computing Science and Engineering
V.I.T. University, Vellore, India.
E-mail: mouli.chand@gmail.com

Abstract: This paper presents a new algorithm for image segmentation problem using the concepts of Euler graphs in graph theory. By treating image as an undirected weighted non-planar finite graph (G), image segmentation is handled as graph partitioning problem. The proposed method locates region boundaries or clusters and runs in polynomial time. Subjective comparison and objective evaluation shows the efficacy of the proposed approach in different image domains.
Keywords: Image Segmentation, Graph theory, Euler Graphs, Cycles.

1 Introduction

Image segmentation can be treated as a graph partitioning problem which is solved by making use of cuts in a weighted graph based on certain criterion. The proposed method deals the image segmentation problem in a diverse manner. An excellent review for image segmentation is available in [8], [9], [15]. Earlier approaches to image segmentation are categorized into three groups: (1) Cluster the low level feature, such as histogram thresholding by [14], k–means / k–centroid by [12], [27] and mixture of Gaussians (MoG) by [2], (2) Edge linking such as dynamic programming by [26], relaxation approach by [16] and saliency network by [25] and (3) Region operations, such as region splitting and merging by [24], [23], region growing methods by [3] and by [13], by [17], by [11] and region competition by [20]. Applications of segmentation are abundant. It is heavily used in medical imaging. For example, segmentation of internal brain nuclei in MRI images as discussed in [38]. This work is aimed to bring robust image segmentation using graph theoretic concepts like Euler graphs and cycles. The proposed method finds the cycles of a given graph so that the image regions are formed by connecting all relevant pixels together. The relevancy of pixels is determined based on two parameters namely, edge weight similarity and node label similarity, which are described in the subsequent sections. The algorithm may end up at a particular stage when there is no possibility of refinement due to constraints imposed on cycle formation. Such paths are tried for further refinement. If the refinement is not possible then those paths are treated as open paths and may be treated as cuts. All the procedures of the proposed method run in polynomial time. The rest of the paper is organized as follows. In Section 2, a brief review on graph based segmentation is discussed. The basic definitions related to Euler graphs and some of its properties are presented in Section 3. In Section 4, the proposed algorithm and the experimental results are presented. Section 5 concludes the work.

2 Graph Approaches

Recently graph based image segmentation has attracted growing interest. Graph Theory and its concepts has been dominating in image processing research. The concepts of graph theory like maximum

flow, maximum clique, shortest path, minimum spanning tree etc have been used for image processing problems. [21] discussed the various types of graph algorithms in computer vision. A special issue on graph based image processing is published in [32]. Early graph-based methods include by [4], [18] and more recent formulations in terms of graph cuts by [22], [31] and spectral methods by [30]. The notion of a connectivity graph was introduced by [19] to allow for image processing on a foveal sensor. This notion is introduced specifically to model the sampling of the macaque retina in [5]. The work of Zahn (1971) presents a segmentation method based on the minimum spanning tree (MST) of the graph. The segmentation criterion in Zahn's method is to break MST edges with large weights. The algorithm proposed by Urquhart (1982) normalizes the weight of an edge using the smallest weight incident on the vertices touching that edge. Work by Wu and Leahy (1993) introduced such a cut criterion, but it was biased toward finding small components. This bias was addressed with the normalized cut criterion developed by Shi and Malik (2000), which takes into account self-similarity of regions. These cut-based approaches to segmentation capture non-local properties of the image, in contrast with the early graph-based methods. Weiss (1999) has shown how the eigenvector-based approximations developed by Shi and Malik relate to more standard spectral partitioning methods on graphs. However, all such methods are too slow for many practical applications. An alternative to the graph cut approach is to look for cycles in a graph embedded in the image plane. [10] described the quality of each cycle is normalized in a way that is closely related to the normalized cuts approach. [7] described an efficient graph-based segmentation in which they defined a predicate for measuring the evidence for a boundary between two regions. Using that predicate, an algorithm is developed which makes greed decisions to produce segmentations that satisfy global properties. The literature in the most recent times reveal many improvements over these existing methods but for comparison and evaluation, the methods by Shi and Malik, Pedro F. Felzenzwalb etc are treated as benchmark works. A method to build a hierarchy of partitions of an image is introduced by [29] in which they build a hierarchy of partitions of an image by comparing in a pairwise manner the difference along the boundary of two components relative to the differences of components' internal differences. They stated the drawback of this method as the maximum and minimum criterion introduced are very sensitive to noise, although in practice it has a small impact. A MST pyramid based segmentation is carried out by [28] using dual graph contraction. For evaluating the segmentation results of the proposed methods with other existing methods, Precision, Recall and F-measure have been implemented since Berkeley Images [34] for segmentation have been evaluated using these three measures. The methods considered for comparison are [35], [36] and [37].

3 Background

Leonhard Euler discussed [6] graphs for the first time while solving the famous Seven Bridges of Königsberg problem. The following are some of the terms and their definitions used in this work. These definitions are taken as they are defined in [1].

3.1 Basic Definitions

Let G(V,E) be the given graph with V and E representing the vertex set and edge set respectively.

Definition 1. A trail that traverses every edge of G is called an Euler trail. It is named as Euler trail because Euler was the first to investigate the existence of such trails in graphs.

Definition 2. An Euler tour is a tour which covers all the edges of G.

Definition 3. A graph is an Euler graph or Eulerian if it contains an Euler tour.

Euler proved the following theorem and corollary through which a graph has Euler tour can be determined. The following characterizations are taken as they are defined and proved in Bondy and Murty (1982).

Theorem 4. *A non-empty connected graph is Eulerian if and only if it has no vertices of odd degree.*

Corollary: A connected graph has an Euler trail if and only if it has at most two vertices of odd degree.

3.2 Extraction / Development of Euler graphs from non-Euler graphs

If a graph does not have an Euler circuit, it still might be interested in knowing how it could be traveled with as few retraced edges as possible (starting and ending at the same vertex). Eulerian can be obtained in two ways. (i) By adding one spurious multiple edge which joins two adjacent odd degree vertices and (ii) By deleting the edges joining two adjacent odd degree vertices.

4 Proposed Method

The Euler graph and its properties are exposed in this work for solving image segmentation problem. The basic idea is that Euler graph is decomposed into edge disjoint cycles. The steps of the proposed method are given below:

```
Step-1: Representation of image as a grid graph
Step-2: Conversion of grid graph into Eulerian
Step-3: Segmentation Procedure
Step-4: Refinement of segments
```

These stages are discussed in detail in the following sub-sections.

4.1 Representation of Image as a Grid Graph

The image to be segmented is represented as a graph $G(V,E)$. To do so, each pixel is treated as a vertex of the graph. Edges are defined based on 8–connectivity of the neighborhood vertices. An edge $(v_i, v_j) \in E$ corresponds to a pair of neighboring vertices. The graph G, thus obtained is an undirected weighted non-planar graph. Clearly, an image of size $N \times N$ contains N^2 vertices, $(N-1)N$ vertical edges, $N(N-1)$ horizontal edges and $2(N-1)^2$ diagonal edges. Thus, in total there are $(N-1)N + N(N-1) + 2(N-1)^2 = 4N^2 - 6N + 2$ edges. Let $M = 4N^2 - 6N + 2$. The graph thus formed is visualized as a grid and hence called as grid graph. A sample grid graph of size 8×8 is shown in Figure 1. The weights are assigned to the edges by using the absolute intensity difference between the adjacent pixels.

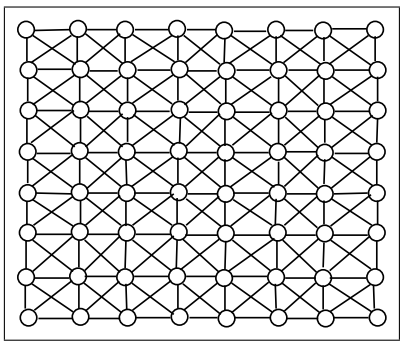


Figure 1: Grid graph of an image

4.2 Conversion of Grid Graph into Eulerian

The grid graph thus obtained is a connected non-Eulerian because some of the vertices have odd degree. The procedure for the conversion to Eulerian guarantees the formation of cycles covering all edges since all the vertices are of even degree. Border vertices are the vertices on the first row, last row, first column and last column. For this reason, the grid graph can be converted to Eulerian so that all vertices have even degree. This can be achieved in two ways. In the first case i.e., by adding one extra multiple edge for each of disjoint pair of adjacent odd degree vertices. The same weight is allocated to both duplicated and original edge to avoid ambiguity. The process is repeated until no such pair exists. In Figure 2, (a) and (d) show two grid graphs of size $4 \times 4$and 5×4 respectively. Figure 2(b) and (c) represent the two possible Euler graphs of (a), (e) and (f) represent the Euler graphs of (d).

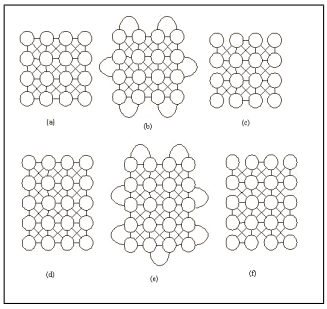


Figure 2: Grid graph and its corresponding Euler graphs

In the second case, instead of adding duplicate edges to the pair of adjacent vertices of odd degree, alternate edges are removed at the boundary to maintain even degree. It is found that there is no loss of information from images by removing such edges because all the edges removed are due to border vertices. In practice, there is not much information available at the border vertices and experimentally it is found that there is no variation in the segments formed in either way.

4.3 Segmentation Procedure

Once the given image is represented as Eulerian, the segmentation procedure is carried out over the Eulerian. The algorithms for image segmentation and segments_formed are given below.

```
1. Color all the edges as white.
2. Call segments_formed procedure.
3. Call regions_refinement procedure.
```

```
1. Select arbitrarily a white colored edge.
2. The selected edge is included in the
        temporary_growing_vector if it satisfies the threshold.
3. If the temporary_growing_vector forms a cycle then the
        closed path is stored in cycles_formed vector.
3a. The cycle formed is treated as a region. In the region
        formed, the edges in the closed path represent the
        boundary edges of the region. The edges present inside
        the region are internal edges of that region. The
        corresponding vertices are called boundary vertices
        and internal vertices respectively.
3b. The boundary edges of the region are colored black and
        internal edges are colored gray.
```

```
4. If the temporary_growing_vector has no cycle then choose
         the next minimum weighted white colored adjacent edge
         satisfying the threshold and goto step-2.
4a. If there is no edge available or which satisfies the
         threshold then backtrack its parent and search for
         next white colored adjacent edge satisfying the
         threshold.
4b. If no parent exits then the temporary_growing_vector is
         stored in open_paths vector. Color all the edges in
         open_paths as black.
4c. else
4d. remove the last included edge in the temporary_growing_vector.
4e. choose the next white colored adjacent edge and goto
      step-2.
5. If all the vertices are not covered or all edges are not
         colored gray or black then goto step-1.
6. If all the vertices are covered in the region either as
         boundary or as internal vertices then it induces the
         initial segmentation for the given threshold.
7. If all the edges are colored either gray or black
         representing the internal or boundary edges then the
         segmentation is subjected for refinement.
```

The algorithm uses a color structure which labels the edges as given below:

- Initially all edges are in WHITE color
- A visited edges is in GRAY color
- An edge in BLACK color indicates that it is a part of the boundary of a region.

The BLACK colored edges are marked permanently so that they are not considered for refinement. Only WHITE and GRAY colored edges are subjected for refinement. The criteria that is imposed on every edge to form a segment is defined in Equation (1). The Equation (1) refers to the difference of the maximum and minimum vertex labels in the cycle formed. In this case, it is used as the difference of the maximum and minimum vertex labels in the temporary_growing_vector.

$$T = \frac{maxv - minv}{2} \tag{1}$$

The algorithm starts by randomly choosing a white colored edge. At the first execution, the edge chosen is included directly in the temporary_growing_vector. Since a cycle cannot be formed with one edge, line 4 is executed where the algorithm tries to choose a white colored edge adjacent to the previously chosen edge. The edge is selected based on the threshold criteria. If no minimum weighted, white colored adjacent edge is available then, the algorithm backtracks to its parent and searches for another minimum weighted white colored adjacent edge. If it finds, then the last included edge in the temporary_growing_vector is removed since the algorithm could not traverse from that edge and adds the newly selected edge into temporary_growing_vector.

Line 3 of the algorithm checks for any cycle in temporary_growing_vector. To check this, BFS algorithm is used. Each cycle is treated as one region. If cycle is formed, then the closed path is stored in cycles_formed vector. The edges of the closed path are colored black indicating that they are boundary vertices. These edges are not chosen for forming any other cycles. The edges present inside the region are colored gray. These edges may be used for forming cycles once the white colored edges are exhausted. This will help in avoiding self overlapping region formation that means that the traversal starts from an

internal edge and traverses to outside the region is termed as self overlapping. Self overlapping is avoided at the initial stage in order to get maximum number of non–overlapping regions but it is carried out in region refinement stage, if necessary.

During the execution of the algorithm, if it chooses a white colored edge outside any region and on its traversal, overlaps the existing region, then it is allowed because the internal edges of one region act like boundary edges of another region.

Another possibility during the traversal is that the temporary_growing_vector may not grow further because no further edge satisfies the criteria at any level (neither at the current edge nor at any of its parent edges), then the temporary_growing_vector stops traversing. By nature, Eulerian guarantees cycle formation but due to the threshold criteria, it may not form cycles all the cases. In such case, the temporary_growing_vector contains an open path and such paths are stored separately in open_paths vector.

In this way, the algorithm tries to traverse until it covers all vertices. This completes the first stage where, it induces an initial segmentation of image.

Refinement of Regions Formed

At this stage, all the edges are labeled to either gray or black. Refinement of black colored edges is not possible because they represent the boundaries of the regions already formed. The gray colored edges are subjected for refinement. The same procedure is used to form regions by choosing any randomly selected gray colored edge and for further traversals.

In this way, the algorithm tries to refine the segmentation for regions formation. Too much of refinement leads to over segmentation and no refinement leads to under segmentation. A moderate level of refinement is necessary. This is controlled by threshold selection.

5 Experimental Results

The proposed method is tested on standard Berkeley Image database. Two trivial synthetic images have been created and tested the algorithm on them. The results of the two synthetic images and the corresponding results are shown in Figure 3. The results presented in Figure 3 are the induced segmentations obtained before refinement process.

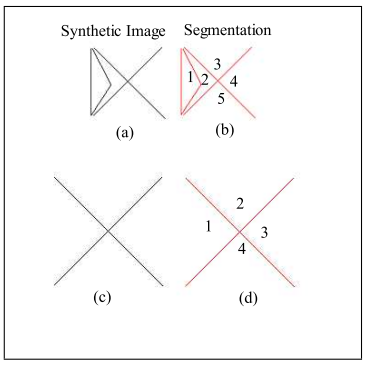


Figure 3: Segmentation results-I of synthetic images

In Figure 3, (a) and (c) are the two synthetic images created and the corresponding segmentations are shown in (b) and (d). These two synthetic images are created in such a way to study the behavior of the algorithm in open_paths case. As mentioned in the algorithm, the temporary_growing_vector stops traversing when there is no suitable edge satisfying the criteria. In such case the path is not closed and hence it is stored in open_paths vector. In Figure 3b, the segmentation result shows two open_paths

(cross lines). The two different ends of the two open_paths are adjacent to one region formed. Thus, segmentation output gives a visualization that there are two closed regions labeled 1 and 2; and three open regions labeled 3, 4 and 5. In Figure 3d, the segmentation output shows two open_paths for which no end is adjacent to any other region. The four open regions formed by the two open_paths are labeled 1,2,3 and 4 in Figure 3d. After applying the refinement procedure, the segments obtained are shown

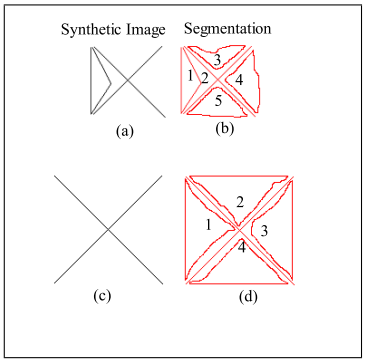


Figure 4: Segmentation results-II of synthetic images

in Figure 4. In Figure 4b, the segmentation output shows five regions labeled. Similarly in Figure 4d, there are 4 closed regions. The refinement process, in these cases, tried to get closed regions and in that process lead to over segmentation. This may be true in real images also. Hence, the refinement procedure is executed depending on the user's choice. The results of some real images taken from Berkeley Image database are shown in Figure 5. In Figure 5, the first and third columns represent the original image and second and fourth columns represent the segmentation result obtained.

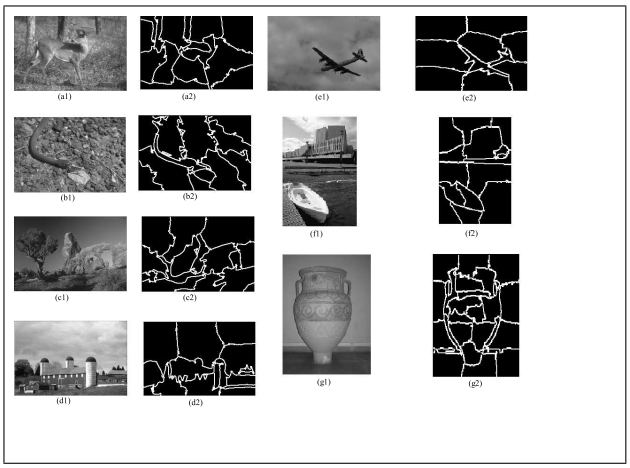


Figure 5: Segmentation results of some sample images in Berkeley Image Database

The proposed algorithm is executed on the 100 images in the database. The results of 50 images have been tabulated in Table 1. From the table, the following observations have been made. It is observed that those images having uniform background or average intensity range obtained best results; images having overlapping of objects or having complex structures, the statistical results are almost equivalent to the other existing methods chosen for comparison and for those images having high overlapping of objects or very dark images which cannot be visualized perfectly with the human eye, the proposed method could not segment the images and the statistical results revals that the F-measure for such images for the existing methods is better compared to the proposed method. The graphical representation of the results is shown in Figure 6 respectively.

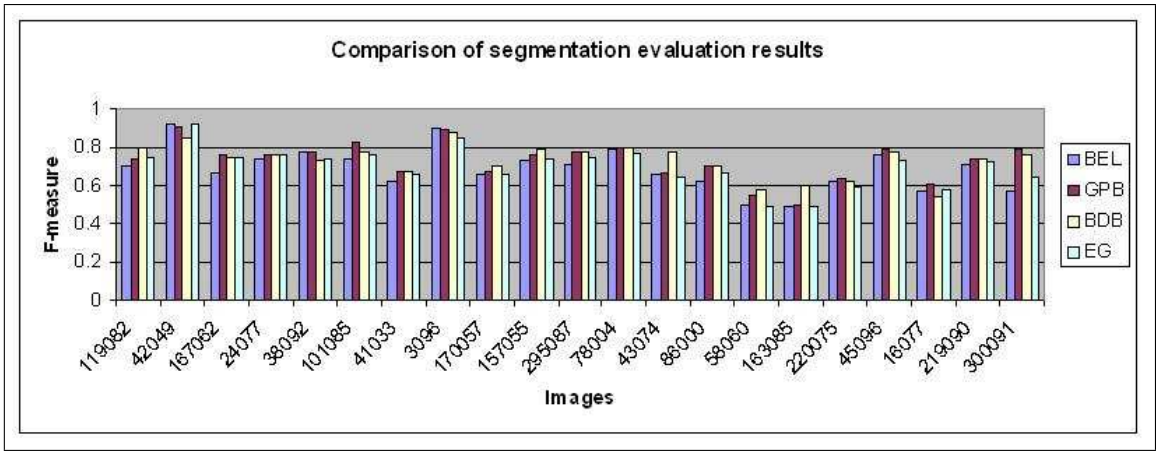


Figure 6: Comparison of segmentation evaluation results

Table 1: Comparison of segmentation evaluation results with other existing methods

Image Name	BEL	GPB	XREN	EG	Image Name	BEL	GPB	XREN	EG
119082	0.7	0.74	0.8	0.75	89072	0.68	0.71	0.71	0.69
42049	0.92	0.91	0.85	0.92	126007	0.72	0.78	0.76	0.75
167062	0.67	0.76	0.75	0.75	296007	0.66	0.69	0.69	0.65
24077	0.74	0.76	0.76	0.76	175032	0.49	0.62	0.63	0.6
38092	0.78	0.78	0.73	0.74	103070	0.68	0.68	0.62	0.65
101085	0.74	0.83	0.78	0.76	285079	0.71	0.72	0.71	0.69
41033	0.62	0.68	0.68	0.66	167083	0.61	0.75	0.75	0.7
291600	0.57	0.61	0.59	0.6	271035	0.73	0.73	0.71	0.71
130026	0.52	0.51	0.47	0.52	12084	0.48	0.52	0.49	0.5
241004	0.85	0.81	0.81	0.85	69015	0.79	0.82	0.79	0.75
147091	0.71	0.77	0.75	0.75	58060	0.5	0.55	0.58	0.49
189080	0.78	0.8	0.77	0.79	163085	0.49	0.5	0.6	0.49
14037	0.65	0.7	0.65	0.71	220075	0.62	0.64	0.62	0.59
62096	0.79	0.79	0.78	0.78	45096	0.76	0.79	0.78	0.73
227092	0.75	0.88	0.85	0.88	16077	0.57	0.61	0.54	0.58
253027	0.63	0.65	0.69	0.68	219090	0.71	0.74	0.74	0.72
229036	0.67	0.76	0.72	0.75	300091	0.57	0.79	0.76	0.65
3096	0.9	0.89	0.88	0.85	156065	0.66	0.67	0.64	0.63
170057	0.66	0.68	0.7	0.66	76053	0.61	0.61	0.62	0.59
157055	0.73	0.76	0.79	0.74	304034	0.47	0.49	0.47	0.41
295087	0.71	0.78	0.78	0.75	86016	0.39	0.52	0.42	0.48
78004	0.79	0.8	0.8	0.77	8023	0.41	0.5	0.42	0.4
43074	0.66	0.67	0.78	0.65	108082	0.43	0.46	0.47	0.43
86000	0.62	0.7	0.7	0.67	69040	0.5	0.55	0.57	0.52

6 Conclusion

In this paper, a novel algorithm for segmenting an image into different regions using Euler graphs has been proposed. The algorithm starts by randomly choosing an edge and tries to form closed regions. In cases, open paths are formed. The color look up table is used for edges to trace their transition. A white color indicates unvisited edge, a gray color indicates visited and may go for refinement and black color indicates visited and marked permanently for no refinement since it is already a part of a region boundary. The procedures discussed run in polynomial time. The MST and cycles method performs better compared to Euler Graph method in terms of precision, recall and F measures.

Bibliography

[1] J.A. Bondy and U.S.R. Murty. *Graph Theory with Applications, Fifth printing*. Elsevier Science Publishing Co., Inc., 52, Vanderbilt Avenue, New York, 1982.

[2] H.Greenspan C.Carson, S.Belongie and J.Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[3] C.R.Brice and C.Fennema. Scene analysis using regions. *Artificial Intelligence*, 1(3–4):205–226, 1970.

[4] C.T.Zahn. Graph theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computation*, 20:68–86, 1971.

[5] E.L.Schwartz. Spatial mapping in the primate sensory projection: Analytic structure and relevance to perception. *Biological Cybernetics*, 25(4):181–194, 1977.

[6] Euler. Solutio problematis ad geometriam situs pertinentis comment. *Academiae Sci. I. Petropolitanae*, 8:128–140, 1736.

[7] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[8] K.S. Fu and J.K. Mui. A survey of image segmentation. *Pattern Recognition*, 13:3–16, 1981.

[9] R.M. Haralick and L.G. Shapiro. Survey, image segmentation techniques. *Computer Vision, Graphics and Image Processing*, 29:100–132, 1985.

[10] I.Jermyn and H.Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):1075–1088, 2001.

[11] L.Vincent and P.Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.

[12] R.Velthuizen L.Hall D.Goldgof L.Clarke M.Clark and M.Silbiger. Mri segmentation using fuzzy clustering techniques. *IEEE Engineering in Medicine and Biology Magazine*, 13(5):730–742, 1994.

[13] O.Monga. An optimal region growing algorithm for image segmentation. *PRAI*, 1(4):351–375, 1987.

[14] N. Otsu. A threshold selection method from grey level histograms. *IEEE Transactions on System, Man and Cybernetics*, 9:62–66, 1979.

[15] N.R. Pal and S.K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26:1277–1294, 1993.

[16] P.Parent and S.W.Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):823–839, 1989.

[17] R.Adams and L.Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.

[18] R.Urquhart. Graph theoretical clustering based on limited neighborhood sets. *Pattern Recognition*, 15(3):173–187, 1982.

[19] P.W.Ong. R.Wallace and E.Schwartz. Space variant image processing. *International Journal of Computer Vision*, 13(1):71–90, 1994.

[20] S.C.Zhu and A.L.Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, 1996.

[21] M.Pelillo S.Dickinson and Ramin Zabih. Introduction to the special section on graph algorithms in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1049–1052, 2001.

[22] J. Shi. and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[23] S.L.Horowitz and T.Pavlidis. A graph-theoretic approach to picture processing. *JACM*, 7(2):282–291, 1976.

[24] S.L.Horowitz and T.Pavlidis. Picture segmentation by a tree traversal algorithm. *JACM*, 23(2):368–388, 1976.

[25] S.Ullman and A.Shaashua. Structural saliency: The detection of globally salients structures using a locally connected network. Technical report, Cambridge, MA, USA, 1988.

[26] U.Montanari. On the optimal detection of curves in noisy pictures. *Communications. ACM*, 14(5):335–345, 1971.

[27] A.R. Weeks and G.E. Hague. Color segmentation in the hsi color space using the k-means algorithm. In *SPIE, p. 143-154, Nonlinear Image Processing VIII, Edward R. Dougherty; Jaakko T. Astola; Eds*, Volume 3026, pages 143–154, 1997.

[28] Y.Haxhimusa and W.Kropatsch. Hierarchy of partitions with dual graph contraction. *Lecture Notes in Computer Science*, 2781:338–345, 2003.

[29] Y.Haxhimusa and W.Kropatsch. Segmentation graph hierarchies. In *Proceedings of Structural, Syntactic, and Statistical Pattern Recognition*, Volume 3138, pages 343–351. LNCS, 2004.

[30] Y.Weiss. Segmentation using eigenvectors: A unifying view. In *Proceedings of the International Conference on Computer Vision*, Volume 2, pages 975–982, 1989.

[31] Z.Wu and R.Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.

[32] Dickinson, S., Pelillo, M. and Zabih, R. Introduction to the special section on graph algorithms in computer vision *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1049–1052, 2001.

[33] D. Martin and C. Fowlkes and D. Tal and J. Malik A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics *Proc. 8th Int'l Conf. Computer Vision*, Volume 2, pages 416–423, 2001.

[34] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

[35] Piotr Dollar, Zhuowen Tu, and Serge Belongie Supervised Learning of Edges and Object Boundaries *Proc. IEEE Computer Vision and Pattern Recognition, CVPR*, June, 2006.

[36] Michael Maire, Pablo Arbelaez, Charless Fowlkes and Jitendra Malik Using Contours to Detect and Localize Junctions in Natural Images *Proc. IEEE Computer Vision and Pattern Recognition, CVPR*, 2008.

[37] Xiaofeng Ren Multi-Scale Improves Boundary Detection in Natural Images *Proc. ECCV Conference*, 2008.

[38] Marius George Linguraru, Miguel *Á* ngel Gonz *á* lez Ballester, Nicholas Ayache Deformable Atlases for the Segmentation of Internal Brain Nuclei in Magnetic Resonance Imaging *International Journal of Computers, Communications & Control*, Volume 2, No. 1, pages 26–36, 2007.

T. N. Janakiraman (born on February 14, 1953) received his Ph.D. in graph theory and its applications from Madras University, India in 1991. He did his P.D.F. from the same university in graph theory and its applications in 1994. He is currently associated with Department of Mathematics, National Institute of Technology, Trichy, India. He has two sponsored research projects to his credit and published around 40 papers in refereed international journals. His main research interests are in graph theory and its applications in digital image processing, wireless ad hoc networks and digital video processing.

P.V.S.S.R. Chandra Mouli (born on May 21, 1976) received his Ph.D. from National Institue of Trichy in 2010. He is currently working as Associate Professor in School of Computing Science and Engineering, VIT University, Vellore. His research interests include Image Segmentation, Pattern Classification and Wireless Ad–hoc networks. He has published 10 refereed research papers in various international journals and conferences. He was co–investigator to research project sponsored by Defence Research Development Organization (DRDO), New Delhi and also worked as a research fellow in another research project sponsored by DRDO, New Delhi, India. He is a life member of ISTE, and also member of CSI. He is reviewer for many international journals.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 3, pp. 325-335

Consensus Problem of Second-order Dynamic Agents with Heterogeneous Input and Communication Delays

C.-L. Liu, F. Liu

Cheng-Lin Liu, Fei Liu
Jiangnan University
Institute of Automation
Wuxi 214122, Jiangsu, People's Republic of China
E-mail: liucl@jiangnan.edu.cn, fliu@jiangnan.edu.cn

Abstract: Consensus problem of second-order multi-agent systems with velocity damping term in agent's dynamics is investigated. Based on frequency-domain analysis, decentralized consensus condition, which depends on the input delays, is obtained for the system based on undirected and symmetric graph with heterogeneous input delays. For the system based on directed graph with both heterogeneous input delays and communication delays, decentralized consensus condition, which is dependent on the input delays but independent on the communication delays, is also obtained. Simulations illustrate the correctness of the results.
Keywords: coordination control, consensus, second-order multi-agent systems, communication delay, input delay

1 Introduction

In the last decade, distributed coordination of multiple autonomous agents has attracted more and more attention from various research communities for its broad application including automated highway systems, air traffic control, congestion control in Internet, etc.

Consensus problem, which is one of the most fundamental and important issues in coordination control of multi-agent systems, requires that the outputs of several spatially distributed agents reach a common value without recurse to a central controller. For the first-order multi-agent systems with agents' dynamics modeled by single integrators and second-order multi-agent systems with agents' dynamics modeled by double integrators, consensus algorithms have been proposed to solve the consensus problem, and sufficient conditions have been obtained for the system converging to the consensus with static or switched interconnection topology [1–4].

Recently, more and more attention has been paid on the delay effect on consensus convergence of multi-agent systems. Generally speaking, two kinds of time delays cannot be negligible in the multi-agent systems. One is communication delay, which is related to the information transmission between neighboring agents. The other is input delay, which is related to the processing and connecting time for the packets arriving at each agent [5].

Consensus problem under communication delays has been extensively studied for the first-order multi-agent systems based on different analysis methods, such as Lyapunov functions analysis [6, 7], frequency-domain analysis [2, 8], method based on delayed and hierarchical graphs [9, 10], method based on difference of maximum value and minimum value [11, 12], etc. However, consensus analysis of second-order multi-agent systems with communication delay is much more difficult, and many existing results are mostly on the synchronous consensus algorithm [13–15], in which self-delays equaling to the corresponding communication delays are introduced for each agent in the coordination control part. Compared with the first-order multi-agent systems, the consensus algorithm without any self-delay, which is called asynchronous consensus algorithm, has not been studied extensively for the second-order multi-agent systems. Using small-μ stability theorem, Yang et al. [16] obtained the frequency-domain

consensus conditions for the second-order multi-agent systems with time-varying communication delays. Based on frequency-domain analysis [17] and Lyapunov-Krasovskii functional method [18], Spong et al. proved that, by choosing proper consensus protocol and control parameters, the second-order multi-agent systems with heterogeneous communication delays can converge to a stationary consensus without any relationship to the delays. Using the properties of nonnegative matrices, Lin and Jia [19] obtained delay-independent sufficient conditions for the second-order discrete-time multi-agent systems with heterogeneous communication delays converging to the stationary consensus under dynamically changing topologies.

To our knowledge, however, the consensus problem under input delays has not attracted much more attention. In some reports, the identical communication delay introduced in the synchronous consensus algorithm can be treated as the identical input delay [2, 14, 15]. Using frequency-domain analysis method, Tian and Liu [5] considered the consensus problem of the first-order multi-agent systems with heterogeneous input delays based on undirected graphs, and obtained the decentralized consensus criterion depending on the input delays. Moreover, the decentralized consensus condition, which depends only on the input delays, is also obtained for the first-order multi-agent systems with both heterogeneous communication delays and input delays based on the digraph in [5]. In [20], Tian and Liu investigated the leader-following consensus problem of the second-order multi-agent systems with heterogeneous input delays and symmetric coupling weights, and the decentralized consensus condition with some prerequisites is obtained for the system converging to the states of the dynamic leader asymptotically. Furthermore, the robustness of the symmetric system with asymmetric weight perturbation is also investigated in [20], and a bound of the largest singular value of the perturbation matrix is obtained as the robust consensus condition.

In this paper, we consider the consensus problem of second-order multi-agent systems with velocity damping term in the agent's dynamics, and analyze the consensus conditions for the system with heterogeneous delays converging to the stationary consensus. Firstly, we investigate the consensus problem for the system based on undirected and symmetric graph with heterogeneous input delays, and a decentralized consensus condition, which is delay-dependent, is obtained by using some early results for the Internet congestion control with heterogeneous communication delays [21]. Then, we study the consensus problem for the system based on general directed graph with both heterogeneous input delays and communication delays by using Greshgorin disc theorem, and another decentralized consensus condition, which depends on the input delays only, is also obtained. This consensus condition is more conservative than the former for the existence of heterogeneous communication delays and the asymmetry of coupling weights, but it can be applied to the systems based on directed graph with asymmetric weights.

2 Preliminaries on Graph Theory

A weighted directed graph (digraph) $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ of order n consists of a set of vertices $\mathcal{V} = \{1, ..., n\}$, a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ and a weighted adjacency matrix $\mathcal{A} = [a_{ij}] \in R^{n \times n}$ with nonnegative adjacency elements a_{ij}. The node indexes belong to a finite index set $\mathcal{I} = \{1, 2, ..., n\}$. An edge of the weighted diagraph $\mathcal{G}$ is denoted by $e_{ij} = (i, j) \in \mathcal{E}$, i.e., e_{ij} is a directed edge from i to j. We assume that the adjacency elements associated with the edges of the digraph are positive, i.e., $a_{ij} > 0 \Leftrightarrow e_{ij} \in \mathcal{E}$. Moreover, we assume $a_{ii} = 0$ for all $i \in \mathcal{I}$. The set of neighbors of node i is denoted by $N_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. In the digraph $\mathcal{G}$, if $(i, j) \in \mathcal{E} \leftrightarrow (j, i) \in \mathcal{E}$, we usually say $\mathcal{G}$ is undirected graph or bidirectional graph. The out-degree of node i is defined as: $deg_{out}(i) = \sum_{j=1}^{n} a_{ij}$. Let $\mathcal{D}$ be the diagonal matrix with the out-degree of each node along the diagonal and call it the degree matrix of $\mathcal{G}$. The Laplacian matrix of the weighted digraph is defined as $L = \mathcal{D} - \mathcal{A}$.

If there is a path in $\mathcal{G}$ from one node i to another node j, then j is said to be *reachable* from i. If not, then j is said to be not reachable from i. If a node is reachable from every other node in the digraph, then

we say it *globally reachable*. A digraph is *strongly connected* if every node in the digraph is globally reachable. An undirected graph is connected if it contains a globally reachable node.

3 Problem Formulation

In a multi-agent system composed of n agents, each agent can be regarded as a node in a digraph, and information flow between neighboring agents can be considered as directed paths between the nodes in the digraph. Thus, the interconnection topology of multi-agent systems can be described as a diagraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$. In this paper, we just consider static topology $\mathcal{G}$, i.e., the connection of the nodes in the diagraph $\mathcal{G}$ does not change with time.

Consider the second-order dynamic agents modeled by

$$\begin{aligned} \dot{x}_i &= v_i, \\ m_i \dot{v}_i &= F_i,\ i \in \mathcal{I}, \end{aligned} \tag{1}$$

where $x_i \in R$ and $v_i \in R$ are the position and the velocity of the agent i respectively, $m_i > 0$ is the mass of the agent i, and F_i is the total force on the agent i. Analogous to [22], the total force F_i in the second-order model (1) consists of two components:

$$F_i = -\alpha_i v_i + u_i,$$

where u_i is the external control input, and $-\alpha_i v_i$ with $\alpha_i > 0$ denotes the velocity damping term caused by the resistance, e.g., the friction. Then, the second-order model (1) becomes

$$\begin{aligned} \dot{x}_i &= v_i, \\ m_i \dot{v}_i &= -\alpha_i v_i + u_i,\ i \in \mathcal{I}. \end{aligned} \tag{2}$$

With non-negligible input delays for the external control, the agents (2) become

$$\begin{aligned} \dot{x}_i(t) &= v_i(t), \\ m_i \dot{v}_i(t) &= -\alpha_i v_i(t) + u_i(t - T_i),\ i \in \mathcal{I}, \end{aligned} \tag{3}$$

where $T_i > 0$ is the input delay of the agent i. For the system (3), we take a consensus protocol based on the agents' position states as follows

$$u_i = \kappa_i \sum_{j \in N_i} a_{ij}(x_j - x_i), \tag{4}$$

where $\kappa_i > 0$, N_i denotes the neighbors of agent i, and $a_{ij} > 0$ is the adjacency element of $\mathcal{A}$ in the digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$. Under the communication delays, the protocol (4) becomes

$$u_i(t) = \kappa_i \sum_{j \in N_i} a_{ij}(x_j(t - \tau_{ij}) - x_i(t)), \tag{5}$$

where τ_{ij} is the communication delay from agent j to agent i.

With the protocol (5), the closed-loop form of the system (3) is

$$\begin{aligned} \dot{x}_i(t) &= v_i(t), \\ m_i \dot{v}_i(t) &= -\alpha_i v_i(t) + \kappa_i \sum_{j \in N_i} a_{ij}(x_j(t - T_i - \tau_{ij}) - x_i(t - T_i)),\ i \in \mathcal{I}. \end{aligned} \tag{6}$$

Remark 1. In [20], Tian and Liu has studied the leader-following consensus problem of the second-order multi-agent systems with heterogeneous input delays under double-consensus algorithm, and obtained the consensus conditions for the system with symmetric and asymmetric weights respectively. Different from [20], we consider the stationary consensus of the second-order dynamic agents (6) with velocity damping term, and analyze the consensus conditions for the system with heterogeneous communication delays and input delays.

4 Consensus Criterion

4.1 Consensus under Heterogeneous Input Delays

In this section, we investigate the consensus problem of multi-agent systems (6) just with heterogeneous input delays only as follows

$$\begin{aligned} \dot{x}_i(t) &= v_i(t), \\ m_i\dot{v}_i(t) &= -\alpha_i v_i(t) + \kappa_i \sum_{j\in N_i} a_{ij}(x_j(t-T_i) - x_i(t-T_i)),\ i \in \mathcal{I}. \end{aligned} \tag{7}$$

Firstly, we give an assumption on the velocity damping coefficient α_i, the mass m_i and input delay T_i in the following.

Assumption 2. $(T_i\frac{\alpha_i}{m_i} - T_j\frac{\alpha_j}{m_j})(T_i - T_j) \le 0,\ \forall i, j \in \mathcal{I}, i \ne j.$

Now, we present some sufficient conditions for second-order multi-agent systems with heterogeneous input delays.

Theorem 3. *Consider the network of n dynamic agents (7) with a static interconnection topology $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ that is undirected (or bidirectional) and connected, and the topology graph has symmetric weights, i.e., $a_{ij} = a_{ji}$. Then, under Assumption 2, all the agents in system (7) asymptotically converge to a stationary consensus, i.e., $\lim_{t\to\infty} x_i(t) = c,\ \lim_{t\to} v_i(t) = 0,\ \forall i \in \mathcal{I}$, where c is a constant, if*

$$\sum_{j\in N_i} a_{ij} < \frac{m_i}{2\kappa_i (G_i^M)^{-1}}, \forall i \in \mathcal{I}, \tag{8}$$

where G_i^M is the gain margin of the transfer function $W_i(s) = \frac{e^{-sT_i}}{s^2 + \frac{\alpha_i}{m_i}s}$.

Before proving Theorem 3, we list two useful lemmas as follows.

Lemma 4. *[23] Let $Q \in C^{n\times n}$, $Q = Q^\star \ge 0$ and $T = \mathrm{diag}\{t_i, t_i \in C\}$. Then*

$$\lambda(QT) \in \rho(Q)\mathrm{Co}(0 \cup \{t_i\}),$$

where $\lambda(\cdot)$ denotes matrix eigenvalue, $\rho(\cdot)$ denotes the matrix spectral radius, and $\mathrm{Co}(\cdot)$ denotes the convex hull.

Based on Remark 4 and Claim 1 in [21], we obtain the following lemma.

Lemma 5. *Suppose that Assumption 2 holds for the frequency response of a family of systems described by*

$$G_i(j\omega) = \frac{G_i^M}{s^2 + j\frac{\alpha_i}{m_i}\omega} e^{-jT_i\omega},\ \ i \in \mathcal{I},$$

where G_i^M is the gain margin of the transfer function $W_i(s) = \frac{e^{-sT_i}}{s^2 + \frac{\alpha_i}{m_i}s}$. Then, $\gamma\mathrm{Co}(0 \cup \{G_i(j\omega), i \in \mathcal{I}\})$ does not contain the point $(-1, j0)$ for any given real number $\gamma \in [0, 1)$ and any $\omega \in (-\infty, \infty)$.

Now, we give the proof of Theorem 3.

The system (7) is rewritten as follows

$$\begin{aligned} \dot{x}_i(t) &= v_i(t), \\ \dot{v}_i(t) &= -\bar{\alpha}_i v_i(t) + \bar{\kappa}_i \sum_{j\in N_i} a_{ij}(x_j(t-T_i) - x_i(t-T_i)),\ i \in \mathcal{I}, \end{aligned} \tag{9}$$

where $\bar{\alpha}_i = \frac{\alpha_i}{m_i}$ and $\bar{\kappa}_i = \frac{\kappa_i}{m_i}$. Taking the Laplace transform of the system (9), we obtain the characteristic equation of the system (9) about $x(t) = [x_1(t), \cdots, x_n(t)]^T$ as follows

$$\det(\mathrm{diag}\{s^2 + \bar{\alpha}_i s, i \in \mathcal{I}\} + \mathrm{diag}\{\bar{\kappa}_i e^{-T_i s}, i \in \mathcal{I}\}L) = 0.$$

Define $D(s) = \det(\mathrm{diag}\{s^2 + \bar{\alpha}_i s, i \in \mathcal{I}\} + \mathrm{diag}\{\bar{\kappa}_i e^{-T_i s}, i \in \mathcal{I}\}L)$, and we will prove that all the zeros of $D(s)$ are on the open left half complex plane or $s = 0$ in the following.

When $s = 0$, $D(0) = \det(\mathrm{diag}\{0^2 + \bar{\alpha}_i 0, i \in \mathcal{I}\} + \mathrm{diag}\{\bar{\kappa}_i e^{-T_i 0}, i \in \mathcal{I}\}L) = \det(\mathrm{diag}\{\bar{\kappa}_i, i \in \mathcal{I}\})\det(L)$, Because the interconnection graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ is connected, 0 is a simple eigenvalue of L [24], i.e., $\mathrm{rank}(L) = n - 1$. Hence, $D(s)$ has only one zero at $s = 0$.

When $s \neq 0$, define $F(s) = \det(I + \mathrm{diag}\{\frac{\bar{\kappa}_i}{s^2 + \bar{\alpha}_i s} e^{-T_i s}, i \in \mathcal{I}\}L)$. We will prove that the zeros of $F(s)$ lie on the open left half complex plane. According to the General Nyquist stability criterion [25], the zeros of $F(s)$ are on the open left half complex plane, if $\lambda(\mathrm{diag}\{\frac{\bar{\kappa}_i}{(j\omega)^2 + j\omega\bar{\alpha}_i} e^{-j\omega T_i}, i \in \mathcal{I}\}L)$ does not enclose the point $(-1, j0)$ for $\omega \in R$.

For the symmetric weights ($a_{ij} = a_{ji}$), we get $L = L^T \geq 0$ according to the definition of the Laplacian matrix. Based on Lemma 4, we get

$$\begin{aligned}
& \lambda(\mathrm{diag}\{\frac{\bar{\kappa}_i}{-\omega^2 + j\bar{\alpha}_i\omega} e^{-jT_i\omega}\}L) \\
= \; & \lambda(\mathrm{diag}\{\frac{G_i^M}{-\omega^2 + j\bar{\alpha}_i\omega} e^{-jT_i\omega} \mathrm{diag}\{\sqrt{\bar{\kappa}_i (G_i^M)^{-1}}\} L \mathrm{diag}\{\sqrt{\bar{\kappa}_i (G_i^M)^{-1}}\}) \\
\in \; & \rho(\mathrm{diag}\{\sqrt{\bar{\kappa}_i (G_i^M)^{-1}}\} L \mathrm{diag}\{\sqrt{\bar{\kappa}_i (G_i^M)^{-1}}\}) Co(0 \cup \frac{G_i^M}{-\omega^2 + j\bar{\alpha}_i\omega} e^{-jT_i\omega}).
\end{aligned}$$

Since the spectral radius of any matrix is bounded by its largest absolute row sum, it follows from the condition (8) that

$$\begin{aligned}
\rho(\mathrm{diag}\{\sqrt{\bar{\kappa}_i (G_i^M)^{-1}}\} L \mathrm{diag}\{\sqrt{\bar{\kappa}_i (G_i^M)^{-1}}\}) &= \rho(\mathrm{diag}\{\bar{\kappa}_i (G_i^M)^{-1}\} L) \\
&\leq \max_{i \in \mathcal{I}} \bar{\kappa}_i (G_i^M)^{-1} (2 \sum_{j \in N_i} a_{ij}) \\
&< 1.
\end{aligned}$$

Therefore, from Lemma 5, we obtain that

$$(-1, 0) \notin \rho(\mathrm{diag}\{\sqrt{\bar{\kappa}_i (G_i^M)^{-1}}\} L \mathrm{diag}\{\sqrt{\bar{\kappa}_i (G_i^M)^{-1}}\}) Co(0 \cup \frac{G_i^M}{-\omega^2 + j\bar{\alpha}_i\omega} e^{-jT_i\omega}),$$

i.e., $\lambda(\mathrm{diag}\{\frac{\bar{\kappa}_i}{(j\omega)^2 + j\omega\bar{\alpha}_i} e^{-j\omega T_i}, i \in \mathcal{I}\}L)$ does not enclose the point $(-1, j0)$ for $\omega \in R$, which implies that the zeros of $F(s)$ are all on the open left half complex plane.

Now, we have proved that $D(s)$ has its zeros on the open left half complex plane except for one zero at $s = 0$. Thus, the state $x_i(t)$ of the system (7) converges to a steady state, i.e., $\lim_{t \to \infty} x_i(t) = x_i^*, i \in \mathcal{I}$, and $\lim_{t \to \infty} v_i(t) = 0, \forall i \in \mathcal{I}$ holds for (7). It is obtained from (7) that $L[x_1^*, \cdots, x_n^*]^T = 0$. Since $\mathrm{rank}(L) = n - 1$ and $L[1, \cdots, 1]^T = 0$ from the definition of the Laplacian matrix L, the roots of $Lx^* = 0$ can be expressed as $x^* = c[1, \cdots, 1]^T$, where c is a constant. Theorem 3 is proved. □

Remark 6. Obviously, the consensus condition (8) in Theorem 3 depends strictly on the Assumption 2 and the symmetry of the coupling weights between agents.

4.2 Consensus under Heterogeneous Input and Communication Delays

In multi-agent systems, the interconnection topology composed of dynamic agents is usually asymmetric, and the communication delays caused by information transmission always exist between neighboring agents. Thus, the Lemma 4 and Lemma 5 which play important roles in the proof of Theorem 3 cannot be applied in these cases. In this section, we will analyze the consensus of the second-order dynamic agents (6) with both heterogeneous input delays and communication delays under general directed interconnection topology.

Theorem 7. *Consider the network of n dynamic agents (6) with a static interconnection topology* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ *that has a globally reachable node. If*

$$\sum_{j\in N_i} a_{ij} < \frac{\alpha_i^2}{2\kappa_i(m_i+\alpha_i T_i)},\ \forall i \in \mathcal{I}, \tag{10}$$

all the agents in the system (6) converge to a stationary consensus asymptotically.

Proof: Firstly, rewrite the system (6) as

$$\begin{aligned} \dot{x}_i(t) &= v_i(t), \\ \dot{v}_i(t) &= -\bar{\alpha}_i v_i(t) + \bar{\kappa}_i \sum_{j\in N_i} a_{ij}(x_j(t-T_i-\tau_{ij}) - x_i(t-T_i)),\ i \in \mathcal{I}, \end{aligned} \tag{11}$$

where $\bar{\alpha}_i = \frac{\alpha_i}{m_i}$ and $\bar{\kappa}_i = \frac{\kappa_i}{m_i}$. Taking the Laplace transform of the system (12), we obtain that the characteristic equation of the system (12) about $x(t) = [x_1(t), \cdots, x_n(t)]^T$ is

$$\det(\mathrm{diag}\{s^2 + \bar{\alpha}_i s, i \in \mathcal{I}\} + \mathrm{diag}\{\bar{\kappa}_i e^{-T_i s}, i \in \mathcal{I}\} L(s)) = 0,$$

where the $n \times n$ matrix $L(s) = \{l_{ij}(s)\}$ is defined by

$$l_{ij}(s) = \begin{cases} -a_{ij}e^{-\tau_{ij}s}, & j \in N_i; \\ \sum_{j\in N_i} a_{ij}, & j = i; \\ 0, & \text{otherwise}, \end{cases}$$

and $L(0) = L$, which is the Laplacian matrix.

Define $\tilde{D}(s) = \det(\mathrm{diag}\{s^2 + \bar{\alpha}_i s, i \in \mathcal{I}\} + \mathrm{diag}\{\bar{\kappa}_i e^{-T_i s}, i \in \mathcal{I}\} L(s))$, and we will prove that all the zeros of $\tilde{D}(s)$ are on the open left half plane or $s = 0$ in the following.

When $s = 0$, $\tilde{D}(0) = \det(\mathrm{diag}\{0^2 + \bar{\alpha}_i 0, i \in \mathcal{I}\} + \mathrm{diag}\{\bar{\kappa}_i e^{-T_i 0}, i \in \mathcal{I}\} L(0)) = \det(\mathrm{diag}\{\bar{\kappa}_i, i \in \mathcal{I}\} \det(L)$, Because the interconnection topology $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ has a globally reachable node, 0 is a simple eigenvalue of L [24]. Hence, $\tilde{D}(0) = 0$, i.e., $\tilde{D}(s)$ has only one zero at $s = 0$.

When $s \neq 0$, define $\tilde{F}(s) = \det(I + \mathrm{diag}\{\frac{\bar{\kappa}_i}{s^2+\bar{\alpha}_i s} e^{-T_i s}, i \in \mathcal{I}\} L(s))$. We will prove the zeros of $\tilde{F}(s)$ lie on the open left half complex plane. According to the General Nyquist stability criterion [25], the zeros of $\tilde{F}(s)$ are on the open left half complex plane, if $\lambda(\mathrm{diag}\{\frac{\bar{\kappa}_i}{(j\omega)^2+j\omega\bar{\alpha}_i} e^{-j\omega T_i}, i \in \mathcal{I}\} L(j\omega))$ does not enclose the point $(-1, j0)$ for $\omega \in R$. Based on the Greshgorin's disc theorem,

$$\begin{aligned} & \lambda(\mathrm{diag}\{\frac{\bar{\kappa}_i}{-\omega^2 + j\bar{\alpha}_i\omega} e^{-jT_i\omega}, i \in \mathcal{I}\} L(s)) \\ \in\ & \bigcup_{i\in\mathcal{I}} \{\zeta :\ \zeta \in C, |\zeta - \frac{\bar{\kappa}_i(\sum_{j\in N_i} a_{ij})}{-\omega^2 + j\bar{\alpha}_i\omega} e^{-jT_i\omega}| \leq \sum_{j\in N_i} |\frac{\bar{\kappa}_i a_{ij}}{-\omega^2 + j\bar{\alpha}_i\omega} e^{-j(T_i+\tau_{ij})\omega}|\} \\ =\ & \bigcup_{i\in\mathcal{I}} \{\zeta :\ \zeta \in C, |\zeta - \frac{\bar{\kappa}_i g_i}{-\omega^2 + j\bar{\alpha}_i\omega} e^{-jT_i\omega}| \leq |\frac{\bar{\kappa}_i g_i}{-\omega^2 + j\bar{\alpha}_i\omega} e^{-jT_i\omega}|\} \end{aligned}$$

holds for $\omega \in R$, where $g_i = \sum_{j\in N_i} a_{ij}$.

Then, $\lambda(\mathrm{diag}\{\frac{\bar{\kappa}_i}{(\mathrm{j}\omega)^2+\mathrm{j}\omega\bar{\alpha}_i}\mathrm{e}^{-\mathrm{j}\omega T_i}, i\in\mathcal{I}\}L(\mathrm{j}\omega))$ does not enclose the point $(-1,\mathrm{j}0)$ for $\omega\in R$ as long as the point $(-a,\mathrm{j}0)$ with $a\geq 1$ does not in the disc $\{\zeta:\ \zeta\in C, |\zeta-\frac{\bar{\kappa}_i g_i}{-\omega^2+\mathrm{j}\bar{\alpha}_i\omega}\mathrm{e}^{-\mathrm{j}T_i\omega}| \leq |\frac{\bar{\kappa}_i g_i}{-\omega^2+\mathrm{j}\bar{\alpha}_i\omega}\mathrm{e}^{-\mathrm{j}T_i\omega}|\}$ for all $\omega\in R$, i.e., $|-a+\mathrm{j}0-\frac{\bar{\kappa}_i g_i}{-\omega^2+\mathrm{j}\bar{\alpha}_i\omega}\mathrm{e}^{-\mathrm{j}T_i\omega}| > |\frac{\bar{\kappa}_i g_i}{-\omega^2+\mathrm{j}\bar{\alpha}_i\omega}\mathrm{e}^{-\mathrm{j}T_i\omega}|$ holds for all $\omega\in R$ when $a\geq 1$.

By calculating, we obtain

$$|-a+\mathrm{j}0-\frac{\bar{\kappa}_i g_i}{-\omega^2+\mathrm{j}\bar{\alpha}_i\omega}\mathrm{e}^{-\mathrm{j}T_i\omega}|^2-|\frac{\bar{\kappa}_i g_i}{-\omega^2+\mathrm{j}\bar{\alpha}_i\omega}\mathrm{e}^{-\mathrm{j}T_i\omega}|^2 = a(a-2\bar{\kappa}_i g_i\frac{\cos(\omega T_i)+\bar{\alpha}_i\frac{\sin(\omega T_i)}{\omega}}{\omega^2+\bar{\alpha}_i^2}).$$

Because $\cos(\omega T_i)\leq 1$ and $\frac{\sin(\omega T_i)}{\omega}\leq T_i$ hold for $\omega\in R$, it follows from (11) that

$$2\bar{\kappa}_i g_i\frac{\cos(\omega T_i)+\bar{\alpha}_i\frac{\sin(\omega T_i)}{\omega}}{\omega^2+\bar{\alpha}_i^2} \leq \frac{2\bar{\kappa}_i g_i(1+\bar{\alpha}_i T_i)}{\bar{\alpha}_i^2} < 1.$$

Hence, $|-a+\mathrm{j}0-\frac{\bar{\kappa}_i g_i}{-\omega^2+\mathrm{j}\bar{\alpha}_i\omega}\mathrm{e}^{-\mathrm{j}T_i\omega}| > |\frac{\bar{\kappa}_i g_i}{-\omega^2+\mathrm{j}\bar{\alpha}_i\omega}\mathrm{e}^{-\mathrm{j}T_i\omega}|$ holds for all $\omega\in R$ when $a\geq 1$.

Now, we have proved that $\tilde{D}(s)$ has its zeros on the open left half complex plane except for a zero at $s=0$. Thus, the state $x_i(t)$ of the system (6) converges to a steady state, i.e., $\lim_{t\to\infty} x_i(t) = x_i^*, i\in\mathcal{I}$, and $\lim_{t\to\infty} v_i(t) = 0, \forall i\in\mathcal{I}$ holds for (6). Then, analogous to the proof of Theorem 3, the system (6) converges to a stationary consensus for the digraph that has a globally reachable node. Theorem 7 is proved. □

Remark 8. In the networks composed of interconnected dynamic systems, the scalability is a important property that needs to be maintained [5, 20, 21, 23]. Obviously, the decentralized consensus conditions (8) and (11) maintain the scalability of the multi-agent network. Without having to redesign the entire network whenever an agent is added or removed, the networks (6) and (7) can achieve the desired collective behavior as long as the local conditions for the agent and its neighbors hold respectively, and the connectedness of the interconnection topology is maintained.

Remark 9. In the consensus analysis of the multi-agent systems, Greshgorin's disc theorem has been extensively used to obtain the communication delay-independent consensus condition for the system with heterogeneous communication delays [8, 17]. In [17], decentralized frequency-domain consensus conditions have been obtained for the multi-agent systems with agents' dynamic modeled by strictly stable linear systems under heterogeneous communication delays. Then, by transformation, the system (6) can be expressed as a special case of the system studied in [17]. However, (11) gives a concrete algebraic criterion, which is convenient for the design of the consensus algorithm.

Remark 10. According to [20] (the inequality (24) there in), the consensus condition (8) in Theorem 3 satisfies

$$\frac{m_i}{2\kappa_i(G_i^M)^{-1}} > \frac{\alpha_i}{2\kappa_i T_i} > \frac{\alpha_i^2}{2\kappa_i(m_i+\alpha_i T_i)}.$$

Thus, under the same conditions, the consensus condition (11) in Theorem 7 is more conservative than the consensus condition (8) given in Theorem 3.

5 Simulation

Example 11. *Consensus under input delays based on symmetric graph.*

Consider a multi-agent network of five dynamic agents described by (7). The interconnection topology is described in Figure 1, and the graph is undirected and connected. The symmetric weights of the edges are: $a_{12}=a_{21}=0.2$, $a_{15}=a_{51}=0.3$, $a_{23}=a_{32}=0.5$, $a_{24}=a_{42}=0.1$, $a_{45}=a_{54}=0.4$. *The input delays of the agents are:* $T_1=0.5$*(s),* $T_2=0.6$*(s),* $T_3=0.8$*(s),* $T_4=0.4$*(s) and* $T_5=0.3$*(s). The velocity*

damping coefficients of the agents are: $\alpha_1 = 1$, $\alpha_2 = \frac{2}{3}$, $\alpha_3 = 0.25$, $\alpha_4 = 2$, $\alpha_5 = 3$, *and the mass of each agent is assumed to be* 1, *i.e.,* $m_i = 1, i = 1, \cdots, 5$. *Thus, the Assumption 2 holds for all the agents. For the transfer functions* $W_i(s) = \frac{e^{-sT_i}}{s^2 + \frac{\alpha_i}{m_i}s}, i = 1, \cdots, 5$, *by using the Matlab simulator, we obtain the gain margins:* $G_1^M \simeq 2.15$, $G_2^M \simeq 1.18$, $G_3^M \simeq 0.32$, $G_4^M \simeq 5.56$ *and* $G_5^M \simeq 11.24$. *According to the condition (8), we obtain that the control parameters* κ_i *satisfy:* $\kappa_1 \in (0, 2.15)$, $\kappa_2 \in (0, 0.74)$, $\kappa_3 \in (0, 0.32)$, $\kappa_4 \in (0, 5.56)$, $\kappa_5 \in (0, 8.03)$, *and we choose* $\kappa_1 = 2$, $\kappa_2 = 0.7$, $\kappa_3 = 0.3$, $\kappa_4 = 2$ *and* $\kappa_5 = 3$. *Then, with the initial states generated randomly, the agents in the system (7) converge to a stationary consensus (see Figure 2).*

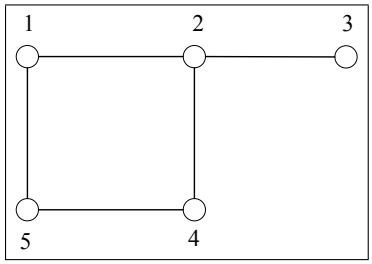


Figure 1: Undirected graph with symmetric weights

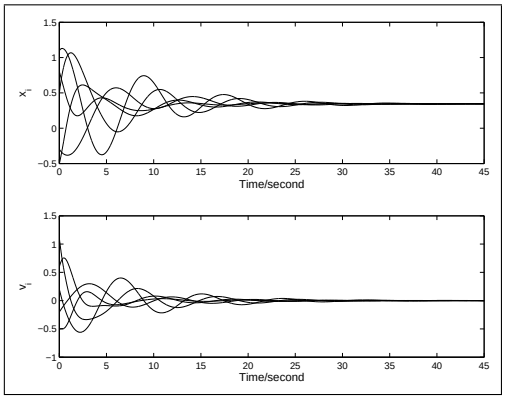


Figure 2: Consensus convergence under input delays

Example 12. *Consensus under input and communication delays based on asymmetric digraph.*

Consider a network of five agents described by (6). The interconnection topology is a digraph described in Figure 3, and the globally reachable node set of the digraph is $\{3, 4\}$. *The weights of the directed edges are:* $a_{12} = 0.2$, $a_{24} = 0.2$, $a_{34} = 0.5$, $a_{43} = 0.1$, $a_{54} = 0.4$, $a_{51} = 0.1$, *and the corresponding communication delays are:* $\tau_{12} = 0.15$*(s),* $\tau_{24} = 0.2$*(s),* $\tau_{34} = 0.5$*(s),* $\tau_{43} = 0.3$*(s),* $\tau_{54} = 0.4$*(s),* $\tau_{51} = 0.2$*(s). The velocity damping coefficients of the agents are:* $\alpha_1 = 1.5$, $\alpha_2 = 2$, $\alpha_3 = 1$, $\alpha_4 = 2$, $\alpha_5 = 3$, *and the mass of each agent is assumed to be* 1. *Choosing the control parameters:* $\kappa_1 = 1$, $\kappa_2 = 2$, $\kappa_3 = 0.5$, $\kappa_4 = 2$, $\kappa_5 = 3$, *we obtain from the condition (11) that the constraints on the input delays are:* $T_1 \in (0, 1.83)$*(s),* $T_2 \in (0, 2)$*(s),* $T_3 \in (0, 1)$*(s),* $T_4 \in (0, 4.5)$*(s) and* $T_5 \in (0, 0.67)$*(s). With* $T_1 = 1$*(s),* $T_2 = 0.8$*(s),* $T_3 = 0.6$*(s),* $T_4 = 2$*(s),* $T_5 = 0.6$*(s), the agents in the system (6) converge to a stationary consensus (see Figure 4).*

6 Conclusions

In this paper, we investigate the consensus problem of second-order multi-agent systems with velocity damping term in the agent's dynamic. Based on the frequency-domain analysis, two sufficient

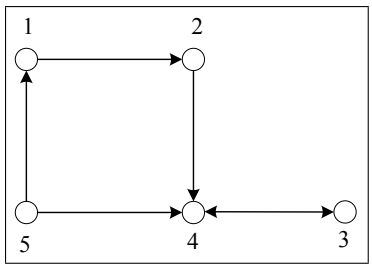


Figure 3: Digraph composed of 5 agents

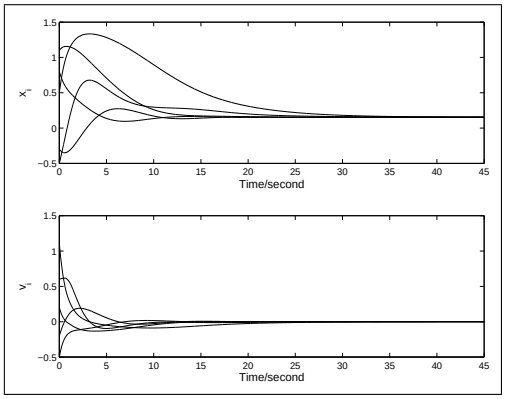


Figure 4: Consensus convergence under input and communication delays

decentralized consensus conditions are obtained. One consensus condition is for the system with heterogeneous input delays based on undirected and symmetric graph, and is dependent on the input delays. The other consensus condition is for the system with both heterogeneous input delays and communication delays based on general directed graph, and depends on the input delays only. Although the later consensus condition is more conservative than the former, it can be applied to the systems based on directed graph with asymmetric coupling weights.

Acknowledgements

This work was supported by Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20090093120006).

Bibliography

[1] A. Jadbabaie, J. Lin, A.S. Morse, Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules, *IEEE Transactions on Automatic Control*, 48(6):988-1001, 2003.

[2] R. Olfati-Saber, R. Murray, Consensus Problems in Networks of Agents with Switching Topology and Time-delays, *IEEE Transactions on Automatic Control*, 49(9):1520-1533, 2004.

[3] W. Ren, E. Atkins, Distributed Multi-vehicle Coordinated Control via Local Information Exchange, *International Journal of Robust and Nonlinear Control*, 17(10-11):1002-1033, 2007.

[4] Y. Hong, L. Gao, D. Cheng, J. Jiang, Lyapunov-based Approach to Multiagent Systems with Switching Jointly Connected Interconnection, *IEEE Transactions on Automatic Control*, 52(5):943-948, 2007.

[5] Y.-P. Tian, C.-L. Liu, Consensus of Multi-agent Systems with Diverse Input and Communication Delays, *IEEE Transactions on Automatic Control*, 53(9):2122-2128, 2008.

[6] W. Wang, J.J.E. Slotine, Contraction Analysis of Time-delayed Communication Delays, *IEEE Transactions on Automatic Control*, 51(4):712-717, 2006.

[7] Y.-G. Sun, L. Wang, Consensus of Multi-agent Systems in Directed Networks with Nonuniform Time-Varying Delays, *IEEE Transactions on Automatic Control*, 54(7):1607-1613, 2009.

[8] J. Wang, N. Elia, Consensus over Network with Dynamic Channels, *Proc. of the 2008 American Control Conference*, Seattle, pp.2637-2642, 2008.

[9] M. Cao, A.S. Morse, B.D.O. Anderson, Reaching an Agreement Using Delayed Information, *Proc. of the 45th IEEE Conference on Decision and Control*, San Diego, pp.3375-3380, 2006.

[10] F. Xiao, L. Wang, Asynchronous Consensus in Continuous-time Multi-agents with Switching Topology and Time-varying Delays, *IEEE Transactions on Automatic Control*, 53(8):1804-1816, 2008.

[11] V.D. Blondel, J.M. Hendrickx, A. Olshevsky, J.N. Tsitsiklis, Convergence in Multi-agent Coordination, Consensus, and Flocking, *Proc. of the 44th IEEE Conference on Decision and Control*, Seville, pp.2996-3000, 2005.

[12] V. Gazi, Stability of an Asynchronous Swarm with Time-dependent Communication Links, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 38(1):267-274, 2008.

[13] J. Hu, Y. Hong, Leader-following Coordination of Multi-agent Systems with Coupling Time Delays, *Physica A*, 374(2):853-863, 2007.

[14] H. Su, X. Wang, Second-order Consensus of Multiple Agents with Coupling Delay, *Proc. of the 7th world Congress on Intelligent Control and Automation*, Chongqing, pp.7181-7186, 2008.

[15] P. Lin, Y. Jia, J. Du, S. Yuan, Distributed Consensus Control for Second-order Agents with Fixed Topology and Time-delay, *Proc. of the 26th Chinese Control Conference*, Zhangjiajie, pp.577-581, 2007.

[16] W. Yang, A.L. Bertozzi, X. Wang, Stability of a Second Order Consensus Algorithm with Time Delay, *Proc. of the 47th IEEE Conference on Decision and Control*, Cancun, pp.2926-2931, 2008.

[17] D.J. Lee, M.K. Spong, Agreement with Non-uniform Information Delays, *Proc. of the 2006 American Control Conference*, Minneapolis, pp.756-761, 2006.

[18] S. Kawamura, M. Svinin (eds.), *Advances in Robot Control: From Everyday Physics to Human-Like Movements*, Berlin: Springer-Verlag, 2006, pp.107-134.

[19] P. Lin, Y. Jia, Consensus of Second-order Discrete-time Multi-agent Systems with Nonuniform Time-delays and Dynamically Changing Topologyies, *Automatica*, 45(9):2154-2158, 2009.

[20] Y.-P. Tian, C.-L, Liu, Robust Consensus of Multi-agent Systems with Diverse Input Delays and Asymmetric Interconnection Perturbations, *Automatica*, 45(5):1374-1353, 2009.

[21] Y.-P. Tian, G. Chen, Stability of The Primal-dual Algorithm for Congestion control, *International Journal of Control*, 79(6):662-676, 2006.

[22] R. Pedrami, B.W. Gordon, Control and Analysis of Energetic Swarm Systems, *Proc. of the 2007 American Control Conference*, New York, pp.1894-1899, 2007.

[23] I. Lestas, G. Vinnicombe, Scalable Robustness for Consensus Protocols with Heterogeneous Dynamics, *Proc. of the 16th IFAC World Congress*, Prague, 2005.

[24] Z. Lin, B. Francis, M. Maggiore, Necessary and Sufficient Graphical Conditions for Formation Control of Unicycles, *IEEE Transactions on Automatic Control*, 50(1):121-127, 2005.

[25] C.A. Desoer, Y.T. Wang, On the Generalized Nyquist Stability Criterion, *IEEE Transactions on Automatic Control*, 25(2):187-196, 1980.

Cheng-Lin Liu was born in China in 1981. He got PHD at Southeast University in 2008. Since 2008, He is a faculty member at Institute of Automation, Jiangnan University, China. His current research interests include Internet congestion control and coordination control of multi-agent systems.

Fei Liu was born in China in 1965. He is a professor at Institute of Automation, Jiangnan University, China. His research interests include the theory and application of advanced process control, process monitoring and diagnose on industrial system, and integrated automatic system for fermentation process.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 3, pp. 336-350

Node Availability for Distributed Systems considering processor and RAM utilization for Load Balancing

A. Menendez LC, H. Benitez-Perez

Antonio Menendez Leonel de Cervantes, Hector Benitez Perez
Universidad Nacional Autonoma de Mexico
Instituto de Investigaciones en Matematicas Aplicadas y Sistemas
Departameto de Ingenieria de Sistemas Computacionales y Automatizacion
Ciudad Universitaria, Mexico D.F.
E-mail: toniomlc@gmail.com, hector@uxdea4.iimas.unam.mx

Abstract: Node-Availability is a new metric that based on processor utilization, free RAM and number of processes queued at a node, compares different workload levels of the nodes participating in a distributed system. Dynamic scheduling and Load-Balancing in distributed systems can be achieved through the Node-Availability metric as decision criterion, even without previously knowing the execution time of the processes, nor other information about them such as process communication requirements.
This paper also presents a case study which shows that the metric is feasible to implement in conjunction with a dynamic Load-Balancing algorithm, obtaining an acceptable performance.
Keywords: Node-Availability, Load Balancing, Distributed systems, High-Performance.

1 Introduction

Load-Balancing is a technique often used to distribute computational load among processors or other resources in order to get a better performance (i.e. optimal resource utilization and small processing time). When performing Load-Balancing (LB) for a Distributed System (DS) it is expected that the resources (particularly the processors) to be evenly used, therefore obtaining a general system performance increase [4]. Several studies have been carried out in terms of performance [10], task allocation [8], communication media [5], dynamic networking [1], mobile performance [5] and so on. However, these strategies depend on previous measures such as execution time of a process or communication requirements, or in standard metrics (e.g. number of processes queued at a node or processor idle time percentage) where availability (i.e. the capacity of a node to process a job) is not observed. In any case the need to measure the performance(i.e. optimal resource utilization and small processing time) of a DS with similar and almost periodic processes is not directly addressed.

In this paper a new metric named Node-Availability is introduced, it takes advantage on existing metrics such as processor and RAM utilization, the number of processes queued at a node and processes communications and compose them to create the new metric. By including several existing metrics in its calculation, Node-Availability is a metric that provides more information of a node in its value, than solely using any of the existing metrics. It compares different workload levels at two or more nodes participating in a distributed system, providing a decision criterion to be implemented in conjunction with a common workload algorithm. Dynamic scheduling and Load-Balancing in distributed systems is achieved through the Node-Availability metric, even without previously knowing the execution time of the processes, nor other information about them such as process communication requirements.

The objective of this paper is to present a metric named Node-Availability [9], how it is constructed and how it allows a DS to execute a set of processes in a balanced manner obtaining fair utilization of the overall system. One of the advantages of using the Node-Availability metric resides in its ability to

perform the load-balancing of a DS without previously knowing the execution times of the processes involved, because if the processing times where known, the scheduling and execution of the processes could be done using proven algorithms [3], [6].

The rest of this document is organized as follows: The metric is described in section 2. The LB algorithm used is presented in section 3. A case study is in section 4. The conclusions and future work are in section 5.

2 Node-Availability

The execution time of a process in a DS is a function determined by the complexity of the process, the communication strategy and by the resources available within the DS. Since this execution time cannot be easily seeing when a DS implementation is performed other strategy needs to be followed. For instance, secondary measurements such as used memory for each node or communication load amongst processors and processes can be followed. The decision of which is the most suitable measure depends entirely on the implementing resources. Considering that a Metric is a quantitative and periodic measurement interpreted in the context of a series of previous equivalent measurements, the metric to estimate the nodes availability is presented, first from the node resources point of view (2.1), followed by the tasks load (2.2) and the communications costs (2.3).

2.1 Availability

One of the most used metrics in terms of distributed computing is availability, defined as the capacity of a node to process a job at a specific time, it can be obtained from several secondary measures like the time consumed by each node (processor) or by communication performance per process.

A DS can be considered as a set of nodes communicating with each other trough a network, where node is defined as the autonomous processing unit, which consists of one processor and RAM (random access memory).

The processes that are executed in a DS generally demand to use the processor and/or memory, they are not characterized by a high input-output demand, so common measurements within nodes are the processor idle time or the percentage of free memory available. When a process demands a memory space larger than the physical RAM, the Operating System provides virtual memory to it, causing the total execution time of the process to be increased.

The percentage of processor and memory available (Figure 1) during a time sample, allows to realize what the Operating System (OS) behavior is, in terms of resource allocation to a process. The OS tries to allocate all the (RAM) memory that a process demands, it also tries to assign the processor all the possible time, to the process being executed.

Figure 1. shows a process execution which takes about 70 seconds, during this time, the operating system assigns the processor to it, resulting in a 0% processor availability. On the other hand, the RAM demand is lower than the capacity, so the availability of it is between 60% and 100%. Figure 2. shows a process demanding an amount of memory lager than the physical memory (RAM) of the node, the Operating System (OS) assigns to the process all the available RAM and then it provides virtual memory to fulfill the memory demand. As it can be seen in Figure 2. the percentage of memory available during the execution of the process is 0%, while the processor availability oscillates at the beginning and towards the end of the process execution. Figures 1 and 2 show the same process executed by the same node. It takes longer to be executed when it demands the use of virtual memory (approx. 10 seconds, Figure 2) than when it is restricted to use RAM (approx. 70 seconds, Figure 1).

The first step in our proposal is to determine the availability of a node in terms of its idle processor time and its free RAM. The highest values for these metrics indicate the most available node and the lowest values indicate the busiest node. These to metrics are multiplied for two reasons, being the first one

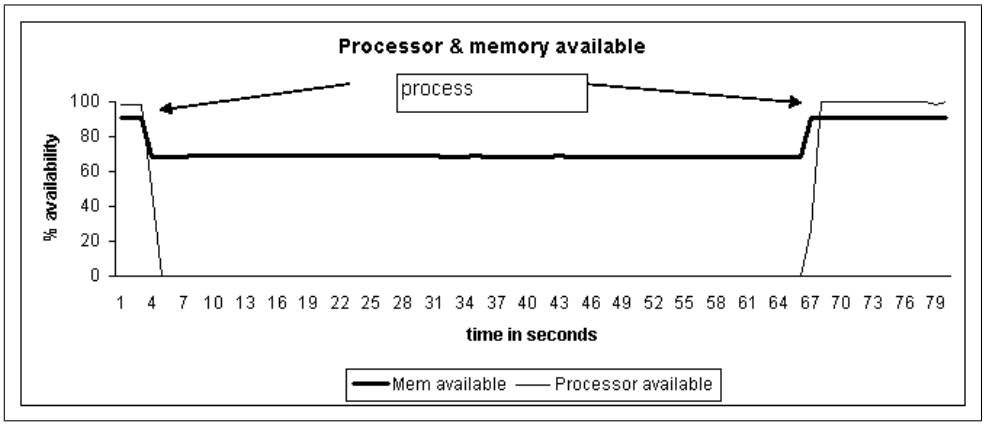


Figure 1: Memory demand within limits of RAM

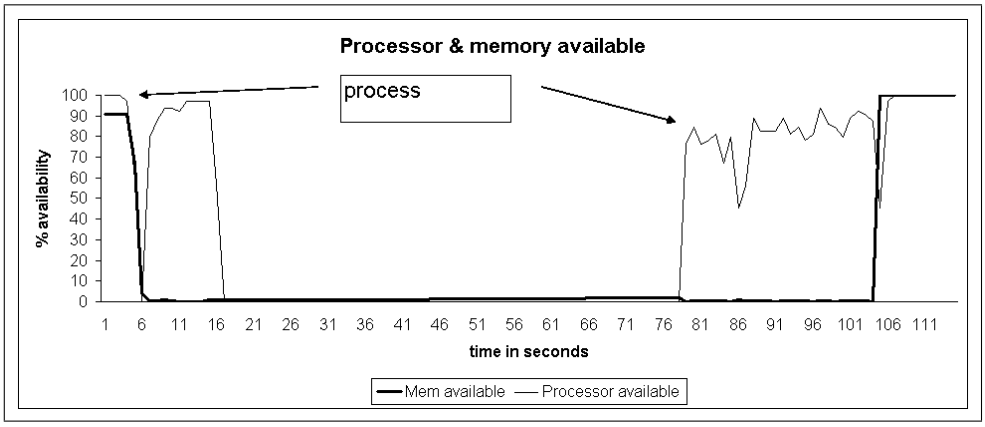


Figure 2: Demanding virtual memory

that since they are percentages the outcome of the product is also a percentage the values cannot be added because the outcome would exceed 100%. The second reason is that if the values are averaged instead of multiplied it is possible to obtain unreal outcomes, for example a node with 100% of idle processor time and 0% of available RAM averages 50%, which in terms of availability should be 0%, because if any of the two resources (processor and memory) is not available no process can be processed. On the other hand, with the proposed approach the availability is real since the value in this same example is 0%.

So the first step to evaluate the node availability of node (i) is calculated by:

$$A_i = \alpha_i \beta_i \tag{1}$$

where: $i \in \{1, 2, \ldots, n\}$ is the node identifier, α_i is the idle processor time percentage of node i, and β_i is the free RAM percentage of node i. As α_i and β_i, are percentages, they can be multiplied to find the availability of the node, giving a value between 0 and 1.

2.2 Number of processes queued at a node

The availability of a node depends not only in its respective α and β values, but also on the number of processes queued, so the previous equation of (Equation 1) must reflect this situation, considering that the number of processes affects the availability of a node in an exponential way. So Equation 1 becomes:

$$B_i = \frac{\alpha_i \beta_i}{e^{\varepsilon_i}} \tag{2}$$

where ε_i is the number of process queued at node i.

2.3 Processes communications

From the communications point of view, a process can have communications with another process in the same node that the process is being executed in, with a process that is being executed in a different node or no communications at all. The communication resources availability and the execution time is different in these three scenarios, therefore it is necessary to differentiate the processes queued in a node based on their communication requirements. So two new variables are introduced: κ_i is the number of processes queued at node i communicating (internally) with processes also queued in node i; and ν_i is the number of processes queued at node i communicating (externally) with processes queued in any node different to i.

As stated previously, the execution time of a process depends (among other things) on its communication requirements, the communication time can be approximated in terms of the type of communication (i.e. internal or external) that a process performs. As each type of communication has a different time impact, two constants are defined: ρ is the internal communication time and τ is the external communication time. The equation of Node-Availability (Equation 2) now becomes:

$$C_i = \frac{\alpha_i \beta_i}{e^{(\varepsilon_i + \rho\,\kappa_i + \tau\,\nu_i)}} \tag{3}$$

From a process communication requirements point of view, all the nodes of the DS are different. A process tends to finish its execution earlier if it communicates with processes running in the same node, therefore depending mostly on the node availability; if the processes are in different nodes, they depend not only on the different nodes availability, but also on the network speed and available bandwidth.

2.4 Number of samples taken and periodicity

Our proposal is to determine the availability of a node in terms of its idle processor time, its free RAM memory, the number of processes queued at the node and their communications. These measures are taken periodically and when δ samples have been read, then, the node availability is calculated obtaining an arithmetic average of the readings. So the final equation of Node-Availability or Φ is:

$$\Phi_{i,j} = \frac{1}{\delta} \sum_{j=1}^{\delta} \frac{\alpha_{i,j}\beta_{i,j}}{e^{(\varepsilon_{i,j}+\rho\kappa_{i,j}+\tau\nu_{i,j})}} \tag{4}$$

where: $i \in \{1,2,\ldots,n\}$ is the node identifier, $j \in \{1,2,\ldots,\delta\}$ is the sample number, $\alpha_{i,j}$ is the idle processor time percentage of node i considering the jth sample, $\beta_{i,j}$ is the free RAM memory percentage of node i considering the jth sample, $\varepsilon_{i,j}$ is the number of processes queued at node i considering the jth sample, $\kappa_{i,j}$ is the number of processes queued at node i with internal communications considering the jth sample, $\nu_{i,j}$ is the number of processes queued at node i with external communications considering the jth sample, ρ is the internal communication time, τ is the external communication time and δ is the number of samples taken of: $\alpha_{i,j}$, $\beta_{i,j}$ and $\varepsilon_{i,j}$ before sending the data to the LB process.

2.5 Optimization characteristics of Φ

In this section we show that the metric Φ presents a global minimum dependent on local values. The metric Φ is characterized per node where the discrete variables $\alpha, \beta, \varepsilon, \kappa$ and ν play the role of representing the system behavior based on the sample taken. Therefore ρ and τ are the communication time factors that can be distinguish as bounded variables and they can be delimited through a local linear optimization strategy. The time values of ρ and τ help to determine some network characteristics such as the type of protocol or the required speed specifications.

The first parameter to be defined is the Load-Balancing factor of the Distributed System(DS), which is defined as:

$$\mu_j = \frac{\Phi_{min,j}}{\Phi_{max,j}} \tag{5}$$

Where $\Phi_{min,j}$ is the Node-Availability value corresponding to the least available node and $\Phi_{max,j}$ is the value corresponding to the most available node within the DS at sample j. It can be noticed that when $\Phi_{min,j}$ and $\Phi_{max,j}$ are similar, μ_j tends to be one if the system is balanced. On the other hand, the DS is as unbalanced as μ_j tends to zero. Based upon this approximation the DS error at sample j can be defined as the DS level of unbalance Ω given by:

$$\Omega_j = \frac{(1-\mu_j)^2}{2} \tag{6}$$

To minimize this error it is necessary to balance the DS around the local loads, having that:

$$if \quad \mu_j \to 1 \quad then \quad \Omega_j \to 0$$

In order to minimize this error, the use of partial derivatives of this equation is pursued in terms of the communication times ρ and τ as shown next:

$$\frac{\partial\Omega_j}{\partial\rho_{max,j}}, \frac{\partial\Omega_j}{\partial\rho_{min,j}}, \frac{\partial\Omega_j}{\partial\tau_{max,j}}, \frac{\partial\Omega_j}{\partial\tau_{min,j}} \tag{7}$$

where: $\rho_{max,j}$ and $\tau_{max,j}$ are the communication times at the most available node, and $\rho_{min,j}$ and $\tau_{min,j}$ are the communication times at the least available node, both cases at sample j.

Now if we take into account the related node and sample values, the global error ˝ is expressed as follows:

$$\begin{aligned}\Omega_j &= \frac{1}{2}\left(\frac{\Phi_{min,j}}{\Phi_{max,j}}\right)^2 = \\ &\frac{1}{2}\left(1-\frac{\alpha_{min,j}\beta_{min,j}}{\alpha_{max,j}\beta_{max,j}}e^{(\varepsilon_{max,j}-\varepsilon_{min,j}+\rho_{max,j}\kappa_{max,j}-\rho_{min,j}\kappa_{min,j}+\tau_{max,j}\nu_{max,j}-\tau_{min,j}\nu_{min,j})}\right)^2\end{aligned} \tag{8}$$

where: *max* corresponds to the most available node, *min* corresponds to the least available node and *j* is the sample number.

Reordering this expression in terms of $\lambda_j^{(1)}$ and $\lambda_j^{(2)}$ as follows:

$$\begin{aligned}\lambda_j^{(1)} &= \frac{\alpha_{min,j}\beta_{min,j}}{\alpha_{max,j}\beta_{max,j}} \\ \lambda_j^{(2)} &= \varepsilon_{max,j}-\varepsilon_{min,j}+\rho_{max,j}\kappa_{max,j}-\rho_{min,j}\kappa_{min,j}+\tau_{max,j}\nu_{max,j}-\tau_{min,j}\nu_{min,j}\end{aligned} \tag{9}$$

Ω can be expressed as:

$$\Omega_j = \frac{1}{2}\left(1-\lambda_j^{(1)}e^{\lambda_j^{(2)}}\right)^2 \tag{10}$$

The partial derivatives are as follows:

$$\begin{aligned}\frac{\partial\Omega_j}{\partial\rho_{min,j}} &= (1-\lambda_j^{(1)}e^{\lambda_j^{(2)}})(\lambda_j^{(1)}e^{\lambda_j^{(2)}})(\kappa_{min,j}) \\ \frac{\partial\Omega_j}{\partial\rho_{max,j}} &= (1-\lambda_j^{(1)}e^{\lambda_j^{(2)}})(\lambda_j^{(1)}e^{\lambda_j^{(2)}})(-\kappa_{max,j}) \\ \frac{\partial\Omega_j}{\partial\tau_{min,j}} &= (1-\lambda_j^{(1)}e^{\lambda_j^{(2)}})(\lambda_j^{(1)}e^{\lambda_j^{(2)}})(\nu_{min,j}) \\ \frac{\partial\Omega_j}{\partial\tau_{max,j}} &= (1-\lambda_j^{(1)}e^{\lambda_j^{(2)}})(\lambda_j^{(1)}e^{\lambda_j^{(2)}})(-\nu_{max,j})\end{aligned} \tag{11}$$

The communication times ρ and τ defined in terms of the next sampling period (j+1) are expressed as:

$$\begin{aligned}\rho_{min,j+1} &= \rho_{min,j}+\eta\frac{\partial\Omega_j}{\partial\rho_{min,j}} \\ \rho_{max,j+1} &= \rho_{max,j}+\eta\frac{\partial\Omega_j}{\partial\rho_{max,j}} \\ \tau_{min,j+1} &= \tau_{min,j}+\eta\frac{\partial\Omega_j}{\partial\tau_{min,j}} \\ \tau_{max,j+1} &= \tau_{max,j}+\eta\frac{\partial\Omega_j}{\partial\tau_{max,j}}\end{aligned} \tag{12}$$

where: η is a design factor where the metric balances the performance of each node depending on its relations amongst (α, β and ε).

2.6 Metric optimization examples

In order to show the effectiveness of this technique two examples are carried out in which the metric Φ is evaluated without performing any load-balancing. The first example has a medium profile processor utilization and is called "relaxed example", the second example has high profile processor utilization and is named "restrictive example". This processor utilization is calculated with the well known metric of processor utilization by a set of periodic tasks called "Processor Utilization Factor" [16]. In both examples a set of 40 periodic tasks is evenly distributed trough 10 nodes (i.e. 4 tasks per node). The theoretical workload that a set of periodic tasks imposes to a processor can be calculated using Equation (13), in which the "Utilization" of a processor is a value under one.

$$U = \sum_{i=1}^{n} \frac{c_i}{p_i} \tag{13}$$

where: U = processor Utilization, c_i = time Consumed by the task i and p_i = Period of task i and n = number of tasks.

In both examples the purpose is to obtain the optimized values for ρ and τ that are used in the next section where the metric Φ is implemented within a load-balancing algorithm.

The setup for the two examples consists of 10 nodes with 4 periodic tasks per node, the periods and consumption times for each task differ in both cases, Table 1 shows the parameters used for the relaxed example and in Table 4 are the periods and consumption times for the tasks in the restrictive example. Notice that the first three tasks (i.e. tasks numbered 1 to 3) are identical in all the nodes, the difference in the workloads of the nodes resides in the fourth task, in which the period is modified. As it can be seen in Table 1, the fourth task at every node has a period equivalent to 8 times the node number (e.g. the period of the fourth task 4 at the node number 1 is 8, the period of task 4 in node 2 is 16 an so on).

The number of samples taken during both examples is 1000, and every 8 (i.e. $\delta = 8$) samples the value of Φ is calculated using Equation (4). The number of tasks per node with internal communications is 2 and one task has external communications.

Table 1: Tasks parameters (relaxed example)

Task Number	Period (p)	Consumption time (C)
1	8	1
2	16	2
3	32	3
4	8 times the number of node	1

The set of tasks assigned to each node according to Table1 has a processor utilization per node as listed in Table 2. This example is called "relaxed" because the maximum utilization of a processor corresponds to node number 1, and it is 0.5 (as can be seen in Table 2).

Following the optimization procedure of ρ and τ explained in the previous section, their final values (final ρ and final τ) after 1,000 samples with δ=8 are shown in Table 2.

A restrictive example is presented where the processor utilization is between 70% and 81%, as can be seen in Table 4.

The difference amongst the "relaxed" and "restrictive" example resides in duration of the periods of the first three tasks at every node, as shown in Table 3, these periods last half the time for the restrictive case, thus imposing a major workload to the processors as can be seen in Table 4. The number of samples taken is again 1000 with a δ of 8. The number of tasks with internal communications is 2 and the number of tasks with external communications is 1. As stated earlier, the main difference with the

Table 2: Processor Utilization and final values of ρ and τ per node (relaxed example)

Node	Utilization factor	Final ρ time	Final τ time
1	0.4688	0.0095	0.0127
2	0.4062	0.0935	0.0673
3	0.3854	0.1033	0.1165
4	0.375	0.1081	0.1406
5	0.3688	0.111	0.1548
6	0.3646	0.1128	0.1642
7	0.3616	0.1142	0.1708
8	0.3594	0.1152	0.1758
9	0.3576	0.1159	0.1797
10	0.3562	0.1166	0.1828

"relaxed" example resides on the "Utilization factor" for all the nodes, which is around 0.75 as shown in Table 4.

Based upon these two examples, final values pf ρ & τ represent the communication characteristics or guarantees that the DS must provide in order to have a balanced system, meaning that when a node has more processor utilization, it needs to take less time in its communications.

Table 3: Tasks parameters (restrictive example)

Task Number	Period (p)	Consumption time (C)
1	4	1
2	8	2
3	16	3
4	8 times the number of node	1

Even tough the value of τ can be seen as a local parameter, it is more common to have or guarantee a global communication time for all the nodes participating in a network, so a unique communication time for the whole network must be considered. If the value chosen for τ is the one corresponding to the minimum value for the τ's amongst all the nodes then the network is more restrictive and therefore the external communications need to be faster. On the other hand the maximum value of τ means that the communications are relaxed respect to the time they take, but the counterpart is that they not help to have a balanced DS since the least available nodes need faster external communications in order to have equivalent Node-Availability values to the most available nodes. It can clearly be noticed that the in order to provide communications at the speed required by the value of τ, the network specifications play an important role. The same reasoning applies to the values of ρ for internal communications.

The range of values for ρ and τ listed in Tables 2 for the relaxed example or in 4 for the restrictive example, indicate respectively the time that internal (ρ) and external (τ) communication must take in order to have a balanced system. Further more they show the benefits of using Φ as a metric, convenient not only to perform such a task as load-balancing, but also useful to determine the optimal duration for process communications, and in the case of τ providing the speed specifications for the communications network.

Whether to choose the minimum, maximum or average values from the restrictive or relaxed case

Table 4: Processor Utilization per node (restrictive example)

Node	Utilization factor	Final ρ time	Final τ time
1	0.8125	0.0167	0.0348
2	0.75	0.1491	0.3455
3	0.7292	0.172	0.4599
4	0.7188	0.1828	0.5138
5	0.7125	0.189	0.5452
6	0.7083	0.1931	0.5657
7	0.7054	0.196	0.5802
8	0.7031	0.1982	0.591
9	0.7014	0.1999	0.5993
10	0.7	0.2012	0.606

for ρ and τ depends strictly on the particular implementation case (i.e. network protocol and processor utilization factor).

3 The High-Low (HILO) algorithm.

In order to perform LB (load-balancing) and load distribution using the presented metric Node-Availability, a simple and well known algorithm, here named High-Low (HILO) is used. The underlying principle in HILO is to determine the most available node and the least available one. The knowledge of these nodes is used by the algorithm to perform its two main methods, the periodic method named Balance and the event triggered method named Distributor (see Figure 3). These two methods are nested depending on the arrival of new processes as shown in Figure 4, in this case the periodic Balance method is executed every period while Distributor is executed only when a new process arrives.

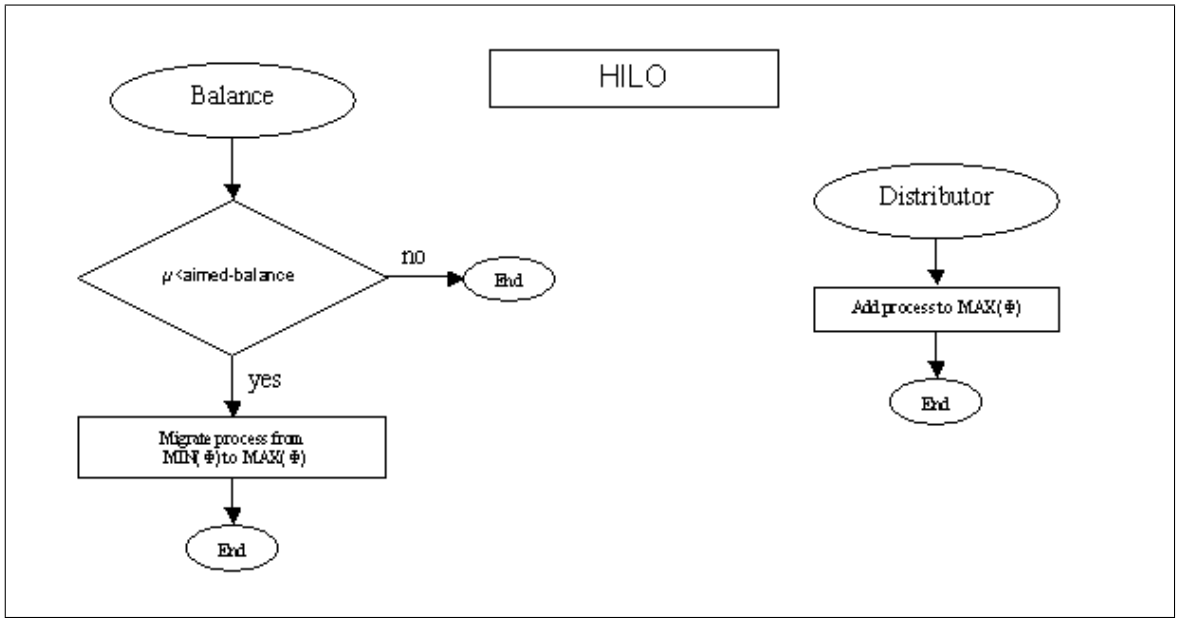


Figure 3: Flow diagram of HILO methods Balance an Distributor

The Balance method is performed as follows: the most available node is found by calculating the Node-Availability of all the nodes participating in the distributed system and selecting the one with the maximum Φ value, so the most available node is: MAX(Φ) and analogously the least available node is MIN(Φ). In order to be able to balance a DS, the HILO algorithm also requires an aimed balance level for it named ξ having $0 < \xi \leq 1$.

With this three values (MAX(Φ), MIN(Φ) and ξ), the periodical Balance method calculates the actual "Load-Balancing factor" μ using Equation (5). If the obtained value of μ is under ξ then the Balance method performs the actual load-balancing. This load-balancing is as simple as removing one process from the queue of the node MIN(Φ) and migrating it to the node named MAX(Φ).

The second method named Distributor is responsible to assign a node to any new process arriving to the distributed system. Once a new process arrives, Distributor sends it to the node MAX(Φ).

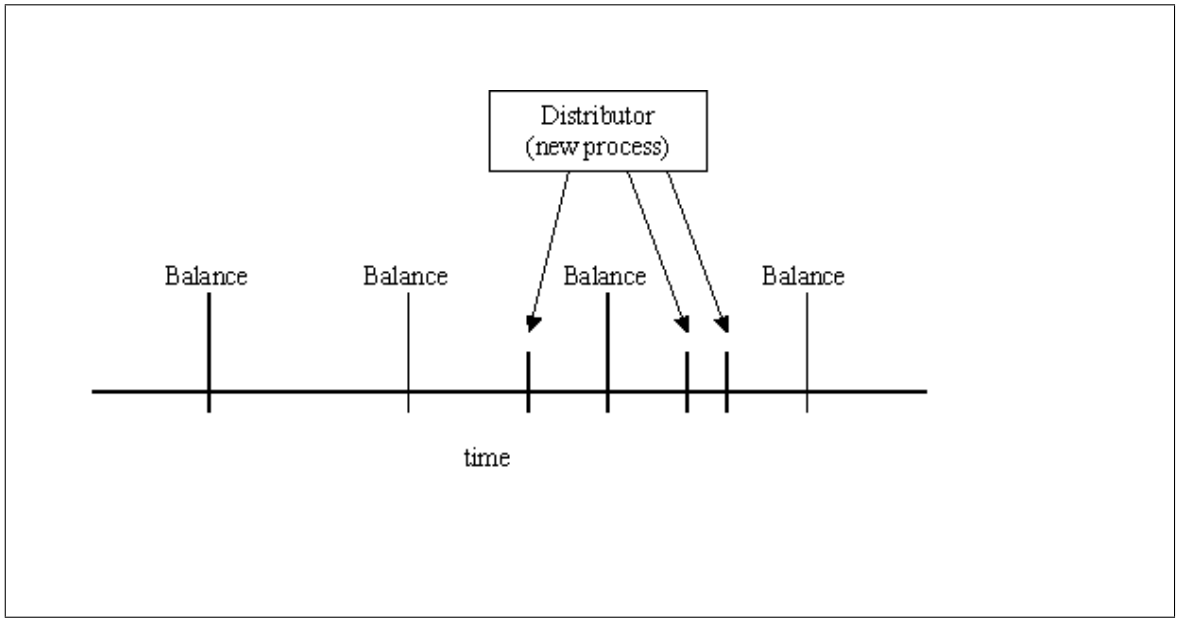


Figure 4: HILO methods: Balance an Distributor

3.1 Pseudo-code of HILO

The algorithm HILO has two methods the first one is a periodic method named "Balance", the second method which is executed every time a new process arrives to the distributed system is named "Distributor".

```
HILO
  Periodically_execute Balance
  if new Process  then Distributor
```

The Balance method:

```
Balance
 if MIN(Φ) / MAX(Φ) < ξ then
       remove Process from MIN(Φ) and send it to MAX(Φ)
```

When activated, the Distributor method sends the arriving (new) process, to the node MAX(Φ):

```
Distributor
 Add_queue MAX(Φ) new Process
```

3.2 Simulation

The HILO algorithm as well as the metric Φ are tested on both: a 16 nodes cluster (Case Study in next section) and on a simulation using Matlab. Figure 5 shows the simulation of this process where 500 samples are taken, the processes are generated between samples 50 and 250 using a Poisson distribution to simulate both; the number of processes ready to be executed and the duration of each one. It can be seeing that the system reaches an absolute balance around sample 400, but during the whole execution of the set of processes, no single node is over-occupied nor idle.

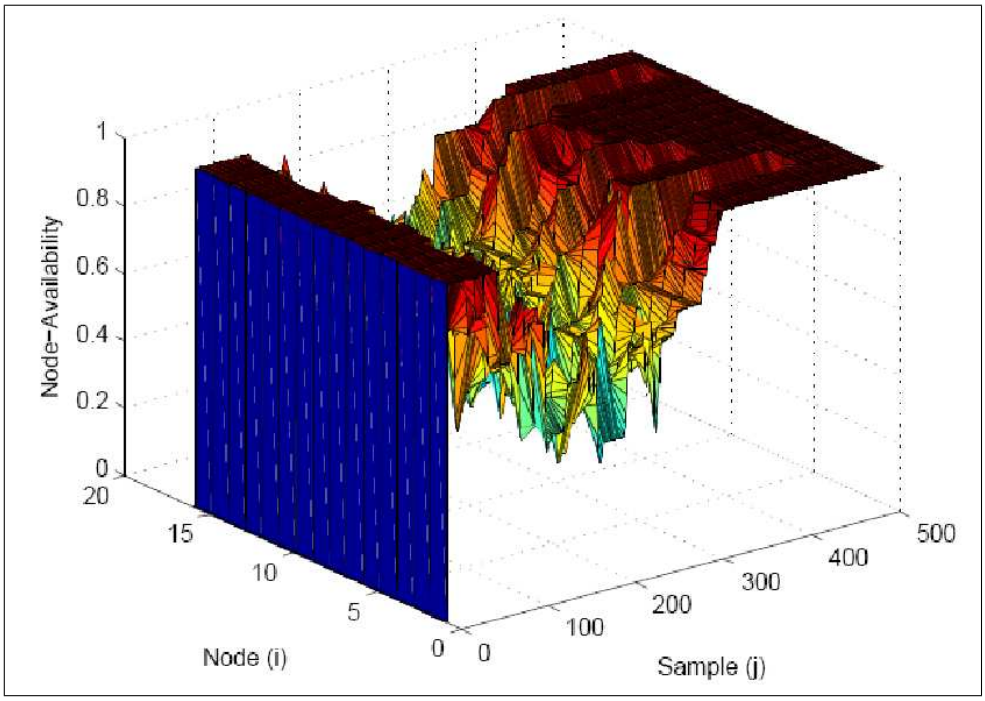


Figure 5: Simulation of a LB process within a cluster, 500 samples are taken and the processes are generated between samples 50 and 250.

The parameters used for the simulation are: Number of nodes n =16, total samples(j) = 500, and $\delta = 4$. The internal communication factor $\rho = 0.01$ and the external communication factor $\tau = 0.02$ are obtained by calculating the average times from the minimum respective values from Table 2 and Table 4. The average consumption time of each process is given by an exponential random distribution with $mean = \delta * 10$ and ξ is 0.66%. The initial value for $\alpha_{i,j}$ is one, this value is updated every time a process is queued at a node i, by decreasing its value by 0.1% per process, the approximated inverse procedure is performed when the process has finished. In terms of $\beta_{i,j}$ (available memory) a similar procedure is performed with a decrement value of 0.12%. These increments/decrements are according to the availability behavior presented in section 2.1. The value for $\varepsilon_{i,j}$ is calculated every sample, based on how many processes are queued per node and the values for $\kappa_{i,j}$ and $\nu_{i,j}$ are random numbers between 0 and $\varepsilon_{i,j}$.

The impact that Φ has on the DS load balance can be seen in Figure 5, outlining that the work load was evenly distributed amongst the 16 nodes during the whole simulation.

4 Case Study

The case study is based on a real geology-specific application, which consists of several similar processes distributed over a cluster. These processes perform a different number of operations locally. The Case study is processed seven times with a different number of processes, each occasion, as shown in Table 5. Every time the algorithm HILO and two common load distribution algorithms: Random and Round-Robin [12–14] are used to execute these 7 different sets of processes. The processes are ready to be executed based on a Poisson distribution, independent for each case study. The implementation details are presented in the 4.1 subsection, and the results in subsection 4.2.

4.1 Implementation

The case study is implemented in a dedicated cluster, which consists of 16 nodes with the following configuration:

One master node with 2 Xeon processors at 2.6 GHz, 1.5 GB RAM and Linux kernel 2.6.8. 15 nodes with Pentium IV processor at 2.6 GHz, 512 MB RAM and Linux kernel 2.6.12. The master node performs the distribution and load-balancing functions of the cluster. The case studies are integrated shown in Table 5.

Table 5: Number of processes per case study

Case study	Number of processes
1	100
2	200
3	300
4	400
5	500
6	1000
7	1500

These processes are independent amongst each other, and to simulate when a process is ready to be executed, a Poisson distribution is used. Every process performs a random number of local sums and string concatenations, both random numbers are generated globally using an exponential distribution for each case study [6], [15]. Furthermore, as both numbers differ, the demands of processor and memory are also different for every process and case study. For these cases κ and ν are equal to zero since there is no communication between processes.

Each set of processes is executed using the Random, Round-Robin and HILO algorithms to distribute the load. The Random algorithm uses a uniform distribution to select the node in which the arriving process is going to be queued. The Round-Robin algorithm sends the arriving process to the nodes in a round-robin manner. The algorithm HILO sends the arriving process to the UN node.

The algorithm HILO uses the following values for the parameters described earlier in this paper, selected (as means of example) in a heuristic manner: $\delta = 4$, $n = 15$ and $\xi = 0.6$.

4.2 Results

The total execution times of the seven sets of processes (listed in Table 1.) were obtained by executing them in the cluster, using the previously listed algorithms for the load distribution. The presented metric Node-Availability, implemented in the HILO algorithm outperforms the other two as can be seen in the

Figure 6 (Algorithm Comparison). Considering the execution time of HILO as 100%, the other two algorithms (Round-Robin and Random) take more time to complete the execution of the same seven sets of processes, this extra time goes from 10% to 65% (i.e. The execution time with Round-Robin or Random algorithms takes from 110% to 165% compared with HILO which is 100%). It can also be noticed in Figure 6 that with the smaller set of processes (i.e. 100) the percentage gain of HILO is larger, meaning that the algorithm is efficient even when the set has a relatively small number of processes.

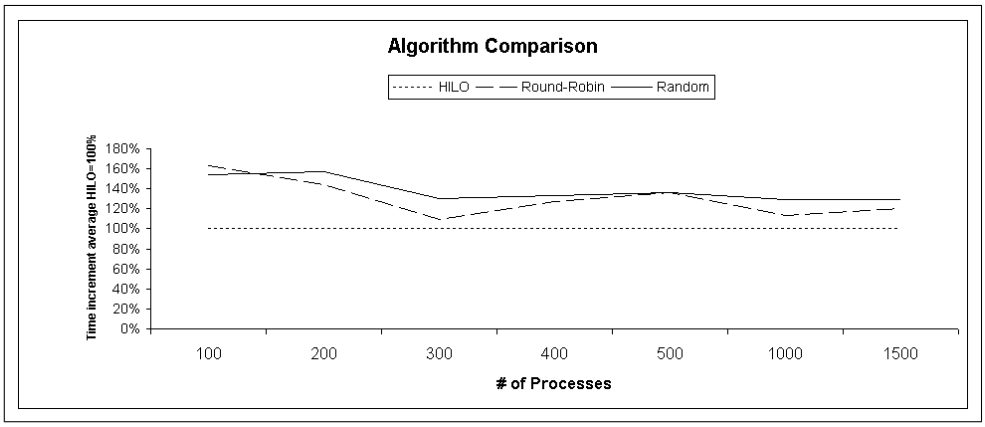


Figure 6: Percentage compared time-efficiency of the algorithm. HILO=100% .

Every occasion that the condition $\mu < \xi$ was fulfilled, a balance was performed. The total numbers of balances that HILO performed are shown in Table 6. The total execution time is expressed in minutes:seconds.

Table 6: Number of balances and total time execution per processes set.

# Processes	# Balances	Total time
100	31	01:32.0
200	86	03:17.8
300	137	04:50.0
400	186	06:27.3
500	226	07:52.0
1,000	456	15:36.0
1,500	676	22:59.9

Figure 7. shows the execution times of the processes set listed in Table5. using the "Random", "Round-Robin" and HILO algorithms, in every case HILO outperforms the other two.

5 Conclusions and future work

The metric “Node-Availability” (Φ), allows performing an efficient LB without previously knowing the execution times of the processes, nor the processes communication requirements. This metric takes into account processor and memory availability every given sample and the estimation of the related communication times per processor and process respectively.

An optimization procedure based on the communications protocol performance is carried out in order to guarantee the suitability of this metric. The time values of ρ and τ obtained after this optimization

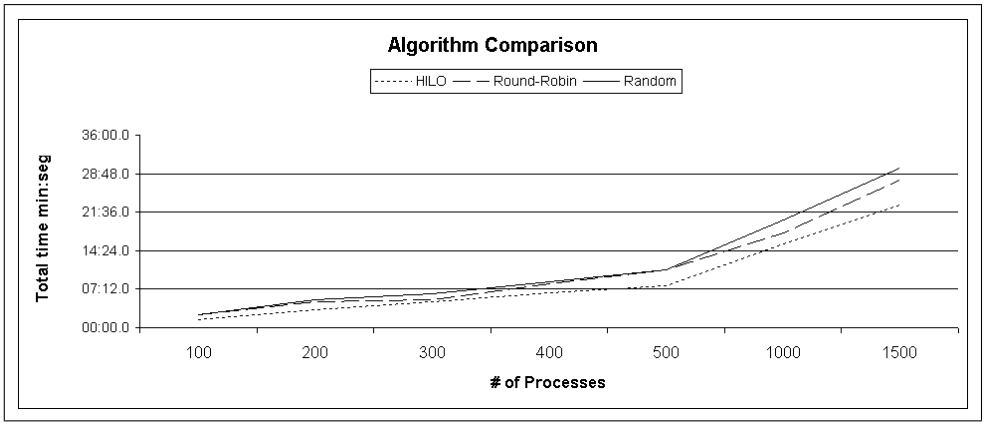


Figure 7: Total time algorithm comparison.

procedure provide the speed specifications of the communications network and the type of protocol required.

Based upon this information the proposed metric (Φ) and HILO perform an efficient response as shown in Figure 6. The results presented here provide a clear idea of the impact of Φ as the criterion metric to perform a load-balancing procedure.

For the case study taken into account, the load-balancing algorithm has a cost in terms of the time taken which can be neglected and is included in the "Total time" column of Table 6.

In terms of HILO algorithm, the ξ factor gives the possibility of balancing either in a very relaxed or strict manner. For instance, if the desired balance factor (ξ) is decreased in a significant way, the number of balances decreases towards zero. On the other hand when the value of ξ tends to 1 the system performs a load-balancing process every δ samples.

Bibliography

[1] J. Bahi, R. Couturier, F. Vernier, Synchronous distributed load balancing on dynamic networks,*Journal of Parallel and Distributed Computing*, 65 pp.1397-1405, Elsevier 2005.

[2] D. Bertsekas, *Constrained Optimization an Lagrange Multiplie Methods*, Academic Press Inc., USA 1992.

[3] J. Chiasson, Z. Tang, J. Ghanem, T. Chaouki,, J. Abdallah, D. Birdwell, M.M. Hayat, H. Jrez, The Effect of Time Delays on the Stability of Load Balancing Algorithms for Parallel Computations, *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, VOL. 13, NO. 6, NOVEMBER 2005 pp. 932-942.

[4] R. F. de Mello, L. J. Senger, L.T. Yang, A Routing Load Balancing Policy for Grid Computing Environments, *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*, 1550-445X/06 IEEE 2006.

[5] P. Ghosh, N. Roy, S.K. Das, K. Basu, A pricing strategy for job allocation in mobile grids using a non-cooperative bargaining theory framework,*Journal of Parallel and Distributed Computing*, 65 pp.1366-1383, Elsevier 2005.

[6] D. Grosu, A. Chronopoulos, Algorithmic Mechanism Design for Load Balancing in Distributed Systems,*IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICSŃPART B: CYBERNETICS*, VOL. 34, NO. 1, FEBRUARY 2004, pp. 77-84.

[7] O. Lee, M. Anshel, I. Chung, Design of an efficient load balancing algorithm on distributed networks by employing symmetric balanced incomplete block design,*IEE Proc.-Commun*, Vol. 151, No. 6, December 2004.

[8] L. Keqin, Job scheduling and processor allocation for grid computing on metacomputers, *Journal of Parallel and Distributed Computing*, 65 pp.1406-1418, Elsevier 2005.

[9] A. Menendez, H. Benitez-Perez, Node Availability for Distributed Systems considering processor and RAM utilization, *Eighth Mexican International Conference on Computer Science*, ENC07, Page(s):131 - 137, DOI:10.1109/ENC.2007.24, 2007.

[10] B. Parhami, Swapped interconection networks: Topological, performance, and robustness attributes, *Journal of Parallel and Distributed Computing*, 65 pp.1443-1452, Elsevier 2005.

[11] M. Perez, A. Sanchez, J. Pea, V. Robles, A new formalism for dynamic reconfiguration of data servers in a cluster, *Journal of Parallel and Distributed Computing*, 65 pp.1134-1145, Elsevier 2005.

[12] H. Sit, K. Ho, H. V. Leong, W. P. R. Luk, L. Ho, An Adaptive Clustering Approach to Dynamic Load Balancing, *Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks (ISPANŐ04)* 1087-4089 IEEE 2004.

[13] D. Takemoto, S. Tagashira, S. Fujita, Partitioning in Content-Addressable Networks Distributed Algorithms for Balanced Zone, *Proceedings of the Tenth International Conference on Parallel and Distributed Systems (ICPADSŐ04)* 1521-9097 IEEE 2004.

[14] Torque Resource Manager http://www.clusterresources.com/pages/products/torque-resource-manager.php 2006.

[15] Z. Zeng, B. Veeravalli, Rate-Based and Queue-Based Dynamic Load Balancing Algorithms in Distributed Systems, *Proceedings of the Tenth International Conference on Parallel and Distributed Systems*, 1521-9097/04 IEEE 2004.

[16] Liu W.S. Jane, *Real-Time Systems*, Prentice Hall, USA, 2000.

Antonio Menéndez LC is a computer science engineer from the Universidad La Salle and currently PhD candidate by the Universidad Nacional Autónoma de México (UNAM). More than 25 years of experience in the Computer and technology industries, leading multi-million projects, as well as in the academic world. Devoted to research for the last years has participate in international congresses of real-time, convergence, hybrid systems, and Computer Science.

Héctor Benítez Pérez is a full time researcher in the IIMAS UNAM (México). He obtained his BSc in electronic engineering at the Engineering Faculty UNAM in 1994 and his PhD at Sheffield University, UK en 1999. His areas of interest are in Real Time Control and Fault Diagnosis.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 3, pp. 351-361

Improving a SVM Meta-classifier for Text Documents by using Naive Bayes

D. Morariu, R. Creţulescu, L. Vinţan

Daniel Morariu, Radu Creţulescu, Lucian Vinţan
"Lucian Blaga" University of Sibiu
Engineering Faculty,Computer Science Department
E. Cioran Street, No. 4, 550025 Sibiu, ROMANIA
E-mail: {daniel.morariu,radu.kretzulescu,lucian.vintan}@ulbsibiu.ro

Abstract: Text categorization is the problem of classifying text documents into a set of predefined classes. In this paper, we investigated two approaches: a) to develop a classifier for text document based on Naive Bayes Theory and b) to integrate this classifier into a meta-classifier in order to increase the classification accuracy. The basic idea is to learn a meta-classifier to optimally select the best component classifier for each data point. The experimental results show that combining classifiers can significantly improve the classification accuracy and that our improved meta-classification strategy gives better results than each individual classifier. For Reuters2000 text documents we obtained classification accuracies up to 93.87%.
Keywords: Meta-classification, Support Vector Machine, Naive Bayes, Text document and Performance Evaluation

1 Introduction

WHILE more and more textual information is available online, effective retrieval is difficult without good indexing and summarization of document content. Document categorization is one solution to this problem. The task of document categorization is to assign a user defined categorical label to a given document. In recent years a growing number of categorization methods and machine learning techniques have been developed and applied in different contexts.

Documents are typically represented as vectors in a features space. Each word in the vocabulary is represented as a separate dimension. The number of occurrences of a word in a document represents the value of the corresponding component in the document's vector.

In this paper we investigate some strategies for combining classifiers in order to improve the classification accuracy. We used classifiers based on Support Vector Machine (SVM) techniques and based on Naive Bayes Theory, respectively. They are less vulnerable to degrade with an increasing dimensionality of the feature space, and have been shown effective in many classification tasks. The SVM classifiers are actually based on learning with kernels and support vectors.

We combine multiple classifiers hoping that the classification accuracy can be improved without a significant increase in response time. Instead of building only one highly accurate specialized classifier with much time and effort, we build and combine several simpler classifiers.

Several combination schemes have been described in the papers [2] and [6]. A usually approach is to build individual classifiers and later combine their judgments to make the final decision. Another approach, which is not so commonly used because it suffers from the "curse of dimensionality" [5], is to concatenate features from each classifier to make a longer feature vector and use it for the final decision. Anyway, meta-classification is effective only if classifiers' synergism can be exploited.

In previous studies combination strategies were usually ad hoc and are implementing strategies like majority vote, linear combination, winner-take-all [2], or Bagging and Adaboost [16]. Also, some rather complex strategies have been suggested; for example in [4] a meta-classification strategy using SVM [15] is presented and compared with probability based strategies.

Section 2 and 3 contains prerequisites for the work that we present in this paper. In sections 4 we present the methodology used for our experiments. Section 5 presents the experimental framework and section 6 presents the main results of our experiments. The last section debates and concludes on the most important obtained results and proposes some further work.

2 Support Vector Machine

The Support Vector Machine (SVM) is a classification technique based on statistical learning theory [13], [15] that was applied with great success in many challenging non-linear classification problems and on large data sets. The SVM algorithm finds a hyperplane that optimally splits the training set. The optimal hyperplane can be distinguished by the maximum margin of separation between all training points and the hyperplane. Looking at a two-dimensional problem we actually want to find a line that "best" separates points in the positive class from points in the negative class. The hyperplane is characterized by a decision function like:

$$f(x) = sign(\langle \vec{w}, \Phi(x) \rangle + b) \tag{1}$$

where $\vec{w}$ is the weight vector, orthogonal to the hyperplane, "b" is a scalar that represents the hyperplane's margin, "x" is the current sample tested, "$\Phi(x)$" is a function that transforms the input data into a higher dimensional feature space and $\langle \cdot, \cdot \rangle$ represents the dot product. *Sign* is the sign function. If $\vec{w}$ has unit length, then $\langle \vec{w}, \Phi(x) \rangle$ is the length of $\Phi(x)$ along the direction of $\vec{w}$. Generally $\vec{w}$ will be scaled by $\|\vec{w}\|$. In the training part the algorithm needs to find the normal vector "$\vec{w}$" that leads to the largest "b" of the hyperplane.

3 Naive Bayes

The Bayes classifier uses the Bayes Theorem which basically computes prior probabilities for a given class based on the probability for a given term to belong to the specified class. The classifier computes the probability for a document to be into a given class.

Bayesian theory woks as a framework for making decision under uncertainty - a probabilistic approach to inference [4] and is particularly suited when the dimensionality of the inputs data is high. Bayes theorized that the probability of future events could be calculated by determining their earlier frequency. Bayes theorem states that:

$$P(Y = y_i \mid X = x_k) = \frac{P(Y = y_i)P(X = x_k \mid Y = y_i)}{P(X = x_k)} \tag{2}$$

where:

$P(Y = y_i)$ - Prior probability of hypothesis Y- Prior

$P(X = x_k)$ - Prior probability of training data X-Evidence

$P(X = x_k \mid Y = y_i)$ - Probability of X given Y- Likelihood

$P(Y = y_i \mid X = x_k)$ - Probability of Y given X- Posterior probability.

The Naive Bayes classifier is based on the simplifying assumption that the attribute values are conditionally independent given target value. In other words the assumption is that, given the target value of the instance, the probability of observing the conjunction $y_1, y_2...y_n$ is just the product of the probabilities for the individual attributes:

$$c_{map} = argmax_{1<i<m} \bar{P}(X_i \mid Y) = argmax_{1<i<m} \bar{P}(X_i) \prod_{j=1}^{n} \bar{P}(y_j \mid X_i) \tag{3}$$

We used the notation $\bar{P}$ for P because we don't know exactly the values of the parameters $\bar{P}(X_i)$ and $\bar{P}(y_j \mid X_i)$. These values can be estimated based on the training set.

We can calculate $\bar{P}(X_i) = \frac{|D_i|}{|D|}$, where D is the set of documents and D_i is a subset of D for each category X_i contained in the category set X.

For training the classifier we consider V as the words vocabulary from documents contained in D, and for any category $X_i \in X$ there is a subset of documents contained in D that belongs to X_i category.

Let Y_i a vector that contains all words extracted form documents from D_i set and n_i numbers of all words occurrence from Y_i. Thus for each word $y_i \in V$ we noted with n_{ij} the total number of word occurrence y_i in Y_i. We can write:

$$\bar{P}(y_j \mid X_i) = \frac{n_{ij} + 1}{n_i + |V|} \tag{4}$$

So, basically the Naive Bayes classifier ignores the possible dependencies, correlations, among the inputs and reduces a multivariate problem to a group of simple independent problems. It is noticed that in a Naive Bayes classifier the number of distinct $P(y \mid X_j)$ terms that must be estimated from the training data is just the number of distinct attribute values multiplied by the number of distinct target values. Therefore it represents a much smaller number than if we would estimate all the $P(y_1, y_2 ... y_n \mid X_j)$ terms that are needed for Bayesian theory.

For extending the SVM and the Naive Bayes classifiers from two-class classification to multi-class classification, typically one of two methods are used: "One versus the rest", where each topic is separated from the remaining topics, and "One versus the one", where a separate classifier is trained for each class pair as in [17]. We selected the first method for two reasons: first, preliminary experiments shows that the this method gives better performance, which might be explained by the fact that the Reuter's database contains strongly overlapping classes and assigns almost all samples in more than one class. Second, overall training time is much shorter for the first method.

4 Meta-Classifier Models

In [12] we presented a meta-classifier based on 8 SVM classifiers that was used to improve the accuracy of the classification for text documents. We use the 3 models presented in [11] to test the classification accuracy of the meta-classifier in the case of introducing a new classifier based of naive Bayes theory. The 3 models are: majority vote, the selection based on the Euclidian distance and selection based on the cosine angle.

4.1 Majority Vote (MV)

This first model for meta-classification is a maladjusted model that obtains the same results in time. The idea is to use all selected classifiers to classify the current document. Each classifier proposes a specified class for this document incrementing the corresponding class-counter. The MV meta-classifier will select the class with the greatest count. If two or more classes with identical counts are obtained we classify the current document in all proposed classes.

4.2 Selection based on Euclidean distance (SBED)

This model selects a classifier based on the current input data. It will learn only data that is incorrectly classified by the selected classifier, because we are expecting to have a smaller number of incorrectly classified input data than the number of correctly classified input data. Thus we create for each classifier a queue which contains all incorrectly classified documents.

When we have an input document (current sample) that needs to be classified, first we randomly chose one classifier. After that we compute the Euclidean distances (equation 5) between the current sample and all the samples that are in that self queue of the selected classifier. If we obtain at least one distance smaller than a predefined threshold we will reject this classifier. Instead we will randomly select another classifier, except for the already rejected one. If all component classifiers are rejected, however, we'll choose the classifier with the greatest Euclidian distance.

$$Eucl(\mathbf{x},\mathbf{x'}) = \sqrt{\sum_{i=1}^{n} ([x]_i - [x']_i)^2} \tag{5}$$

where $[x]_i$ represents the value from entry i of the vector **x**, and **x** and **x'** represent the input vectors.

After selecting the optimal classifier we'll used it to classify the current sample. If the selected classifier succeeds to correctly classify the current document, nothing is done. Otherwise we'll put the current document into the corresponding queue. To see if the document is correctly or incorrectly classified we compare our proposed (computed) class with the Reuters proposed class.

The training is as follows: the meta-classifier analyzes the training set and each time when a document is incorrectly classified, the pattern is added to the selected classifier queue. In the second step, the validation step, we test the classification accuracy using the validation set. In the testing step the characteristics of the meta-classifier remains unchanged. Because after each training part the characteristics of meta-classifier might change, we repeated these two steps many times. After 14 steps we obtain good results and the classification accuracy have not substantially increasing after that.

4.3 Selection based on cosine angle (SBCOS)

The cosine angle is another possibility to compute the document similarity, often used to calculate text similarities. The formula to compute the cosine angle θ between two input vectors x and x' is:

$$cos\theta = \frac{\langle x, x' \rangle}{|x| \cdot |x'|} = \frac{\sum_{i=1}^{n} [x]_i [x']_i}{\sqrt{\sum_{i=1}^{n} [x]_i^2} \cdot \sqrt{\sum_{i=1}^{n} [x']_i^2}} \tag{6}$$

where $[x]_i$ represents the value from entry i of a vector x.

This method is similar with the SBED method with two modifications. Obvious, the first modification is that the similarity between documents is computed using the cosine angle between input vectors. The second modification is that the classifier is not randomly selected. Instead doing this we constantly take into consideration all available classifiers. We compute the cosine between the current sample and all incorrect classified samples that are in the self queues. It is chosen the classifier that obtains the maximum cosine value.

5 Experimental Framework

5.1 The Dataset

Our experiments are performed on the Reuters-2000 collection [14], which has 984Mb of newspapers articles in a compressed format. Collection includes a total of 806,791 documents, with news stories published by Reuters Press covering the period from 20.07.1996 through 19.07.1997. The articles have 9822391 paragraphs and contain 11522874 sentences and 310033 distinct root words. Documents are pre-classified according to 3 categories: by the Region (366 regions) the article refers to, by Industry Codes (870 industry codes) and by Topics proposed by Reuters (126 topics, 23 of them contain no articles). Due to the huge dimensionality of the database we will present here results obtained using a subset of data. From all documents we selected the documents for which the industry code value is

equal to "System software". We obtained 7083 files that are represented using 19038 features and 68 topics. A document is represented as a vector of words, applying a stop-word filter (from a standard set of 510 stop-words) and extracting the word stem [1]. Entire set of 7083 documents is represented as word frequency matrix where each row represents a single document and each column represents a single word [3]. From these 68 topics we have eliminated those topics that are poorly or excessively represented. Thus we eliminated those topics that contain less than 1% documents from all 7083 documents in the entire set. We also eliminated topics that contain more than 99% samples from the entire set, as being excessively represented. After doing so we obtained 24 different topics and 7053 documents, that were split randomly in training set (4702 samples) and testing set (2351 samples). In the feature extraction part we take into consideration both the article and the title of the article.

5.2 Kernel Types for SVM

The idea of the kernel trick is to compute the norm of the difference between two vectors in a higher dimensional feature space without representing those vectors in the new feature space. In practice we observed that by adding a constant bias to the kernel we obtained improved classifying results. For more details please consult [7] and [10].

We used in our selected SVM classifiers two types of kernels each of them with different parameters [10]. For the polynomial kernel we varied the degree and for the Gaussian kernel we changed the parameter C according to the following formulas (x and x' being the input vectors):

- **Polynomial**

$$k(x,x') = (2 \cdot d + \langle x \cdot x' \rangle)^d \tag{7}$$

  d being the only parameter to be modified and represents the degree of the kernel,

- **Gaussian (radial basis function RBF)**

$$k(x,x') = exp(-\frac{\|x-x'\|^2}{n \cdot C}) \tag{8}$$

  C being the classical parameter and n being the new parameter, introduced by us, representing the number of elements from the input vectors that are greater than 0.

For feature selection with Support Vector Machine method we used the polynomial kernel with degree 1 [8] and [9].

5.3 Representing the input data

Also in our selected classifier we used different representation of the input data. After extensive experiments [8], [10] we have seen that different types of kernels work better with different types of data representation. We represented the input data in three different formats. In the following formulas $n(d,t)$ is the number of times that term t occurs in document d, and $n(d,\tau)$ is the maximum frequency occurring in document d.

- **Binary representation** - in the input vector we store "0" if the word doesn't occur in the document and "1" if it occurs without considering the number of occurrences.

- **Nominal representation** - we compute the value of the weight using the formula:

$$TF(d,t) = \frac{n(d,t)}{max_\tau n(d,\tau)} \tag{9}$$

- **Cornell SMART representation** - we compute the value of the weight using the formula:

$$TF(d,t) = \begin{cases} 0 & if n(d,t) = 0 \\ 1 + \log(1 + \log(n(d,t)) & otherwise \end{cases} \quad (10)$$

These are later called as BIN, NOM or CS.

6 Experimental Results

In [11] it is presented a meta-classifier based on 8 of SVM classifiers that was used to improve the classification accuracy of text documents. The maximum classification accuracy, 87.11%, was obtained by a single SVM type classifier with a second degree for polynomial kernel and Cornell Smart data representation. In [12] there are presented and tested several types of SVM classifiers based on polynomial kernel and Gaussian kernel, with different forms of representation. From all the tested classifiers eight distinct SVM classifiers were included in the meta-classifier. The eight classifiers were chosen based on the obtained classification accuracy.

We decided to incorporate also a Bayes classifier in the meta-classifier presented above. As a result the new meta-classifier has 9 classifiers. We run again the tests for all the 3 models of the meta-classifier. We also calculated the maximum theoretical limit that could be reached by the new developed meta-classifier. Thus, the introduction of the Bayes classifier in the meta-classifier increases this maximum limit to 98.63% (against 94.21% as it was without Bayes classifier). This fact provides an opportunity to obtain better classification accuracy.

6.1 Selection based on the vote majority

Using this method, the obtained classification accuracy is 86.09%, for the 9 classifiers system. The classification accuracy dropped against the obtained value with 8 classifiers (86.38%), leading to a fall of 0.29%. This may occur because the Bayes classifier obtains - on the entire set of tests - an accuracy of only 81.32%, incorrectly classifying quite a lot of documents (439). Consequently, it seems to "help" the meta-classifier by selecting the 7 cases of wrong categories because Bayes reinforced the wrong vote. The results are shown in Fig. 1.

6.2 Selection based on Euclidian distance (SBED)

We present the results obtained with the new meta-classifier consisting of 9 classifiers in comparison with the meta-classifier presented in [11]. There are presented only the first 14 steps since after this number of steps the classification accuracy is not substantially amended. As in [11] the threshold for the first 7 steps was chosen equal to 2.5 and the threshold for the last 7 steps was chosen equal to 1.5. A step represents a training process followed by a testing process. Fig. 1 shows the results for the meta-classifiers with 8 and 9 classifiers.

For the meta-classifier with 9 classifiers, the results are weaker than those obtained by the meta-classifier with 8 classifiers. This poorer accuracy is due to the poor accuracy of the Bayes classifier (81.32%) compared with the SVM, and due to the fact that the classifiers are randomly selected as we already explained.

Besides, it could be observed that the classification accuracy for the meta-classifier with 9 classifiers has also decreasing trends. This may be due to the fact that a classifier that correctly classifies a document d_1 could incorrectly classify a document d_2 that is quite similar with d_1. From this reason - when running again the tests set - this classifier has not been selected for the classification of d_1 (because it gave poor results for d_2) and then looking for other classifier (which may in turn be classified wrong).

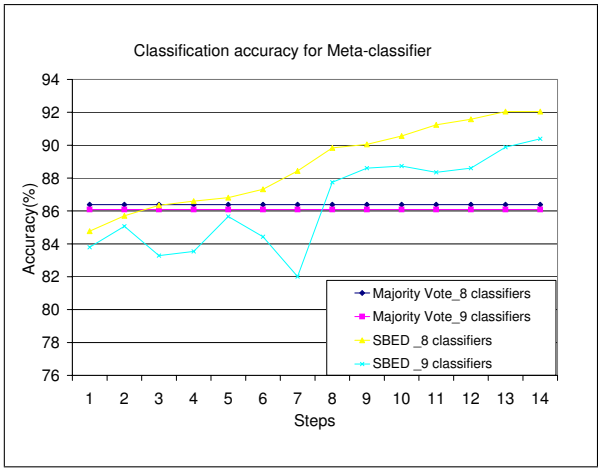


Figure 1: Meta-classifier accuracy for Majority Vote and SBED

6.3 Selection based on Cosine (SBCOS)

Fig. 2 shows the results, obtained with the new meta-classifier containing 9 classifiers compared with the meta-classifier with 8 classifiers. It presents only the first 14 steps, because after this number of steps, the classification accuracy is not substantially amended. Similar to [11], the threshold for the first 7 steps was chosen equal to 0.8 and the threshold for the last 7 steps was chosen equal to 0.9. In addition, this figure shows the maximum limit that can be achieved by the new meta-classifier (98.63%).

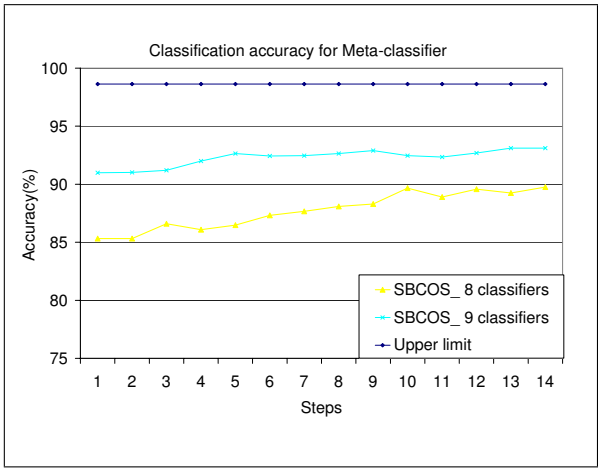


Figure 2: Meta-classifier accuracy for SBCOS and upper limit

6.4 The modification of the meta-classifier when it fails to select a class

The meta-classifier presented in [11], in the case of a document d_n that needs to be classified, takes each classifier and calculates the distance between d_n and each document from the error queue respectively. If the calculated distance is less than the established threshold, the meta-classifier will not use the classifier to classify the document d_n. On the other hand, if all classifiers are rejected by this method, the meta-classifier will choose the one that has the greatest distance achieved. Thus, the meta-classifier will compute (predict) the class specified by the classifier even if it has a high probability to be wrong. For this reason we modified the meta-classifier in such a way that when all classifiers are rejected, the "forced" selected classifier will not choose the class with the highest value (it will fail anyway because

it is prone to falsely classify that type of documents). Instead, it will choose the class immediately following in the list of classes which it predicts. This can be done because the classifier returns different values for each class. In this case, we will not select (predict) the class with the maximum value; instead of doing this, we will select (predict) the class that have the second value in the list, if the difference between the maximum value and the second one is not greater that 0.5. In this case, the classifier would specify another class for the current document d_n. By implementing this change, the results of the meta-classifier consisting of 9 classifiers have been improved. In the following paragraphs, we will call this meta-classifier: the modified meta-classifier with 9 classifiers.

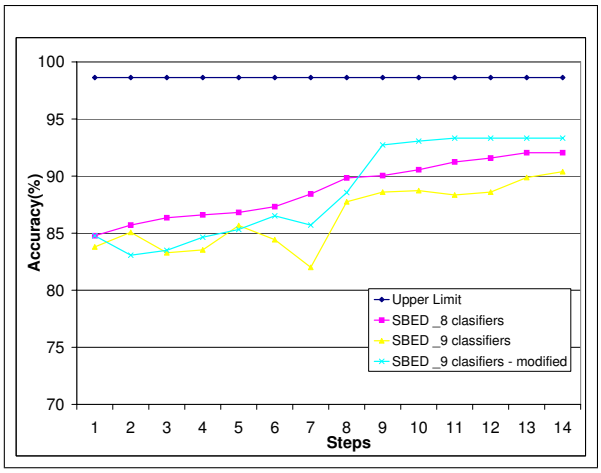


Figure 3: Classification accuracy for modified meta-classifier- SBED

Fig.3 shows the results obtained with the modified meta-classifier with 9 classifiers vs. the meta-classifier with 8 SVM classifiers and the unmodified meta-classifier with 9 classifiers respectively, all using SBED method. For the selection based on majority voting, learning not being involved, the modifications of the meta-classifier will produce no influence on the final outcome. Besides each figure it is shown the maximum limit that can be obtained by the meta-classifier with 9 classifiers.

The results obtained by the modified meta-classifier with 9 classifiers based on Euclidian distance were improved. Thus, it has been obtained a classification accuracy of 93.32% compared with the unmodified meta-classifier which achieved only 90.38%. Recall that under the same conditions, the meta-classifier with 8 SVM type classifiers [11] obtained a maximum classification accuracy of only 92.04%. In the first 7 steps, the classification accuracy of the modified meta-classifier with 9 classifiers is almost identical to that of the unmodified meta-classifier with 9 classifiers. This happens because during the first steps we can find a classifier that can be selected to classify the current document. After the first 9 steps we put in the queue of each classifier the documents with problems and only after these steps all the basic classifiers could be rejected and the proposed modification to the meta-classifier might be applied. Therefore, during the first steps the results between the two meta-classifiers are slightly different because they always choose a random classifier of the existing 9. With this random selection at different running time we obtain small different results.

Fig. 4 exposes results obtained by meta-classifier using SBCOS method.

In this case, the classification accuracy for the new meta-classifier has improved from 93.10% to 93.87%. Note that the meta-classifier with 8 SVM classifiers under the same conditions, obtained a classification accuracy of only 89.74%.

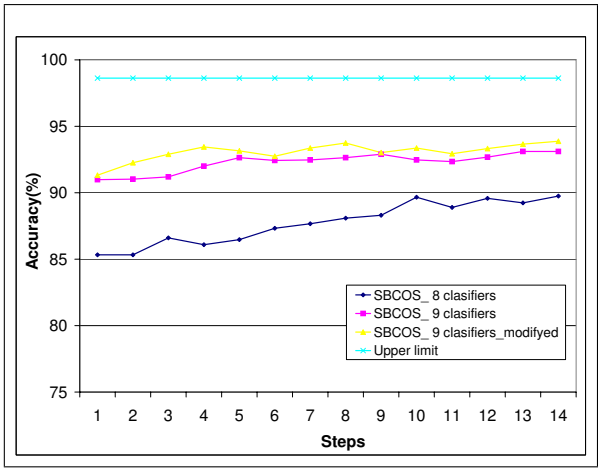


Figure 4: Classification accuracy for modified meta-classifier - SBCOS

7 Conclusions and Further Work

Building up on the meta-classifier presented in [11], based on 8 SVM components, we add to these a new Bayes type classifier which leads to a significant improvement of the upper limit that the meta-classifier can reach. Thus, the meta-classifier upper limit has increased from 94.21% when using 8 SVM classifiers [11] to 98.63% when using the 8 SVM classifiers plus the Bayes classifier.

Furthermore, we presented results for all three models of meta-classifiers: majority voting (MV), selection based on Euclidian distance (SBED) and selection based on Cosine (SBCOS). In the case of MV, we obtained a classification accuracy of only 86.09%, which is 0.29% lower than when using only 8 classifiers.

Moreover, in the case of the 9-classifiers SBED meta-classifier we obtained even lower results, on average dropping from 92.04% to 90.38%. In the case of the 9-classifiers SBCOS, the classification accuracy of the meta-classifier has increased from 89.74% to 93.10%.

Finally, we considered that if there is any suspicion that the class to be predicted will not be the correct one, then the classifier should predict a different class. The latest will be the next class in the list of classes only if it is sufficiently close to the first predicted class. These change led to a substantial improvement of the meta-classifier with 9 classifiers. We have performed only experiments with the meta-classifier with 9 classifiers because only in this situation a maximum accuracy of 98.63% could be reached. In the case of the SBED meta-classifier, we obtained an average classification accuracy of 93.32%. This accuracy is with 2.94% greater than the best accuracy obtained without changing the class selection method. In the case of the augmented SBCOS meta-classifier we similarly improved the accuracy from 93.10

In the future, an interesting natural extension of our work may be a more adaptive and intelligent meta-classifier that uses a neural network for choosing the classifier that will be used in classifying the current document. The meta-classifier could learn the incorrect / correct document classified and, without using the error queues, could optimally and faster select the optimal classifier that will be used.

Acknowledgements This work was partially supported by the Romanian National Council of Academic Research (CNCSIS) through the grant CNCSIS no. 485/2009-2011.

Bibliography

[1] S. Chakrabarti, Mining the Web- Discovering Knowledge from hypertext data, *Morgan Kaufmann Press*, 2003.

[2] N. Dimitrova, L. Agnihotri and G. Wei, Video Classification Based on HMM Using Text and Face, *Proceedings of the European Conference on Signal Processing*,Vol. XVII, pp. 1373-1376, Finland, 2000.

[3] J. Engler, A. Kusiak, Mining Authoritativeness of Collaborative Innovation Partners, *International Journal of Computers, Communications and Control*, Vol. V, No. 1, pp. 42-51, 2010.

[4] D. Lewis, Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval, *ATT Lab Research*, NJ, Vol. 1398, pp. 4-15, USA, 1998.

[5] W.H. Lin, A. Houptmann, News Video Classification Using SVM-based Multimodal Classifier and Combination Strategies, *Proceedings of the tenth ACM international conference on Multimedia*, pp. 323-326, 2002.

[6] W.H. Lin , R. Jin, A. Houptmann, A Meta-classification of Multimedia Classifiers, *International Workshop on Knowledge Discovery in Multimedia and Complex Data*, Taiwan, 2002.

[7] D. Morariu, L. Vintan, A Better Correlation of the SVM kernel's Parameters, *Proceeding of the 5th RoEduNet International Conference*, Sibiu, pp. 244-249, June 2006.

[8] D. Morariu, L. Vintan, V. Tresp, Feature Selection Methods for an Improved SVM Classifier, *Proceedings of the 14th International Conference on Computational and Information Science*, pp. 83-89, Prague, August 2006.

[9] D. Morariu, L. Vintan, V. Tresp, Evolutionary Feature Selection for Text Documents Using the SVM, *Proceeding of the 3rd International Conference on Neural Computing and Patter Recognition*, pp. 215-221, Barcelona, October 2006.

[10] D. Morariu, Classification and Clustering using Support Vector Machine, *2nd PhD Report, University "Lucian Blaga" of Sibiu*, September, 2005, http://webspace.ulbsibiu.ro/daniel.morariu/html/Docs /Report2.pdf.

[11] D. Morariu, L. Vintan, V. Tresp, Meta-Classification using SVM Classifiers for Text Documents, *The 3rd International Conference on Neural Computing and Patter Recognition*, pp. 222-227, Barcelona, October 2006.

[12] D. Morariu, Text Mining Methods based on Support Vector Machine, *MatrixRom*, Bucharest, 2008.

[13] C. Nello, J. Swawe-Taylor, An introduction to Support Vector Machines, *Cambridge University Press*, 2000.

[14] Reuters Corpus: http://about.reuters.com/researchandstandards/corpus/. Released in November 2000.

[15] B. Schoelkopf, A. Smola, Learning with Kernels. Support Vector Machines, *MIT Press*, London, 2002.

[16] G. Siyang, L. Quingrui, M. Lin, Meta-classifier in Text Classification, http://www.comp.nus.edu.sg/ zhouyong/papers/cs5228project.pdf.

[17] R. Stoean, C. Stoean, M. Preuss, D. Dumitrescu, Evolutionary Multi-class Support Vector Machine for Classification, *International Journal of Computers, Communications and Control, 1(S):423-428, 2006*.

Daniel I. Morariu PhD, was born at September 17th 1974 in Sighişoara, Romania. He graduates "Lucian Blaga" University of Sibiu, obtaining a M.Sc. in Computer Engineering, and a Ph.D. in Computer Science from the same university. The PhD title is "Contributions to Automatic Knowledge Extraction from Unstructured Data", PhD supervisor Professor Lucian N. VINTAN. The PhD program was partially supported from scientific and financial point of view by SIEMENS Corporate Technology from Munich. At present he is a full-time lecture at "Lucian Blaga" University of Sibiu, Engineering Faculty, Computer Science department. He published over 12 scientific papers in international conference from Romania, Czech Republic, and Spain.

R. Creţulescu was born at August 8th 1968 in Sibiu, Romania. He graduates "Babeş-Bolyai" University of Cluj-Napoca and obtained a M.Sc. in Computer Engineering, from the "Lucian Blaga" University of Sibiu. At the present time he is a full-time lecture at University of Sibiu, Engineering Faculty, Computer Science department. He published over 6 scientific papers in international conferences from Romania, Finnland and Greece.

Profesor Lucian N. Vinţan PhD ("Lucian Blaga" University of Sibiu, RO) is an active researcher in Advanced Computer Architecture, Context Prediction in Ubiquitous Computing Systems, Text Documents Classification, etc. He is a member of Academy of Technical Sciences from Romania, European Commission Expert in Computer Science, and Visiting Researcher Fellow at University of Hertfordshire, UK. Professor Vintan published 8 books and over 100 scientific papers (Romania, USA, Italy, UK, Portugal, Hungary, Austria, Germany, Poland, China etc.). For his merits, Professor Vintan obtained the "Tudor Tanasescu" Romanian Academy Award. He introduced some well-known original architectural concepts in Computer Architecture domain (Dynamic Neural Branch Prediction, Pre-Computed Branches, Value Prediction focused on CPU's Context, etc.), recognized, cited and debated through over 80 papers published in many prestigious international conferences and scientific reviews (ACM, IEEE, IEE, etc.)

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 3, pp. 362-374

Fuzzy Filtering of Sensors Signals in Manufacturing Systems with Time Constraints

A. Mhalla, N. Jerbi, S. C. Dutilleul, E. Craye, M. Benrejeb

Anis M'halla, Nabil Jerbi, Mohamed Benrejeb
Ecole Nationale d'Ingénieurs de Tunis
Unité de recherche LARA-Automatique
BP 37, Le Belvédère, 1002 Tunis, Tunisie
E-mail: anis.mhalla@enim.rnu.tn, nabil.jerbi@isetso.rnu.tn, mohamed.benrejeb@enit.rnu.tn

Anis M'halla, Simon Collart Dutilleul, Etienne Craye ** Ecole Centrale de Lille
Laboratoire d'Automatique, Génie Informatique et Signal
Cité Scientifique, BP 48, 59651 Villeneuve d'Ascq, France
E-mail: simon.collart_dutilleul@ec-lille.fr, etienne.craye@ec-lille.fr,

Abstract: The presented work is dedicated to the supervision of manufacturing job-shops with time constraints. Such systems have a robustness property towards time disturbances. The main contribution of this paper is a fuzzy filtering approach of sensors signals integrating the robustness values. This new approach integrates a classic filtering mechanism of sensors signals and fuzzy logic techniques. The strengths of these both techniques are taken advantage of the avoidance of control freezing and the capability of fuzzy systems to deal with imprecise information by using fuzzy rules. Finally, to demonstrate the effectiveness and accuracy of this new approach, an example is depicted. The results show that the fuzzy approach allows keeping on producing, but in a degraded mode, while providing the guarantees of quality and safety based on expert knowledge integration.
Keywords: Alarm filtering, fuzzy logic, symptoms generation, robustness, time constraints, manufacturing.

1 Introduction

In general, the detection of failure symptoms related to the process elements requires a development of a system model to be supervised [1]. This model can be either a normal functioning model or dysfunction model. If a model is adopted, two basic mechanisms are used for detection. The first one consists of comparing the evolutions of the observed system with those of the process model, or with those of normal functioning signatures evolving in real time with the system. The second one is based on observing known failures signatures. These signatures describe historical or theoretical failures known from the process or the process elements. Without the supervision of a system model, the adopted strategy consists of an exploitation of the information given by the sensors and the detectors at a local level of the process [2]. Sometimes, in manufacturing workshops with time constraints, the information given by sensors signals is dubious and the symptoms generated are vague. Furthermore, the validation interval associated to each sensor signal is not always exact, this is the case where the temporal windows are badly defined. These reasons bring us to use fuzzy logic which is based on an approximate reasoning able to take into account the uncertainty and the inaccuracy of knowledge. This paper is an extension of Jerbi work [3]. In [3] was proposed an integration of the robustness in the filtering mechanism of sensors signals. This mechanism, presented by Toguyeni in 1992, aims at generating symptoms for the diagnosis [4]. Our main contribution is a fuzzy filtering mechanism of sensors signals. This paper is organised as follows. The first part summarizes the proposed filtering mechanism of sensors signals taking into account the robustness intervals. The second part introduces a new fuzzy filtering approach where

fuzzy logic and filtering of sensors signals techniques are integrated. In order to show the effectiveness of this approach, in the third part, an illustrative example is outlined and the results are discussed. Finally, conclusions of this work are given.

2 Robustness integration in the filtering of sensors signals

2.1 Symptoms generation

Permanently, the state of the process model is updated by the evolutions caused by the control and the sensors signals. These sensors signals are sent by the controlled system in response to a control request. The mechanism developed for the detection of failures symptoms is based on the impact study of sensors signals on the process model, called reference model, and on that of the control. These two models do not make it possible to characterize all failures symptoms of a controlled system, for example, the absence of sensors signals (this can be the case, when a sensor is not functional or the control request was not carried out). In order to take into account these problems, mechanisms of "watchdogs" were integrated in the control and process models. These mechanisms are based on two dates provided by the scheduling task: the beginning date as soon as possible noted $\Delta t_{m/CRi}$ and the completion date noted $\Delta t_{M/CRi}$ of the control operation [4]. The idea consists of modelling any operation from a temporal approach. At each operation A_i is associated a sensor signal CR_i. To each sensor signal CR_i is associated a temporal interval $[\Delta t_{m/CRi},\ \Delta t_{M/CRi}]$ (figure 1). The report CR_i is valid only inside this window. $\Delta t_{m/CRi}$ and $\Delta t_{M/CRi}$ are defined relatively to the beginning of the operation A_i (Start-Event). The filtering principle is to position the temporal window of each sensor signal CR_i when its Start-Event was received.Two types of symptoms are distinguished:

- Symptoms type I noted S_i^1: This class of symptoms corresponds to awaited sensor signal which is not received at $\Delta t_{M/CRi}$. The detection mechanism of this symptom type corresponds to the traditional mechanism of watchdog, but implemented in a separate way of the control.
- Symptoms type II noted S_i^2: It is generated by the occurrence of a sensor signal which is not expected. Two cases are considered, the first one corresponds to an action but its report occurs before the validation interval. The second one corresponds to the occurrence of a report in absence of any control which can create it.

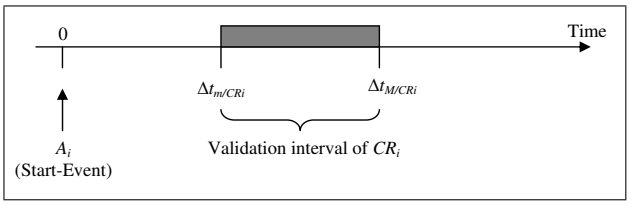


Figure 1: Operation associated model [4]

2.2 Symptoms generation

The robustness of a system can be defined as its ability to preserve the specifications facing some expected or unexpected variations. So, the robustness of a system characterizes its capacity to deal with disturbances [6]. It is interpreted into different specializations. The passive robustness is based upon variations included in validity time intervals.There is no control loop modification to preserve the required specifications. On the other hand, active robustness uses observed time disturbances to modify the control loop in order to satisfy these specifications. Therefore, the robustness intervals must be

integrated in the filtering mechanism of sensors signals. In [3], a filtering mechanism of sensors signals integrating the robustness values is proposed. This mechanism allows the integration of the robustness results in the symptoms generation and the classification of the various actions reports. This classification is very useful for the supervision because it makes it possible to preserve the production function. This constitutes an enhancement of the filtering mechanism. Five time intervals, figure 2, are defined, namely: $I_{1i} = [\Delta t_{m''/CR_i},\ \Delta t_{m'/CR_i}[$, $I_{2i} = [\Delta t_{m'/CR_i},\ \Delta t_{m/CR_i}[$, $I_{3i} = [\Delta t_{m/CR_i},\ \Delta t_{M/CR_i}[$, $I_{4i} = [\Delta t_{M/CR_i},\ \Delta t_{M'/CR_i}[$ and $I_{5i} = [\Delta t_{M'/CR_i},\ \Delta t_{M''/CR_i}[$.
The margin of passive robustness is available in $I_{2i} \cup I_{4i}$ whereas the margin of active robustness is in $I_{1i} \cup I_{5i}$. Several cases can arise:

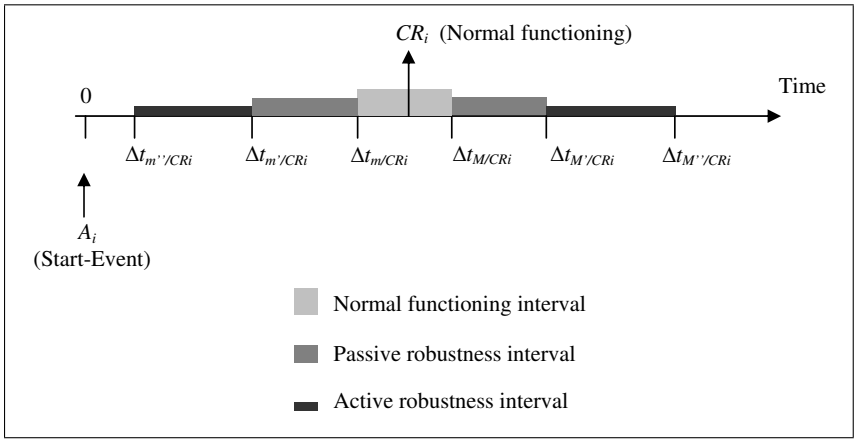


Figure 2: Robustness integration in the operation associated model [3]

- If there are absence of order (not A_i) and presence of CR_i, there are freezing of the control and generation of a symptom S_i^2.
- If the sensor signal CR_i arrives in the time interval $[0,\ \Delta t_{m''/CR_i}[$, there are freezing of the control and generation of a symptom S_i^2.
- If the sensor signal CR_i arrives in the time interval I_{1i}, there are change of the control (active robustness to an advance) and memorizing a symptom S_i^2.
- If the sensor signal CR_i arrives in the time interval I_{2i}, there is no change of the control (passive robustness to an advance) but only a memorizing of a symptom S_i^2.
- If the sensor signal CR_i arrives in the time interval I_{3i}, the behavior of the system is normal.
- At the instant $\Delta t_{M/CR_i}$, there is automatically memorizing of a symptom S_i^2.
- If the sensor signal CR_i arrives in the time interval I_{4i}, it is a case of passive robustness to a delay. The symptom S_i^1 is already memorized.
- If the sensor signal CR_i arrives in the time interval I_{5i}, a change of the control is necessary (active robustness to a delay).
- At the instant $\Delta t_{M''/CR_i}$, there is freezing of the control.

Therefore, the robustness intervals are integrated in the filtering mechanism of sensors signals. It makes it possible to continue the production in a degraded mode. However the assumptions formulated in [3] are very restrictive. It is natural to consider different scenarios where the temporal specifications of the

process are not fulfilled, nevertheless the production can continue. The next section presents a fuzzy filtering mechanism of sensors signals which introduces a finer classification of abnormal functioning and integrates the vague knowledge of the robustness intervals in the interpretation of sensors signals, coming from the workshop, for the generation of symptoms. The objective is to avoid the freezing of the control when the time disturbance is in the robustness intervals.

3 Fuzzy filtering of sensors signals

3.1 Introduction

Fuzzy logic is a mathematical tool that allows us to approach an unknown function by means of linguistic descriptions. Nevertheless, the linguistic information is a feature of the human reasoning and not of the mechanical components or programs. In consequence, this tool has achieved widespread applications and success in many areas such as control, supervision, image filtering and communications [6–9]. The essential characteristics of fuzzy logic are as follows [10]:

- Exact reasoning is viewed as a limiting case of approximate reasoning.
- Everything is a matter of degree.
- Any logical system can be fuzzified.
- Knowledge is interpreted as a collection of elastic or, equivalently, fuzzy constraint on a collection of variables.
- Inference is viewed as a process of propagation of elastic constraints.

Fuzzy logic calls upon a base of dubious knowledge, modelled by the sequence of fuzzy rules. This technique seems very promising thanks to its potential of use in dynamic monitoring and supervision, with the possibility of remaining human operator, by taking into account its way of reasoning and offering an interesting traceability. Before proceeding, we define some important terms.

Definition 1. [11]: A fuzzy set F in a universe of discourse U is characterized by a membership function $\mu_F : U \rightarrow [0,1]$

Definition 2. [11]: A linguistic variable x in a universe of discourse U is characterized by: $T(x) = \left\{T_x^1, T_x^2, ..., T_x^k\right\}$ and $M(x) = \left\{M_x^1, M_x^2, ..., M_x^k\right\}$ where $T(x)$ is the term set of x, that is the set of names of linguistic values of x with each value T_x^i being a fuzzy number with membership function M_x^i on U.

3.2 Basic Structure of a Fuzzy System

Figure 3, shows the basic structure of a conventional fuzzy system. Such system can be seen as consisting of four basic building blocks: Fuzzification, Fuzzy rule set, Inference method and Defuzzification. Let us examine these building blocks in details:

- The fuzzification transforms a numerical input variable in a fuzzy set described by linguistic expressions.
- Fuzzy rules set: the fuzzy IF-THEN rule expresses a fuzzy implication relation between the fuzzy sets of the premise and the fuzzy sets of the conclusion.
- The inference makes it possible to implement, on the basis of fuzzy rules, the logical dependence between input variables and output fuzzy variables [12].

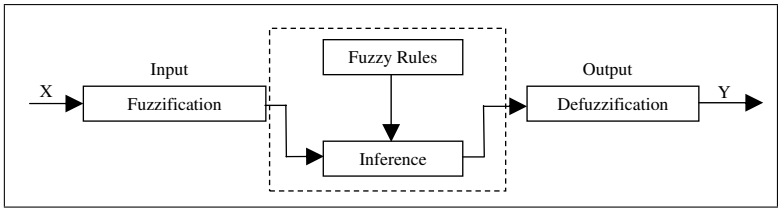


Figure 3: Fuzzy set stages

- The defuzzification transforms the output fuzzy set in a numerical variable.

Following the above definitions, the input vector X which includes the input state linguistic variables x_i's, and the output state vector Y which includes the output state linguistic variables y_i's, can be defined as:

$$X = \left[x_i, U_i, \left\{\mathrm{T}^1_{\mathrm{x_i}}, \mathrm{T}^2_{\mathrm{x_i}}, ..., \mathrm{T}^{\mathrm{k}}_{\mathrm{x_i}}\right\}, \left\{\mathrm{M}^1_{\mathrm{x_i}}, \mathrm{M}^2_{\mathrm{x_i}}, ..., \mathrm{M}^{\mathrm{k}}_{\mathrm{x_i}}\right\}\right]_{\ i=[1,...,n]} \quad (1)$$

$$Y = \left[y_i, U_i, \left\{\mathrm{T}^1_{\mathrm{y_i}}, \mathrm{T}^2_{\mathrm{y_i}}, ..., \mathrm{T}^{\mathrm{k}}_{\mathrm{y_i}}\right\}, \left\{\mathrm{M}^1_{\mathrm{y_i}}, \mathrm{M}^2_{\mathrm{y_i}}, ..., \mathrm{M}^{\mathrm{k}}_{\mathrm{y_i}}\right\}\right]_{\ i=[1,...,m]} \quad (2)$$

The fuzzifier, in figure 3, is a mapping from an observed input space to fuzzy sets in certain input universe of discourse. So, a specific value $x_i(t)$ at the time t is mapped to the fuzzy set T^1_x with degree $\mathrm{M}^1_x(x_i(t))$ and to the fuzzy set T^2_x with degree $\mathrm{M}^2_x(x_i(t))$ and so on.

Fuzzification of the input and output variables

If we want to introduce linguistic information, we have to define an interface. This interface is denominated fuzzification, and it translates the sensor measurements into linguistic concepts. To carry out such transformation, the fuzzification resorts to a characteristic function called membership function. The aim of fuzzification is to produce initial membership functions. Therefore, the universe of discourse U of the input and output variables are divided into fuzzy subsets.
The first step consists of choosing the input and output variables. This choice depends on the parameters available and the type of application [13]. In order to produce initial membership functions, the input and output spaces are divided into fuzzy regions. In our example, we have two inputs and two outputs variables, all membership functions are represented by trapezoidal forms. The sensor signal (CR_i) and the occurrence of the Start-Event (A_i) are considered as inputs variables. However, the type of symptom (S_i) and the Control Decision (CD) are considered as outputs ones.
The second step consists of defining the universe of discourse which can take each variable. Then, we define the fuzzy sets associated to the inputs and outputs variables and their corresponding membership functions. Thus, the universe of discourse is divided into intervals at which a descriptive label is associated. This last choice is based on the experiment of the operator.

- **Fuzzification of sensor signal (CR_i)**

We define, figure 4, thirteen time intervals, namely: $I_{1i} = [0,\ \Delta t_{m''/CR_i}[$, $I_{2i} = [\Delta t_{m''/CR_i},\ \Delta t_{m''_1/CR_i}[$, $I_{3i} = [\Delta t_{m''_1/CR_i},\ \Delta t_{m'_1/CR_i}[$, $I_{4i} = [\Delta t_{m'_1/CR_i},\ \Delta t_{m'_2/CR_i}[$, $I_{5i} = [\Delta t_{m'_2/CR_i},\ \Delta t_{m_1/CR_i}[$, $I_{6i} = [\Delta t_{m_1/CR_i},\ \Delta t_{m_2/CR_i}[$, $I_{7i} = [\Delta t_{m_2/CR_i},\ \Delta t_{M_1/CR_i}[$, $I_{8i} = [\Delta t_{M_1/CR_i},\ \Delta t_{M_2/CR_i}[$, $I_{9i} = [\Delta t_{M_2/CR_i},\ \Delta t_{M'_1/CR_i}[$, $I_{10i} = [\Delta t_{M'_1/CR_i},\ \Delta t_{M'_2/CR_i}[$, $I_{11i} = [\Delta t_{M'_2/CR_i},\ \Delta t_{M''/CR_i}[$, $I_{12i} = [\Delta t_{M''/CR_i},\ \Delta t_{M''_1/CR_i}[$ and $I_{13i} = [\Delta t_{M''_1/CR_i},\ +\infty[$.
The full set intervals is summarised in table 1.

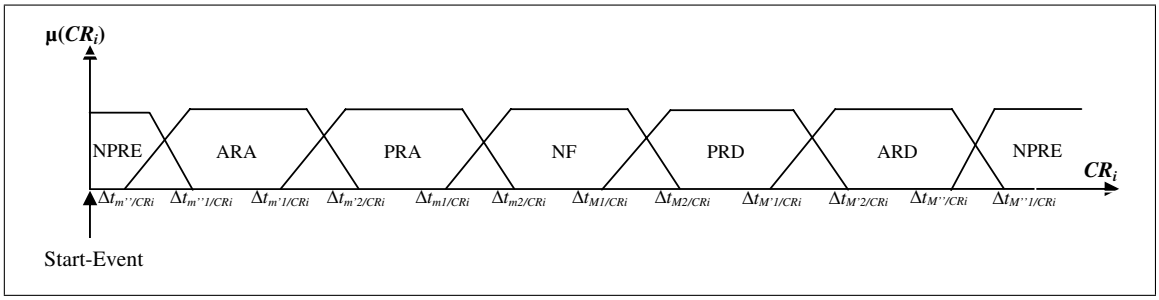


Figure 4: Fuzzy robustness integration in the operation associated model

Table 1: Linguistic variables associated to the input CR_i

T_{CR_i}	Linguistic variable
$T^1_{CR_i}$	CR_i arrives in the interval $I_{1i} = [0,\ \Delta t_{m''/CR_i}[$
$T^2_{CR_i}$	CR_i arrives in the interval $I_{2i} = [\Delta t_{m''/CR_i},\ \Delta t_{m''_1/CR_i}[$
$T^3_{CR_i}$	CR_i arrives in the interval $I_{3i} = [\Delta t_{m''_1/CR_i},\ \Delta t_{m'_1/CR_i}[$
$T^4_{CR_i}$	CR_i arrives in the interval $I_{4i} = [\Delta t_{m'_1/CR_i},\ \Delta t_{m'_2/CR_i}[$
$T^5_{CR_i}$	CR_i arrives in the interval $I_{5i} = [\Delta t_{m'_2/CR_i},\ \Delta t_{m_1/CR_i}[$
$T^6_{CR_i}$	CR_i arrives in the interval $I_{6i} = [\Delta t_{m_1/CR_i},\ \Delta t_{m_2/CR_i}[$
$T^7_{CR_i}$	CR_i arrives in the interval $I_{7i} = [\Delta t_{m_2/CR_i},\ \Delta t_{M_1/CR_i}[$
$T^8_{CR_i}$	CR_i arrives in the interval $I_{8i} = [\Delta t_{M_1/CR_i},\ \Delta t_{M_2/CR_i}[$
$T^9_{CR_i}$	CR_i arrives in the interval $I_{9i} = [\Delta t_{M_2/CR_i},\ \Delta t_{M'_1/CR_i}[$
$T^{10}_{CR_i}$	CR_i arrives in the interval $I_{10i} = [\Delta t_{M'_1/CR_i},\ \Delta t_{M'_2/CR_i}[$
$T^{11}_{CR_i}$	CR_i arrives in the interval $I_{11i} = [\Delta t_{M'_2/CR_i},\ \Delta t_{M''/CR_i}[$
$T^{12}_{CR_i}$	CR_i arrives in the interval $I_{12i} = [\Delta t_{M''/CR_i},\ \Delta t_{M''_1/CR_i}[$
$T^{13}_{CR_i}$	CR_i arrives in the interval $I_{13i} = [\Delta t_{M''_1/CR_i},\ +\infty[$

The margin of active robustness is available in $(I_{2i} \cup I_{3i} \cup I_{4i}) \cup (I_{10i} \cup I_{11i} \cup I_{12i})$, whereas the margin of passive robustness is in $(I_{4i} \cup I_{5i} \cup I_{6i}) \cup (I_{8i} \cup I_{9i} \cup I_{10i})$. From a functional point of view, there are six intervals of use in which it is possible to prove the validity: intervals of No Proof of Robustness Existance (NPRE), Normal Functioning (NF), Passive Robustness to an Advance (PRA), Passive Robustness to a Delay (PRD), Active Robustness to an Advance (ARA) and Active Robustness to a Delay (ARD). If the functioning is abnormal, there is duality of advance and delay scenarios.

- **Fuzzification of a Start-Event (A_i)**

Figure 5, shows the different set of the input variable (Start-Event A_i). The full set of linguistic variables associated to each membership is summarised in table 2.

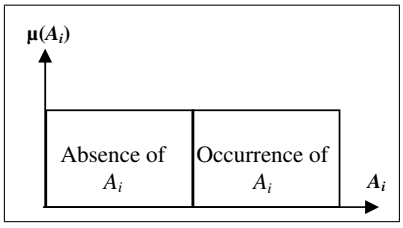


Figure 5: Membership functions of A_i

Table 2: Linguistic variables associated to the input Start - Event

T_{A_i}	Linguistic variable
$T^1_{A_i}$	Occurrence of the Start-Event A_i
$T^2_{A_i}$	Absence of the Start-Event A_i

- **Symptoms fuzzification**

Figure 6, shows an uniform distribution of fuzzy logic membership functions associated to the output " type of symptom ". Similarly, table 3 shows linguistic variables associated to the output " type of symptom ".

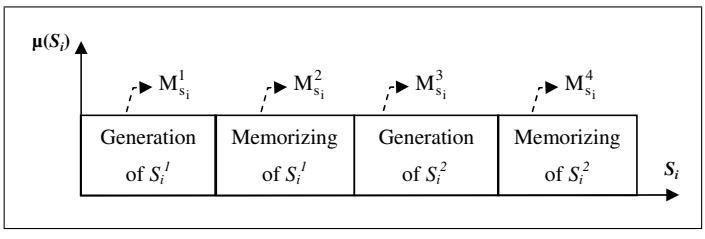


Figure 6: Membership functions of S_i

Table 3: Linguistic variables associated to the output “ type of symptom"

T_{S_i}	Linguistic variable
$T^1_{S_i}$	Generation of a symptom S^1_i
$T^2_{S_i}$	Memorizing of a symptom S^1_i
$T^3_{S_i}$	Generation of a symptom S^2_i
$T^4_{S_i}$	Memorizing of a symptom S^2_i

- **Fuzzification of Control Decision (CD)**

The three fuzzy set for the output CD are chosen as indicated in figure 7. Hence the three membership functions, uniformly distributed, are denoted M^1_{CD}, M^2_{CD} and M^3_{CD}. The linguistic variables are summarised in table 4. The integration of the approach generation of symptoms and the classification of the reports of various actions allows a qualitative description of fuzzy variables. These variables have balanced values of truth, pertaining to the interval [0, 1].

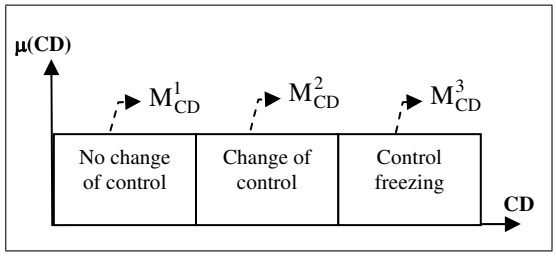


Figure 7: Membership functions of CD

Table 4: Linguistic variables associated to the output Control Decision

T_{CD}	Linguistic variable
T^1_{CD}	No Change of control
T^2_{CD}	Change of control
T^3_{CD}	Control freezing

Definition of fuzzy rules

Next we have to evaluate the Rules. The rules associate the input variables with the output ones by means of linguistic terms, and according to their physical properties [14]. The rules can present different structures (MIMO, MISO, SISO, ...), although the most common one is the Multiple Inputs Multiple Outputs (MIMO). We have used this structure to build fuzzy rules, and its arrangement is:

$$R^j_{MIMO}: \text{IF } (\ x_1 \text{ is } T_{x_1}) \text{ AND } ... \text{ AND } (\ x_p \text{ is } T_{x_p}) \tag{3}$$

$$\text{THEN } (\ y_1 \text{ is } T_{y_1}) \text{ AND } ... \text{ AND } (\ y_q \text{ is } T_{y_q}).$$

Being p the number of input variables, q the number of output variables whereas T_{x_i} and T_{y_i} represent their respective fuzzy sets for j^{th} rule.
The preconditions of R^j_{MIMO}, form a fuzzy set $(T_{x_1} \times T_{x_2} \times ... \times T_{x_p})$ and the consequent of R^j_{MIMO} is the union of q independent outputs [11]. So, the rule can be represented by a fuzzy implication:

$$R^j_{MIMO}: (T_{x_1} \times T_{x_2} \times ... \times T_{x_p}) \rightarrow (T_{y_1} + ... + T_{y_q}) \tag{4}$$

where "+" represents the union of independent variables.The fuzzy rules are merely a series of IF-THEN statements. These statements are usually derived by an expert to achieve optimum results. Thus, according to (3) we can formulate the rules as following:
Rule 1: IF there are absence of order (not A_i) AND presence of CR_i, THEN there are freezing of the control and generation of a symptom S^2_i.
Rule 2: IF the sensor signal CR_i arrives in the time interval $[0,\ \Delta t_{m''/CR_i}[$ AND the Start-Event A_i is occurred, THEN there are freezing of the control and generation of a symptom S^2_i.
Rule 3: IF the sensor signal CR_i arrives in the time intervals $(I_{2i} \cup I_{3i} \cup I_{4i})$ AND the Start-Event A_i is occurred, THEN there are change of control (ARA) and memorizing of a symptom S^2_i.
Rule 4: IF the sensor signal CR_i report arrives in the time intervals $(I_{4i} \cup I_{5i} \cup I_{6i})$ AND the Start-Event A_i is occurred, THEN there are no change of the control (PRA) and memorizing of a symptom S^2_i.
Rule 5: IF the sensor signal CR_i arrives in the time intervals $(I_{6i} \cup I_{7i} \cup I_{8i})$ AND the Start-Event A_i is occurred, THEN the behaviour of the system is normal (no change of the control).
Rule 6: IF the sensor signal CR_i arrives in the time intervals $(I_{8i} \cup I_{9i} \cup I_{10i})$ AND the Start-Event A_i is occurred, THEN it is the case of passive robustness to a delay (no change of the control) and memorizing of a symptom S^1_i.
Rule 7: IF the sensor signal CR_i arrives in the time intervals $(I_{10i} \cup I_{11i} \cup I_{12i})$ AND the Start-Event A_i is occurred, THEN a change of the control is necessary (ARD) and memorizing of a symptom S^1_i.
Rule 8: IF the sensor signal CR_i arrives in the time interval CR_i in the time interval I_{13i},THEN there are freezing of the control and memorizing of a symptom S^1_i.
Since the two outputs (Symptoms and Control Decision) of MIMO rule are independent, the general rule structure of MIMO fuzzy system can be represented as a collection of multiple-input and single-output (MISO) fuzzy systems by decomposing the above rules into q (q=2) subrules with as the single consequent of the j^{th} subrule. Therefore, the inference engine matches the rule preconditions in the fuzzy rule base with the input state linguistic terms and performs implication. In this subsection, for clarity, we will consider MISO system in the following analysis.
It is interesting to mention that each fuzzy rule just controls a part of the function to approach. These parts are denominated "patches", and they are the result of the localization property of the fuzzy basis functions. Nevertheless, from the fuzzy rules base (Rule 1, Rule 2, ..., Rule 8), we need to model numerically the operators AND and THEN. The fuzzy systems define an intermediate stage denominated Inference. The Inference is the part of the fuzzy systems that carries out an isomorphism between propositional logic and the Set and the Algebraic Theories. However, it is not valid whatever relation between the logical and math operators. In concrete, if they want to be equivalent, their logical and math tables for the crisp values $\{0, 1\}$ have to be the same. The selected inference method is the Mamdani type which is known as the max-min method.

Defuzzification

After inferring all rules, the fuzzy systems need to fusion them.This is the main goal of the defuzzification step, and it constitutes the last part of all fuzzy systems.This fusion is not unique, although the Centre Of Area (COA) defuzzifier is the widespread one, figure 8. In the COA method, the fused

measurement output y^* is obtained as:

$$y^* = \frac{\sum_{y \in Y} \mu_F(y).y}{\sum_{y \in Y} \mu_F(y)} \tag{5}$$

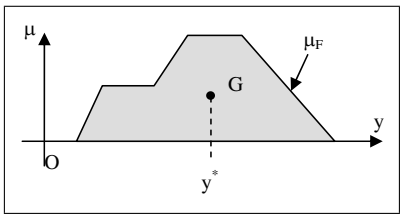


Figure 8: Centre Of Area defuzzifier

4 Illustrative example

To demonstrate the effectiveness and accuracy of the fuzzy filtering approach, an example with two fuzzy rules is outlined. Consider the following fuzzy rules base:
Rule 2: IF the sensor signal CR_i arrives in the time interval $[0,\ \Delta t_{m''/CR_i}[$ AND the Start-Event A_i is occurred, THEN there are freezing of the control and generation of a symptom S_i^2.
Rule 3: IF the sensor signal CR_i arrives in the time intervals $(I_{2i} \cup I_{3i} \cup I_{4i})$ AND the Start-Event A_i is occurred, THEN there are change of control (ARA) and memorizing of a symptom S_i^2.

- Each rule use the operator "AND" in the premise, since it is an AND operation, the minimum criterion is used (Mamdani inference method), and the fuzzy outputs corresponding to these rules are represented by figure 9 and figure 10.

- Next we perform defuzzification to convert our fuzzy outputs to a single number (crisp output), various defuzzification methods were explored to select the best one for this particular application. According to the relation(5),the weighted strengths of each output member function are multiplied by their respective output membership function center points and summed. Finally, this area is divided by the sum of the weighted member function strengths and the result is taken as the crisp outputs. In practice, there are two fuzzy outputs to defuzzify (Symptoms and Control Decision). To obtain a numerical output, we can take the COA of each fuzzy output, named G_{SY} and G_{CD}. The measures of the two CAO using specific values of sensor signal and Start-Event are summarized in table 5.

Table 5, shows the measures obtained by using defuzzification method mentioned above.Analysing the data, it is noted that the first and the third cases represent a classic filtering mechanism of sensors signals, integrating the robustness values described in [3]. The second case, using fuzzy filtering approach, gives better results than the two cases previously analysed. These cases reveal that the proposed approach is able to avoid control freezing (the COA G_{CD} belongs to the membership function "change of control") same if the sensor signal arrives in the "No Proof of Robustness Existance (NPRE)" interval. Therefore, the fuzzy filtering approach makes it possible to continue the production in a degraded mode providing the guarantees of quality and safety. Consequently, the intelligent fuzzy logic control strategy, based on expert knowledge, provides the avoidance of control freezing if the time disturbance is in the robustness intervals.

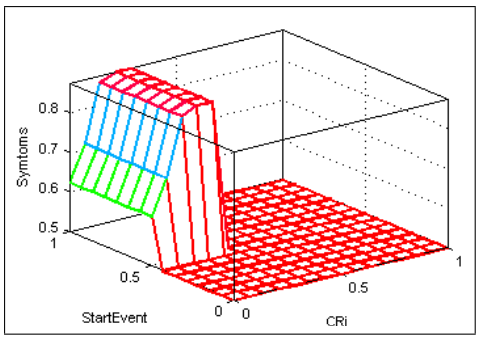


Figure 9: Three-dimensional trapezoidal membership function: Symptoms= $f(Cr_i, A_i)$

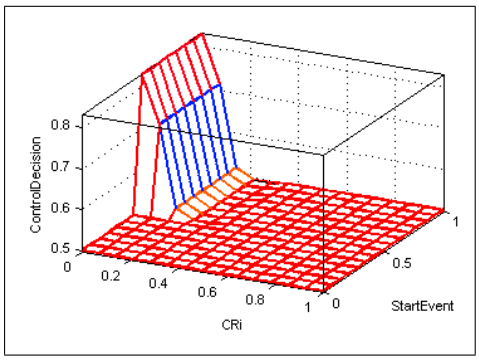


Figure 10: Three-dimensional trapezoidal membership function: Control Decision= $f(Cr_i, A_i)$

Table 5: Measures of G_{SY} and G_{CD}

Variables		Measures	
	First Case	Second Case	Third case
CR_i	CR_i arrives in the interval I_{1i}	CR_i arrives in the interval I_{2i}	CR_i arrives in the interval I_{3i}
A_i	A_i is occurred	A_i is occurred	A_i is occurred
G_{CD}	$G_{CD} \in M^3_{CD}$	$G_{CD} \in M^2_{CD}$	$G_{CD} \in M^1_{CD}$
G_{CD}	$G_{CD} \in M^3_{Si}$	$G_{CD} \in M^4_{Si}$	$G_{CD} \in M^4_{Si}$

5 Conclusion

This paper deals with supervision of manufacturing workshops with time constraints. A new approach integrating a classic filtering mechanism of sensors signals and fuzzy logic techniques has been presented. This approach exploits the advantages that both techniques have: the avoidance of control freezing using robustness knowledge and the ability of fuzzy systems to deal with imprecise information by using fuzzy rules.

In this new approach, an enhancement technique based on various combinations of fuzzy logic linguistic statements in the form of IF-THEN rules, based on expert knowledge, makes it possible to continue the production in a degraded mode providing the guarantees of quality and safety.The establishment of fuzzy logic is interesting, but it is necessary to call upon the human expertise, in an environment of uncertainty and imprecision, able to formulate and to transmit its knowledge for decision making.

The results obtained in the illustrative example show that this fuzzy approach is effective in situations where the sensor measurement is contaminated with different kind of noises. In this case, the temporal windows associates to each sensor signal are badly defined.
Conventionally, the selection of fuzzy IF-THEN rules often relies on a substantial amount of heuristic observation to express proper strategy's knowledge. Obviously, it is difficult for human experts to examine all the input-output data from a complex system to find the suitable number of rules within the fuzzy systems. For this reason, a fuzzy system with neural network's learning ability is required. A new approach using Neural Fuzzy Filter (NFF), based upon a neural network's learning ability and fuzzy IF-THEN rule structure can be developed in order to supervise critical time manufacturing job-shops.
This fuzzy filtering approach shows how the knowledge of the robustness could make the supervision more efficient, by introducing two events (Start-Event and sensor signal). A chronicle recognition approach, using the additional information provided by the occurrences of intermediate events, is a challenging technique for performing early diagnosis.

Bibliography

[1] A. Boufaied, A. Subias, and M. Combacau, Distributed Fault Detection with Delays Consideration, 15^{th} International Workshop on Principles of Diagnosis, Carcassonne, June 2004.

[2] A. Boufaied, A. Subias, and M. Combacau, The Distributed time constraints verification modelled with time Petri nets, *17^{th} IMACS Word Congress on Scientific Computation, Applied Mathematics and Simulation* (IMACS'05), Paris, July 2005.

[3] N. Jerbi, S. Collart Dutilleul, E. Craye, and M. Benrejeb, Time Disturbances and Filtering of Sensors Signals in Tolerant Multi-product Job-shops with Time Constraints, *International Journal of Computers, Communications & Control*, Vol. 1, No. 4, pp. 61 – 72, 2006.

[4] A. Toguyeni, *Surveillance et diagnostic en ligne dans les ateliers flexibles de l'industrie manufacturière*, Ph.D. Thesis, Université des Sciences et Technologies de Lille, November 1992.

[5] N. Jerbi, S. Collart Dutilleul, E. Craye, and M. Benrejeb, Robust Control of Multiproduct Job-shops in Repetitive Functioning Mode, *IEEE Conference on Systems, Man, and Cybernetics* (SMC'04), The Hague, Vol. 5, pp. 4917 – 4922, October 2004.

[6] N. Sawaya, and B. Ghaddar, A Fuzzy Logic Approach for Adjusting the Contention Window Size in IEEE 802.11e Wireless Ad hoc Networks, *IEEE International Symposium on Communication, Control, and Signal Processing* (IEEE ISCCSP), Marrakech 2006.

[7] J. Bas, A. Pérez, and M. Lagunas, Differential fuzzy filtering for adaptive line enhancement in spread spectrum communications, *Signal Processing Journal*, Vol. 86, Issue 5, pp 984 – 1009, May 2006.

[8] D. Van De Ville, M. Nachtegael, D. Van der Weken, E. Kerre, and W. Philips, Noise Reduction by Fuzzy Image Filtering, *IEEE Transactions on Fuzzy System*, Vol. 11, No. 4, pp. 429 – 436, August 2003.

[9] R. Mikut, A. Lehmann, and G. Bretthauer, Fuzzy Stability Supervision of Robot Grippers, *IEEE International Conference on Fuzzy Systems*, Budapest, Vol. 3, pp. 1473 – 1478, July 2004.

[10] L. Zadeh, Knowledge Representation in Fuzzy Logic, *IEEE Transactions on Knowledge And Data Engineering*, Vol. 1, No. 1, pp. 89 – 100, March 1989.

[11] C. Teng Lin, An Adaptive Neural Fuzzy Filter and Its Applications, *IEEE Transactions on Systems Man and Cybernetics*, Vol. 27, No. 4, pp. 635 – 656, August 1997.

[12] F. Lotte, A. L´ecuyer, F. Lamarche, and B. Arnaldi, Studying the Use of Fuzzy Inference Systems for Motor Imagery Classification, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 15, No. 2, June 2007.

[13] Z. Shafiq, F. Muddassar, and S. Khayam, A Comparative Study of Fuzzy Inference Systems, *Neural Networks and Adaptive Neuro Fuzzy Inference Systems for Portscan Detection* (EvoWorkshop), pp. 52 – 61, 2008.

[14] J.L. Castro, J.M. Benitez, and I. Requena, Are artificial neural networks black boxes?, *IEEE Transaction on Neural Networks*, Vol. 8, No. 5, pp.1156 – 1164, September 1997.

Anis M'halla was born in Mahdia, Tunisia in 1980. He obtained the Engineer degree in electro-Mechnical engineering from the "Ecole Nationale d'Ingénieurs de Sfax (ENIS) " and obtain the master degree in automatic and industrial Maintenance from the "Ecole Nationale d'Ingénieur de Monastir" in 2006. He is currently preparing the Ph.D. degree in automatic and computer science within the framework of LAGIS-EC-Lille and LARA-ENIT cooperation. His research is related to robustness and supervision of multi-product job-shops with time constraints.

Nabil Jerbi was born in Tunis, Tunisia, in 1970. He obtained the Engineer degree in electrical engineering from the Ecole Nationale d'Ingénieurs de Tunis (ENIT), in 1994. Also, he received the Aggregation Certificate from the Ecole Supérieure des Sciences et Techniques de Tunis and the Master degree in automatic and signal treatment from ENIT, in 2001 and 2003, respectively. He is currently an assistant at "Institut supérieur des Sciences Appliquées et de Technologie de Kairouan ". His research interests include robustness and supervision of multi-product job-shops with time constraints.

Pr. Simon Collart Dutilleul obtained a Ph.D. degree in Electronics, Electrotechnics and Automatic Control from Université de Savoie in 1997. He is currently a Professor at Ecole Centrale de Lille, and a member of the research team on Discrete Event Systems in the LAGIS laboratory (Laboratory of Control Engineering, Computer Science & Signal), where he studies specifically time constrained systems.

Pr. Etienne Craye was born in Roubaix (France) in 1961; he obtained in 1984 the Engineer Diploma of the "Institut Industriel du Nord" (French "Grande Ecole") and the same year his Master Degree in Computer Sciences. He obtained a Ph.D. in Automatic control for Manufacturing and Discrete Events systems in 1989 and his "Habilitation ŕ Diriger des Recherches" in 1994. He is now, since 1995, Professor at the Ecole Centrale de Lille and in the same time the Director of this institution. Pr. Craye is currently working on Monitoring and Supervision of Fault Tolerant Systems. Specially, reconfiguration and working mode management are today studied by taken into account on one-hand failures and on the other hand the flexibilities of the system architecture. The objective is to be able to go on with the production and not to reconsider the objectives.

Pr. Mohamed Benrejeb was born in Tunisia in 1950. He obtained the Diploma of "Ingénieur IDN" (French "Grande Ecole") in 1973, The Master degree of Automatic Control in 1974, the PhD in Automatic Control of the University of Lille in 1976 and the DSc of the same University in 1980. Full Professor at "Ecole Nationale d'Ingénieurs de Tunis" since 1985 and at "Ecole Centrale de Lille" since 2003, his research interests are in the area of analysis and synthesis of complex systems based on classical and non conventional approaches.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 3, pp. 375-384

A Swarm Intelligence Approach to the Power Dispatch Problem

D.C. Secui, I. Felea, S. Dzitac, L. Popper

Dinu Călin Secui, Ioan Felea, Simona Dzitac, Laurenţiu Popper
University of Oradea, Faculty of Energy Engineering, Romania
E-mail: csecui@uoradea.ro, ifelea@uoradea.ro, simona.dzitac@gmail.com,
director@perfect-service.ro

Abstract: This paper examines how two techniques of the Particle Swarm Optimization method (PSO) can be used to solve the Economic Power Dispatch (EPD) problem. The mathematical model of the EPD is a nonlinear one, PSO algorithms being considered efficient in solving this kind of models. Also, PSO has been successfully applied in many complex optimization problems in power systems. The PSO techniques presented here are applied to three case studies, which analyze power systems having four, six, respectively twenty generating units.
Keywords: economic dispatch problem, constrained optimization, particle swarm optimization.

1 Introduction

An important issue in optimizing the power systems is the economic power dispatch. This problem consists in determining the power generated by the plant units of a system in order to minimize the total generation cost of units, taking into account the active power balance and the constraints imposed to the capability of the units.

The mathematical optimization model is nonlinear, where both the objective function and the restrictions imposed by equations describing the system functionality are nonlinear.

The mathematical model can be solved using conventional optimization techniques such as the lambda iteration method, the gradient method and others [1, 2], if constraints are considered as linear, the objective function is continuous and the domain of the values is convex. Some disadvantages that arise in these situations - long solving time, objective function discontinuity etc. - may be overcome by applying the artificial intelligence techniques. The most common optimization techniques based upon artificial intelligence used for solving economic power dispatch problems are: the genetic algorithm [3–6], the Hopfield neural networks [2, 7], the differential algorithm [8], the evolutionary programming [9, 10], fuzzy-optimization [12, 13], tabu search [14], particle swarm optimization [15, 16, 27, 28]. Also, the EPD can be formulated as a multi-objective optimization problem [11, 13, 17].

TWe mention that the particle swarm optimization method was successfully applied to other optimization problems, such as optimal power flow [18–20], reactive power optimization and voltage control [23], power loss reduction in distribution systems [24], network reconfiguration [25], unit commitment problem [26], due to its good convergence, low computational time and good quality solutions.

In this paper there are proposed two versions of applying PSO method for solving EPD, comparing the results obtained for three systems consisting of four, six and respectively, twenty generating units.

The paper is structured as follows: section 2 exposes the EPD problem, section 3 and 4 describe the proposal for solving this problem using PSO algorithm, section 5 presents the results obtained through the application of PSO algorithm for three power systems, and section 6 outlines the conclusions drawn.

2 Formulation of the economic power dispatch problem

We consider a power system containing n generating units, each unit having its own generated power P_j, j = 1...n. The total load required in the system is considered to be known and equal to P_D. The fuel

cost $(F_j(P_j))$ for each generator j is represented by a quadratic function:

$$F_j(P_j) = a_j \cdot P_j^2 + b_j \cdot P_j + c_j \tag{1}$$

where: a_j, b_j and c_j are fuel cost coefficients of generator j;
P_j represent the power of generator j.

EPD solution consists in determining the Pj powers of generating units, so that the total fuel cost of the entire system to be minimal, respecting the restriction of power balance on the overall system and the inequality restrictions for each unit j.

The objective function is:

$$\min F = \sum_{j=1}^{n} F_j(P_j) \tag{2}$$

The problem constraints are given by relations (3) and (4):

$$F_p = \sum_{j=1}^{n} P_j - P_D - \Delta P = 0 \tag{3}$$

$$P_j^{min} \leq P_j \leq P_j^{max}, j = 1, 2, ..., n \tag{4}$$

Where: P_j^{min} and P_j^{max} represent the maximum and the minimum operation limits of a generator j;

The power loss at the level of the entire system is a quadratic function in relation to variables P_j and it is calculated by using constant B coefficient formula:

$$\Delta P = \sum_{i=1}^{n} \sum_{j=1}^{n} P_i \cdot B_{ij} \cdot P_j + \sum_{i=1}^{n} B_{0i} \cdot P_i + B_{00} \tag{5}$$

Where B_{ij} is an element of the loss coefficient matrix, B_{i0} is the element i of the loss coefficient vector, and B_{00} is the loss coefficient constant.

3 Presentation of different Particle Swarm Optimization techniques

PSO is a heuristic algorithm, used for solving nonlinear and noncontinuous optimization problems, being introduced by Kennedy and Eberhart [22], in 1995. Since then several techniques in applying the PSO method have been developed, but in the current paper only two techniques are presented, namely the classical PSO (PSO Classical) and PSO with time varying acceleration coefficients (PSO Accelerated).

Classical PSO: To search for the optimal solution in a space with the dimension n, PSO uses a population of NP particles. For a given particle i within NP population, vector solutions at a certain iteration k are represented by X_i^k=(x_{i1}^k, x_{i2}^k,..., x_{ij}^k,..., x_{in}^k). In any optimization process, switching from one solution (x_{ij}^k) to another solution (x_{ij}^{k+1}) is accomplished by using the velocity of particles, represented by the vector V_i^k=(v_{i1}^k, v_{i2}^k,..., v_{ij}^k,..., v_{in}^k), according to the relation:

$$X_i^{k+1} = X_i^k + V_i^{k+1}, i = 1, 2, ..., NP \tag{6}$$

The updated velocity of the particle in the next iteration (k+1) is given by the relation:

$$X_i^{k+1} = \omega \cdot V_i^k + c_1 \cdot r_1 \cdot (Pbest_i^k - X_i^k) + c_2 \cdot r_2 \cdot (Gbest^k - X_i^k) \tag{7}$$

Where: V_i^k, V_i^{k+1} represent the velocity vector of particle i at iteration k, respectively $k+1$;
X_i^k, X_i^{k+1} represent the solution vector of particle i at iteration k, respectively $k+1$;

$Pbest_i^k$ represent the best solution vector of particle i, until iteration k;

$Gbest^k$ represent the vector corresponding to the best solution of the group, until iteration k;

c_1 and c_2 are coefficients corresponding to cognitive and social behavior;

r_1 and r_2 are random numbers between 0 and 1, and w is the inertia weight factor determined using the relation:

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{k_{max}} \cdot k \tag{8}$$

Where ω_{max} and ω_{min} represent the initial, respectively final weights, k_{max} is maximum iteration number and k is current iteration number for the algorithm.

PSO with time varying acceleration coefficients tries to improve the global search in the early stages of the optimization process and to accelerate the convergence of the particles to the global optimum in the final part of the process. In this case, the calculation of the velocity (V_i^{k+1}) and of the solution (X_i^{k+1}) for the next iteration is done with relations (6) and (7), and according to [21], the coefficients c_1 and c_2 are determined by the relations:

$$c_1 = (c_{1f} - c_{1i}) \cdot \frac{k}{k_{max}} + c_{1i} \tag{9}$$

$$c_2 = (c_{2f} - c_{2i}) \cdot \frac{k}{k_{max}} + c_{2i} \tag{10}$$

Where c_{1i}, c_{2i}, c_{1f} and c_{2f} are initial and final weights for cognitive and social acceleration coefficients.

4 The methodology based on PSO for solving the EPD problem

The implementation of PSO techniques for solving the EPD problem involves the following steps: Step1. *Initialization of the parameters and of PSO solution.* The PSO algorithm parameters are set in reference to ω_{max}, ω_{min}, k_{max}, number of particles (NP), coefficients c_1 and c_2 (for PSO Classical), respectively, c_{1i},c_{2i}, c_{1f} and c_{2f} (PSO Accelerated). Initially a population of *NP* particles is randomly formed. Each particle defines a possible solution to the problem, which should respect the constraints given by relations (3) and (4).

Step 2. *Evaluation of the objective function F and of the auxiliary function f.* The problem contains an equality restriction shown by relation (3). Thus, the auxiliary function f is formed, using the relation:

$$f = F + \alpha \cdot F_p^2 \tag{11}$$

Where α is the penalty factor.

For each particle and each iteration the values of function f will be calculated, and by comparing them solutions $Pbest_i$ and *Gbest* are selected. At the end of the optimization process, functions f and F will have approximately equal values, according to the calculation error admitted by choosing factor α.

Step 3. *Update velocity and solution.* The minimum (V_j^{min}) and the maximum limits (V_j^{max}) of the velocity for each generating unit j are calculated:

$$V_j^{max} = \beta \cdot (P_j^{max} - P_j^{min}) \text{ and } V_j^{min} = -V_j^{max} \tag{12}$$

Where factor β was considered between 0.05 and 0.1.

The update of the particle position and velocity is done with relations (6) and (7). For each solution (X_i)) it is verified if the components x_{ij} satisfy the constrain (4). If the constrain is satisfied, then the

Unit	P^{min}[MW]	P^{max}[MW]	a[\$/MW^2]	b[\$/MW]	c[\$]
1	30	120	0.00875	18.24	750
2	50	160	0.00754	18.87	680
3	50	200	0.00310	19.05	650
4	100	300	0.00423	17.90	900

Table 1: Cost coefficients and limits of generated powers for a thermal power plant with four units (CS4)

Unit	P^{min}[MW]	P^{max}[MW]	a[\$/MW^2]	b[\$/MW]	c[\$]
1	100	600	0.001562	7.92	561
2	100	400	0.00194	7.85	310
3	50	200	0.00482	7.97	78
4	140	590	0.00139	7.06	500
5	110	440	0.00184	7.46	295
6	110	440	0.00184	7.46	295

Table 2: Cost coefficients and limits of generated powers for a thermal power plant with six units (CS6)

calculated value for x_{ij} is kept. Otherwise x_{ij} is set with the value nearest to the limit of the domain (P_j^{max} or P_j^{min}).

The vectors *Pbest* and *Gbest* are obtained based on the evaluation of the auxiliary function f and on the comparison of the f values calculated in two consecutive steps. If the new value of function f is better than the previous value of f for previous *Gbest*, then *Gbest* is set at the new value. Similarly, *Pbest* vector is updated.

Step 4. *Stopping process*. In this paper the criterion of stopping the calculation process is given through achieving the maximum number of iterations set.

5 Numerical examples and simulation results

In this section three case studies on how EPD solving by applying the two PSO techniques (PSO Classical and PSO Accelerated) are presented. The objective function is given by the relation (2) and restrictions (3) and (4). All case studies were implemented in Mathcad, on a personal computer having a 1.58 GHz processor and 896 MB of RAM.

5.1 Description of tested systems

Case study 1 - four unit system (CS4). The first system (CS4) is a thermal power plant having four generating units, where the total power losses (ΔP) are considered zero. The data for the four generators (cost coefficients and limits of generated powers) are presented in Table 1. The total power demanded in the system is P_D=520 MW.

Case study 2 - six unit system (CS6). The second system (CS6) is a thermal power plant with six generating units, where the total (ΔP) is considered zero. The data for the six generators (cost coefficients and limits of generated powers) are presented in Table 2 [5]. The total power demanded in the system is P_D=1800 MW.

Case study 3 - twenty unit system (CS20). The third system contains twenty units, and the demanded power in the system is P_D=2500MW. The data for the generators and the values of the coefficients B_{ij} are available in [9].

Scenarios	c_{1i}	c_{2f}	Best F [$/hour]	Worst F [$/hour]	Average F [$/hour]	Standard deviation F [$/hour]
1	1.5	1.5	12919.76	12920.77	12919.84	0.026
2	1.5	2	12919.76	12920.36	1291981	0.015
3	1.5	2.5	12919.76	12921.01	12919.84	0.035
4	2	1.5	12919.77	12921.00	12919.83	0.028
5	2	2	12919.76	12920.04	12919.79	0.007
6	2	2.5	12919.76	12920.42	12919.80	0.016
7	2.5	1.5	12919.76	12922.35	12919.84	0.052
8	2.5	2	12919.76	12920.17	12919.80	0.009
9	2.5	2.5	12919.76	12920.27	12919.80	0.013

Table 3: The influence of the coefficients c1i,c2f upon the results obtained through PSO Accelerated, for one hundred trials (CS4)

5.2 PSO parameters and PSO convergence

PSO parameters involved in the calculation process may affect the algorithm performances and quality of the solutions. For the system having four generators (CS4) the parameters were set to values: $W_{min} = 0.4$, $W_{max} = 1$, $c_1 = 2.75$, $c_2 = 1.75$, NP=6, $k_{max} = 15$ (PSO Classical) and respectively $W_{min} = 0.4$, $W_{max} = 1$, $c_{1i} = 2$, $c_{2i} = 0.4$, $c_{1f} = 0.4$, $c_{2f} = 2$, NP=6, $k_{max} = 15$ (PSO Accelerated).

In case of PSO Accelerated algorithm, in order to assess the influence of the coefficients on EPD problem solving, 100 distinct trials were performed, noting the best value for F (Best F), the worst value for F (worst F), the average value for F (Average F) and the standard deviation of cost F. Table 3 shows the values Best F, Worst F, Average F and standard deviation F, considering that the values of coefficients (c_{1i}, c_{2f}) vary between limits $c_{1i}, c_{2f} \in [1.5, 2.5]$, and the values of coefficient (c_{1f}) and (c_{2i}) are constant and equal to $c_{1f} = c_{2i} = 0.4$.

Regarding the four unit system (CS4), in Table 3 it can be noticed that coefficient changes do not affect the solutions, but the best results are obtained for scenario 5, where $c_{1i} = 2$, $c_{2f} = 2$.

For the six generator system (CS6) in both PSO Classical and PSO Accelerated algorithm, the coefficients have little influence upon the outcomes, considering that they vary within the limits c1,c2 [1, 4], respective c1i,c2f [1.5, 2.5] and c1f, c2i [0.1, 0.4]. The results are shown in Table 4, the parameters being set to values: wmin=0.4, wmax=1, c1=2, c2=2, NP=15, kmax=30 (PSO Classical), and wmin=0.4, wmax=0.9, c1i=2.5, c2i=0.2, c1f=0.4, c2f=1.6, NP=15, kmax=30 (PSO Accelerated).

In case of the twenty unit system (CS20), PSO Classical was applied taking into account the following settings: w1=0.4, w2=1.1, c1=2, c2=2, NP=500, kmax=200 and =0.1; for PSO Accelerated it was considered: w1=0.4, w2=1.1, c1i=2.5, c2f=2, c1f=0.4, c2i=0.2, NP=500, kmax=200 and =0.1.

The number of particles that constitute the population is another important factor in the PSO algorithm. In case of the four unit system, for both methods (PSO Classical and PSO Accelerated), the changes in cost F in relation to particles number (NP) were graphically represented, considering the number of iterations set at k_{max}=15 (Fig. 1). Also, in Fig. 2 the variation of cost F in relation to the number of iterations (k_{max}) is represented, considering the number of particles set to NP=10.

Analyzing the diagram shown in Fig. 1 it is found that for both algorithms (PSO Classical and PSO Accelerated) the best solution (Best F) is obtained considering a population consisting of 5-6 particles. Fig. 2 shows that the converging process towards the best value of F is obtained after fifteen iterations for both algorithms, the initial solution being different.

In the six unit system, increasing the number of variables involves a higher number of particles (NP=15) and a higher number of iterations to (k_{max}=30) in order to obtain the same solution, presented in Table 4.

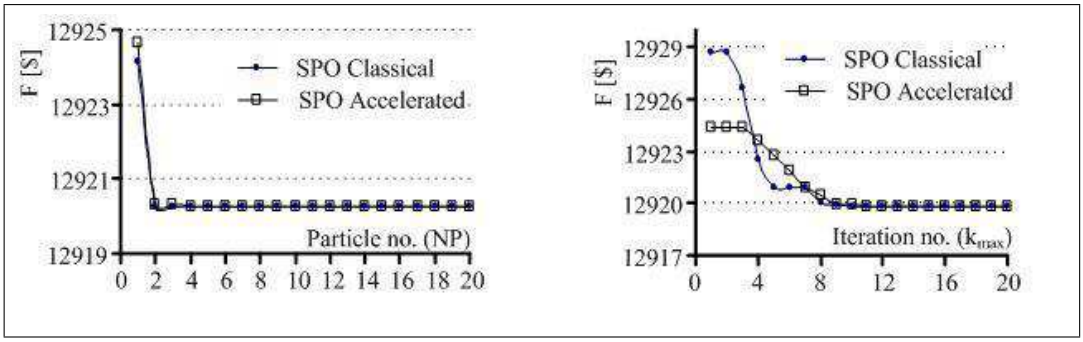


Figure 1: F convergence with particles number (NP)/CS4; Figure 2. F convergence with k_{max} for CS4

Algorithms	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	Cost F [$/hour]
FLCGA [3]	250.49	215.43	109.92	572.84	325.66	325.66	16585.85
PGA [5]	248.14	217.74	75.20	587.80	335.56	335.56	16579.33
AECGA [6]	248.07	217.73	75.30	587.70	335.60	335.60	16579.33
IHN [7]	248.08	217.74	75.18	587.90	335.55	335.55	16579.33
PSO Classical	247.95	218.44	75.16	587.58	335.43	335.44	16579.33
PSO Accelerated	248.00	217.71	75.16	588.02	335.52	335.59	16579.33

Table 4: The comparison of the results obtained for the system with six generating units (CS6)

To assess the efficiency of PSO algorithms, they are compared to other four algorithms previously presented using the same data, available in [5]. It can be seen that both PSO Classical, and PSO Accelerated algorithm reach the same cost F as the algorithms presented in [3], [5], [6] and [7]. The resulting solution and the cost value are comprised in Table 4.

5.3 The assessment of the solutions

The quality of the solutions was assessed by determining the values Best F, Worst F, Average F and standard deviation F considering 100 trials.

FFor the system with four generating units the results are presented in Table 5. In order to assess the convergence process and quality of the solutions the average of cost values F (Average F - Fig.3) and its standard deviation (Standard deviation F - Fig.4) for each iteration are determined, considering only one trial. The system considered in Fig.3 and Fig.4 proves that both algorithms converge quickly toward the best solution for F, the curves presenting a continuous decrease in relation to the number of iterations.

The best solution obtained in 100 trials, using algorithms PSO Classical and PSO Accelerated, is presented in Table 5, together with the results of the gradient method.

In case of the six units system (CS6), the values Best F, Worst F, Average F and standard deviation F considering 100 trials are presented in Table 6.

Algorithm robustness was tested starting from different initial solutions, randomly obtained, and retaining the best value of F for a number of algorithm trials within the [1-50] interval.

For the system having twenty units (CS20) the following values were obtained, considering 100 trials (Table 7). In Table 8 is presented the solution for the twenty units system, and the comparison of the results with those obtained in [2].

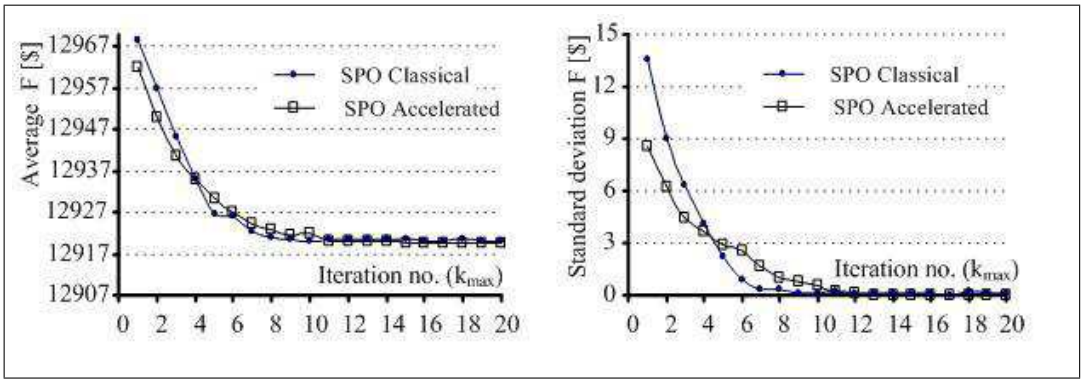


Figure 3. Average variation of F with k_{max} (CS4); Figure 4. Variation of standard dev. F with k_{max}(CS4)

Unit power output [MW]	PSO Classical	PSO Accelerated	Gradient method
P1	88.554	92.536	92.493
P2	65.340	65.539	65.559
P3	134.662	130.293	130.431
P4	231.444	231.632	231.517
Total power [MW]	520.00	520.00	520.00
Total generated cost (Best F)[$/hour]	12919.96	12919.76	12919.76
CPU time (s)	< 1	< 1	< 1

Table 5: The best solutions (Best F) obtained for the system with four generating units (CS4)

Algorithm	Best F [$/hour]	Worst F [$/hour]	Average F [$/hour]	Standard deviation F [$/hour]
PSO Classical	16579.33	16582.64	16579.51	0.0650
PSO Accelerated	16579.33	16581.93	16579.49	0.0362

Table 6: The values Best F, Worst F, Average F and standard deviation F for CS6

Algorithm	Best F [$/hour]	Worst F [$/hour]	Average F [$/hour]	Standard deviation F [$/hour]
PSO Classical	62457.1805	62469.64	62462.45	0.4346
PSO Accelerated	62456.4380	62466.63	62461.05	0.3340

Table 7: The values Best F, Worst F, Average F and standard deviation F for CS20

The value of cost F obtained in Table 7 presents a smaller dispersion for both PSO algorithms, showing a high quality of the solutions. The calculation of power losses in the transmission lines was achieved with an error of 0.03 MW. It is also seen that PSO Accelerated algorithm gets a solution as good as the algorithms presented in [2], but PSO Classical leads to a farther solution, comparing to the solutions presented in Table 8.

Unit power output [MW]	PSO Accelerated	PSO Classical	Lambda-iteration method [2]	Hopfield neural network [2]
P1	511.1808	514.2219	512.7805	512.7804
P2	171.9583	163.3928	169.1033	169.1035
P3	125.9410	125.7172	126.8898	126.8897
P4	99.6666	99.4738	102.8657	102.8656
P5	114.9377	112.8359	113.6836	113.6836
P6	75.1378	76.8715	73.5710	73.5709
P7	113.2613	109.2679	115.2878	115.2876
P8	116.2341	116.8953	116.3994	106.3994
P9	101.5174	100.5633	100.4062	100.4063
P10	102.5556	113.8036	106.0267	106.0267
P11	150.6753	154.2216	150.2394	150.2395
P12	292.5836	289.6717	292.7648	292.7647
P13	120.2476	118.0356	119.1154	119.1155
P14	34.9866	38.2194	30.8340	30.8342
P15	117.3186	115.2359	115.8057	115.8056
P16	36.1563	36.4369	36.2545	36.2545
P17	68.7329	69.8707	66.8590	66.8590
P18	83.7616	81.9408	87.9720	87.9720
P19	98.9061	103.8655	100.8033	100.8033
P20	56.4137	51.6055	54.3050	54.3050
Total power [MW]	2592.1728	2592.1468	2591.9670	2591.9669
Total generation cost [$/hour]	62456.4380	62457.1805	62456.6391	62456.6341
Total lines losses [MW]	92.1728	92.1468	91.9670	91.9669

Table 8: The best solutions (Best F) obtained for the system with twenty generating units (CS20)

6 Conclusions

In this paper, the economic power dispatch problem is solved using two PSO techniques, namely, PSO Classical and PSO Accelerated. Both techniques are effective in solving this problem, but PSO Accelerated leads to a better quality of solutions and a lower computing time.

For EPD problems with small number of variables and linear restrictions, the classical solving techniques (gradient method, lambda iteration method) are also applicable, obtaining the same results as PSO. For EPD with nonlinear restrictions, PSO techniques are more effective, having a better convergence, robustness and stability, indicated by low values of standard deviation. The number of particles and the number of iterations required to obtain stable solutions are related to a reduced computing time.

PSO techniques are also compared with other techniques, such as Hopfield neural network, the results being almost identical for our applications.

Bibliography

[1] Lee, F.N. and Breipohl, A.M., *Reserve contrained economic dispatch with prohibited operating zones*, IEEE Transaction Power Systems, Vol. 8 (1), pp: 246-254, 1993

[2] Su, C.T. and Lin C.T., *New Approach with a Hopfield Modeling Framework to Economic Dispatch*, IEEE Transaction Power Systems, Vol. 15 (2), pp: 541-545, 2000

[3] Song, Y.H., Wang, G.S., Wang, P.V. and Johns, A.T., *Environmental/Economic Dispatch Using Fuzzy Logic Controlled Genetic Algorithms*, IEE Generation, Transmission and Distribution, Vol. 144(4), pp: 377-382, 1997

[4] Chiang C.L., *Improved Genetic Algorithm for Power Economic Dispatch of Units with Valve-Point Effects and Multiple Fuels*, IEEE Trans. Power Systems, Vol.20(4), pp: 1690-1699, 2005

[5] Yalcinoz, T., Altun, H. and Uzam, M., *Economic Dispatch Solution Using a Genetic Algorithm Based on Arithmetic Crossover*, IEEE Power Tech Conference, Vol.2, pp: 4, Porto, 2001

[6] Song, Y.H. and Chou C.S.V., *Advanced Engineered Conditioning Genetic Approach to Power Economic Dispatch*, IEE Generation, Transmission and Distribution, Vol.144(3),pp: 285-292, 1997

[7] Yalcinoz, T. and Short M.J., *Large scale Economic Dispatch Using an Improved Hopfield Neural Network*, IEE Generation, Transmission and Distribution, Vol. 144(2), pp: 181-185, 1997

[8] Coelho, Ld.S. and Mariani, V.C., *Combining of Chaotic Differential Evolution and Quadratic Programming for Economic Dispatch Optimization with Valve-Point Effect*, IEEE Transactions on Power Systems, Vol. 21(3), pp: 1465- 1465. 2006.

[9] Sinha, N., Chakrabarti, R. and Chattopadhyay P.K., *Evolutionary Programming Techniques For Economic Load Dispatch*, IEEE Transaction Evolutionary Computation, Vol.7(1), pp: 83-94, 2003

[10] Venkatesh, P., Gnanadass, R. and Padhy N.P, *Comparison And Application Of Evolutionary Programming Techniques To Combined Economic Emission Dispatch With Line Flow Constraints*, IEEE Transactions on Power Systems, Vol.18(2), pp: 688-697, 2003

[11] Singh, L. and Dhillon, J.S., *Fuzzy Satisfying Multiobjective Thermal Power Dispatch Based On Surrogate Worth Trade-Off Method, Electric Power Components And Systems*, Vol. 36(1), pp: 93-108, 2008

[12] Attaviriyanupap, P., Kita, H., Tanaka, E. and Hasegawa J., *A Fuzzy-Optimization Approach to Dynamic Economic Dispatch Considering uncertainties*, IEEE Trans. Power Systems, Vol.19(3), pp: 1299-1307, 2004

[13] Brar, Y.S., Dhillon, J.S. and Kothari, D.P., *Multiobjective Load Dispatch By Fuzzy Logic Searching Weightage Pattern, Electric Power Systems Research*, Vol. 63, pp: 149-160, 2002

[14] Lin, W.M., Cheng, F.S. and Tsay, M.T., *An Improved Tabu Search For Economic Dispatch With Multiple Minima*, IEEE Transaction Power Systems, 17(1), pp: 108-112, 2002

[15] AlRashidi, M.R. and El-Hawary M.E., *Hybrid Particle Swarm Optimization Approach for Solving the Discrete OPF Problem Considering the Valve Loading Effects*, IEEE Transaction Power Systems, Vol. 22 (4), pp: 2030-2038, 2007

[16] Gaing, Z.L., *Particle Swarm Optimization to Solving the Economic Dispatch Considering the Generator Constraints*, IEEE Transaction Power Systems, Vol. 18 (3), pp: 1187-1195, 2003

[17] Rotar, C., Dumitrescu, D. and Lung, R.I., *Guided hyperplane evolutionary algorithm, Proceedings of the 9th annual conference on Genetic and evolutionary computation*, London, pp: 884 - 891, 2007

[18] Yumbla, P.E.O., Ramirez, J.M. and Coello, C.A.C., *Optimal Power Flow Subject to Security Constraints Solved With a Particle Swarm Optimizer*, IEEE Transaction Power Systems, Vol. 23(1), pp: 33-40, 2008

[19] Makeechev, V.A., Soukhanov, O.A. and Sharov, Y.V., *Hierarchical Algorithms Of Functional Modelling For Solution Of Optimal Operation Problems in Electrical Power Systems*, International Journal of Electrical Power & Energy Systems, Vol. 30 (6-7), pp: 415-427, 2008

[20] Singh, L. and Dhillon J.S., *Secure Multiobjective Real And Reactive Power Allocation Of Thermal Power Units*, International Journal Of Electrical Power & Energy Systems, Vol. 30, pp: 594-602, 2008

[21] Ratnaweera, A., Halgamuge, S.K. and Watson, H,C., *Self-Organizing Hierarchical Particle Swarm Optimizer With Time Varying Acceleration Coefficients*, IEEE Transaction Evolutionary Computation, Vol. 8(3), pp: 240-255, 2004

[22] Kennedy, J.F., Eberhart, R.C. and Shi, R.C., *Swarm Intelligence*, San Francisco (CA, USA): Morgan Kaufmann Publishers, 2001.

[23] Yoshida, H., Kawata, K., Fukuyama, Y., Takayama, S. and Nakanishi, Y., *A Particle Swarm Optimization For Reactive Power And Voltage Control Considering Voltage Security Assessment*, Transactions of the Institute of Electrical Engineers of Japan. B, Vol. 119-B, 12, pp: 1462-1469, 1999

[24] Gavrilas, M., Iovanov, O. and Sfintes, C.V., *Enhanced Particle Swarm Optimization Method For Power Loss Reduction In Distribution Systems*, 19th International Conference on Electricity Distribution, Vienna, paper 0088 (4 pp), 2007

[25] Liu, Y. and Gu X., *Skeleton-Network Reconfiguration Based on Topological Characteristics of Scale-Free Networks and Discrete Particle Swarm Optimization*, IEEE Transaction Power Systems, Vol. 22 (3), pp: 1267-1274, 2007

[26] Ting, T.O., Rao, M.V.C. and Loo, C.K., *A Novel Approach for Unit Commitment problem via an Effective Hybrid Particle Swarm Optimization*, IEEE Transaction Power Systems, Vol. 21 (1), pp: 411-417, 2006

[27] Krishna Teerth Chaturvedi, Manjaree Pandit, Laxmi Srivastava, *Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch*, International Journal of Electrical Power & Energy Systems, Vol. 31 (6), pp: 249-257, 2009

[28] Leandro dos Santos Coelho, Chu-Sheng Lee, *Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches*, International Journal of Electrical Power & Energy Systems, Vol. 30 (5), pp: 297-307, 2008

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 3, pp. 385-397

Robust Control of Particle Size Distribution in Aerosol Processes

Z. Xiang

Zhengrong Xiang
School of Automation
Nanjing University of Science and Technology
Nanjing, 210094, People's Republic of China
E-mail: xiangzr@mail.njust.edu.cn

Abstract: This paper deals with a comprehensive study on robust control of particle size distribution of fractal agglomerate in aerosol processes with simultaneous chemical reaction, nucleation, condensation and coagulation. Firstly, a general aerosol process is described by population balance and mass and energy balances, which describes the evolution of particle size distribution, continuous phase species and temperature of the aerosol system, respectively. A lognormal moment approximations of the population balance model is then presented. Then, the robust state feedback controller is designed for the aerosol process with some unknown uncertainties, the proposed controller is composed of an nominal control term and a robust control term so that it only ensures the stability of the closed-loop system, but also attenuates the effect of the unknown uncertainties on the system. A high-gain observer is adopted to estimate state variables required in the on-line implementation of the state feedback. Finally, the proposed robust controller is applied to an aerosol process for achieving an aerosol size distribution with desired geometric average particle diameter, the simulation results show the robustness properties of the controller with respect to parametric model uncertainty and unmodeled dynamics.

Keywords: particle size distribution, aerosol process, population balance, nonlinear systems, robust control

1 Introduction

Aerosol process is of interest in many problems of ecology, atmospheric physics, and mechanics of multiphase system[1-3]. The evolution of an aerosol size distribution is modeled by population balance equations, which give rise to nonlinear partial integro-differential equation systems. The nonlinearities usually arise from complex reaction, nucleation, condensation and coagulation rates and their nonlinear dependence on temperature. A variety of solution techniques have been developed to address the complexity at various levels. In most cases, one needs to resort to numerical solutions. One of the standard numerical techniques is to discretize the population balance equation using finite difference/element methods (see[4-8], and the references cited therein), but these methods suffer from extremely large computational requirements which cannot be accommodated by conventional computers. Sectional models offer a computationally less demanding solution by approximating the continuous size distribution by a finite number of sections within which the particle size distribution (PSD) function is assumed to be constant [9]. On the other hands, under appropriate simplifying assumptions, analytical solutions have been developed to solve the population balance equation [10-13].

Recently, the issue of population balance mode-based feedback control of PSD has received considerable attention [14-16]. Previous work in this area include stability analysis and the application of the conventional control schemes to crystallizers and emulsion polymerization processes [17]. Unfortunately, conventional control schemes perform poorly in the face of severe process nonlinearities, and may even

lead to destabilization of the closed-loop system. These limitations of conventional control schemes have motivated research efforts towards synthesizing nonlinear model-based feedback controllers on spatially-homogeneous aerosol processes with application to size distribution control in continuous crystallizers [18,19], and titania aerosol reactor [20,21].

However, most real aerosol systems have uncertainties include unknown or partially known time-varying process parameters, exogenous disturbance, and unmodeled dynamics. It is well known that the presence of uncertain variables and unmodeled dynamics, if not taken into account in the controller design, may be lead to severe deterioration of the nominal closed-loop performance or even to closed-loop instability. Therefore, it is very important thing to study robust control of nonlinear aerosol systems with uncertainty.

This paper focuses on robust control of particle process described by uncertain population balances. The objective is to develop a general method for the synthesis of practically implementable robust nonlinear controller that explicitly handle time-varying uncertain variables (such as unknown process parameters and disturbances) and unmodeled dynamics. The robust nonlinear controller enforces stability in the closed-loop system and attenuation of the effect of uncertain variables on the system, and achieve PSD with desired characteristics. To present robust output feedback control techniques that are applicable to a broad range of aerosol systems, we choose to focus on an aerosol process that involves simultaneous chemical reaction, nucleation, condensation and coagulation rather than a specific aerosol.

The remainder of this paper is organized as follows: In Section 2, a general aerosol process is described by population balance and mass and energy balances, which describes the evolution of particle size distribution, continuous phase species and temperature of the aerosol process, respectively. A log-normal moment approximations of the population balance model is presented. In Section 3, the robust state feedback controller is designed for the aerosol system with some unknown uncertainties including unknown or partially known time-varying process parameters, exogenous disturbance, and unmodeled dynamics, the proposed controller is composed of a robust control term and a nominal control term so that it only ensures the stability of the closed-loop system, but also attenuates the effect of the unknown uncertainties on the system. A high-gain observer is adopted to estimate state variables required in the on-line implementation of the state feedback. In Section 4, the proposed robust controller is applied to a general aerosol process for achieving an aerosol size distribution with desired geometric average particle diameter, the simulation results show the robustness properties of the controller with respect to parametric model uncertainty and unmodeled dynamics. Finally, conclusions are given in Section 5.

2 Mathematical model of aerosol process

2.1 Spatially-homogeneous aerosol process

Consider a general aerosol process which described by the following nonlinear partial integro-differential equation [22]:

$$\frac{\partial n(v,t)}{\partial t}+\frac{\partial (G(\bar{x},v)n)}{\partial v}-I(v^*)\delta(v-v^*)$$

$$=\frac{1}{2}\int_0^v \beta(\bar{x},v-\bar{v},\bar{v})n(v-\bar{v},t)n(\bar{v},t)d\bar{v}-n(v,t)\int_0^\infty \beta(v,\bar{v})n(\bar{v},t)d\bar{v} \tag{1}$$

where the term $n(v,t)$ represents the PSD function at time t , v is the particle volume, $I(v^*)$ is the nucleation rate, $G(\bar{x},v)$ and $\beta(\bar{x},v-\bar{v},\bar{v})$ are the diffusional condensation growth function and the Brownian coagulation kernel of agglomerates, $\delta(\cdot)$ is the standard Dirac function.

A mathematical model which predicts the time evolution of the concentrations of species and temperature of the gas phase has the following form [21]:

$$\frac{d\bar{x}}{\partial t} = f(\bar{x}) + g(\bar{x})u(t) + \bar{A}\int_0^\infty a(\eta, v, \bar{x})dv \tag{2}$$

where $\bar{x}(t)$ is an n-dimensional vector of state variables that depend on time, $\bar{A}$ is constant matrix, $f(\bar{x}), g(\bar{x}), a(\eta, v, \bar{x})$ are nonlinear vector functions and $u(t)$ is the time-varying manipulated input (e.g. wall temperature). The term $\bar{A}\int_0^\infty a(\eta, v, \bar{x})dv$ accounts for mass and heat transfer from the continuous phase to all the particles in the population.

The diffusional condensation growth function $G(\bar{x}, v)$ and the Brownian coagulation kernel of agglomerates $\beta(\bar{x}, v-\bar{v}, \bar{v})$, for the free molecule size regimes, are represented by

$$G_{FM}(\bar{x}, v) = B_1 v^{1/3}(S-1), B_1 = (36\pi)^{1/3} v_0 n_s \left(\frac{k_B T}{2\pi m_0}\right)^{1/2},$$

$$\beta_{FM}(\bar{x}, v, \bar{v}) = B_2 \left(v^{1/D_f} + \bar{v}^{1/D_f}\right)^2 \sqrt{\frac{1}{v} + \frac{1}{\bar{v}}}, B_2 = (3/4\pi)^{2/D_f - 1/2} (6k_B T/\rho)^{1/2} r^{2-6/D_f} \tag{3}$$

and for the near-continuum and the continuum regimes

$$G_c(\bar{x}, v) = B_3 v^{1/3}(S-1), B_1 = (48\pi^2)^{1/3} C_f v_0 n_s, C_f = \frac{\lambda}{3}\left(\frac{8k_B T}{\pi m_0}\right)^{1/2}$$

$$\beta_c(\bar{x}, v, \bar{v}) = B_2 \left(v^{1/D_f} + \bar{v}^{1/D_f}\right)\left[\frac{C(v)}{v^{1/D_f}} + \frac{C(\bar{v})}{\bar{v}^{1/D_f}}\right], B_4 = \frac{2k_B T}{3\mu} \tag{4}$$

In the above equations, S is the saturation ratio, T is the absolute temperature, D_f is the fractal (Hausdorff) dimension, C_f is the condensable vapor diffusivity, λ is the mean free path of the gas ($\lambda = v\pi M_w / 2k_B T N_{av}$, where v and M_w are the kinematic viscosity and molecular weight of the fluid, respectively, and N_{av} is the Avogadro's number), μ is the viscosity of the fluid, n_s is the monomer concentration at saturation ($n_s = P_s/k_B T$, where P_s is the saturation pressure), m_0 is the monomer mass, v_0 is the monomer volume, ρ is the particle density, r is the particle radius, $C(v) = 1 + B_5 \lambda / r$ is the Cunningham slip correction factor, and $B_5 = 1.257$.

The nucleation rate $I(v^*)$ is assumed to follow the classical Becker-Doring theory and is given by the following expression [23]:

$$I = n^2 s_0 \left(\frac{k_B T}{2\pi m_0}\right)^{1/2} S^2 \left(\frac{2}{9\pi}\right)^{1/3} \Sigma^{1/2} \exp(-k^* \ln\frac{S}{2}) \tag{5}$$

where s_0 is the monomer surface area and k^* is the number of monomers in the critical size nucleus which is given by:

$$k^* = \frac{\pi}{6}\left(\frac{4\Sigma}{\ln S}\right)^3 \tag{6}$$

where $\Sigma = \gamma v_0^{2/3}/k_B T$ and γ is the surface tension.

Remark 1. It is found experimentally that in many cases of practical interest, the total number of primary particles N_p in an agglomerate is related to characteristic radius R through a power law expression $N_p \sim R^{D_f}$, where the exponent D_f is called the fractal dimension. This is usually true in a statistical sense after averaging over many agglomerates with the same N_p. The value of D_f depends on the details of the agglomerate formation process. For compact agglomerates we have $D_f \to 3$, while for chain-like agglomerates we have $D_f \to 1$.

Remark 2. It is noticed that the collision kernel β_{FM} in (3) is in a form that appears to be rather difficult to expand into a series with a manageable number of terms. The following approximation can be made [11];

$$\left(\frac{1}{v}+\frac{1}{\bar{v}}\right)^{1/2}=b\left(\frac{1}{v^{1/2}}+\frac{1}{\bar{v}^{1/2}}\right)$$

The variables for b depend on the initial geometric standard deviation σ_0 and on the fractal dimension D_f .

Remark 3. The manipulated input $u(t)$ enters system (1)-(2) through (2), this assumption is usually satisfied in most practical applications where the wall temperature is chosen as the manipulated input.

2.2 Moment model

In this subsection, we present moment model of size distribution of fractal agglomerates in aerosol processes with simultaneous chemical reaction, nucleation, condensation and coagulation. Many experimental results and numerical calculation indicate that aerosol PSDs can be adequately described by unimodal lognormal functions. This result makes it possible to develop a moment model for the aerosol process in terms of the three leading moments of the size distribution. The log-normal size distribution function is written as:

$$n(v,t)=\frac{1}{3v}\frac{N(t)}{\sqrt{2\pi}\ln\sigma(t)}\exp\left[\frac{-\ln^2\{v/v_g(t)\}}{18\ln^2\sigma(t)}\right]$$

where $N(t)$ is the total number concentration of particles, $\sigma(t)$ is the geometric standard deviation, and $v_g(t)$ is the geometric number mean particle volume. The kth moment of the particle size distribution is written as:

$$M_k(t)=\int_0^\infty v^k n(v,t)dv$$

where k is an arbitrary real number. According to the properties of a log-normal function, any moment can be written in terms of M_1, v_g and σ as follows

$$M_k=M_1 v_g^{k-1}\exp\left\{\frac{9}{2}(k^2-1)\ln^2\sigma\right\} \tag{7}$$

Obviously, the three leading moments are sufficient to generate the lognormal PSD. Using the same derivation procedure as the one in [21], we can easily obtain the following equations, which describe the time evolution of the three leading moments (*i.e.* the zeroth, first and second moments) of the size distribution for the free molecule size.

$$\frac{dM_0}{dt}=I-bB_2(M_{2/D_f-1/2}M_0+2M_{1/D_f}M_{1/D_f-1/2}+M_{2/D_f}M_{-1/2}) \tag{8}$$

$$\frac{dM_1}{dt}=Iv^*+B_1(s-1)M_{2/D_f-1/2} \tag{9}$$

$$\frac{dM_2}{dt}=Iv^{*2}+2B_1(s-1)M_{1+2/D_f}$$
$$+2bB_2(M_{2/D_f+1/2}M_0+2M_{1+1/D_f}M_{1/D_f+1/2}+M_{1+2/D_f}M_{1/2}) \tag{10}$$

Similar to the case of the free molecule regime, the dynamics of the zeroth and second moments of the aerosol size distribution in the continuum size regime is described by the following equations:

$$\frac{dM_0}{dt}=I-B_4[M_0^2+2M_{1/D_f}M_{-1/D_f}+B_5\lambda\left(\tfrac{4}{3}\pi\right)^{1/5}(M_0M_{-1/D_f}+M_{1/D_f}M_{-1+1/D_f})] \tag{11}$$

$$\frac{dM_1}{dt}=Iv^*+B_3(s-1)M_{1/D_f} \tag{12}$$

$$\frac{dM_2}{dt} = Iv^{*2} + 2B_3(s-1)M_{1+1/D_f}$$

$$+2B_4[M_1^2 + M_{1+1/D_f}M_{1-1/D_f} + +B_5\lambda\left(\tfrac{4}{3}\pi\right)^{1/3}(M_1M_{1-1/D_f} + M_{1/D_f}M_{1+1/D_f})] \tag{13}$$

Table 1 Dimensionless variables

$N = M_0/n_s$, Aerosol concentration
$V = M_1/n_s v_0 d$, Aerosol volume
$V_2 = M_2/n_s v_0^2$, Second aerosol moment
$\tau = (2\pi m_0/k_B T)^{1/2}/ns_0$, Characteristic time for particle growth
$K = (2k_B T/3\mu)n_s\tau$, Coagulation coefficient
$K_{n_0} = \lambda/r_0$, Monomer Knudsen number
$I' = I/(n_s/\tau)$, Nucleation rate
$R_r' = R_r/(n_s/\tau)$, Reaction rate group
$v_g' = v_g/v_0$
$r_g' = r_g/r_0$
$\theta = t/\tau$

The moment equations for the free molecule size and continuum size regimes may be combined to describe the aerosol dynamics over the entire particle size spectrum by using the harmonic average of the dimensionless coagulation rates in the free molecule size and continuum size regimes [23]. Using the dimensionless variables listed in Table 1, we have to the following equations:

Zeroth moment

$$\frac{dN}{d\theta} = I' - \xi N^2 \tag{14}$$

where

$$\xi = \frac{\xi_{FM}\xi_c}{\xi_{FM}+\xi_c}, \quad \xi_{FM} = r_g'^{1/2} b\left[\exp\left(\tfrac{25}{8}\ln^2\sigma\right) + 2\exp\left(\tfrac{5}{8}\ln^2\sigma\right) + \exp\left(\tfrac{1}{8}\ln^2\sigma\right)\right]$$

$$\xi_c = K\left[1 + \exp(\ln^2\sigma) + B_5\left(\frac{K_{n_0}}{r_g'}\right)\exp\left(\frac{1}{2}\ln^2\sigma\right)(1+\exp(2\ln^2\sigma)\right]$$

First moment

$$\frac{dV}{d\theta} = I'k^* + \eta(S-1)N \tag{15}$$

where

$$\eta = \frac{\eta_{FM}\eta_c}{\eta_{FM}+\eta_c}, \quad \eta_{FM} = v_g'^{2/3}\exp(2\ln^2\sigma)$$

$$\eta_c = \frac{4K_{n_0}}{3}v_g'^{1/3}\exp\left(\frac{1}{2}\ln^2\sigma\right)$$

Second moment

$$\frac{dV_2}{d\theta} = I'k^{*2} + 2\varepsilon(S-1)V + 2\zeta V^2 \tag{16}$$

where

$$\varepsilon = \frac{\varepsilon_{FM}\varepsilon_c}{\varepsilon_{FM}+\varepsilon_c}, \quad \zeta = \frac{\zeta_{FM}\zeta_c}{\zeta_{FM}+\zeta_c}$$

$$\varepsilon_{FM} = v_g'^{1/2} b\exp(8\ln^2\sigma), \quad \varepsilon_c = \frac{4K_{n_0}}{3}v_g'^{1/3}\exp\left(\frac{7}{2}\ln^2\sigma\right)$$

$$\zeta_{FM} = r_g'^{1/2} b\exp\left(\tfrac{3}{2}\ln^2\sigma\right)\left[\exp\left(\tfrac{25}{8}\ln^2\sigma\right) + 2\exp\left(\tfrac{5}{8}\ln^2\sigma\right) + \exp\left(\tfrac{1}{8}\ln^2\sigma\right)\right]$$

$$\zeta_c = K\left[1 + \exp(\ln^2\sigma) + B_5\left(\frac{K_{n_0}}{r_g'}\right)\exp\left(-\frac{1}{2}\ln^2\sigma\right)(1+\exp(-2\ln^2\sigma)\right]$$

The rate of change of S can be obtained from a monomer balance and is given by:

$$\frac{dS}{d\theta} = R_r' N_{av} - I' k^* - \eta(S-1)N \tag{17}$$

2.3 Mathematical model of aerosol process

We consider a typical aerosol process in a cylindrical volume with diameter D_T. Under the assumption of lognormal aerosol size distribution, the dimensionless model that describes the evolution of the first three moments of the distribution, along with the saturation ratio, reactant concentrations and fluid temperature, takes the following form:

$$\begin{aligned}
\frac{dN}{d\theta} &= I' - \xi N^2 \\
\frac{dV}{d\theta} &= I' k^* + \eta(S-1)N \\
\frac{dV_2}{d\theta} &= I' k^{*2} + 2\varepsilon(S-1)V + 2\zeta V^2 \\
\frac{dS}{d\theta} &= C\bar{C}_1\bar{C}_2 - I' k^* - \eta(S-1)N \\
\frac{d\bar{C}_1}{d\theta} &= -A_1\bar{C}_1\bar{C}_2 \\
\frac{d\bar{C}_2}{d\theta} &= -A_2\bar{C}_1\bar{C}_2 \\
\frac{d\bar{T}}{d\theta} &= B\bar{C}_1\bar{C}_2\bar{T} + E\bar{T}(\bar{T}_w - \bar{T})
\end{aligned} \tag{18}$$

where $\bar{C}_1$ and $\bar{C}_2$ are the dimensionless reactant concentrations, $\bar{T}, \bar{T}_w$ are the dimensionless fluid temperature and the dimensionless temperature of the heat transferring medium at the system boundary, respectively. A_1, A_2, B, C, E are dimensionless quantities, their explicit form are given in Table 2.

Table 2 Dimensionless variables for the model

$A_1 = \tau k P_0 y_{20}/RT_0$
$A_2 = \tau k P_0 y_{10}/RT_0$
$B = \tau k P_0 \Delta H_\tau y_{10} y_{20}/RT_0^2 C_p$
$C = N_{av}\tau k y_{10} y_{20}(P_0/RT_0)^2/n_s$
$E = 4URT_0\tau/D_T C_p P_0$
$\bar{C}_i = y_i/y_{i0}\bar{T}$
$\bar{T} = T/T_0$
$\bar{T}_w = T_w/T_0$

We formulate the control problem as one of tracking the geometric average particle diameter of the aerosol system, by manipulating the wall temperature, *i.e.*

$$y(t) = d_{pg}(t), \quad u(t) = \bar{T}_w(t) - \bar{T}_{ws}$$

where $\bar{T}_{ws} = T_{ws}/T_0 = 1$.

We write the model (18) in the following form:

$$\frac{d\tilde{x}}{dt} = \tilde{f}(\tilde{x}) + \tilde{g}(\tilde{x})u(t), y = \tilde{h}(\tilde{x}) \tag{19}$$

where the explicit form of $\tilde{x}, \tilde{f}(x), \tilde{g}(\tilde{x})$ can be obtained by comparing (18) and (19).

Remark 4. In addition to being highly nonlinear, the real aerosol processes have uncertainties including unknown or partially known time-varying process parameters, exogenous disturbance, and unmodeled dynamics. Therefore, the real aerosol systems can be described by the following systems.

$$\frac{d\tilde{x}}{dt} = \tilde{f}(\tilde{x}) + \Delta\tilde{f}(\tilde{x}) + (\tilde{g}(\tilde{x}) + \Delta\tilde{g}(\tilde{x}))u(t), y = \tilde{h}(\tilde{x}) \tag{20}$$

where $\Delta\tilde{f}(\tilde{x})$ and $\Delta\tilde{g}(\tilde{x})$ represent uncertainties caused by unknown or partially known time-varying process parameters, exogenous disturbance, and unmodeled dynamics.

3 Robust controller design

In this section, we will begin with the design of the robust controller. Our objective is to design a robust output feedback controller which guarantees boundedness of all variables for the closed-loop system and tracking of a given reference signal y_d .

By differentiating the output $y(t)$ with respect to t , we obtain the following nonlinear system represented by the differential equation

$$\begin{array}{l} y^{(r)} = f(y, y^{(1)}, \cdots, y^{(r-1)}) + \Delta f(y, y^{(1)}, \cdots, y^{(r-1)}) + \\ (g(y, y^{(1)}, \cdots, y^{(r-1)}) + \Delta g(y, y^{(1)}, \cdots, y^{(r-1)}))u \end{array} \tag{21}$$

where r is the relative order (The definition can be found in [24], and omitted here for brevity) of the output $y(t)$ with respect to the manipulated input $u(t)$.

By taking $x_1 = y, x_2 = y^{(1)}$ up to $x_r = y^{(r-1)}$, we can represent the augmented system by the state space model

$$\begin{array}{l} \dot{x} = Ax + B(v + D_1(x)) \\ y = x_1 \end{array} \tag{22}$$

where $x = (x_1, x_2, \cdots, x_r)^T, \quad v = f(x) + g(x)u, \quad \delta_1(x) = \Delta g(x)/g(x),$

$D_1(x) = \Delta f(x) + \delta_1(x)(v - f(x))$, (A, B) is controllable canonical pairs of the form

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \in R^{r\times r}, B = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \in R^{r\times 1}$$

Assuming the states is available for feedback, taking

$$\begin{array}{l} e_1 = y - y_r = x_1 - y_r \\ e_2 = \dot{y} - \dot{y}_r = x_2 - \dot{y}_r \\ \vdots \\ e_r = y^{(r-1)} - y_r^{(r-1)} = x_r - y_r^{(r-1)} \end{array}$$

and $e = (e_1, e_2, \cdots, e_r)^T$, we rewrite (22) as

$$\dot{e} = Ae + B(v + D_1(e + Y_d) - y_d^{(r)}) \tag{23}$$

where $Y_d = (y_d, \dot{y}_d, \cdots, y_d^{(r-1)})^T$.

Obviously, if $D_2(\cdot)=0$ (*i.e.* $\Delta f(x)=\Delta g(x)=0$), we can use the following nominal state feedback control to achieve our objective.

$$u_1=\frac{1}{g(x)}(-Ke+y_d^{(r)}-f(x)) \tag{24}$$

where matrix K is chosen such that the matrix $A_m=A-BK$ is Hurwitz, *i.e.*, all its eigenvalues are in the left open plane.

However, due to the presence of uncertainties $\Delta f(x)$ and $\Delta g(x)$, a nominal state feedback control alone cannot ensure the stability of the closed-loop system, it is necessary to add a robust compensator to deal with it.

$$u_2=-\frac{1}{g(x)(1-\Delta_1(x))}\frac{B^TPe\mu^2(x)}{\|B^TPe\|\mu(x)+\varepsilon_1\exp(-\varepsilon_2 t)} \tag{25}$$

where $\varepsilon_1,\varepsilon_2>0$, $P=P^T>0$ is a solution of the following Lyapunov equation

$$P(A-BK)+(A-BK)^TP=-Q,\quad \forall Q^T=Q>0$$

$$\mu(x)=\Delta_1(x)\left(\left|y_d^{(r)}-Ke\right|+|f(x)|\right)+|\Delta f(x)|,\quad \Delta_1(x)=\sup_{x\in R^r}|\delta_1(x)|$$

The robust state feedback controller we proposed is as follows

$$u=u_1+u_2 \tag{26}$$

Theorem 5. *Suppose that $\Delta_1(x)=\sup\limits_{x\in R^r}|\delta_1(x)|<1$, consider system (22) under the robust state feedback controller (26), the closed-loop system have the following properties:*

(1) The state of the closed-loop system is bounded.

(2) The output of the closed-loop system satisfy $\lim\limits_{t\to\infty}|y-y_d|=0$.

Proof: Let

$$v_2=-\frac{1}{(1-\Delta_1(x))}\frac{B^TPe\mu(x)^2}{\|B^TPe\|\mu(x)+\varepsilon_1\exp(-\varepsilon_2 t)}$$

The closed-loop system under the robust state feedback controller (26) is as follows

$$\dot{e}=(A-BK)e+B(v_2+D_1(x)) \tag{27}$$

Consider the Lyapunov function candidate

$$V(e)=e^T(t)Pe(t)$$

The derivative of $V(e)$ along the trajectories of the system is given by

$$\dot{V}(e)=2e^TP(A-BK)e+2e^TPB(v_2+D_1(x)) \tag{28}$$

Obviously, we have

$$|D_1(x)|\le|\Delta f(x)|+\Delta_1(x)(|v|+|f(x)|)$$

$$\le\Delta_1(x)|v_2|+|\mu(x)|$$

Therefore

$$2e^TPB(v_2+D_1(x))$$

$$\le-2(1-\Delta_1(x))\left\|B^TPe\right\|\cdot|v_2|+2\left\|B^TPe\right\|\mu(x)$$

$$\leq \frac{2B^T Pe\mu(x)\varepsilon_1 \exp(-\varepsilon_2 t)}{\|B^T Pe\|\,\mu(x)+\varepsilon_1 \exp(-\varepsilon_2 t)} \leq 2\varepsilon_1 \exp(-\varepsilon_2 t) \tag{29}$$

Substituting (29) into (28), we have

$$\dot{V}(e) \leq -\lambda_{\min}(Q)\,\|e\|^2 + 2\varepsilon_1 \exp(-\varepsilon_2 t) \tag{30}$$

where $\lambda_{\min}(Q)$ denotes the minimum eigenvalue of Q . Using the lemma in [25], we have

$$\lim_{t\to\infty} e = 0$$

This completes the proof. □

The on-line implementation of the controller (26) requires that the values of the state variables x are known. Unfortunately, x may be not be known in many practical applications. One way to address this problem is to use state observer to estimate x. Here we use the high-gain state observer as

$$\begin{aligned} \dot{\hat{e}}_i &= \hat{e}_{i+1} + \tfrac{\alpha_i}{\varepsilon}(e_1 - \hat{e}_1), \quad 1 \leq i \leq r-1 \\ \dot{\hat{e}}_n &= \tfrac{\alpha_n}{\varepsilon}(e_1 - \hat{e}_1) \end{aligned} \tag{31}$$

where $\hat{e} = \hat{x} - Y_d$, $\hat{x}$ denote the estimate state of the state x, ε is a small positive parameter, the constants $\alpha_i > 0$ are chosen such that the roots of

$$s^r + \alpha_1 s^{r-1} + \cdots + \alpha_{r-1} s + \alpha_r = 0$$

have negative real parts.

The state feedback controller (26) and the state observer (31) can be combined to yield the following nonlinear robust output feedback controller:

$$\begin{aligned} \dot{\hat{e}}_i &= \hat{e}_{i+1} + \tfrac{\alpha_i}{\varepsilon}(e_1 - \hat{e}_1), \quad 1 \leq i \leq r-1 \\ \dot{\hat{e}}_n &= \tfrac{\alpha_n}{\varepsilon}(e_1 - \hat{e}_1) \end{aligned}$$

$$u = \frac{1}{g(\hat{e}+Y_d)}\left(y_d^{(r)} - K\hat{e} - f(\hat{e}+Y_d) - \frac{1}{(1-\Delta_1(\hat{e}+Y_d))}\frac{B^T P\hat{e}\mu^2(\hat{e}+Y_d)}{\|B^T P\hat{e}\|\,\mu(\hat{e}+Y_d)+\varepsilon_1 \exp(-\varepsilon_2 t)}\right) \tag{32}$$

Theorem 6. *Suppose that* $\Delta_1(x) = \sup_{x\in R^r} |\delta_1(x)| < 1$, *consider system (22) under the robust state feedback controller (32), the closed-loop system have the following properties:*

(1) The state of the closed-loop system is bounded.

(2) The output of the closed-loop system satisfy $\lim_{t\to\infty} |y - y_d| = 0$.

We can prove Theorem 2 by following the proof line of Theorem 1, the detailed proof is omitted here.

4 Simulation study

The performance of the nonlinear robust output feedback controller (28) was tested through simulations. The values of the process parameters are shown in Table 3.

Table 3 Model parameters for the simulation study

$D_T = 0.05m$, Process diameter
$P_0 = 1atm$, Process pressure
$T_0 = 298K$, Process pressure
$Y_{10} = Y_{20} = 40ppm$, Initial mole fraction of reactants
$U = 10.4Jm^{-2}s^{-1}K^{-1}$, Overall coefficient of heat transfer
$\Delta H_r = 175.7KJml^{-1}s^{-1}$, Heat of reaction
$C_p = 29.1Jmol^{-1}K^{-1}$, Heat capacity of process fluid
$M_w = 14.0 \times 10^{-3}kgmol^{-1}$, Molecular weight of process fluid
$k = 11.4m^3mol^{-1}s^{-1}$, Reaction constant
$\mu = 3.5 \times 10^{-6}kgm^{-1}s^{-1}$, Viscosity of process fluid
$\gamma = 0.08Nm^{-1}$, Surface tension
$v_0 = 5.33 \times 10^{-29}m^3$, Molecular weight of process fluid
$R = 8.314Jmol^{-1}K^{-1}$, Universal gas constant
$N_{av} = 6.023mol^{-1}$, Avogadro' constant
$k_B = 1.38 \times 10^{-23}JK^{-1}$, Boltzmann's constant

Several simulations were performed to evaluate the disturbance attenuation and set-point tracking capabilities of the robust nonlinear controller, as well as its robustness with respect to uncertainties in model parameters and unmodeled dynamics.

The objective of these simulations is to show that the use of proposed robust control allows producing an aerosol product with a desired geometric average particle diameter ($d_{pg} = 0.18\mu m$) within a given type of batch reactor even in the presence of uncertainty in the process model parameters. Figure 1 presents the closed-loop trajectory (dashed line) and closed-loop trajectory (solid line) for d_{pg} under robust nonlinear controller and nonlinear controller, respectively, and the corresponding manipulated input trajectory in the case of 10% error in the value of the rate constant k. It is clear that the robust nonlinear control regulates the output d_{pg} to the desired point value (*i.e.* $d_{pg} = 0.18\mu m$), and attenuate the effect of time-varying uncertainty on the process. However, nonlinear control cannot guarantee the output d_{pg} to the desired point values in the presence of uncertainty. There is an error between steady-state value and the desired point one. Figure 2 presents the closed-loop trajectory (dashed line) and closed-loop profile (solid line) for d_{pg} under robust nonlinear controller and nonlinear controller, and the corresponding manipulated input trajectory in the case of a 8% decrease in the parameters. Again, the robust nonlinear controller allows achieving an aerosol product with the desired d_{pg}.

Also we simulated the robustness properties of the proposed controller with respect to parametric model uncertainty and unmodeled dynamics in the presence of a set-point change, the robust nonlinear controller was found to have very good robustness properties, keeping the output on the set-point.

Remark 7. It is worth to point out that the proposed approach for the design of robust nonlinear controllers is applicable to most aerosol systems for which the hypothesis of unimodal lognormal aerosol size distribution for long times is valid.

5 Conclusions

In this paper, we present a comprehensive study on robust control of particle size distribution of fractal agglomerate in aerosol processes with simultaneous chemical reaction, nucleation, condensation and coagulation. Initially, a general aerosol process is presented by population balance and mass and energy balances, which describes the evolution of particle size distribution, continuous phase species and temperature of the aerosol system, respectively. A lognormal moment approximations of the population balance model is then presented. Then, the robust state feedback controller is designed for the aerosol system with some unknown uncertainties, the proposed controller is composed of a robust control term and an nominal control term so that it only ensures the stability of the closed-loop system, but also atten-

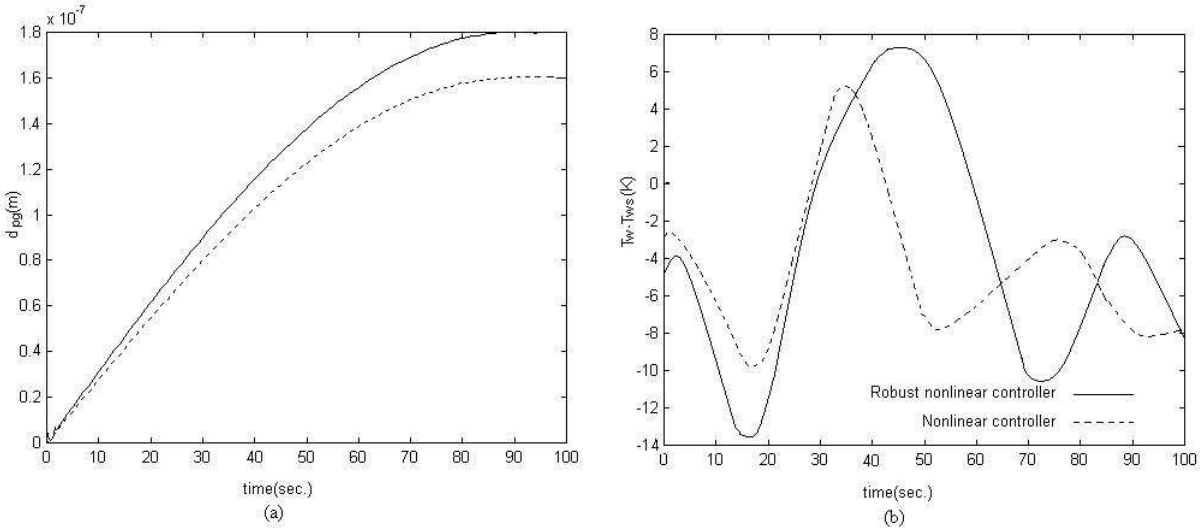


Figure 1: (a) d_{pg} using the robust nonlinear controller and nonlinear controller, (b) The corresponding manipulated input trajectory

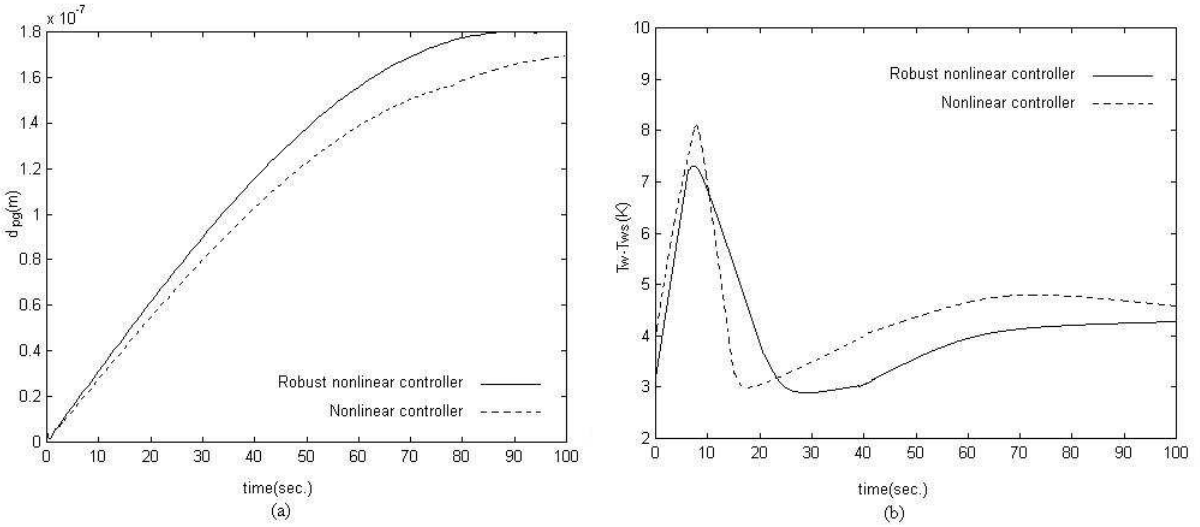


Figure 2: (a) d_{pg} using the robust nonlinear controller and nonlinear controller, (b) The corresponding manipulated input trajectory

uates the effect of the unknown uncertainties on the system. A high-gain observer is adopted to estimate state variables required in the on-line implementation of the state feedback. Finally, the proposed robust controller is applied to an aerosol process for achieving an aerosol size distribution with desired geometric average particle diameter, the simulation results show the robustness properties of the controller with respect to parametric model uncertainty and unmodeled dynamics.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No. 60974027).

Bibliography

[1] J. M. Hidy and J. R. Brock, *The Dynamic of Aerocolloidal System*, Oxford, Pergamon Press, 1970.

[2] V. M. Voloshuk, and Y. S. Sedunov,*Coagulation phenomena in disperse systems*, Moscow: Moskovskiy Inghenerno Fizicheskiy Institut, 1975 (in Russia)

[3] V. N. Piskunov, *Physical phenomena in disperse systems,* Leninggrad: Gidrometeoizdat Publishers, 1991 (in Russia)

[4] M. J. Hounslow, A discretized population balance for continuous systems at steady-state. *American Institute of Chemical Engineers Journal,* 36, pp. 106-116,1990

[5] P. J. Hill, and K. M. Ng, New discretization procedure for the breakage equation. *American Institute of Chemical Engineers Journal*, 41 , pp. 1204-1216, 1995

[6] P. J. Hill, and K. M. Ng, New discretization procedure for the agglomeration equation. *American Institute of Chemical Engineers Journal*, 42, pp. 727-741, 1996

[7] N. V. Mantzaris , P. Daoutidis, and F. Srienc, Numerical solution of multivariable cell population balance models. Parts: I, II and III. *Computers and Chemical Engineering*, 25, pp. 1411-1481, 2001

[8] M. Nicmanis, and M. J. Hounslow, Finite-element methods for steady-state population balance equations. *American Institute of Chemical Engineers Journal*, 44, pp. 2258-2272, 1998

[9] J. D. Landgrebe, and S. E. Pratsinis, A discrete sectional model for particulate production by gas phase chemical reaction and aerosol coagulation in the free molecular regime. *Journal of Colloid Interface Science*, 139, pp. 63-86, 1990

[10] E. Otto, H. Fissan, S. H. Park, and K. W. Lee, The Log-Normal Size Distribution Theory of Brownian Aerosol Coagulation for the Entire Particle Size Range. Part I: Analytical Solution Using the Harmonic Mean Coagulation Kernel, *Journal of Aerosol Science*, Vol. 30. No. 1 pp. 3-16, 1999

[11] S. H. Park, and K. W. Lee, Analytical Solution to change in size distribution of polydisperse particles in closed chamber due to diffusion and sedimentation, *Atmospheric Environment*, Vol. 36. pp. 5459-5467, 2002

[12] S. H. Park, and K. W. Lee, Change in particle size distribution of fractal agglomerates during Browian coagulation in the free-molecule regime, *Journal of Colloid and Intreface Science*, Vol. 246. pp. 85-91, 2002

[13] V. N. Piskunov, and A. I. Golubev, The Generalized Approximation Method for Modelling Coagulation Kinetics ¨C Part I: Justification and Implementation of the Method, *Journal of Aerosol Science* , Vol. 33, pp. 51-63,2002

[14] P. Daoutidis, and P. D. Christofides, Dynamic feedforward/output feedback control of nonlinear processes, *Chemical Engineering Science*, Vol. 50, pp. 1889-2007, 1995

[15] A. Kalani, and P. D. Christofides, Nonlinear control of spatially-in homogeneous aerosol processes, *Chemical Engineering Science*,Vol. 54, pp. 2669-2678, 1999

[16] T. J. Crowley, E. S. Meadows, E. Kostoulas, and F. J. Doyle III, Control of Particle Size Distribution Described by a Population Balance Model of Semibatch Emulsion Polymerization, *Journal of Process Control*, Vol.10, 419-432, 2000.

[17] D. Semino, and W. H. Ray, Control of systems described by population balance equations-II. Emulsion polymerization with constrained control action, *Chemical Engineering Science*, Vol.50, pp. 1825-1839, 1995.

[18] T. Chiu, and P. D. Christofides, Nonlinear control of particulate processes, *American Institute of chemical Engineers Journal*,Vol. 45, pp. 1279-1297, 1999.

[19] N. H. El-Farra, T. Chiu, and P. D. Christofides, Analysis and control of particulate processes with input constraints, *American Institute of Chemical Engineering Journal*, 47, pp. 1849-1865, 2001.

[20] A. Kalani, and P. D. Christofides, Modeling and control of a titania aerosol reactor, *Aerosol Science and Technology*, 32, pp. 369-391, 2000.

[21] A. Kalani, and P. D. Christofides, Simulation, estimation and control of size distribution in aerosol processes with simultaneoue reaction, nucleation, condensation and coagulation, *Computers and Chemical Engineering*, 26, pp. 1153-1169, 2002.

[22] S. K. Friedlander , *Smoke, Dust and Haze: Fundamentals of Aerosol Dynamics*, New York: Oxford University Press, 2000.

[23] S. E. Pratsinis, Simultaneous nucleation, condensation, and coagulation in aerosol reactors, *Journal of Colloid Interface Science*, 124, pp. 416-426, 1988.

[24] A. Isidori, *Nonlinear control systems: an introduction* (Second Edition). Berlin, Heidelberg, Springer-verlag, 1989.

[25] E. Yaz, Stabilizing compensator design for uncertain nonlinear systems, *Systems and Control Letter*, 21, pp.11-17, 1993

Zhengrong Xiang was born in China, in 1969. He obtained his Ph.D. degree in Control Theory and Control Engineering from the Nanjing University of Science and Technology, China, in 1998. Since 2001, he has been an associate professor in the School of Automation. He is an IEEE member. His main research interests include nonlinear control, robust control, intelligent control, and switched systems. He has published significantly on the subjects with over 80 technical papers in journals and conferences. E-mail: xiangzr@mail.njust.edu.cn.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 3, pp. 398-409

Role-Based Access Control for the Large Hadron Collider at CERN

I. Yastrebov

Ilia Yastrebov
1. European Organization for Nuclear Research
Switzerland, 1211 Geneva 23, and
2. Joint Institure for Nuclear Research
Russia, 141980 Dubna, 6 Jolio-Curie
E-mail: ilia.yastrebov@cern.ch

Abstract: Large Hadron Collider (LHC) is the largest scientific instrument ever created. It was built with the intention of testing the most extreme conditions of the matter. Taking into account the significant dangers of LHC operations, European Organization for Nuclear Research (CERN) has developed multi-pronged approach for machine safety, including access control system. This system is based on role-based access control (RBAC) concept. It was designed to protect from accidental and unauthorized access to the LHC and injector equipment.
This paper introduces the new model of the role-based access control developed at CERN and gives detailed mathematical description of it. We propose a new technique called dynamic authorization that allows deploying RBAC gradually in the large systems. Moreover, we show how the protection for the very large distributed equipment control system may be implemented in efficient way. This paper also describes motivation of the project, requirements and overview of the main components: authentication and authorization.

Keywords: Software development, role-based access control, information security, equipment protection.

1 Introduction

The Large Hadron Collider was built by the European Organization for Nuclear Research (CERN) with the intention of testing various predictions of high-energy physics, including the existence of the hypothesized Higgs boson and of the large family of new particles predicted by super-symmetry [1]. The LHC is the world's largest and highest-energy particle accelerator. It is contained in a circular tunnel with a circumference of 27 kilometers, at a depth ranging from 50 to 175 meters underground [2]. The LHC uses some of the most powerful dipoles and radio-frequency cavities in existence. The size of the tunnel, magnets, cavities and other essential elements of the machine represent the main constraints that determine the design energy of 7TeV per proton beam [3].

The energy stored in the LHC magnets and beam is enormous, and the potential for crippling machine damage is a serious concern. That's why European Organization for Nuclear Research (CERN) has developed a multi-pronged approach for machine safety [4]:

- Hardware Protection
  - LHC Beam Interlock System
  - Powering Interlock System
- Software Interlock System
- Role-Based Access Control

- Prevents unauthorized access to equipment
- Provide logging to detect errant settings

Role-Based Access Control (RBAC) is an approach to restrict system access to authorized users. Within an organization, roles are created for various job functions. The permissions to perform certain operations are assigned to specific roles. Members of staff (or other system users) are assigned particular roles, and through those role assignments acquire the permissions to perform particular system functions [5]. RBAC is a preventative and therefore inexpensive way to protect the accelerator equipment. It keeps users from making the wrong settings or from logging into the application. Other machine protection systems such as interlocks are reactive and once triggered it is expensive to recover operations.

RBAC is also used to ensure machine stability during a run. Once the equipment is fine tuned and beam is in the machine, an error setting can disrupt operations for hours and lose valuable data. It is important to mention that RBAC is not a security system against hackers; it is designed only to prevent well meaning people from making the wrong setting, and unauthorized users who have no credentials from running the control applications. The last motivation is that RBAC implements logging for each setting protected by access rules. This is crucial during commissioning and debugging. Each setting can be traced and bugs in the sequencer or operations can be caught and corrected [6].

As part of the problem solution a new mathematical model of the role-based access control (hereinafter CERN-RBAC) was proposed. Equipment protection subsystem based on this model has been successfully implemented and deployed at CERN. The subject of this paper is to describe the new concept of CERN-RBAC for distributed equipment control system. The paper also demonstrates the practical implementation of the major system components, and presents the results of the comprehensive testing.

2 Mathematical Model

Role-based access control, as formalized in 1992 by David Ferraiolo and Rick Kuhn [7], has become the predominant model for advanced access control. In 2000, the Ferraiolo-Kuhn model was integrated with the framework of Sandhu et al. [8] to create a unified model for RBAC, published as the NIST RBAC model [9] and adopted as an ANSI/INCITS standard in 2004. Today, most information technology vendors have incorporated RBAC into their product lines, and the technology is finding applications in areas ranging from health care to defense, in addition to the mainstream commerce systems for which it was designed [5]. We propose a new model of CERN-RBAC for the distributed control system. This concept is based on the standard model and preserves the advantages of scalable security administration that RBAC-style models offer. Moreover it significantly extends standard RBAC model according to specific requirements and yet offers the flexibility to specify complex access restrictions based on the dynamic security attributes. The new CERN-RBAC model is quite general and flexible and could be used in many other areas for equipment access control. Below we give a formal mathematical description of the model in terms of sets and relations.

U – A set of users, $\{u_1, u_2, \ldots, u_n\}$. The user is either a human user or a computer program.
R – A set of roles, $\{r_1, r_2, \ldots, r_n\}$. Role is a job function which defines an authority level.
P – A set of permissions, $\{p_1, p_2, \ldots, p_n\}$. Permission (access rule) is an approval of a mode of access to a resource.
UA – User assignment: operation which assigns concrete roles to the users.

$$UA \subseteq U \times R \tag{1}$$

PA – Permission assignment: $R \rightarrow 2^p$ – function, defining a set of access rules for each role. This condition must be met at that:

$$\forall p \in P, \exists r \in R : p \in PA(r) \tag{2}$$

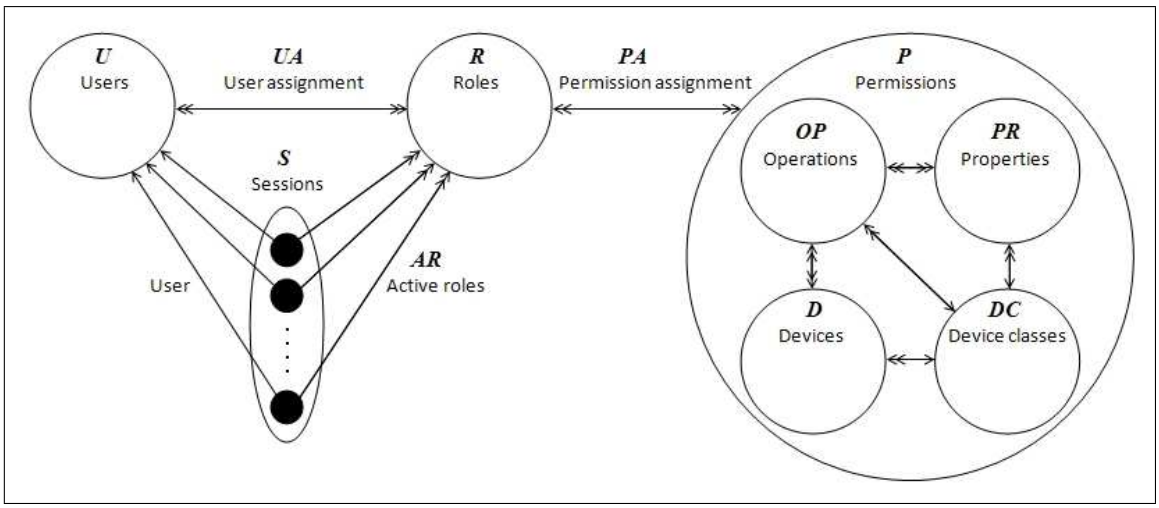


Figure 1: Mathematical model of CERN-RBAC

S – A set of sessions, $\{s_1, s_2, \ldots, s_n\}$. Session (subject) is mapping of one user to possibly many roles. The double-headed arrow from the session to R in Figure 1 indicates that multiple roles are simultaneously activated. The permissions available to the user are the union of permissions from all roles activated in that session. Each session is associated with a single user, as indicated by the single-headed arrow from the session to U in Figure 1. This association remains constant for the life of a session.

$$user(s_i) = S \rightarrow U \tag{3}$$

AR – A set of the subject's active roles (which can change with time). For each subject, the active role is the one that the subject is currently using.

$$AR(s_i) = S \rightarrow 2^R \tag{4}$$

$$AR(s_i) \subseteq \{r \in R \mid (user(s_i), r) \in UA\} \tag{5}$$

Sessions are under the control of individual users. As far the model is concerned, a user can create a session and choose to activate some subset of the user's roles. Roles active in a session can be changed at the user's discretion. The session terminates at the user's initiative. A system may also terminate a session if it is inactive for too long [8].

CERN-RBAC Model

In the previous sections we described the main elements of the RBAC model included in the standard [9]. However the standard model is quite general and abstract to protect the equipment against unauthorized access. The standard concept allows defining permissions only for simple objects. Without further extension it cannot describe more complex interaction. Moreover, the specificity of the equipment requires taking into account its current working mode. That is the authorization algorithm must be dynamic. We propose a new model of the role-based access control for the distributed control system, called CERN-RBAC.

Within this model device is a named entity in the control system. Device can be controlled via its properties. Each property has a name and value. The set of properties specific for the group of homogenous devices forms a device class. The model also defines operations for work with all environments in the system. These are get, set and monitor operations, that allows reading value from device, writing value to device and monitor the device property correspondingly.

OP – A set of operations, $\{get, set, monitor\}$.

PR – A set of device properties, $\{pr_1, pr_2, \ldots, pr_n\}$.
DC – A set of device classes, $\{dc_1, dc_2, \ldots, dc_n\}$.

$$DCA = DC \rightarrow 2^{PR} \tag{6}$$

D – A set of devices (equipments), $\{d_1, d_2, \ldots, d_n\}$. Now we give a function that for every device from the set D defines associated device class:

$$class(d_i) = D \rightarrow DC \tag{7}$$

The properties set for concrete device (APR) are defined as:

$$APR(d_i) = \{pr \in PR \mid (class(d_i), pr) \in DCA\} \tag{8}$$

A set of transactions, which user can execute working within the distributed control system:

$$T = \{(op, pr, d) \mid op \in OP, pr \in PR, d \in D, pr \in APR(d)\} \tag{9}$$

This implies that interaction of user and control system comes to execution of operations from the set OP for the properties of some devices. In terms of proposed model, permission defines for the given role a set of transactions potentially executable by the user owning that role. Function defining a set of transactions for each permission (transaction assignment):

$$TA = P \rightarrow 2^T \tag{10}$$

The predicate $exec(s_i, t_j)$ is true if subject can execute transaction at the current time, otherwise it is false. Subject can execute transaction only if it has active roles:

$$\forall s \in S, t \in T, (exec(s,t) \Rightarrow AR(s) \neq o) \tag{11}$$

Authorization is the function of specifying access rights to resources or services. In our case subject can execute transaction from the set T only if this transaction is authorized for at least one active role of the subject:

$$\forall s \in S, t \in T, (exec(s,t) \Rightarrow t \in TA(PA(AR(s)))) \tag{12}$$

This rule ensures that users can execute only transactions for which they are authorized. Because the conditional is "only if", this rule allows the possibility that additional restrictions may be placed on transaction execution. That is, the rule does not guarantee a transaction to be executable just because it is in the set of transactions potentially executable by the subject's active role [10].

Dynamic Authorization

Equipment functions in different modes. While a device is working in the test mode it's often necessary to allow access for wider range of users than during normal operation. In such cases an expansion of the access rules is not always desirable and appropriate. Firstly, it may loosen up on the system security, and secondly it requires significant administrative costs. We believe the introduction of different working modes of authorization assists to problem solving. In this case the authorization algorithm will take into account not only access rules, but also the current working mode of the equipment.

CERN has a lot of exposed equipments due to the size of the LHC, which contains hundreds of thousands different devices with dozens of properties each. Thus it takes a lot of time to design access rules for all devices. As far as we cannot force equipment specialists to define these rules in a single day, we have to propose the flexible solution to regulating access to non-protected equipments. This will allow us to deploy the access control system step-by-step, without breaking existing infrastructure. It's the crucial requirement for the access control system at CERN.

Dynamic authorization is the algorithm of authorization taking into accounts not only access rules, but also internal state of the authorization subject. Moreover this approach defines default privileges for unprotected environment and non-authenticated users.

CP a set of checking policies considered by dynamic authorization algorithm, {no-check,lenient,strict}. A checking policy is defined at the level of every device and can be changed at runtime. The function mapping each device from the set D to the associated checking policy:

$$policy(d_i) = D \rightarrow CP \tag{13}$$

Now let's introduce the predicate of dynamic authorization as $\psi(s, op, pr, d, policy(d))$, which will be true if subject can execute operation from OP on concrete device property.

$$\forall s \in S, op \in OP, pr \in PR, d \in D : (exec(s,t) \Rightarrow \psi(s, op, pr, d, policy(d))) \tag{14}$$

Now we define a predicate which is true if property is considered to be protected.

$$protected(op \in OP, pr \in PR, d \in D) = 1 \Leftrightarrow (op, pr, d) \in T \tag{15}$$

That is property is protected if there is at least one access rule which restricts access for the given operation. Below we give a description of the dynamic authorization algorithm for each checking policy.

No-check policy grants access for each property without any checking. Typically this policy is used at design stage, when the device interface is not fixed and there are no access rules yet. This policy is also used during testing phase if needed to permit equipment access for some additional users for a short period of time. This mode could be useful for system debugging because or for other activity when it's required to disable CERN-RBAC authorization checks.

$$\psi(no-check) = \begin{cases} 1, protected(pr,d) = 1; \\ 1, protected(pr,d) = 0. \end{cases} \tag{16}$$

Lenient checking policy implements relaxed authorization. For the protected properties algorithm gives access only if corresponding access rule permits so. That is in order to deal with protected properties user must be authenticated in the system. For unprotected properties access is not restricted for any users. Typically this policy is used at the testing stage, when access rules exist only for the most critical settings of the equipment. Some of the devices work in this mode permanently, because sometimes it's desirable to restrict access only to significant settings while keeping others unprotected.

$$\psi(lenient) = \begin{cases} (op, pr, d) \in TA(PA(AR(s))), protected(pr,d) = 1; \\ 1, protected(pr,d) = 0. \end{cases} \tag{17}$$

Strict checking policy implies the most exacting verification. It always requires users to be authenticated in the system; otherwise user's requests will be blocked. Access for the protected properties is granted only if there is an associated access rule. For unprotected properties access is permitted only for reading property value and monitoring operation. Setting new value even for unprotected property is forbidden. This checking policy is the strictest in existence at CERN. Our final goal is to propagate this mode as wide as possible, because it provides best security. All critical equipments are supposed to work in this mode.

$$\psi(strict) = \begin{cases} 0, s = 0; \\ (op, pr, d) \in TA(PA(AR(s))), protected(pr,d) = 1; \\ 1, op \in \{get, monitor\}, protected(pr,d) = 0; \\ 0, op = set, protected(pr,d) = 0. \end{cases} \tag{18}$$

Thereby role-based access control concept with dynamic authorization solves the problem of unprotected properties and legacy applications working without authentication. Introducing this approach we

get a system that works dynamically and its behavior can be easily changed at runtime. One of the main advantages of this approach is that it allows deploying CERN-RBAC step-by-step without interruption of the existing software. This requirement is very important for such a huge project like LHC. Our final goal is to protect every single device, but it's impossible to accomplish this job even in one year, taking into account number of environments. We believe that our model of CERN-RBAC could be a good example of the flexible system and could be useful in many other applications for very large distributed systems. Based on the new model of CERN-RBAC we developed and deployed software products. In the following chapters we overview the main technical requirements of the product, and describe implementation of its main components.

3 System Requirements

To distinguish authentication from the closely related term authorization, the shorthand notations A1 (authentication) and A2 (authorization) are occasionally used. In the present paper we list only the most important requirements of the authentication and authorization components of the CERN-RBAC system [11]. Both parts are implemented independently, as two different systems, which do not interact in any way other than passing the users' credentials from A1 to A2.

Authentication requirements:

- Encryption: the credentials used to authenticate the user shall be encrypted when sent over the network.
- Hardware Independence: the method of authentication shall be independent of specialized hardware such as a card reader, finger print reader etc.
- Quick and simple: the method of authentication must be straightforward for the users.
- Authentication Method: authentication shall be done via user name and password from a personal CERN account. The software should allow flexibility for future implementations of Kerberos and/or X.509 certificates.
- Authentication by Location should be implemented as an additional authentication method. This will allow users to be authenticated without providing credentials from a very limited set of trusted machines located in the CERN Control Centre (CCC).
- Role activation mechanism should be implemented, which will allow users to pick up active roles from the set of available roles.

Authorization requirements:

- Subject of authorization: it shall be possible to restrict access i.e. define access privileges for the following operations on each device property: read, monitor, and write.
- Permission Administration: the permissions shall be defined during the design/deployment phase, and authorized administrators shall be able to edit the permissions at any time.
- Logging/tracking: CERN-RBAC shall keep track of all write actions.
- Performance: authorization shall be fast and shall not hinder the performance of the middleware.

4 Authentication

The purpose of the CERN-RBAC authentication system is to verify the digital identity of a principal (which is either a human user or a program). If the authentication succeeds its result is a digitally signed authentication token that is returned to the application [12]. The program can use the token whenever it needs to interact with various parts to the control system. For example, the token can be provided as one of the arguments in a remote call to set a device. Front-ends and the middleware that are receiving such calls will verify the token, thus confirming the identity of the remote party, and can use it as a base for authorization.

The CERN-RBAC authentication token is a short-term uniform substitute of the real credentials. It gets issued by a central service that can reliably verify the user's identity. Various recipients of the tokens can validate them quickly and easily, and use for making authorization decisions [6]. Figure 2 demonstrates the authentication process.

1) Application sends login request to CERN-RBAC.
2) CERN-RBAC checks the user location in the database (for authentication by location).
3) If location is trusted, then the new token immediately created and returned to the user (authentication by location).
4) Otherwise CERN-RBAC sends to the application the login dialog for credentials or certificate. The user types in the name and password or chooses the certificate and sends it to CERN-RBAC.
5) CERN-RBAC verifies the credentials with CERN Account System (NICE).
6) If the check was successful CERN-RBAC retrieves roles from the CERN-RBAC database, generates a new token and signs it with private key to prevent any modifications.
7) CERN-RBAC returns to the application a digitally signed token containing the user roles.

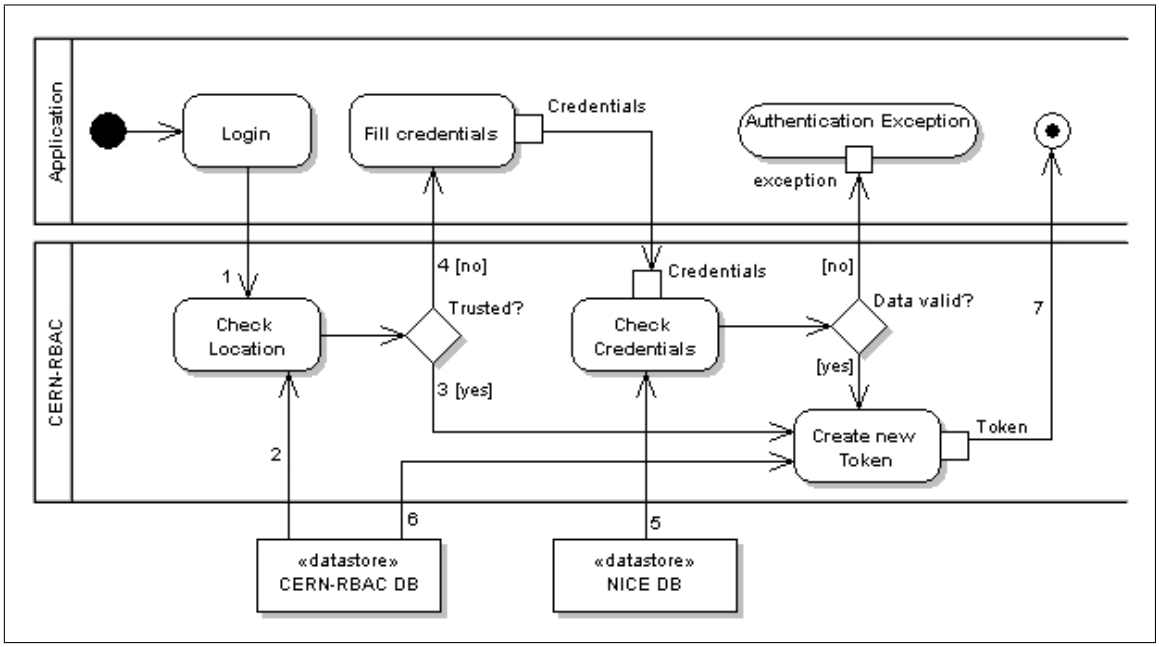


Figure 2: Authentication process

The A1 server is implemented in Java programming language. It receives authentication requests via HTTP from multiple clients, returning back either an authentication token, or an error code. Each request from a user contains its credential in some form. All requests are atomic, so no session information is cached by the server. The SSL/TLS protocol is generally used over HTTP to protect the communication

between the two parties, and to authenticate the client's X.509 certificate, if such is provided.

The client side is organized as a library implemented both in C++ and Java, which can be used by other applications or application frameworks. This library provides a function that should be called in order to obtain the authentication token from the server. Java implementation also provides several standard GUI components, such as a login dialog, role picker dialog, and others. Basically, that authentication client-side library can be used in most application without changes. C++ version of the client library is more lightweight because it does not provide any GUI.

Types of Authentication

- By the user name and password. The user names and passwords are checked against the central NICE account database, via a dedicated web service. No user account information is stored in the CERN-RBAC own database.
- By a X.509 certificate. If the user's X.509 certificate is available, it can be applied in the standard client authentication mechanism of TLS/SSL protocol. Then, the certificate information is used to look up the user name in the CERN-RBAC database.
- By the network address (also called Authentication by location). Certain clients can be authenticated by their IP addresses, using a lookup table in the CERN-RBAC database. Normally, the address authentication is permitted only for a very limited number of machines, such as control room consoles.
- By using an existing authentication token. Any existing token can be used to request a new one, providing that the original token is not expired, bears valid signature, and was issued to the same location address. The validity time of the new token will not exceed the validity time of the original one.

5 Authorization

Middleware in the Control System

Controls Middleware (CMW) is a software infrastructure delivered and managed by CERN Beams Department/Controls group. Its goal is to provide a generic way of accessing LHC-era accelerator devices [13]. CMW provides access to devices from application programs in a distributed heterogeneous control system. It allows interconnecting applications and devices implemented in Java or C++, and running under Unix or Windows platforms.

The CMW design reflects the Accelerator Device Model in which devices, named entities in the control system, can be controlled via properties. Each property has a name and a value. The model defines several basic device access methods (get, set, monitor); by invoking these methods, applications can read, write and subscribe to the property values. CMW is based on a client-server model. Accelerator devices are implemented in device servers, and client applications access them using the CMW client API. CMW provides location transparent access to devices: servers can run anywhere in the controls network and devices can be moved from server to server without clients becoming aware of it.

Authorization

CERN-RBAC authorization library is the part of each device server. In most cases, authorization occurs once the application makes a request to get, set, or monitor a property via CMW protocol. This request is made from the application via the CMW client to the CMW server. The token obtained at authentication is passed to the CMW client. There the digital signature is verified, and if valid, the token is sent to the CMW server. If the token is not valid a meaningful exception message is returned to the

sender. The CMW looks up the permission in the access map, and depending on access rights either grants access or blocks the request.

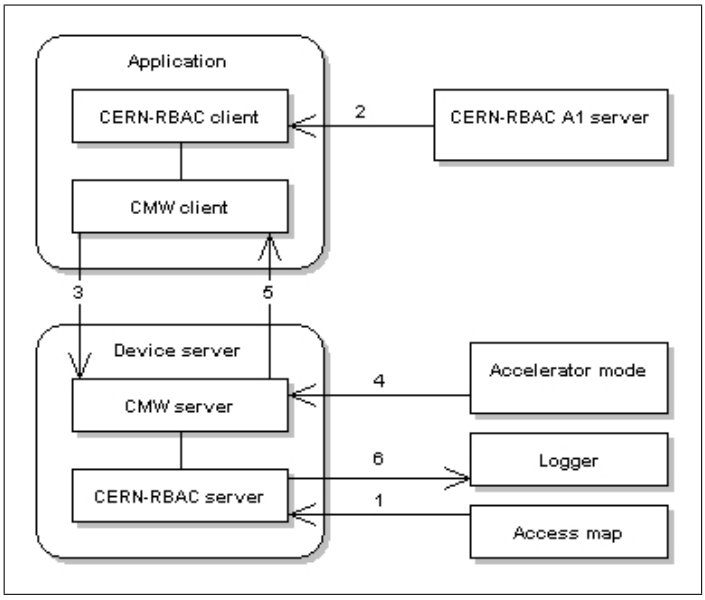


Figure 3: Authorization process

Authorization process:
1) Access map loaded from the local file on device server startup.
2) Client application authenticates on the CERN-RBAC A1 server and obtains a valid token.
3) Token is passed to the CMW client and then to the device server via CMW protocol.
4) CMW server receives the accelerator mode from timing source.
5) Dynamic authorization algorithm verifies user permissions and either allows to execute operation or thrrows an exception.
6) Result of authorization process logged for auditing.

6 Access Rules

Decision whether a particular operation is valid or not is dependent on a set of access rules. They are specified by an equipment specialist for every device class, stored and managed centrally in the Controls database. Every CMW server can read access rules (referred to as access map), relative to device classes it is providing access to, through a tab-separated text file located in the Network File System. This file mirrors the access rules located centrally in the database, and has a digitally signed by the server to prevent any modifications.

Access map is read by CMW server on its start-up. In addition, access map can be reloaded upon a distant call from CMW client. It usually happens when the set of related access rules has been altered by equipment specialist. Access rules are parameterized using all factors relative to the authorization process. More specifically, an equipment specialist has to specify the following fields to define an access rule: device class, property, device name, role for which access is permitted, application, location of the remote client, operational mode of the accelerator, and operation on the property (get, set or monitor). Apart from specific values, an equipment specialist can put a wildcard '*' in any of the fields except device class and operation. This interpreted as all values fit.

Authorization process is performed for each transaction and therefore should work as fast as possible

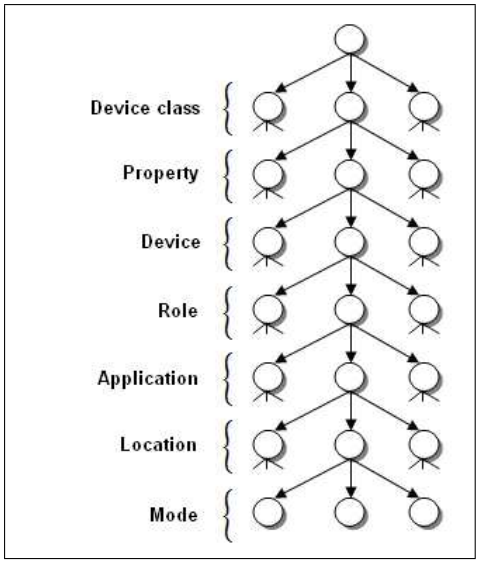


Figure 4: Access Rules structure

and should not slow down the performance of the system significantly. As an authorization time is a concern for the CMW operation, it was decided to represent the access map in the CMW server in the form of trees, a separate one for every operation type. Figure 4 presents the structure of access map tree.

Nodes placed with the same distance from the tree root group authorization argument of one type. Authorization is performed by traversing the tree, appropriate in respect to the operation type, from its root to a leaf, trying at each level to match exact authorization parameter or to find a wildcard '*'. If this succeeds, the access is granted. Such a structuring of the access map allows us to assure authorization with time complexity of O(log n) instructions, where n is the number of access rules [14].

7 Performance and Tests

An important concern with any design is its performance. If CERN-RBAC would double the time to make a setting, nobody would use it. The system also had to scale with a large number of access rules. How does a 2000-rule access map on a front-end impacts the search for the right permission? Another potential performance hit was logging each request on a protected property, and the 3-tier vs. 2-tier performance.

LHC controls can run in 2-tier mode, meaning the application and the client are on one machine and the database and devices are accessed directly on the second machine. This configuration is used when in developer testing. However when the system is in operation the configuration is usually 3-tier. Meaning the client is on a middle tier. A common client is shared by several applications and consolidates requests. In 2-tier, a dedicated client ensures that each request is made from the same application. The CERN-RBAC token can be validated once per session and the credentials can be used for each subsequent request.

In 3-tier, the requests can come from any application at random times. The simplest design is to verify the token for each request. But how does this affect performance? We ran performance tests to answer these questions on a front-end with a 400 MHz. Power PC with Linux, and here are the results:

- The size of the access map has very little effect on performance due to sophisticated and optimized search algorithms. Our test result show a 0.02 ms increase between a 20 rule and 2000 rule access map.

- Logging each request also has very little effect on the performance. Our tests show the difference between having logging on vs. having it off is 0.003 ms.
- 3-tier token verification on every request has a larger impact on performance than the other two concerns. The key size is the most contributing factor. A DSA, 1024 bit key takes 5 ms to validate. A RSA 512 bit key takes 0.150 ms.
- For a 2000 rule access map in a 2-tier configuration the average turn around time or a request is 0.7 ms. In a 3-tier configuration it is 2.7 ms. At this time, this is acceptable according to the requirements.

8 Conclusions and Future Works

The CERN-RBAC approach was successfully implemented based on the proposed model. The system successfully passed many centrally organized tests. Results of the tests prove that the proposed model and design concepts are valid. We also measured the performance of the implemented system and show that the overhead is acceptable. This allows us to assume that the proposed model of the CERN-RBAC could be used in many other areas where access control is needed for large distributed systems.

Currently the RBAC system is released in a production version and used by virtually every equipment device at CERN. Thanks to the algorithm of the dynamic authorization we propagate CERN-RBAC step-by-step, without interruption of the legacy subsystems. In the future we expect to extend functionality of the CERN-RBAC software and the area of its applicability.

Bibliography

[1] CERN: Why the LHC. http://public.web.cern.ch/public/en/LHC/WhyLHC-en.html (2008), Accessed 1 August 2009.

[2] Wikipedia: Large Hadron Collider. http://en.wikipedia.org/wiki/Large_Hadron_Collider (2008), Accessed 1 August 2009.

[3] CERN: What is LHCb. CERN FAQ LHC: the guide. CERN Communication Group. http://cdsmedia.cern.ch/img/CERN-Brochure-2008-001-Eng.pdf (2008). Accessed 9 December 2008.

[4] Wenninger J.: Operational challenges of the LHC. http://irfu.cea.fr/Phocea/file.php?class=std&file=Seminaires/1595/Dapnia-Novc07-partB.ppt. (2007), Accessed 1 August 2009

[5] Wikipedia: Role Based Access Control, http://en.wikipedia.org/wiki/Role_based_access_control (2009), Accessed 1 August 2009

[6] Petrov A., Schumann C., Gysin S.: User Authentication for Role-Based Access Control. *Proceedings of ICALEPCS 2007*

[7] Ferraiolo D.F., Kuhn D.R.: Role Based Access Control. *15th National Computer Security Conference*, Baltimore, USA, (1992)

[8] Sandhu R., Coyne E. J., Feinstein H. L., Youman C. E.: Role-Based Access Control Models. *IEEE Computer* 29 (2): 38-47, (1996)

[9] Sandhu R., Ferraiolo D.F., Kuhn D.R.: The NIST Model for Role Based Access Control: Toward a Unified Standard, *5th ACM Workshop Role-Based Access Control*, 47-63, (2000)

[10] Ferraiolo D.F., Cugini J.A., D. Kuhn D.R.: Role-Based Access Control (RBAC): Features and Motivations. *Proceedings of 11th Annual Computer Security Application Conference*, New Orleans, LA, 241-248, (1995)

[11] Gysin S., Kostro K., Kruk G., Lamont M., Lueders S., Sliwinski W., Charrue P.: Role-Based Access for the Accelerator Control System in the LHC Area - Requirements, EDMS Id 769302, (2006)

[12] Charrue P. et al.: Role Based Access Control for the Accelerator Control System in the LHC Era - Design. EDMS Id 805654, (2007)

[13] Kostro K., Baggiolini V., Calderini F., Chevrier F., Jensen S., Swoboda R., Trofimov N.: Controls Middleware - the New Generation, *EPAC*, Paris, France, p. 2028. (2002)

[14] Kostro K., Gajewski W., Gysin S.: Role-Based Authorization in Equipment Access at CERN. *Proceedings of ICALEPCS*, (2007)

Ilia Yastrebov (b. September 29, 1981) received his M.Sc. in Information Technology (2004). Currently he is a PhD student in Computer Science at Ulyanovsk State University, Russia. He has been working at European Organization for Nuclear Research since 2005 as a software developer for access control. His current research interests include different aspects of Access Control and Distributed Systems. He has 12 papers, 4 conferences participation.

Author index