IntechOpen

Automation Control

Theory and Practice

Edited by A D Rodić

AUTOMATION & CONTROL
- Theory and Practice

Edited by

A. D. Rodić

Automation and Control - Theory and Practice
http://dx.doi.org/10.5772/163
Edited by A D Rodić

First published in Croatia, 2009 by INTECH d.o.o.
eBook (PDF) Published by IN TECH d.o.o.
Place and year of publication of eBook (PDF): Rijeka, 2019.
IntechOpen is the global imprint of IN TECH d.o.o.
Printed in Croatia

Legal deposit, Croatia: National and University Library in Zagreb

Additional hard and PDF copies can be obtained from orders@intechopen.com

Automation and Control - Theory and Practice
Edited by A D Rodić
p. cm.
ISBN 978-953-307-039-1
eBook (PDF) ISBN 978-953-51-5858-5

Preface

Automation is the use of control systems (such as numerical control, programmable logic control, and other industrial control systems), in concert with other applications of information technology (such as computer-aided technologies [CAD, CAM, CAx]), to control industrial machinery and processes, reducing the need for human intervention. In the scope of industrialization, automation is a step beyond mechanization. Whereas mechanization provided human operators with machinery to assist them with the muscular requirements of work, automation greatly reduces the need for human sensory and mental requirements as well. Processes and systems can also be automated.

Automation plays an increasingly important role in the global economy and in daily experience. Engineers strive to combine automated devices with mathematical and organizational tools to create complex systems for a rapidly expanding range of applications and human activities.

Many roles for humans in industrial processes presently lie beyond the scope of automation. Human-level pattern recognition, language recognition, and language production ability are well beyond the capabilities of modern mechanical and computer systems. Tasks requiring subjective assessment or synthesis of complex sensory data, such as scents and sounds, as well as high-level tasks such as strategic planning, currently require human expertise. In many cases, the use of humans is more cost-effective than mechanical approaches even where automation of industrial tasks is possible.

Specialized industrial computers, referred to as programmable logic controllers (PLCs), are frequently used to synchronize the flow of inputs from (physical) sensors and events with the flow of outputs to actuators and events. This leads to precisely controlled actions that permit a tight control of almost any industrial process.

Human-machine interfaces (HMI) or computer human interfaces (CHI), formerly known as man-machine interfaces, are usually employed to communicate with PLCs and other computers, such as entering and monitoring temperatures or pressures for further automated control or emergency response. Service personnel who monitor and control these interfaces are often referred to as stationary engineers.

Different types of automation tools exist:

- ANN - Artificial neural network
- DCS - Distributed Control System
- HMI - Human Machine Interface
- SCADA - Supervisory Control and Data Acquisition

- PLC - Programmable Logic Controller
- PAC - Programmable Automation Controller
- Instrumentation
- Motion control
- Robotics

Control theory is an interdisciplinary branch of engineering and mathematics that deals with the behavior of dynamical systems. Control theory is

- a theory that deals with influencing the behavior of dynamical systems
- an interdisciplinary subfield of science, which originated in engineering and mathematics, and evolved into use by the social, economic and other sciences.

Main control techniques assume:

- Adaptive control uses on-line identification of the process parameters, or modification of controller gains, thereby obtaining strong robustness properties.
- A Hierarchical control system is a type of Control System in which a set of devices and governing software is arranged in a hierarchical tree. When the links in the tree are implemented by a computer network, then that hierarchical control system is also a form of a Networked control system.
- Intelligent control use various AI computing approaches like neural networks, Bayesian probability, fuzzy logic, machine learning, evolutionary computation and genetic algorithms to control a dynamic system.
- Optimal control is a particular control technique in which the control signal optimizes a certain "cost index". Two optimal control design methods have been widely used in industrial applications, as it has been shown they can guarantee closed-loop stability. These are Model Predictive Control (MPC) and Linear-Quadratic-Gaussian control (LQG).
- Robust control deals explicitly with uncertainty in its approach to controller design. Controllers designed using robust control methods tend to be able to cope with small differences between the true system and the nominal model used for design.
- Stochastic control deals with control design with uncertainty in the model. In typical stochastic control problems, it is assumed that there exist random noise and disturbances in the model and the controller, and the control design must take into account these random deviations.

The present edited book is a collection of 18 chapters written by internationally recognized experts and well-known professionals of the field. Chapters contribute to diverse facets of automation and control. The volume is organized in four parts according to the main subjects, regarding the recent advances in this field of engineering.

The first thematic part of the book is devoted to automation. This includes solving of assembly line balancing problem and design of software architecture for cognitive assembling in production systems.

The second part of the book concerns with different aspects of modeling and control. This includes a study on modeling pollutant emission of diesel engine, development of a PLC program obtained from DEVS model, control networks for digital home, automatic control of temperature and flow in heat exchanger, and non-linear analysis and design of phase locked loops.

The third part addresses issues of parameter estimation and filter design, including methods for parameters estimation, control and design of the wave digital filters.

The fourth part presents new results in the intelligent control. That includes building of a neural PDF strategy for hydroelectric station simulator, intelligent network system for process control, neural generalized predictive control for industrial processes, intelligent system for forecasting, diagnosis and decision making based on neural networks and self-organizing maps, development of a smart semantic middleware for the Internet , development appropriate AI methods in fault-tolerant control, building expert system in rotary railcar dumpers, expert system for plant asset management, and building of a image retrieval system in heterogeneous database.

The content of this thematic book admirably reflects the complementary aspects of theory and practice which have taken place in the last years. Certainly, the content of this book will serve as a valuable overview of theoretical and practical methods in control and automation to those who deal with engineering and research in this field of activities.

The editors are greatfull to the authors for their excellent work and interesting contributions. Thanks are also due to the renomeus publisher for their editorial assistance and excellent technical arrangement of the book.

December, 2009

A. D. Rodić

Contents

IV. Intelligent Control

1

Assembly Line Balancing Problem Single and Two-Sided Structures

Waldemar Grzechca
The Silesian University of Technology
Poland

1. Introduction

The manufacturing assembly line was first introduced by Henry Ford in the early 1900's. It was designed to be an efficient, highly productive way of manufacturing a particular product. The basic assembly line consists of a set of workstations arranged in a linear fashion, with each station connected by a material handling device. The basic movement of material through an assembly line begins with a part being fed into the first station at a predetermined feed rate. A station is considered any point on the assembly line in which a task is performed on the part. These tasks can be performed by machinery, robots, and/or human operators. Once the part enters a station, a task is then performed on the part, and the part is fed to the next operation. The time it takes to complete a task at each operation is known as the process time (Sury, 1971). The cycle time of an assembly line is predetermined by a desired production rate. This production rate is set so that the desired amount of end product is produced within a certain time period (Baybars, 1986). In order for the assembly line to maintain a certain production rate, the sum of the processing times at each station must not exceed the station's cycle time (Fonseca et al, 2005). If the sum of the processing times within a station is less than the cycle time, idle time is said to be present at that station (Erel et al,1998). One of the main issues concerning the development of an assembly line is how to arrange the tasks to be performed. This arrangement may be somewhat subjective, but has to be dictated by implied rules set forth by the production sequence (Kao, 1976). For the manufacturing of any item, there are some sequences of tasks that must be followed. The assembly line balancing problem (ALBP) originated with the invention of the assembly line. Helgeson et al (Helgeson et al, 1961) were the first to propose the ALBP, and Salveson (Salveson, 1955) was the first to publish the problem in its mathematical form. However, during the first forty years of the assembly line's existence, only trial-and-error methods were used to balance the lines (Erel et al,, 1998). Since then, there have been numerous methods developed to solve the different forms of the ALBP. Salveson (Salveson, 1955) provided the first mathematical attempt by solving the problem as a linear program. Gutjahr and Nemhauser (Gutjahr & Nemhauser, 1964) showed that the ALBP problem falls into the class of NP-hard combinatorial optimization problems. This means that an optimal solution is not guaranteed for problems of significant size. Therefore, heuristic methods have become the most popular techniques for solving the problem. Author of this book chapter underlines the importance of the final results estimation and proposes for single and two-sided assembly line balancing problem modified measures.

2. Two-sided Assembly Lines

Two-sided assembly lines (Fig. 1.) are typically found in producing large-sized products, such as trucks and buses. Assembling these products is in some respects different from assembling small products. Some assembly operations prefer to be performed at one of the two sides (Bartholdi, 1993).

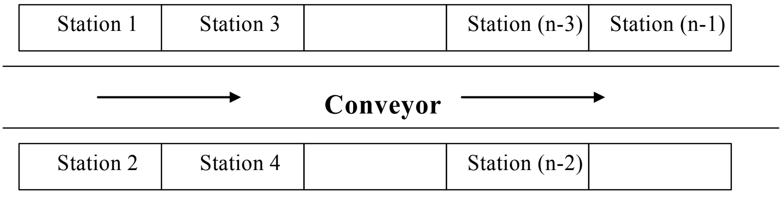


Fig. 1. Two-sided assembly line structure

Let us consider, for example, a truck assembly line. Installing a gas tank, air filter, and toolbox can be more easily achieved at the left-hand side of the line, whereas mounting a battery, air tank, and muffler prefers the right-hand side. Assembling an axle, propeller shaft, and radiator does not have any preference in their operation directions so that they can be done at any side of the line. The consideration of the preferred operation directions is important since it can greatly influence the productivity of the line, in particular when assigning tasks, laying out facilities, and placing tools and fixtures in a two-sided assembly line (Kim et al, 2001). A two-sided assembly line in practice can provide several substantial advantages over a one-sided assembly line (Bartholdi, 1993). These include the following: (1) it can shorten the line length, which means that fewer workers are required, (2) it thus can reduce the amount of throughput time, (3) it can also benefit from lowered cost of tools and fixtures since they can be shared by both sides of a mated-station, and (4) it can reduce material handling, workers movement and set-up time, which otherwise may not be easily eliminated. These advantages give a good reason for utilizing two-sided lines for assembling large-sized products. A line balancing problem is usually represented by a precedence diagram as illustrated in Fig. 2.

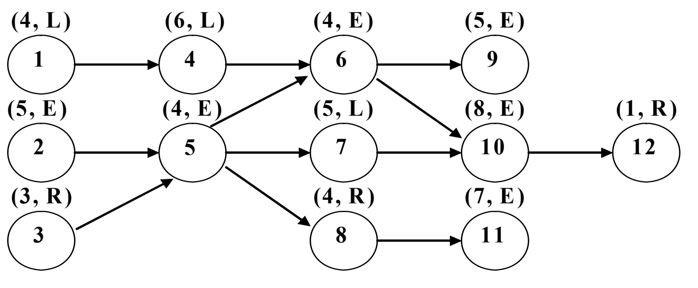


Fig. 2. Precedence graph

A circle indicates a task, and an arc linking two tasks represents the precedence relation between the tasks. Each task is associated with a label of (t_i, d), where t_i is the task processing time and d (=L, R or E) is the preferred operation direction. L and R, respectively, indicate that the task should be assigned to a left- and a right-side station. A task associated with E can be performed at either side of the line. While balancing assembly lines, it is generally needed to take account of the features specific to the lines. In a one-sided assembly line, if precedence relations are considered appropriately, all the tasks assigned to a station can be carried out continuously without any interruption. However, in a two-sided assembly line, some tasks assigned to a station can be delayed by the tasks assigned to its companion (Bartholdi, 1993). In other words, idle time is sometimes unavoidable even between tasks assigned to the same station. Consider, for example, task j and its immediate predecessor i. Suppose that j is assigned to a station and i to its companion station. Task j cannot be started until task i is completed. Therefore, balancing such a two-sided assembly line, unlike a one-sided assembly line, needs to consider the sequence-dependent finish time of tasks.

3. Heuristic Methods in Assembly Line Balancing Problem

The heuristic approach bases on logic and common sense rather than on mathematical proof. Heuristics do not guarantee an optimal solution, but results in good feasible solutions which approach the true optimum.

3.1 Single Assembly Line Balancing Heuristic Methods

Most of the described heuristic solutions in literature are the ones designed for solving single assembly line balancing problem. Moreover, most of them are based on simple priority rules (constructive methods) and generate one or a few feasible solutions. Task-oriented procedures choose the highest priority task from the list of available tasks and assign it to the earliest station which is assignable. Among task-oriented procedures we can distinguish immediate-update-first-fit (IUFF) and general-first-fit methods depending on whether the set of available tasks is updated immediately after assigning a task or after the assigning of all currently available tasks. Due to its greater flexibility immediate-update-first-fit method is used more frequently. The main idea behind this heuristic is assigning tasks to stations basing on the numerical score. There are several ways to determine (calculate) the score for each tasks. One could easily create his own way of determining the score, but it is not obvious if it yields good result. In the following section five different methods found in the literature are presented along with the solution they give for our simple example. The methods are implemented in the Line Balancing program as well. From the moment the appropriate score for each task is determined there is no difference in execution of methods and the required steps to obtain the solution are as follows:

STEP 1. Assign a numerical score $n(x)$ to each task x.

STEP 2. Update the set of available tasks (those whose immediate predecessors have been already assigned).

STEP 3. Among the available tasks, assign the task with the highest numerical score to the first station in which the capacity and precedence constraints will not be violated. Go to STEP 2.

The most popular heuristics which belongs to IUFF group are:

IUFF-RPW Immediate Update First Fit – Ranked Positional Weight,

IUFF-NOF Immediate Update First Fit – Number of Followers,
IUFF-NOIF Immediate Update First Fit – Number of Immediate Followers,
IUFF-NOP Immediate Update First Fit – Number of Predecessors,
IUFF-WET Immediate Update First Fit – Work Element Time.

3.2 Two-sided Assembly Line Balancing Heuristic Method

A task group consists of a considered task i and all of its predecessors. Such groups are generated for every un–assigned task. As mentioned earlier, balancing a two–sided assembly line needs to additionally consider operation directions and sequence dependency of tasks, while creating new groups (Lee et al, 2001). While forming initial groups IG(i), the operation direction is being checked all the time. It's disallowed for a group to contain tasks with preferred operation direction from opposite sides. But, if each task in initial group is E – task, the group can be allocated to any side. In order to determine the operation directions for such groups, the rules (direction rules DR) are applied:
DR 1. Set the operation direction to the side where tasks can be started earlier.
DR 2. The start time at both sides is the same, set the operation direction to the side where it's expected to carry out a less amount of tasks (total operation time of unassigned L or R tasks).
Generally, tasks resulting from "repeatability test" are treated as starting ones. But there is exception in form of first iteration, where procedure starts from searching tasks (initial tasks IT), which are the first ones in precedence relation. After the first step in the first iteration we get:

IG (1) = {1}, Time{IG (1)} = 2, Side{IG (1)} = 'L'
IG (2) = {2}, Time{IG (2)} = 5, Side{IG (2)} = 'E'
IG (3) = {3}, Time{IG (3)} = 3, Side{IG (3)} = 'R

where:
Time{IG(i)} – total processing time of i^{th} initial group,
Side{IG(i)} – preference side of i^{th} initial group.
To those who are considered to be the first, the next tasks will be added, (these ones which fulfil precedence constraints).
Whenever new tasks are inserted to the group i, the direction, cycle time and number of immediate predecessors are checked. If there are more predecessors than one, the creation of initial group j comes to the end.
First iteration – second step

IG (1) = {1, 4, 6}, Time{IG (1)} = 8, Side{IG (1)} = 'L'
IG (2) = {2, 5}, Time{IG (2)} = 9 , Side{IG (2)} = 'E'
IG (3) = {3, 5} , Time{IG (3)} = 7 , Side{IG (3)} = 'R'

When set of initial groups is created, the last elements from those groups are tested for repeatability. If last element in set of initial groups IG will occur more than once (groups pointed by arrows), the groups are intended to be joined – if total processing time (summary time of considered groups) is less or equal to cycle time. Otherwise, these elements are deleted.
In case of occurring only once, the last member is being checked if its predecessors are not contained in Final set FS. If not, it's removed as well. So far, FS is empty.
First iteration – third step

IG (1) = {1, 4}, Time{IG (1)} = 4, Side{IG (1)} = 'L'

IG (2) = {2, 3, 5}, Time{IG (2)} = 12, Side{IG (2)} = 'R'

Whenever two or more initial groups are joined together, or when initial group is connected with those one coming from Final set – the "double task" is added to initial tasks needed for the next iteration. In the end of each iteration, created initial groups are copied to FS.

First iteration – fourth step

FS = { (1, 4); (2, 3, 5) },
Side{FS (1)} = 'L', Side{FS (2)} = 'R'
Time {FS(2)} = 12, Time {FS(1)} = 14,
IT = {5}.

In the second iteration, second step, we may notice that predecessor of last task coming from IG(1) is included in Final Set, FS(2). The situation results in connecting both groups under holding additional conditions:

Side{IG(1)} = Side{FS(2)},
Time + time < cycle.

After all, there is no more IT tasks, hence, preliminary process of creating final set is terminated.

The presented method for finding task groups is to be summarized in simplified algorithm form. Let U denote to be the set of un – assigned tasks yet and IG_i be a task group consisting of task i and all its predecessors (excluded from U set).

STEP 1. If U = empty, go to step 5, otherwise, assign starting task from U.

STEP 2. Identify IG_i . Check if it contains tasks with both left and right preference operation direction, then remove task i.

STEP 3. Assign operation direction Side{ IG_i } of group IG_i . If IG_i has R-task (L-task), set the operation direction to right (left). Otherwise, apply so called direction rules DR.

STEP 4. If the last task i in IG_i is completed within cycle time, the IG_i is added to Final set of candidates FS(i). Otherwise, exclude task i from IG_i and go to step 1.

STEP 5. For every task group in FS(i), remove it from FS if it is contained within another task group of FS.

The resulting task groups become candidates for the mated-station

FS = {(1,4), (2,3,5,8)}.

The candidates are produced by procedures presented in the previous section, which claim to not violate precedence, operation direction restrictions, and what's more it exerts on groups to be completed within preliminary determined cycle time. Though, all of candidates may be assigned equally, the only one group may be chosen. Which group it will be – for this purpose the rules helpful in making decision, will be defined and explained below:

AR 1. Choose the task group FS(i) that may start at the earliest time.

AR 2. Choose the task group FS(i) that involves the minimum delay.

AR 3. Choose the task group FS(i) that has the maximum processing time.

In theory, for better understanding, we will consider a left and right side of mated – station, with some tasks already allocated to both sides. In order to achieve well balanced station, the AR 1 is applied, cause the unbalanced station is stated as the one which would probably involve more delay in future assignment. This is the reason, why minimization number of stations is not the only goal, there are also indirect ones, such as reduction of unavoidable delay. This rule gives higher priority to the station, where less tasks are allocated. If ties occurs, the AR 2 is executed, which chooses the group with the least amount of delay among the considered ones. This rule may also result in tie. The last one, points at relating work

within individual station group by choosing group of task with highest processing time. For the third rule the tie situation is impossible to obtain, because of random selection of tasks. The implementation of above rules is strict and easy except the second one. Shortly speaking, second rule is based on the test, which checks each task consecutively, coming from candidates group FS(i) – in order to see if one of its predecessors have already been allocated to station. If it has, the difference between starting time of considered task and finished time of its predecessor allocated to companion station is calculated. The result should be positive, otherwise time delay occurs.
Having rules for initial grouping and assigning tasks described in previous sections, we may proceed to formulate formal procedure of solving two – sided assembly line balancing problem (Kim et. al, 2005).
Let us denote companion stations as *j* and *j'*,
D(i) – the amount of delay,
Time(i) – total processing time (Time{FS(i)}),
S(j) – start time at station j,
STEP 1. Set up j = 1, j' = j + 1, S(j) = S(j') = 0, U – the set of tasks to be assigned.
STEP 2. Start procedure of group creating (3.2), which identifies
FS = {FS(1), FS(2), …, FS(n)}. If FS = ∅, go to step 6.
STEP 3. For every FS(i), i = 1,2, … , n – compute D(i) and Time(i).
STEP 4. Identify one task group FS(i), using AR rules in Section 3.3
STEP 5. Assign FS(i) to a station j (j') according to its operation direction, and update S(j) = S(j) + Time(i) + D(i). U = U – {FS(i)}, and go to STEP 2.
STEP 6. If U ≠ ∅, set j = j' + 1, j' = j + 1, S(j) = S(j') = 0, and go to STEP 2, otherwise, stop the procedure.

4. Measures of Final Results of Assembly Line Balancing Problem

Some measures of solution quality have appeared in line balancing problem. Below are presented three of them (Scholl, 1999).
Line efficiency (LE) shows the percentage utilization of the line. It is expressed as ratio of total station time to the cycle time multiplied by the number of workstations:

$$LE = \frac{\sum_{i=1}^{K} ST_i}{c \cdot K} \cdot 100\% \tag{1}$$

where: K - total number of workstations,
c - cycle time.
Smoothness index (SI) describes relative smoothness for a given assembly line balance. Perfect balance is indicated by smoothness index 0. This index is calculated in the following manner:

$$SI = \sqrt{\sum_{i=1}^{K} \left(ST_{max} - ST_i\right)^2} \tag{2}$$

where:
ST_{max} - maximum station time (in most cases cycle time),

ST_i - station time of station i.
Time of the line (LT) describes the period of time which is need for the product to be completed on an assembly line:

$$LT = c \cdot (K - 1) + T_K \quad (3)$$

where:

c - cycle time,
K -total number of workstations,
T_K – processing time of last station.

The final result estimation of two-sided assembly line balance needs some modification of existing measures (Grzechca, 2008).

Time of line for TALBP

$$LT = c \cdot (Km - 1) + Max\{t(S_K), t(S_{K-1})\} \quad (4)$$

where:
Km – number of mated-stations
K – number of assigned single stations
$t(S_K)$ – processing time of the last single station

As far as smoothness index and line efficiency are concerned, its estimation, on contrary to LT, is performed without any change to original version. These criterions simply refer to each individual station, despite of parallel character of the method.
But for more detailed information about the balance of right or left side of the assembly line additional measures will be proposed:
Smoothness index of the left side

$$SI_L = \sqrt{\sum_{i=1}^{K} (ST_{maxL} - ST_{iL})^2} \quad (5)$$

where:
SI_L- smoothness index of the left side of two-sided line
ST_{maxL}- maximum of duration time of left allocated stations
ST_{iL}- duration time of i-th left allocated station

Smoothness index of the right side

$$SI_R = \sqrt{\sum_{i=1}^{K} (ST_{maxR} - ST_{iR})^2} \quad (6)$$

where:
SI_R- smoothness index of the right side of two-sided line,
ST_{maxR}- maximum of duration time of right allocated stations,
ST_{iR}- duration time of i-th right allocated station.

5. Numerical examples

An numerical example from Fig. 3. will be considered. The number of tasks, precedence graph and processing times are known and there are given in Table 1.The cycle time is 10.

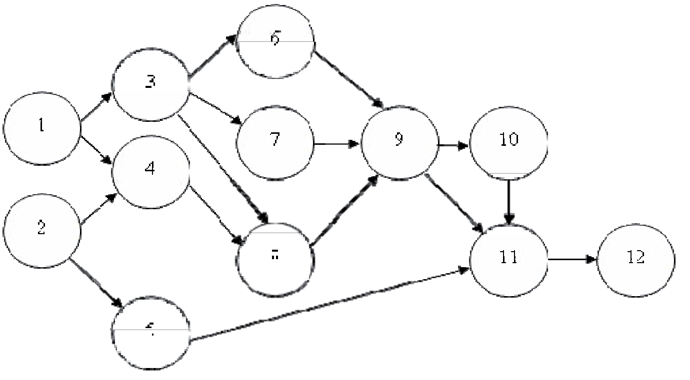


Fig. 3. Precedence graph for single line

Number of task	Processing Time	Weight	Positional Rank
1	1	29	2
2	5	27	1
3	4	28	3
4	3	22	4
5	2	11	5
6	6	24	7
7	3	21	6
8	1	19	8
9	7	18	11
10	6	11	9
11	3	5	10
12	2	2	12

Table 1. Input data of numerical example – IUFF Ranked Positional Weight

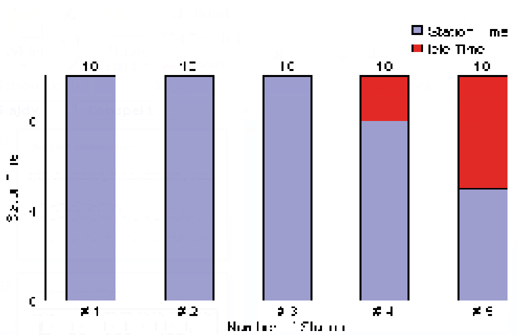


Fig. 4. Assembly line balance for IUFF-RPW and IUFF-NOF methods

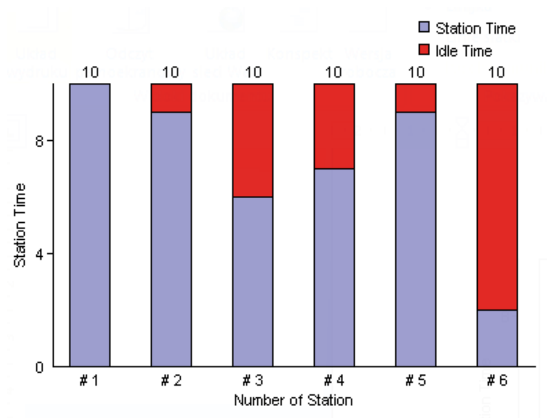


Fig. 5. Assembly line balance for IUFF-NOP and IUFF-NOIF methods

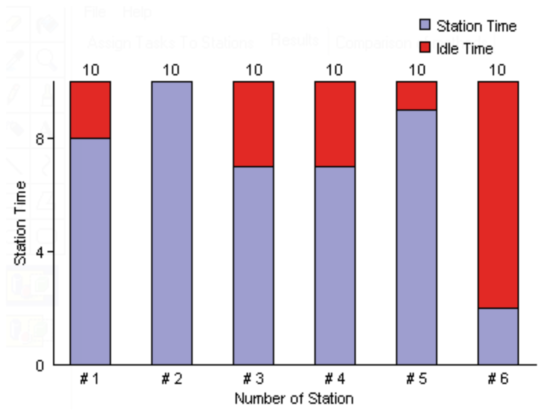


Fig. 6. Assembly line balance for IUFF-WET method

Method	**K**	**Balance**	**LE**	**SI**	**LT**
IUFF-RPW	5	S1 – 1, 3, 2 S2 – 6, 4, 8 S3 – 7, 9 S4 – 10, 5 S5 – 11, 12	86%	5,39	45
IUFF-NOF	5	S1 – 1, 3, 2 S2 – 6, 4, 8 S3 – 7, 9 S4 – 10, 5 S5 – 11, 12	86%	5,39	45
IUFF-NOIF	6	S1 – 1, 2, 3 S2 – 5, 4, 7, 8 S3 – 6 S4 – 9 S5 – 10, 11 S6 – 12	71,67%	9,53	52

IUFF-NOP	6	S1 – 1, 2, 3 S2 – 5, 4, 7, 8 S3 – 6 S4 – 9 S5 – 10, 11 S6 – 12	71,67%	9,53	52
IUFF-WET	6	S1 – 2, 5, 1 S2 – 3, 6 S3 – 4, 7, 8 S4 – 9 S5 – 10, 11 S6 – 12	71,67%	9,33	52

Table 2. Results of balance for IUFF methods

Number of task	Processing Time	Position (Constraints)
1	4	L
2	5	E
3	3	R
4	6	L
5	4	E
6	4	E
7	5	L
8	4	R
9	5	E
10	8	E
11	7	E
12	1	R

Table 3. Input data of numeraical example – two-sided line from Fig. 2.

The results of heuristic procedure for the example from Fig. 2 and cycle time c=16 are given in a Gantt chart – Fig. 7.

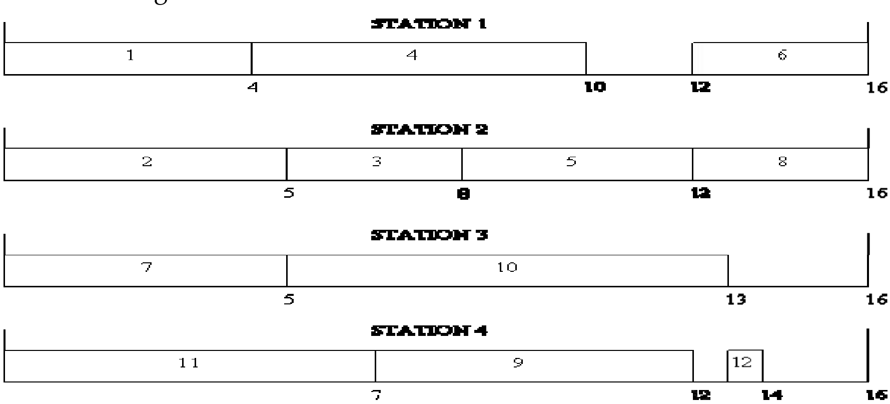


Fig. 7. Gantt chart of assembly line balance of two-sided structure (Fig. 2.)

Before presenting performance measures for current example, it would be like to stress difference in estimation of line time form, resulting from restrictions of parallel stations. In two – sided line method within one mated-station, tasks are intended to perform its operations at the same time, as it is shown in example in Fig. 7., where tasks 7, 11 respectively are processed simultaneously on single station 3 and 4, in contrary to one – sided heuristic methods. Hence, modification has to be introduced to that particular parameter which is the consequence of parallelism. Having two mated-stations from Fig. 7, the line time LT is not 3*16 + 13, as it was in original expression. We must treat those stations as two double ones (mated-stations), rather than individual ones S_k (4). As far as smoothness index and line efficiency are concerned, its estimation, on contrary to LT, is performed without any change to original version. These criterions simply refer to each individual station, despite of parallel character of the method. But for more detailed information about the balance of right or left side of the assembly line additional measures (5) and (6) was proposed (Grzechca, 2008).

Name	Value
LE	84,38%
LT	30
SI	4,69
SI_R	2
SI_L	3

Table 4. Numerical results of balance of two-sided assembly line structure

6. Conclusion

Single and two-sided assembly lines become more popular in last time. Therefore it is obvious to consider these structures using different methods. In this chapter a heuristic approach was discussed. Single assembly line balancing problem has very often difficulties with the last station. Even optimal solution (100 % efficiency of workstations except the last one is impossible to accept by production engineers in the companies. Different heuristic methods allow to obtain different feasible solutions and then to choose the most appropriate result. Two-sided assembly line structure is very sensitive to changes of cycle time values. It is possible very often to get incomplete structure of the two-sided assembly line (some stations are missing) in final result. We can use different measures for comparing the solutions (line time, line efficiency, smoothness index). Author proposes additionally two measures: smoothness index of the left side (SI_L) and smoothness index of the right side (SI_R) of the two-sided assembly line structure. These measurements allow to get more knowledge about allocation of the tasks and about the balance on both sides.

7. References

Bartholdi, J.J. (1993). Balancing two-sided assembly lines: A case study, *International Journal of Production Research,* Vol. 31, No,10, pp. 2447-2461

Baybars, I. (1986). A survey of exact algorithms for simple assembly line balancing problem, *Management Science,* Vol. 32, No. 8, pp. 909-932

Erel, E., Sarin S.C. (1998). A survey of the assembly line balancing procedures, *Production Planning and Control,* Vol. 9, No. 5, pp. 414-434

Fonseca D.J., Guest C.L., Elam M., Karr C.L. (2005). A fuzzy logic approach to assembly line balancing, *Mathware & Soft Computing,* Vol. 12, pp. 57-74

Grzechca W. (2008) Two-sided assembly line. Estimation of final results. *Proceedings of the Fifth International Conference on Informatics in Control, Automation and Robotics ICINCO 2008, Final book of Abstracts and Proceedings,* Funchal, 11-15 May 2008, pp. 87-88, CD Version ISBN: 978-989-8111-35-7

Gutjahr, A.L., Neumhauser G.L. (1964). An algorithm for the balancing problem, *Management Science*, Vol. 11,No. 2, pp. 308-315

Helgeson W. B., Birnie D. P. (1961). Assembly line balancing using the ranked positional weighting technique, *Journal of Industrial Engineering,* Vol. 12, pp. 394-398

Kao, E.P.C. (1976). A preference order dynamic program for stochastic assembly line balancing, *Management Science*, Vol. 22, No. 10, pp. 1097-1104

Lee, T.O., Kim Y., Kim Y.K. (2001). Two-sided assembly line balancing to maximize work relatedness and slackness, *Computers & Industrial Engineering,* Vol. 40, No. 3, pp. 273-292

Salveson, M.E. (1955). The assembly line balancing problem, *Journal of Industrial Engineering,* Vol.6, No. 3. pp. 18-25

Scholl, A. (1999). *Balancing and sequencing of assembly line*, Physica- Verlag, ISBN 9783790811803, Heidelberg New-York

Sury, R.J. (1971). Aspects of assembly line balancing, *International Journal of Production Research*, Vol. 9, pp. 8-14

2

A Software Architecture for Cognitive Technical Systems Suitable for an Assembly Task in a Production Environment

Eckart Hauck, Arno Gramatke and Klaus Henning
ZLW/IMA, RWTH Aachen University
Germany

1. Introduction

In the last years production in low-wage countries became popular with many companies by reason of low production costs. To slow down the development of shifting production to low-wage countries, new concepts for the production in high-wage countries have to be developed.
Currently the highly automated industry is very efficient in the production of a small range of products with a large batch size. The big automotive manufacturer like GM, Daimler and BMW are examples for this. Changes in the production lines lead to high monetary investments in equipment and staff. To reach the projected throughput the production is planned completely in advance. On the other side a branch of the industry is specialized in manufacturing customized products with small batch sizes. Examples are super sports car manufacturers. Due to the customization the product is modified on a constant basis. This range of different manufacturing approaches can be aggregated in two dilemmas in which companies in high wage countries have to allocate themselves.
The first dilemma is between value orientation and planning orientation. Value orientation implies that the manufacturing process involves almost no planning before the production phase of a product. The manufacturing process is adapted and optimized during the production. The antipode to value orientation is planning orientation. Here the whole process is planned and optimized prior to manufacturing.
The second dilemma is between scale and scope. Scale means a typical mass production where scaling effects make production more efficient. The opposite is scope where small batch sizes dominate manufacturing.
The mainstream automotive industry represents scale focused production with a highly planning oriented approach due to the grade of automation involved. The super sports car manufacturer is following a scope approach which is more value oriented.
These two dilemmas span the so called polylemma of production technology (Fig. 1) (Brecher et al, 2007). The reduction of these dilemmas is the main aim of the cluster of excellence "Integrative Production Technology for High-Wage Countries" of the RWTH Aachen University. The research vision is the production of a great variety of products in

small batch sizes with costs competitive to mass production under the full exploitation of the respective benefits of value orientation and planning orientation.
To reach the vision four core research areas were identified. These areas are “Individualized Production Systems”, “Virtual Production Systems”, “Hybrid Production Systems” and “Self-optimizing Production Systems”. Self-optimizing production systems try to realize value orientated approaches with an increase in the planning efficiency by reusing gained knowledge on new production conditions.
The research hypothesis is that only technical systems which incorporate cognitive capabilities are capable of showing self-optimizing behavior (Heide 2006). In addition to that these Cognitive Technical Systems can reduce the planning efforts required to adapt to changes in the process chain (Brecher et al. 2007). In this chapter a software architecture for such a Cognitive Technical System will be described and a use case in the context of assembly processes will be presented. Section 2 will deal with the definition of the terms “Self optimization”, “Cognition” and “Cognitive Technical System”. Section 3 deals with related work in the context of Cognitive Technical Systems and the involved software architectures. The fourth section describes an excerpt of the functional as well as the non-functional requirements for a Cognitive Technical System. Afterwards the software architecture will be presented. The sixth section will introduce an assembly use case and the chapter closes with a final conclusion in section 7.

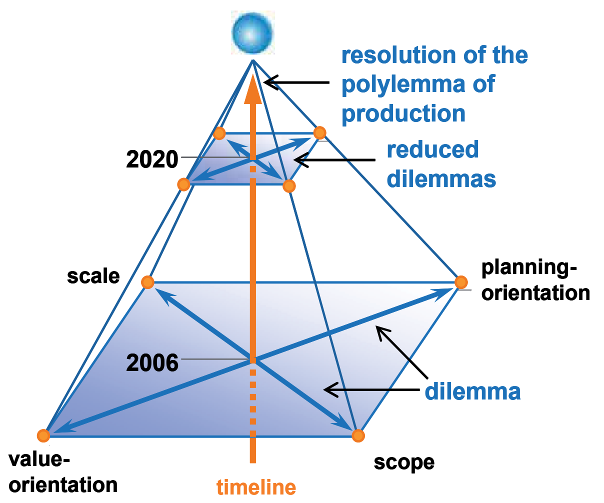


Fig. 1. Polylemma of Production Technology

2. Definition of Terms

2.1 Self-Optimization

Self-optimization in the context of artificial systems includes three joint actions. At first the current situation has to be analyzed and in a second step the objectives have to be determined. These objectives can be contradictive. In this case a tradeoff between the objectives has to be done by the system. The third step is the adaption of the system behavior. A system can be accounted for a self-optimizing system if it is capable to analyze and detect relevant modifications of the environment or the system itself, to endogenously modify its objectives in response to changing influence on the technical system from its surroundings, the user, or the system itself, and to autonomously adapt its behavior by means of parameter changes or structure changes to achieve its objectives (Gausemeier 2008). To adapt itself, the system has to incorporate cognitive abilities to be able to analyze the current situation and adjust system behavior accordingly.

2.2 Cognition

Currently, the term "Cognition" is most often thought of in a human centered way, and is not well defined in psychology, philosophy and cognitive science. In psychology Matlin (2005) defines cognition in the context of humans as the "acquisition, storage, usage and transformation of knowledge". This involves many different processes. Zimbardo (2005) accounts, among others, "perception, reasoning, remembering, thinking decision-making and learning" as essential processes involved in cognition. These definitions cannot be easily transferred to artificial systems. A first approach to the definition of "Cognition" in the context of artificial systems is given in Strasser (2004). She based her definition on Strube (1998):

"Cognition as a relatively recent development of evolution, provides adaptive (and hence, indirect) coupling between the sensory and motor sides of an organism. This adaptive, indirect coupling provides for the ability to learn (as demonstrated by conditioning, a procedure that in its simplest variant works even in flatworms), and in higher organisms, the ability to deliberate".

Strasser develops from this the definition that a system, either technical or biological, has to incorporate this flexible connectivity between input and output which implies the ability to learn. Thereby the term learning is to be understood as the rudimentary ability to adapt the output to the input to optimize the expected utility. For her, these are the essential requirements a system has to incorporate to account for being cognitive.

This definition is to be considered when the term "Cognition" is used. It should be stated that this definition is the lower bound for the usage of the term. Therefore this does not imply that a system which can be accounted as "Cognitive" is able the incorporate human level cognitive processes.

2.3 Cognitive Technical Systems

With the given definition for "Self-optimization" and "Cognition" the term "Cognitive Technical System" can be defined. Also a description of the intermediate steps involved in development towards a Cognitive Technical System able of incorporating cognitive processes of a higher level can be given.

The definition used in the context of this chapter is that an artificial system, which incorporates cognitive abilities and is able to adapt itself to different environmental changes, can be accounted as Cognitive Technical System.

Fig. 2 shows the different steps towards a Cognitive Technical System capable of cognition on a higher level. As cognitive processes of a higher level the communication in natural language and adaption to the mental model of the operator can be named. Also more sophisticated planning abilities in unstructured, partly observable and nondeterministic environments can be accounted as cognitive processes on a higher level.

The current situation of human machine interaction in the context of a production environment is as follows (left part of the picture): Cognitive processes occur only in humans. The technical system, which consists of an Interaction System and a Technological Application System, is not cognitive in the sense of Strasser (2004). The interaction happens in a classical way via a human-machine-interface embedded in the Interaction System. The output of the Technological Application System is evaluated and optimized by the human operator only. This also means that only the human operator can reflect about the output of the Technological Application System and improve it by adapting the parameters of the processes via the human-machine-interface.

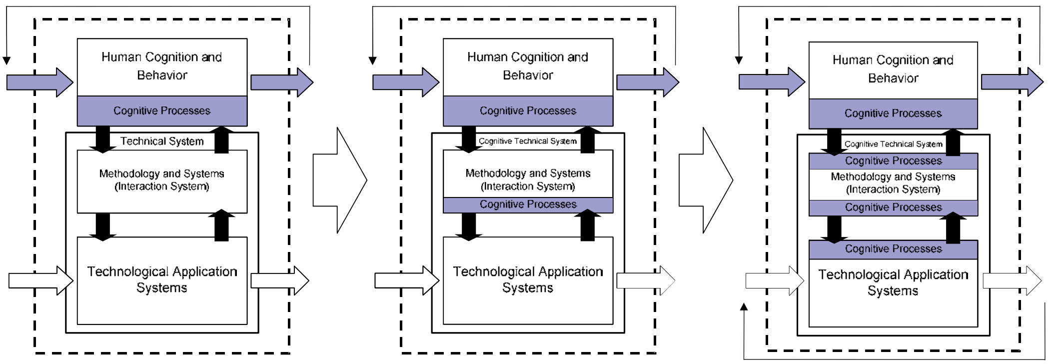


Fig. 2. Steps in the development of Cognitive Technical Systems

The intermediate step (middle of the picture) is the incorporation of basic cognitive processes in the Interaction System. The technical system could now be accounted for a Cognitive Technical System. The cognitive processes can involve reasoning and decision making. The Cognitive Technical System has to incorporate a knowledge base upon which decisions can be derived. These cognitive processes are embedded in the Interaction System which communicates with the Technological Application System but also controls it.

The right part of the picture shows a visionary Cognitive Technical System. Such a system incorporates cognitive processes on all levels, which means that the human-machine-interaction is based on multimodal communication. In addition to that the human-machine-interface adapts itself to the mental model of the human operator during the communication process. This increases the efficiency of the communication process dramatically. Therefore the human-machine-interface incorporates cognitive abilities. In addition the Interaction System incorporates cognitive processes in the communication with the Technological Application System which also embeds cognitive capabilities. This means that the communication can be alleviated to a higher level. The systems only exchange concepts of a certain kind and the subsequent tasks are derived by the system itself. A human related communication which corresponds to such an exchange of concepts would be the task of writing a report. This involves many steps to be enacted by the receiver of the task which are not communicated. Nonetheless is the result in accordance to the “intentions” of the human who gave the task.

In addition such a system would be able to evaluate the output of the process and with that the parameters which lead to it. This enables self-optimizing behavior. The evaluation process is depictured as a feedback system from the Cognitive Technical System (lower right

part). In relation to manufacturing processes, a Cognitive Technical System can control entities of the Technological Application System like robots, belt conveyors, etc. to conduct different tasks. Also a multitude of Cognitive Technical Systems can cooperate to control a process chain and optimize it as a whole in respect to the optimization objectives given to the systems.
Task descriptions can be given to the system in a more abstract way. A possible description could be the shape of the product or certain product properties. With this description the Cognitive Technical System derives the needed steps to produce the desired product. A description on such an abstract level is hugely underspecified which corresponds to a task description given from one human to another (Hägele 2008). To derive the missing information needed to solve the task a massive knowledge base is mandatory.

3. Related Work

Due to the vast research efforts in different fields like artificial intelligence, software engineering, electrical engineering, etc. this section does not intent to give a complete overview, but present a selection of related work with the focus on software architectures. As possible application fields for Cognitive Technical Systems the autonomous vehicle control, manufacturing environments as well as service robotics can be identified. There are many more fields which are not evaluated further in this context.
In the field of autonomous vehicle control the DARPA grand challenges in 2005 and 2007 showed that the control of ground vehicles in a semi-unstructured environment with the constraints of following the rules of the road is possible (Montemerlo et al. 2006). The software architectures used in these Cognitive Technical Systems followed a multi layer approach with the extensive use of state machines (Urmson et al. 2007).
In autonomous robots many architectural approaches are proposed (Karim 2006, Konolige 1998, Gat 1998 et al.). These software architectures focus on the combination of a deliberative part for the actual planning process with a reactive part for motion control (Putzer 2004).
In production technology, the cluster of excellence "Cognition for Technical Systems" (CoTeSys) is researching software architectures for Cognitive Technical Systems in production environments (Ding et al. 2008). The research focuses on the implementation of cognitive abilities in safety controllers for plant control. In this context the human machine cooperation is the main evaluation scenario.
All described approaches do not focus on the application of Cognitive Technical Systems in an assembly operation.

4. Requirements

4.1 Functional Requirements

This section describes the functional requirements for a Cognitive Technical System suitable to act in a production environment. A functional requirement is a requirement which can be noticed during the operation of the system (Sommerville 2007).
The functional requirements that a Cognitive Technical System must fulfill are the capability to process different sensor inputs (visual, tactile or electric sensors) and aggregate them to extract essential information. Based on this information, the Cognitive Technical System must process the information and find the next best action concerning the current

environmental state and the given objective. To change the environment according to the next action derived by the Cognitive Technical System, external entities like robot actuators or conveyor belts have to be controlled.
The Cognitive Technical System must interact with a human operator via a human-machine-interface. The actual design of the human-machine-interface is not part of the functional requirements (Cockburn 2003) but is specified in the non-functional requirements. To derive a decision out of the received information, the system must have a knowledge base which contains the domain knowledge. Also the procedural knowledge about the different operations it has at his disposal, for changing its environment must be stored.
The environment of a production facility adds a further functional requirement for a Cognitive Technical System. The communication via different protocols with machinery like programmable logic controllers (PLC) and multi-axis robots has to be ensured.

4.2 Non-Functional Requirements

Non-Functional requirements are defined as requirements which specify criteria that can be used to judge the operation of a system, rather than specific behaviors (Sommerville 2007). The non functional requirements for the human-machine-interface derive from DIN ISO 9355-1 and can be separated in 14 categories, which will not be described here in detail.
The requirements for the software architecture are partly derived from ISO 9126. The following categories are considered the essential ones and will be described in more detail:

- Modularity, Extendibility, Flexibility
- Robustness and Reliability
- Response times
- Information- and Datamanagement
- External communication
- User Interaction

Modularity, Extendibility, Flexibility
The software architecture of a Cognitive Technical System suitable for an assembly task in a production environment has to meet the requirements of modularity, extendibility and flexibility. Modularity in this context means, that components can be interchanged without redesigning the whole system. This concerns the user interface, the different controller components and the decision making components. This demands the encapsulation of single functionalities within components and the usage of well defined interfaces between them. The software architecture must be extendable in the sense that new components can be integrated without much effort. This satisfies also the requirement of flexibility.
Robustness and Reliability
In a production environment the requirements for the reliability and the robustness of a system are high. The technical system must have a high reliability because of the high costs of a possible production stop in case of a system failure. Because of this certain safety measures must be implemented in the Cognitive Technical System. This can be realized through redundancy of components or by fault tolerant code. This also ensures a high robustness.
Response times
In a production environment processes are optimized for high throughput. This puts further constraints on the software architecture of such a system. The response time must be low enough to react to sudden changes in the environment. The deliberative part of the

Cognitive Technical System can not derive decisions in real time due to the amount of knowledge processed. Therefore the overall response time of the system has to be ensured by a mechanism which does not depend on deliberative decision making.

Information- and Datamanagement

The information flow in the Cognitive Technical System is quite extensive. The sensory information has to be processed and routed to the concerning components. The software architecture has to incorporate an internal communication to feed the information to the components. In addition, storage of the data in different repositories has to be ensured due to the high bandwidth and the amount of accumulated data.

External communication

The Cognitive Technical System has to communicate with the different entities in a production environment. These can be physical entities like robots and programmable logic controller, but also different bus protocols (CAN-Bus and Process Field Bus (PROFIBUS)) have to be supported by the respective interfaces. Also a simple extendibility of these interfaces must be possible.

User Interaction

The Cognitive Technical System has to ensure the communication with the user of the system. The user input has to be processed and the decisions of the Cognitive Technical System have to be presented to the user.

4.3 Conclusion

The functional and non-functional requirements for the system influence the design of the software architecture. Especially the requirements of a production environment by demanding a low response time of the system define the software architecture. Furthermore the reliability is an important requirement.

5. Software Architecture

5.1 Multilayer approach

To meet the functional and non-functional requirements a software architecture for a Cognitive Technical System suitable for assembly tasks has to incorporate multiple components.

The system has to work with different levels of abstractions. This means that the deliberative mechanism cannot work on the direct sensor data received from the Technological Application System. Therefore an abstraction of the received data is necessary. This demands a component which can aggregate the received information for the deliberative mechanism. To meet the requirement of a low response time a control mechanism has to be incorporated which can act without waiting for the deliberative mechanism to respond. Also, the Cognitive Technical System has to be able to control the production facilities as well as ensure a human machine communication. Especially the concepts of modularity and reliability were the driving factors for the chosen approach. To meet these requirements a multilayer approach for the software architecture of the system was chosen (Gat 1998).

Fig. 3 shows the software architecture embedded in the human-machine-interaction. The Cognitive Technical System incorporates the Technological Application System as well as the Interaction System. The software architecture separates the Interaction System into four layers which incorporate the different mechanisms required. The Presentation Layer incorporates the human machine interface and an interface for the modification of the knowledge base. The Planning Layer is the deliberative layer in which the actual decision for the next action is made. The Coordination Layer provides services to the Planning Layer which can be invoked by the latter to start action execution. The Reactive Layer is responsible for a low response time of the whole system in case of an emergency situation. The Knowledge Module contains the necessary domain knowledge of the system.

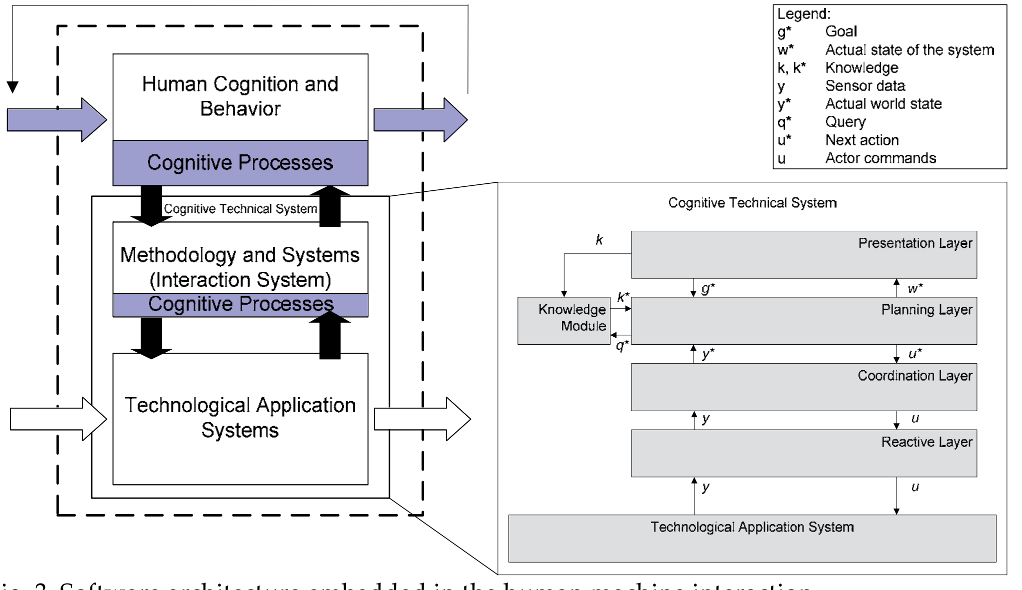


Fig. 3. Software architecture embedded in the human machine interaction

At the beginning the human operator gives the desired goal to the Cognitive Technical System via the Presentation Layer. This goal g^* is then transferred to the Planning Layer where the next action u^* is derived based on the actual world state y^* and the desired goal g^*. The actual world state is based on the measured variables y from the sensors in the Technological Application System which are transferred via the Reactive Layer. In the Coordination Layer y is then aggregated to y^*. To derive y^*, the sensor data y at a discrete time $t \in R^{\geq 0}$ is taken into account. $y(t) \in R^{n_y}$ denotes the current vector of the current measured variables at time t. This vector is then transformed in the world state $y^*(t)$. This means that the base on which all decisions in the Planning Layer are made is the actual world state y^* at a certain time t. Therefore the decision process must not take too long, because the state of the Technological Application System can have changed significantly in the meantime.

The next best action u^* derived in the Planning Layer is sent back to the Coordination Layer, where the abstract description of the next best action u^* is translated into a sequence of actor commands u, which are sent via the Reactive Layer to the Technological Application System. There, the sequence of commands is executed and the changed environmental state is measured again by the sensors. If the new measured variables y of the Technological Application System indicate an emergency situation the Reactive Layer ensures a low

response time. Then the sensor data is processed directly in the Reactive Layer and the according actor commands are executed.
Fig. 4 shows the software architecture in more detail. The different layers and their components will be described in more detail in the following section.

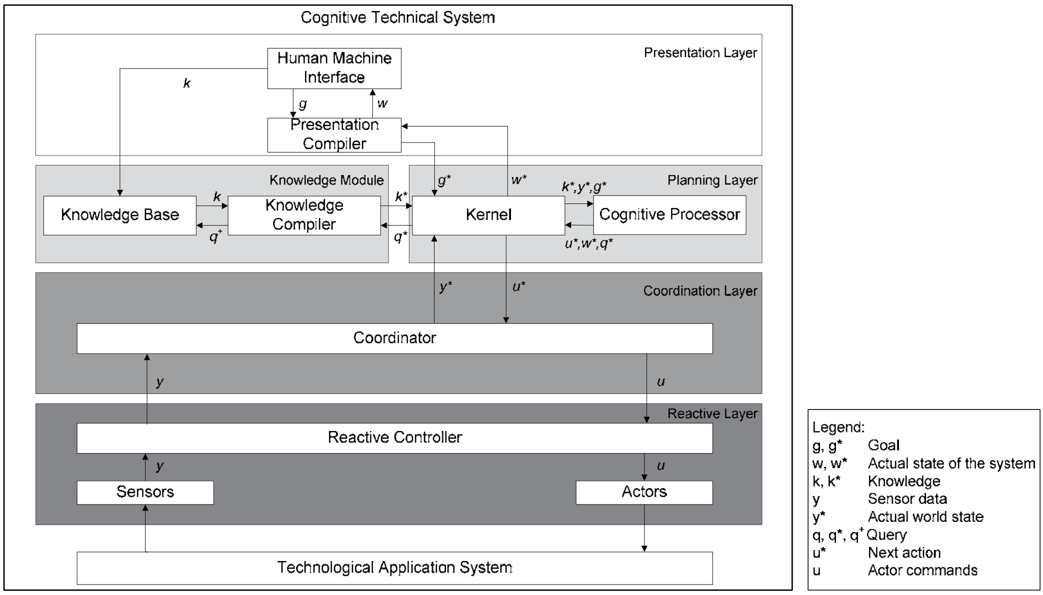


Fig. 4. Software Architecture of the Cognitive Technical System with components based on a multilayer approach

5.2 Presentation Layer

The Presentation Layer is responsible for the interaction with the user. It incorporates the human-machine-interface which is designed for the special requirements given by interacting with a technical system with cognitive capabilities.
The domain knowledge k is encoded in a fixed representational formalism. One possibility is the structuring of k in an ontology. The knowledge engineer encodes the domain knowledge specifically to the task the system has to enact. This is done prior to the system start. During the operation of the system a human operator is interacting with the system. This operator specifies a task description g, which is transferred to the component Presentation Compiler. In case of an assembly task the description g can be the description of the shape of parts to be assembled and the description of the location and forms of the parts in the final assembly. This can be done, but is not restricted to, using a graphical representation, e. g. a CAD program. The Presentation Compiler has to translate this task description g into a goal state g^* which can be interpreted by the Cognitive Processor of the Planning Layer.
Due to the changing environment the behavior of a cognitive system is not perfectly predictable in advance. Therefore, the actual state of the system should always be transparent to the operator. The actual state w^* of the system is given to the Presentation Compiler, where w^* is aggregated to a human interpretable machine feedback w which is then transferred to the operator via the Human Machine Interface.

5.3 Planning Layer

The Planning Layer contains the core elements that are responsible for decision-finding. It contains the Kernel and the Cognitive Processor as components. The Kernel distributes the signal flows in the Planning Layer. The Cognitive Processor computes the next best action u^* based on the goal state g^* and the current world state y^*. If the Cognitive Processor cannot derive a next best action it can send a query q^* for more information to the Knowledge Module.

The Kernel component then invokes the action execution according to the action returned by the Cognitive Processor. In case of a request for more information, the Kernel queries the Knowledge Base for actions applicable on the objects in y^*. According to the actual processor used, the Knowledge Base returns the knowledge k^* via the Knowledge Compiler. The additional knowledge is then considered in the computation of the next best action. Fig. 5 shows the activity diagram for the Cognitive Processor. In the rare case that the Cognitive Processor could not find an action and the Knowledge Base could not return k^*, the Cognitive Processor queries the human operator for the next action. The user can then either give the next action or change the environmental state. This means that the user changes the environment physically without telling the system explicitly about this. The system then recognizes the new environmental state via the measured variables y, reasons about the new world state y^* and derives the next best action u^* based on y^*.

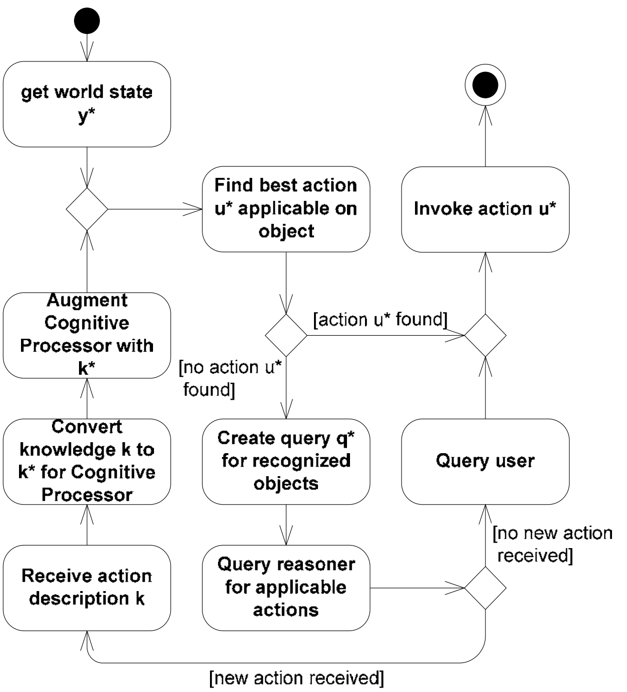


Fig. 5. Activity diagram of possible actions of the Cognitive Processor

Several architectures have been developed for the understanding of the human control behavior. The EPIC (Executive-Process Interactive Control) architecture combines cognitive and perceptual operations with procedural task analysis (Keiras 2004). The different interconnected modules, called processors, operate in parallel. The Soar architecture is a cognitive architecture based on the "unified theory of cognition" (Newell 1994), which aims to model general intelligence (Laird 1996). It models behavior as selection and application of operators to a state. A state represents the current situation of knowledge and problem-solving, and operators transfer knowledge from one state to another. At runtime, Soar tries to apply a series of operators in order to reach a goal (Laird 1996). Control in Soar refers to conflict solution and is implemented as a deliberate and knowledge-based process. ACT-R control is regarded as an automatic process by using an automatic conflict resolution strategy (Johnson 1998). Of these architectures the Soar architecture was chosen as the Cognitive Processor (Hauck 2008).

Soar is a rule based production system. Rules are fired if they match elements of the inner representation of the current $\boldsymbol{y}^*$ and modify this representation. Via input- and output-links Soar is capable of communication with its environment, e.g. to retrieve a new world state or invoke actions. In addition, a combination of the Soar architecture with a classical planning algorithm like Fast Forward (Hoffmann 2001) is currently investigated. This provides the ability to exploit the capabilities of Soar but also enables the generation of a quick plan to solve a task.

5.4 Coordination Layer

The Coordination Layer is the executable layer of the Cognitive Technical System. It provides executable services to the Planning Layer. These services correspond to the actions the Cognitive Processor can invoke. The Coordinator in the Coordination Layer also processes the measured variables $\boldsymbol{y}$ received from the Reactive Controller via the Reactive Layer and aggregates this information to the current world state $\boldsymbol{y}^*$.

Also, the Coordinator component receives the next action $\boldsymbol{u}^*$ to be executed. The abstract service invoked by $\boldsymbol{u}^*$ is a sequence of actor commands $\boldsymbol{u}$. A simple example is the stapling process of two blocks. Provided the positions of the two are known, the service `move(blockA,blockB)` then invokes the sequence of moving the actor, e. g. a robot, to the position of blockA, grasping it and transferring it to the position of blockB and releasing it. $\boldsymbol{u}$ is stored in the Coordinator component and will be executed with parameters given by $\boldsymbol{u}^*$. $\boldsymbol{u}$ is then executed in the Technological Application System via the Reactive Layer. That way, the Planning Layer is exculpated from the details of the robot movements, e. g. the exact coordinates of the block-locations, etc., which leads, due to a reduced problem space, to faster decisions.

5.5 Reactive Layer

The Reactive Layer and in it the component Reactive Controller is responsible for the low level control of the system. The vector of the measured variables $\boldsymbol{y}$ is observed for values which indicate a possible emergency situation. The Reactive Controller responds then with the according actor commands $\boldsymbol{u}$.

This ensures low response times in case of an emergency. The Reactive Controller cannot ensure a safe behavior for the system as a whole. This means if a wrong actor command sequence is sent to the actors in the Technological Application System the Reactive Controller does not check this sequence for potential consequences for the Technological

Application System according to the current state. This has to be done by the Cognitive Processor.

5.6 Knowledge Module

The Knowledge Module contains the Knowledge Base which contains the necessary domain knowledge for the Cognitive Technical System to perform the desired task. The domain knowledge $\boldsymbol{k}$ in the Knowledge Base has to be translated in a form which is interpretable by the Cognitive Processor. This is done by a Knowledge Compiler, which consists of two components: The Reasoner and the Mediator. The Reasoner queries the Knowledge Base and receives additional knowledge $\boldsymbol{k}$. This knowledge is then translated into an intermediate format $\boldsymbol{k'}$ and transferred to the Mediator. The Mediator then compiles the knowledge $\boldsymbol{k'}$ into the syntax $\boldsymbol{k^*}$ which is then processed by the Cognitive Processor. Fig. 6 shows the signal flows and the involved components. In case of an additional information request $\boldsymbol{q^*}$ by the Cognitive Processor the Mediator first translates $\boldsymbol{q^*}$ in $\boldsymbol{q'}$ and the Reasoner accesses the Knowledge Base to infer the requested information.

For assembly tasks, the domain knowledge has to contain the involved actors controlled by the Cognitive Technical System. The formalism used for the domain knowledge is the Web Ontology Language (OWL) (Smith 2004). To store the procedural knowledge, which is used by the cognitive processor in form of production rules the original form is not sufficient. Therefore, an extension to the OWL, the Semantic Web Rule Language (SWRL) (Horrocks et al. 2004) in combination with a description formalism for the direct representation of procedural knowledge in the ontology is used.

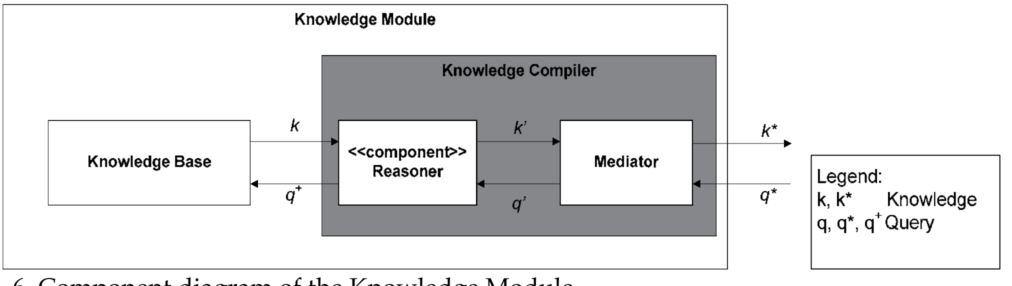


Fig. 6. Component diagram of the Knowledge Module

5.7 Conclusion

The multilayer approach ensures the encapsulation of different information abstractions in different layers. The components in the Planning Layer operate with the highest abstraction of information. The Cognitive Processor invokes the corresponding service according to the next best action. The different services manipulate the environment without dealing with the low level constraints given by the used actors. The Coordination Layer contains the service description in form of sequences of actor commands, which the Reactive Layer than executes and controls.

Due to this approach, the system can deal with a continuously changing environment and adapt itself to it. The system is hybrid in a double fold sense of the word. It connects a continuously stream of input signals with their discrete representation in states and includes reactive and deliberative components.

6. Example: Control of an Assembly Cell

The schematic layout of a robot cell, which is controlled by the Cognitive Technical System is shown in Fig. 7. It consists of two robots and a transport system, which transfers the parts via a conveyor belt. The first robot grasps the incoming parts and puts it on the conveyor. The parts colors and contours are identified by an object recognition software via a CCD camera. If the part is needed for the assembly at hand, the second robot grasps the part and transfers it either to the assembly area in case the part is needed immediately, or to the buffer area. In case that an object is not needed the conveyor transports the object to the leaving part container and it is being discharged. The second robot is equipped with a three finger robot hand to conduct complex gripping operations.

The first evaluations of the Cognitive Technical System will only involve parts with a simple contour, like blocks, spheres etc. This is necessary due to the fact that the object recognition as well as the color recognition would take much longer for complex objects. The system has to adapt to different states without the possibility to preplan the whole assembly process. Therefore the feeding of the parts is stochastic. In addition the actual world state will be repeatedly checked to evaluate if the internal representation in the Cognitive Technical System corresponds to the environmental state.

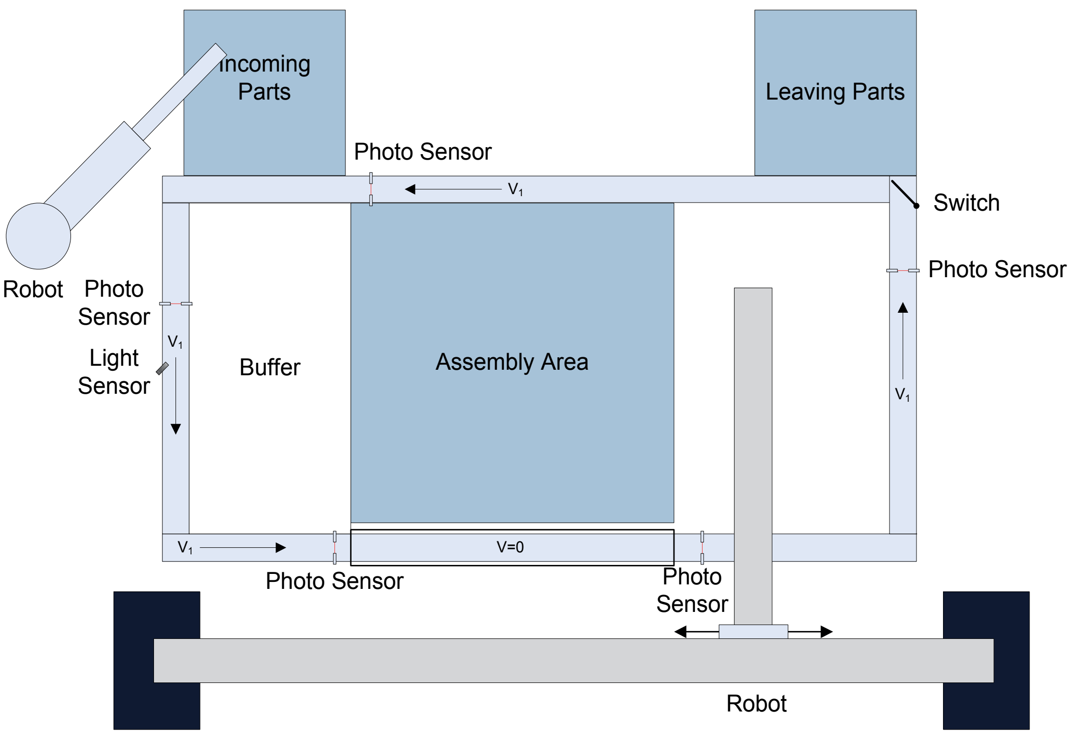


Fig. 7. Schematic of the assembly cell used for the application of the Cognitive Technical System

Possible reasons for unexpected changes in the environmental state can be:

- Erroneous identification of a part
- Dropping or misplacement of a part by the robot
- Changes in the current assembly

Erroneous identification of a part can lead to a false building order for the whole assembly and affect the outcome of an assembly operation significantly. A drop of a part can happen if the three finger robot hand grasps an object wrong or the object falls during the transfer operation. The last possible change in an environmental state is the change of the assembly. This is a scenario where the machine works in cooperation with a human. The change will then be noticed by the system via the measured variables $\boldsymbol{y}$. This is not focus of the current research, but has to be considered for future applications.

Therefore, the Cognitive Technical System has to check the actual world state periodically to prevent the consequences arising out of these changes in the environmental state. To evaluate the system, a simple assembly task will be conducted by the system. The most simplistic geometry is a tower of blocks but this will be extended to the realize of more complex geometries.

7. Conclusion and Future Work

The multilayer approach for a Cognitive Technical System suitable of conducting assembly tasks in a production environment is a feasible one. The software architecture meets the different functional as well as non-functional requirements a production environment has towards such a system. The current work focuses on the implementation of the software architecture and simulation of the environmental states. Future work will include the connection to the assembly cell and the application of the system to more complex object contours.

For interested readers the following links are recommended:

http://www.zlw-ima.rwth-aachen.de/forschung/projekte/exzellenzcluster/index.html

http://www.production-research.de

8. Acknowledgements

The authors would like to thank the German Research Foundation DFG for the support of the depicted research within the Cluster of Excellence "Integrative Production Technology for High-Wage Countries".

9. References

Brecher, C. et al. (2007). *Excellence in Production,* Apprimus Verlag, ISBN: 3940565008, Aachen

Cockburn, A. (2003). Writing effective use cases, Addison Wesley, ISBN: 9780201702255, London

Ding, H. et al. (2008). A Control Architecture for Safe Cognitive Systems, *10. Fachtagung Entwurf komplexer Automatisierungsysteme,* Magdeburg, April 2008, Gat, E. (1998). On Three-Layer Architectures in *Artificial Intelligence and Mobile Robots*, Kortenkamp D., Bonnasso R., Murphy R., (Ed.), pp. 195-211, AAAI Press, ISBN: 9780262611374, Cambridge

Gausemeier, J. (2008). *Towards a Design Methodology for Self-optimizing Systems*, Springer Verlag, ISBN: 978-1-84628-004-7, London

Hauck, E.; Gramatke, A. & Henning,K. (2008). Cognitive technical systems in a production environment, *Proceeding of the 5th international Conference on Informatics in Control, Automation and Robotics,* pp. 108-113, ISBN: 9789898111326, Madeira, May 2008

Hägele, M. (2008). Industrial robotics, In: *Handbook of Robotics*, Siciliano, B., Khatib, O., (Eds.), pp. 963-986, Springer Verlag, ISBN: 9783540239574, London

Heide, A. & Henning, K.(2006). The cognitive car - A roadmap for research issues in the automotive sector, *Proceedings of the 9th IFAC Symposium on Automated Systems Based on Human Skill And Knowledge*, ISBN: 9783902661050, Nancy, May 2006,

Hoffmann J. & Nebel B. (2001). *The FF Planning System: Fast Plan Generation Through Heuristic Search.* Journal of Artificial Intelligence Research, Vol. 14, (2001), pp.253-302, ISSN:11076 – 9757

Horrocks, I. et al. (2004). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/

Johnson, T.R. (1998). A comparison of ACT-R and SOAR. In: Schmid, U., Krems& J.,Wysotzki, F. (Eds.) *Mind modeling*, pp. 17–38, Papst, ISBN: 3933151252, Lengerich

Karim, S. et al. (2006). A Hybrid Architecture Combining Reactive, Plan Execution and Reactive Learning, *Proceedings of the 9th Biennial Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, China, August 2006

Konolige, K. & Myers, K. (1998). The saphira architecture for autonomous mobile robots. In: Kortenkamp et al. (Eds.) *Artificial intelligence and mobile robots: case studies of successful robot systems*, pp.211-242, MIT Press, ISBN: 0262611376, Cambridge

Kieras, D. & Meyer, D. (2004). *EPIC Architecture – Principle of Operation*, Univ. of Michigan, Ann Arbor

Laird, J.E.; Lehman, J.F. & Rosenbloom P. (1996). A gentle introduction to Soar, an architecture for human cognition, In: *Invitation to Cognitive Science*, MIT Press, Boston

Matlin, M. W. (2005). *Cognition*, Wiley& Sons, ISBN: 0471427780, New York

Montemerlo, M. et al. (2006). Winning the DARPA Grand Challenge with an AI robot, *Proceedings of the AAAI National Conference on Artificial Intelligence,* pp. 982-986, ISBN 9781577352815, Boston, July 2006, AAAI, Boston

Newell, A. (1994). *Unified theories of cognition*, Harvard University Press, ISBN: 9780674921016, Cambridge

Putzer, H. (2004). *Ein uniformer Architekturansatz für kognitive Systeme und seine Umsetzung in ein operatives Framework,* Dissertation, München

Smith, M. et al. (2004). *OWL Web Ontology Language Guide,* http://www.w3.org/TR/2004-/REC-owl-guide-20040210/*OWL Web Ontology Language Guide,* http://www.w3.org/TR/2004/REC-owl-guide-20040210

Sommerville, I. (2007). *Software Engineering,* Addison Wesley, ISBN: 9780201398151, London

Strasser, A. (2004). *Kognition künstlicher Systeme*, Ontos Verlag, ISBN:393720296X, Frankfurt

Strube, G. (1998). Modelling motivation and action control in cognitive systems. In *Mind modeling: a cognitive science approach to reasoning, learning and discovery*, Schmid, U., Krems, J., & Wysocki, F. (Eds.), pp 89-108, Pabst, ISBN 159326044X, Berlin

Urmson, C. et al. (2007). *Tartan Racing: A Multi-Modal Approach to the DARPA Urban Challenge*, Pittsburgh

Zimbardo, P. & Gerrig, R. (2005). *Psychology and Life*, Pearson, ISBN: 0205428673, Boston

3

Two stage approaches for modeling pollutant emission of diesel engine based on Kriging model

El Hassane Brahmi, Lilianne Denis-Vidal, Zohra Cherfi, Nassim Boudaoud and Ghislaine Joly-Blanchard
University of technology of Compiegne
France

1. Introduction

The automotive industry faces the competing goals of producing better performing vehicles and keeping development time with low costs. It is crucial for the manufacturers to be able to produce fuel-economic vehicles, which respect pollutant emissions standards, and which meet the customers expectations. Accordingly, the complexity of the engines responses we have to optimize and the number of the parameters to control during the design stage, have increased rapidly, in the last years.
In order to deliver vehicles, which respond to these requirements, in a reasonable time scale, companies use design of experiments (DOE) (Schimmerling et al., 1998) in one side, and modelling, in the other side. DOE is a power tool, but the cost of the experiments and their duration, particularly in the field of pollutant emissions, can be a limit to their use in automotive industry.
The engine developers use two main approaches to model engine behaviour. The first one is based on chemical and physical models, via differential system. This approach is not the subject of this article, because we have not such models. Furthermore, even when these models are available, generally, they are time-consuming, impractical for multi-objective optimisation routines, and fail to capture all the trends in the engine system described by measured data (like Zeldovich model). All this, is particularly true when the number of the control parameters is large and engine responses are complex.
Statistical modelling based on carefully chosen measured data of engine performance, according to an experimental design is an important alternative technique.
Strategies based on Lolimot (Castric et al., 2007) (Local Linear Model Tree) and Zeldovich mechanisms (Heywood, 1988) have been developed in order to predict emissions of NOx. In the first case, the corresponding model can lead to singular points, which reduces the precision of the results. In the second case, the results are not satisfactory enough.
The literature presents several methods based on statistical trainings such as neural networks. This method gives good results, even in the nonlinear case. However, it is not adapted to our case, because it requires a great number of experiments to obtain a

significant estimate of its parameters, and we are very limited by the small experiments number which the industrialist is able to realize. The techniques of the design of experiments (Cochran & Cox, 1957) were conceived to deal with this kind of problems. On the other hand, recent works (Sacks et al. 1989; Bates et al. 1996; Koehler & Owen, 1996) suggest that the polynomial models are not adapted to the numerical experiments. For example, a surface of response of order two is not enough flexible to model a surface admitting several extrema.
The aim of this paper is to present the result that we have obtained in the field of pollutants emissions prediction. These results were obtained without the increase of the number of the experiments that the industrialist can do. We call upon a sophisticated statistical model resulting from the field of geostatistic named Kriging.
We use this method, through two approaches, in order to improve the prediction of NOx (nitrogen oxide) emissions, and fuel consumption.
In the first stage, we estimate the response directly from the controllable factors like main injection timing, pressure in common rail. This can be assimilated to a black box modelling.
In the second stage, we propose an innovative approach that allows us to predict the response from a functional data. More precisely, we estimate the engine performance from signals like pressure and cylinder temperature. These signals are obtained from a model of combustion. The main advantage of the second approach is that it allows us to include a physical knowledge of combustion. This lack of knowledge is often criticized in the case of black boxes models.
The Kriging method is very well adapted for the second approach which predicts engine responses from the state variables (signals) obtained from a physical model of combustion. This means that this method can be recommended in cases where we have a lot of decision variables, with a small number of experiences. This is due to the fact that the method is based on the study of the similarity of the response of interest in different sites, in relation to the distance separating them. We recall that, Software such as R and Matlab contain a toolbox to use this method. But unfortunately, the latter are restricted to less than 3 dimensions. Adapting the method to higher dimensions has been considered.
To implement this second approach, a model reduction is needed. This reduction will be made in two steps:
1) Reducing the number of state variables from 10 to 2 by the study of correlations.
2) Reducing the number of points in each signal using the theory of Fourier.
Once the reduction is made, the Kriging can be applied to the model obtained.
This paper is organized as follows: In the second section, we describe the engine behaviour and recall the importance of controlling pollutant emissions. In the third section, the ordinary Kriging techniques are recalled. In the fourth section, two different approaches for modelling our problem are proposed. An efficient reduction model strategy is considered in order to apply the Kriging method. Finally, the Kriging method is applied to the reduced model. In the last section, numerical results are given followed by a short discussion.

2. Engine calibration

The engine calibration is a process which aims at defining the values of the engine control parameters. During the ten last years, the set of themes “engine calibration” took an important place in the process of the development of the internal combustion engines.

Indeed, under the impulse of the standards more and more severe, the car manufacturers are brought to integrate more and more elaborated technologies in the power unit.
Under these conditions, the principles of control used in the past (cartographic approach, buckles open…) are not enough sufficient. Indeed, the output variables (quantity injected, advances in lighting or in injection) were primarily given starting from two variables of entry (speed and load). Today, the use of new technologies in the conception of engines, in order to reduce the pollutant emissions, as for example EGR (exhaust Gas Recirculation) (Pierpont et al. 1995), multiply the number of parameters to control, as we can see it in Fig.1. This figure shows the exponential evolution of number of setting parameters due to the hardware complexity increase. This makes the cartographic approach impracticable. Moreover, this kind of approach does not take into account the dynamic of system.
The main drawback of this evolution is the increase of the difficulty to understand the engine behavior. To deals with all the parameters, we use a Kriging model which we define in the next section.

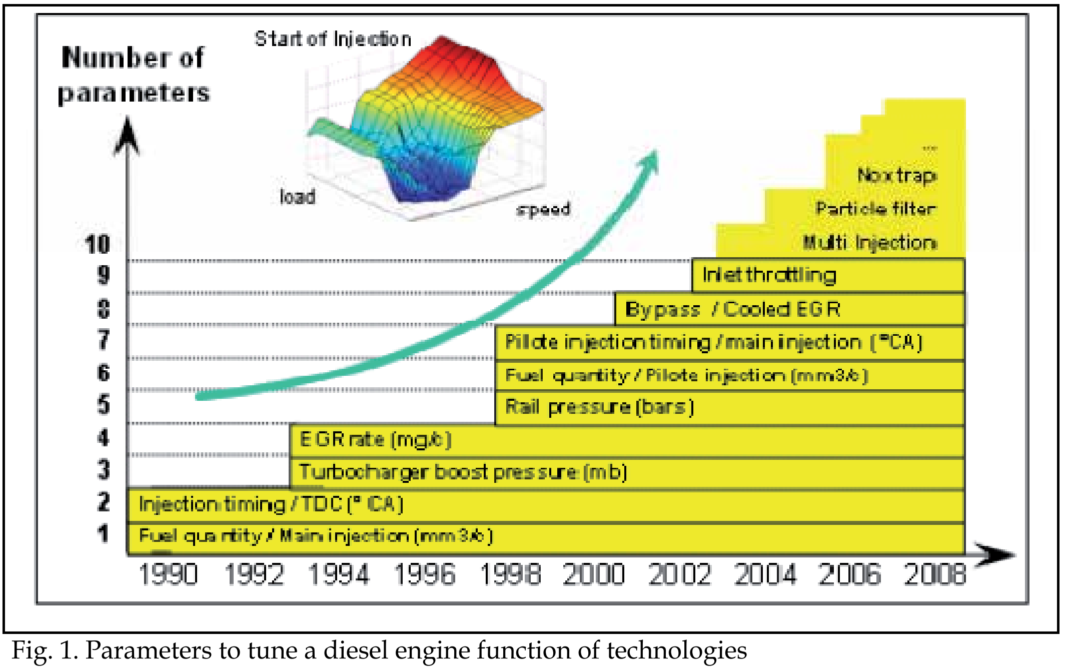


Fig. 1. Parameters to tune a diesel engine function of technologies

3. Ordinary Kriging Techniques

Kriging methods are used frequently for spatial interpolation of soil properties (Krige, 1951; Matheron, 1963). Kriging is a linear least squares estimation algorithm. It is a tool for interpolation. The aim is to estimate the value of an unknown real function Z at point x_0^*, given the values of function Z at some other points $x_i \in R^d$ for each $i = 1, \ldots, n$.

3.1 Ordinary Kriging

The ordinary Kriging estimator $\hat{Z}(x_0^*)$ is defined by:

$$\hat{Z}(x_0^*) = \sum_{i=1}^{n} \lambda_i \, Z(x_i) \tag{1}$$

Where n is the number of surrounding observations $Z(x_i)$ and λ_i is the weight of $Z(x_i)$. The weights should sum to unity in order to make the estimator unbiased:

$$\sum_{i=1}^{n} \lambda_i = 1 \tag{2}$$

The weights are also determined such that the following Kriging variance is minimal under the constraint given by the equation 2:

$$var\left(\hat{Z}(x_0^*) - Z(x_0^*)\right)$$

This leads to a classical optimization problem with equality constraint. The Lagrange multiplier theory is used in order to work out this problem. This gives a linear system which must solved (Davis, 1986).

3.2 Variogram

The variogram is a function representing the spatial dependency. It is obtained from the stationarity definition. In fact, this stationarity hypothesis is an indispensable condition for the use of the Kriging method.

In the case of ordinary Kriging the expression of the variogram is obtained from the following definition of intrinsic stationarity:

1) $E\big(Z(x_i + h) - Z(x_i)\big) = 0 \;\; \forall\, i \in \{1, \dots, n\}$

2) $Var\big(Z(x_i + h) - Z(x_i)\big) = 2\gamma(h) \;\; \forall\, i \in \{1, \dots, n\}\; and\; h$

More precisely, the expression of the theoretical variogram, is deduced from the second condition of intrinsic stationarity. This condition means that the variation of a data set is only dependent on distance r between two locations, where the variables values are $Z(x_i + h)$ and $Z(x_i)$ with $r = |h|$.

Note that the variogram, $\gamma(h)$, is a function of the separation between points h and not a function of the specific location $(x_i, x_i + h)$**.** This mathematical definition is a useful abstraction, but not easy to apply to observed values.

Consider a set of n observed data: $\{(x_1, z_1), (x_2, z_2), \dots (x_n, z_n)\}$, where x_i is the location of observation i and z_i is the associated observed value**.** There are $\frac{n(n-1)}{2}$ unique pairs of observations**.** For each of these pairs we can calculate the associated separation distance: $r = |x_i - x_j|$

To infer the variogram from observed data, we will then use the common formula for the experimental variogram (Cressie, 1993).

$$\hat{\gamma}(r) = \frac{1}{2N(r)} \sum_{N(r)} \left[Z(x_i) - Z(x_j)\right]^2 \tag{3}$$

Where:

$$N(r) = \left\{(i, j) \quad such\; that\; \left|x_i - x_j\right| = r \;\right\}$$

$N(r)$ is the pair number of $Z(x_i + h)$ and $Z(x_i)$; $\hat{\gamma}(r)$ is the experimental variogram.

3.3 Variogram Modeling

The experimental variogram presented in equation 3, estimates the theoretical variogram, for only a finite number of distances. Moreover, it does not necessarily form a valid variogram. This means, that maybe, it does not concern a negative conditionally function. Indeed, this condition is necessary to ensure the positivity of the variance of a sum of random variables (Christakos, 1984).

The experimental variogram is then modeled by a function of negative conditional type and is defined for all distances. This modeling makes the Kriging possible. A variogram model should be fitted to such variogram.

A model must be selected among the various forms of the variogram models which exist in the literature and adjusted of experimental variogram (Arnaud & Emery, 2000). This means that the parameters of the model must be estimated. This adjustment can be done graphically, but it is usually done with an estimation method such as the weighted least squares or maximum likelihood method.

Once the variographic model is chosen, and its parameters are estimated, we compute the weights λ_i which appear in (1) by solving the following system:

$A\Lambda = B$, with

$$A = \begin{bmatrix} \gamma(r_{11}) & \gamma(r_{12}) & \cdots & \gamma(r_{1n}) & 1 \\ \gamma(r_{21}) & \gamma(r_{22}) & \cdots & \gamma(r_{2n}) & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \gamma(r_{n1}) & \gamma(r_{n2}) & \cdots & \gamma(r_{nn}) & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (4)$$

$$\Lambda = [\lambda_1 \quad \lambda_2 \quad \ldots \quad \lambda_n \quad \lambda]^T$$

And

$$B = [\gamma(r_{01}) \quad \gamma(r_{02}) \quad \ldots \quad \gamma(r_{0n}) \quad 1]^T$$

Where λ is the Lagrange multiplier.

$\gamma(.)$ is the variogram model used for adjusting the experimental variogram.

r_{ij} is the distance between the locations x_i and x_j.

The variance of the estimate s_i^2 i.e. the square of the standard error at each point is obtained by the relationship:

$$s_i^2 = \Lambda^T.B$$

If we assume that the estimation errors are normally distributed around the true value, then the probability that the true value will be in $Z(x_i) \pm s_i$ is 68 %, while the probability that the true value will be in $Z(x_i) \pm 2s_i$ is 95 %, (Davis, 1986).

3.4 Kriging Emulator Validation

The true test of the quality of the fitted emulator model is its ability to predict the response at untried factor values. In order to maximally exploit the data to aid model fitting, the emulators are validated using leave-one-out cross validation. This process involves taking the fitted model and re-fitting it to a subset of non used experimental data.

More precisely, for an experiment with d design factors $y = y_1, \ldots, y_d$, the set of n experimental design points $X = x_1, \ldots, x_n$ and responses $Z = z_1, \ldots, z_n$, contain the information used to build the Kriging model. A cross validation involves predicting at each design point in turn when that point is left out of the predictor equations. Let $\hat{Z}(x_i)$ be the estimate of the $Z(x_i)$ based on all the design points except x_i. The prediction error (the estimated root mean square error, RMSE) is then calculated as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[Z(x_i) - \hat{Z}(x_i)\right]^2} \tag{5}$$

An index of the accuracy of the emulator is made by expressing as a percentage of the range of the response Z,

$$RMSE = 100 \times \frac{RMSE}{\max(Z) - \min(Z)} \tag{6}$$

4. Two approaches to model engine responses

In this section we present two stage approaches based on Kriging method for the prediction of NOx (nitrogen oxide) emissions, and fuel consumption.

In the first stage, we estimate the response directly from the controllable factor like main injection timing, pressure in common rail (black box).

In the second stage, we propose an innovative approach that allows us to predict the engine response from signals, like pressure and cylinder temperature (states variables of combustion chamber).

4.1 First approach

We recall that the first approach consists to build a Kriging model from the controllable parameters. Thus, the Kriging was trained on about 300 input/ output sets of points generated by using D-optimal design method. The training examples cover engine speeds from 1000 rpm to 5000 rpm in 250 rpm intervals, and load vary from 1 to 23 bar. The data was generated to cover the cycle point of the engine map in order to construct a global Kriging emulator. For this reason, our model takes into account the engine speed among the following control parameters:

Prail : rail pressure,	- Main: Main injection quantity,
Mpil1: pilot1 injection quantity,	- Pmain: Main injection timing,
Mpil2: pilot2 injection quantity,	- Ppil2: pilot2 injection timing,
Ppil1: pilot1 injection timing,	-VNT: turbine vane position,
VEGR: EGR valve position,	- Volet: position component of admission,

4.2 Second approach

4.2.1 Modelling

In the second approach, we propose an innovative approach that allows us to predict the response from signals like pressure and cylinder temperature (states variables of combustion chamber).

More precisely, we decompose the problem of estimation of the engine responses, into two steps sub problems (Fig.2):

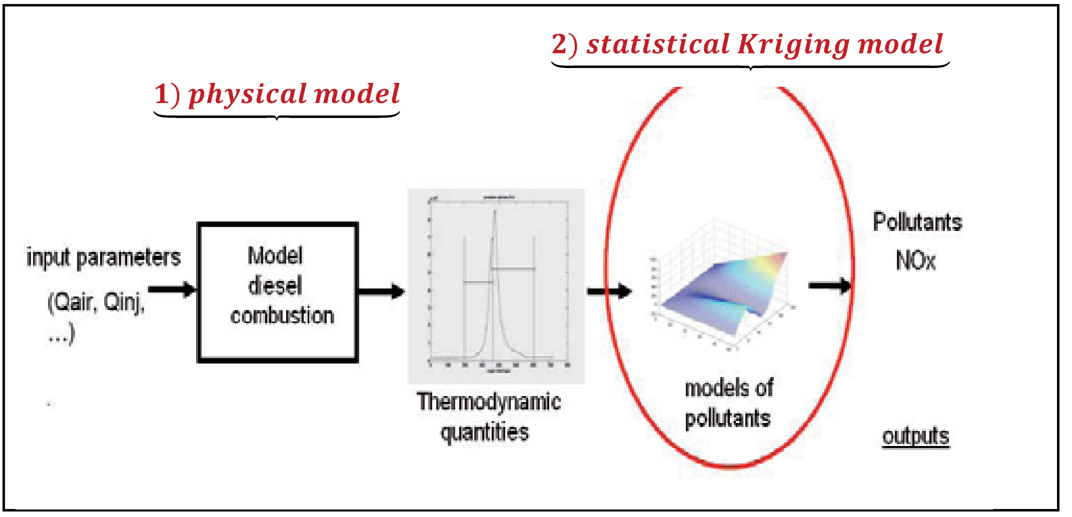


Fig. 2. Coupling of the pollutants and consumption models with the combustion model.

1) The first step consists in simulating the various thermodynamic quantities from a physical model. In this work, we use the model developed by (Castric et al., 2007), which takes into account the input parameters. It leads to have a good representation of the experimental results. This model allows us to generate the following thermodynamic quantities:

The cylinder low pressure (the alone quantity that we can measure)
The temperature in the cylinder,
The temperature of the fresh gas in the cylinder
The temperature of the mixed gas in the cylinder,
The temperature of the burned gas in the cylinder,
The mass of the fresh gas in the cylinder,
The mass of the entrained gas in the cylinder,
The mass of the burned gas in the cylinder,
The turbulence in the motor,
The fuel vapor mass.

We precise, that each signal is represented by a vector of 1334 components.

2) The second step consists in building a statistical Kriging model, from the 11 thermodynamics quantities generated by the model of combustion.

It is true that the advantage of this procedure is that, it allows us to include a physical knowledge of combustion. But this approach requires a great time of computing. Indeed to build Kriging from 11 signals, can pose a serious problem in memory capacity and the computing time can be considerable. Thus, to be able to implement this procedure, a reduction of the model is essential.

4.2.2 Model reduction

The data of the first model can be directly used for the Kriging. It is not the case for the second one. In the last case the data have to be reduced.
The reduction process begins by studying the different correlations between the state variables and their corresponding p-value. The chosen criterion consists in testing the p-value: if it is less than to 0.05, the correlation is significant. This analysis allows us to retain only two state variables: the cylinder pressure P and the mixed gas temperature in the cylinder, Te.
In the second step, the number of components of the two remaining signals is reduced. This is accomplished by using the discrete Fourier transform. The function fft of Matlab returns the discrete Fourier transform (DFT) of a vector, computed with a fast Fourier transform (FFT) algorithm. After calculating the coefficients, a minimum number of them are retained. This allows us to reproduce the initial signal, with a relative error approximately less than 0.02.
The reduction of the number of points of each signal is tantamount to minimize the number of Fourier coefficients representing that signal. The two retained signals representing respectively the cylinder pressure and the temperature of the mixed gas in the cylinder, have been reduced to a 40 Fourier coefficients. Each signal has been reconstructed from the 40 kept coefficients, with an acceptable relative error. The following table.1 presents the relative error committed, for the reconstruction of the two signals from the 40 coefficients selected:

Relative error = $\frac{\|S-S_{rec}\|}{\|S\|}$

S: is the experimental signal
S_{rec} is the reconstruction of the signal S using the fast Fourier transformation.
$\| . \|$ is the Euclidian norm.

Type of signal	relative error
the cylinder low pressure	0.01
the temperature of the mixed gas in the cylinder	0.02

Table 1. Relative error committed for the reconstruction of two signals.

Figure 3 shows the experimental signals, resulting from the combustion and their reconstruction by using the fft Matlab function.

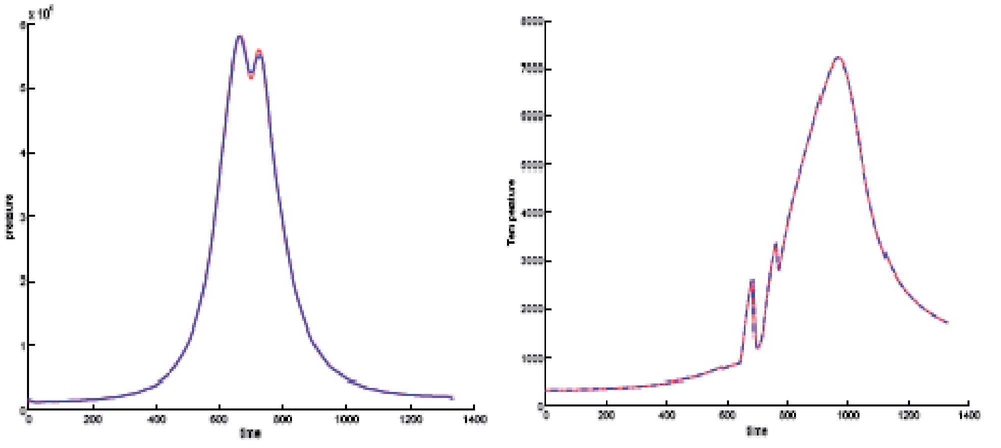


Fig. 3 Rebuilding of the measured signals (red curve) by using the discrete Fourier transform (blue curve)

Such reduction makes the Kriging possible. The considered entries of the model are:

$$\begin{cases} P_{cyl,i}(t) \rightarrow [c_1(P), \dots, c_n(P)]_i \\ T_{cyl,i}(t) \rightarrow [c_1(T), \dots, c_n(T)]_i \end{cases}$$

Where:

i is the index that corresponds to the ith operating point of engine. An operation point of the engine is defined by engine speed and engine torque

$c_n(f)$ is the kept Fourier coefficient for the signal f, which corresponds to the ith operating point of engine.

5. Application to the estimation of engine responses

In the previous section, we have presented and explained the two approaches used in this work, in order to model a behavior of diesel engine. Then, this section will be devoted to present the results respectively obtained by each approach, for the estimation of each response.

5.1 Numerical results using the first approach

We recall that the construction and the modeling of the experimental variogram is the most important step in the Kriging method. Thus, in this part, we will start by giving the chosen model.

Variogram fitting:

Variography modeling is a critical step and most difficult in the construction of a Kriging model. For this reason, several models were adjusted and then compared. It was difficult to select the better model graphically. The cross validation facilitates the work. It allows us to select the one, which minimizes the root mean square error.

For the NOx, the retained model is a Gaussian model which is expressed by the equation:

$$\gamma(r) = c_0 + c\left(1 - exp\left(-\frac{r^2}{a^2}\right)\right) \qquad as \qquad r \geq 0 \tag{7}$$

The value of the model parameters was founded using the least square method. So, we obtain: c_0=10.929, c=1.829, a=309.559.

For the Consumption, the model used is an exponential model, given by the equation:

$$\gamma(r) = c_0 + c\left(1 - exp\left(-\frac{r}{a}\right)\right) \qquad as \qquad r \geq 0 \tag{8}$$

This leads to: c_0=7.106, c=2.415, a=477.444.

Where:

r is the distance

c_0 is the Nugget effect

$c_0 + c$ is the sill correspond to the variance of $Z(x)$

$\sqrt{3a}$ and 3a are the range (the distance at which the variogram reaches the sill) for the Gaussian and exponential model respectively (Baillargeon et al., 2004).

Figures 4 shows the experimental variogram (red points), and Gaussian model (blue curve) corresponding to NOx response.

Figures 5 shows the experimental variogram (red points), and exponential model (blue curve) corresponding to consumption response.

The variogram is the tool which quantifies the spatial correlation of the response in of interest. It measures the variability of NOx and consumption as a function of distance. We notice that, when the distance reaches the range $535 \approx \sqrt{3a}$ (Fig.4) and $1300 \approx 3a$ (Fig.5), the variation becomes stationary. This explains why we can have a similar behavior of consumption and NOx on two different operating points, thus with a pattern of different control parameters

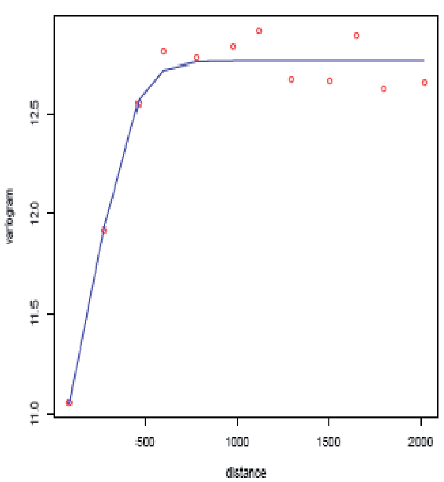


Fig. 4. Experimental and Gaussian model variogram in the case of NOx

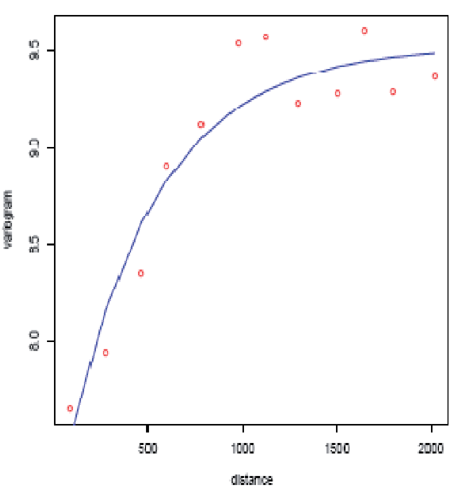


Fig. 5. Experimental and exponential model variogram in the case of consumption

Fig. 4. and Fig. 5. Experimental and model variogram

Figures 6 and 7 show the Cross-validation plots for the Kriging model, corresponding to the Gaussian and exponential variogram respectively. The plots contain the measured, the Kriging estimated value and a 10% errors bands.
The accuracy of predictions was similar for both validation data. Accuracy was good for both of the responses and still within 10% for the majority of operating conditions.
By against, graph 7 presents some observations which are poorly estimated. This is because they are far from the cloud of points used for the adjustment. This bad estimate is also due to the experimental design used. The classical and optimal designs, in particular the D-optimal, are not suitable for Kriging, which is based on measuring similarity between sites. Indeed, the D-optimal design allows to test just a small number of levels for each variable and tend to generate points on the edges of the experimental field (Koehler & Owen, 1996). This distribution of points, which is optimal to fit a polynomial model, cannot pick up any irregularities inside the experimental field and lead to some poorly estimated points. To address this problem, we recommend to use an appropriate designs for Kriging. Class 'space filling designs', such as Latin hypercubes, provide a good spatial distribution of points and is well adapted for modeling by Kriging (Stein, 1987), (McKay et al., 2000).

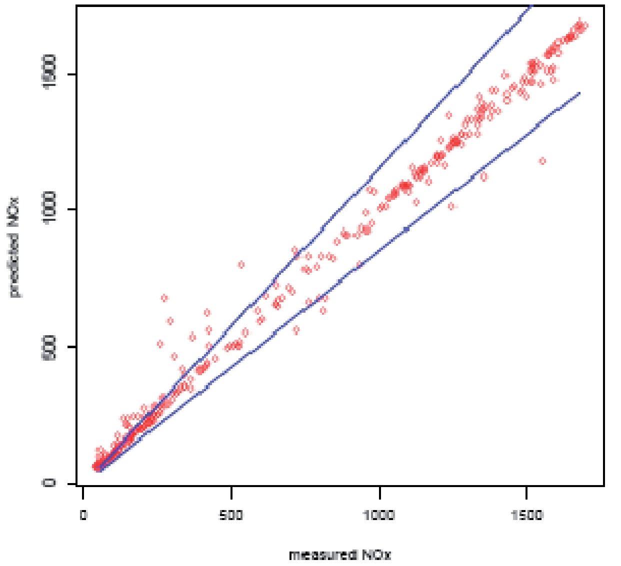


Fig. 6. Measured and Kriging predicted NOx [ppm] with ± 10% error bands

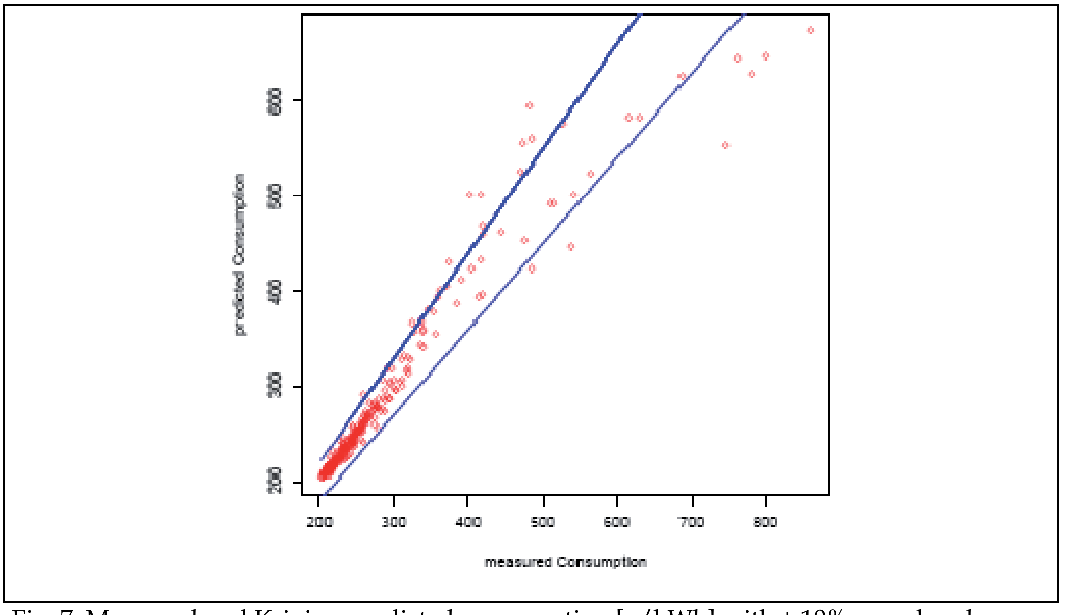


Fig. 7. Measured and Kriging predicted consumption [g/kWh] with ± 10% error bands

The emulator model is fitted to each response in turn and the RMSE, percentage RMSE are recorded. These results are presented in Table2. The percentage RMSE results show that the model has a %RMSE less than 7% of the range of the response data. This indicates roughly, that if the emulator is used to predict the response at a new input setting, the error of prediction can be expected to be less than 7%, when compared with the true value.

	NOx	Consumption
RMSE	61.4	40.63
%RMSE	3.84	6.19

Table 2. Kriging RMSE end %RMSE for each response: first approach case

5.2 Numerical results using the second approach

This subsection is devoted to the presentation of the numerical results obtained in the case of the second modeling. More precisely, we give the mathematical model used to adjust the experimental variogram.

Variogram fitting:

The experimental variogram and the model which adjusts it for each response, were obtained by the same way that we have used in the first approach case.

For the NOx, the model used is a power model given by equation:

$$\gamma(r) = c_0 + cr^a \quad as \quad r \geq 0 \;\; and \;\; 0 \leq a < 2 \tag{9}$$

The value of the model parameters was founded using the least square method.

So, c0=997.28, c=0.00018, a=1.52.

In this case the variogram does not show a sill. This means that the variance does not exist.

For the consumption, the model used is an exponential model given by equation:

$$\gamma(r) = c_0 + c\left(1 - exp\left(-\frac{r}{a}\right)\right) \quad as \quad r \geq 0 \tag{10}$$

So c_0=5193, c=0.0327 10^5, a=5.953610^5

Where:

r is the distance.

c_0 is the Nugget effect.

$c_0 + c$ is the sill correspond to the variance of $Z(x)$.

3a is the range (the distance at which the variogram reaches the sill) for the exponential model (Baillargeon et al., 2004).

Figures 8 shows the experimental variogram (red points), and power model (blue curve) corresponding to NOx response.

Figures 9 shows the experimental variogram (red points), and exponential model (blue curve) corresponding to consumption response.

We notice that when the distance reaches the range 1.786110^6 $\approx 3a$ (Fig. 9), the variation becomes stationary. In other term, this means that there is no correlation beyond the distance 3a. This explains that we have a similar behavior of consumption on two different operating points, thus with a pattern of different control parameters.

Let us notice that the model used here for the variogram of NOx, is of power type, contrary to what we had made in the first approach, where the Gaussian model was retained.

This explains that different engine configurations, lead to different behavior of the NOx. More details will be given in the section 6.

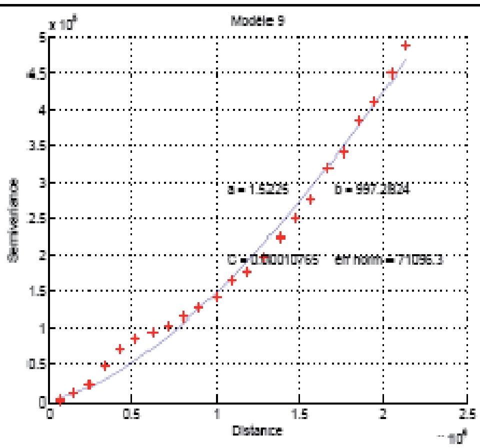


Fig. 8. Experimental and power model variogram in the case of NOx

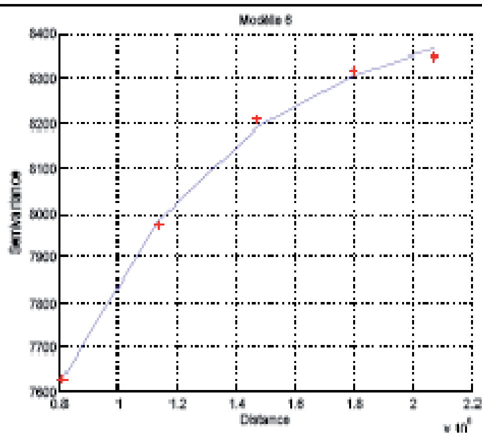


Fig. 9. Experimental and exponential model variogram in the case of consumption

Fig. 8. and Fig. 9. Experimental and model variogram

Figures 10 and 11 show the cross-validation plots for the Kriging model, corresponding to the power and exponential variogram respectively. The plots contain the measured, the Kriging estimated value and a 10% errors bands.

As we can see it, the accuracy of the predictions is similar for both response and still within 10% for the majority of operating conditions.

We just notice that in the second approach, the accuracy of the predictions is improved for the two responses, compared to the first approach. This improvement is very clear for the consumption estimation.
We can explain this improvement, by the fact that in the second approach, we include thermodynamic quantities such as the pressure, for the prediction of the two responses. The inclusion of these quantities allows to bring back an additional knowledge for the prediction of the both responses. Indeed, this knowledge results from the fact, that these quantities represent the states variables of our system, and they characterize the behavior of combustion in the internal of the combustion chamber.

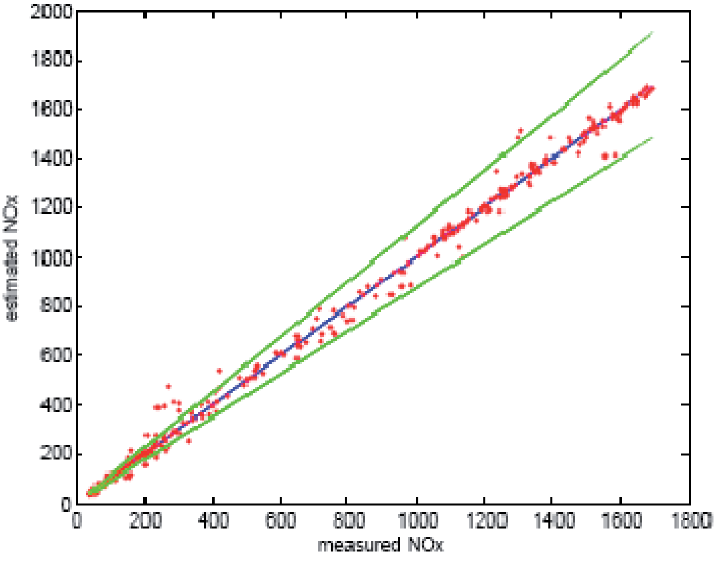


Fig. 10. Measured and Kriging predicted NOx [ppm] with ± 10% error bands

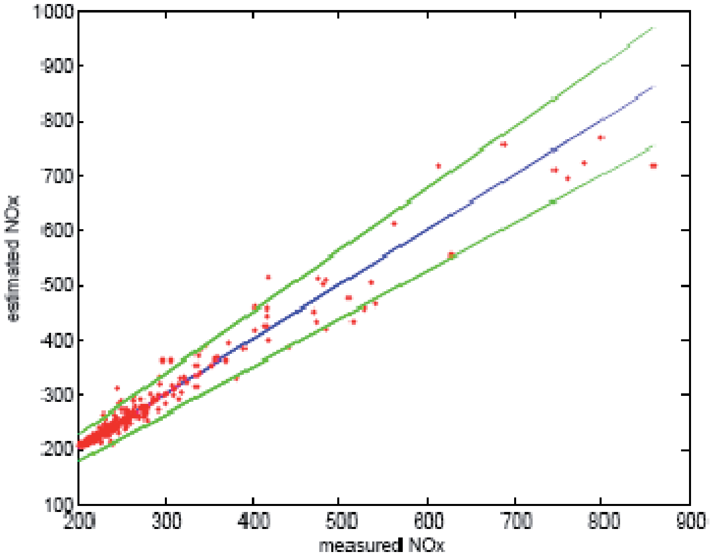


Fig. 11. Measured and Kriging predicted consumption [g/kWh] with ± 10% error bands

The emulator model is fitted to each response in turn and the RMSE, percentage RMSE are recorded. These results are presented in Table3. The percentage RMSE results show that the model has a %RMSE less than 4% of the range of the response data. This indicates roughly, that if the emulator is used to predict the response at a new input setting, the error of prediction can be expected to be less than 4%, when compared with the true value.

	NOx	Consumption
RMSE	40.51	19.99
%RMSE	2.45	3.04

Table 3. Kriging RMSE end %RMSE for each response: second approach case

6. Comparison and discussion

We recall that in the section 4, we have presented two different approaches, based on the Kriging model. In this section we will try to make a comparison between these two approaches, and discuss the advantages and inconvenient of each of them.

Case of NOx:

A legitimate question, which we could ask in the case of the estimate of NOx, is the following one:

Why do we obtain a variogram of power type in the second approach, while we had obtained a Gaussian variogram in the first approach, and the pressure is obtained with the same parameters of control?

In fact, the power variogram obtained in the second approach is a better representation of the true behavior of the emissions of NOx. Indeed, the interpretation of the power variogram suggests that the variability of the response increases with the distance between

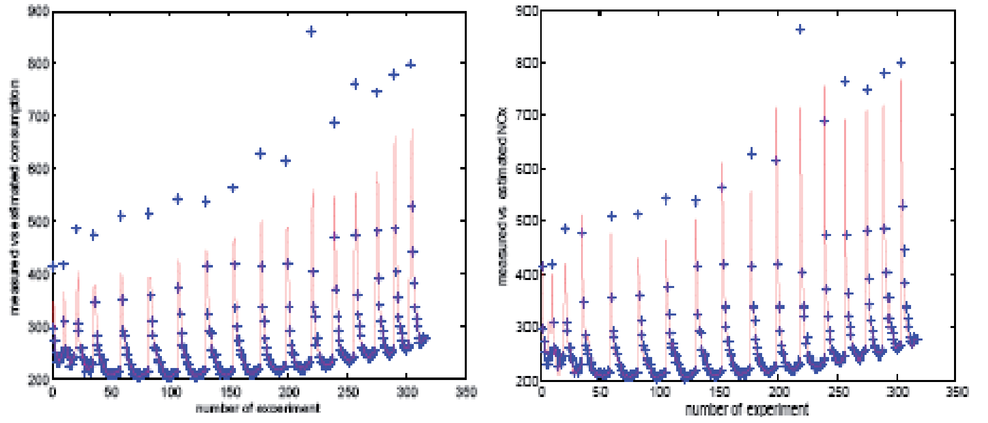
the points. This interpretation joins the opinion of the experts, who say that for two various engine configurations, the quantity of the corresponding NOx emissions will be also different.
Obtaining a Gaussian variogram in the first approach, is explained by the fact that the speed parameter of the engine take a raised values compared to the other control parameters. For example, if we take the first and the second line of the table 5, which correspond to two different engine speeds, we notice that the behavior of NOx is similar. However, the distance between these two points, is very tall (caused by the engine speed) which explains the sill on the variogram of the first approach.
Fortunately, this change in the behavior of variogram does not have an influence on the prediction of NOx. But the interpretation of the variogram in the first approach can lead us to make false conclusions. Indeed, in the case of the first approach, the variogram makes us believe that the quantity of the NOx emissions remains invariant when we consider very different configurations of control parameters. This does not reflect reality. In the case, where we wish to use the variogram, to understand how a response varies. We advise to check the values of the data, or to standardize the factors of the model.

N	Prail	Main	Mpil1	Mpil2	Pmain	Ppil1	Ppil2	VNT	VEGR	Volet	NOx
1000	407,7	5,9	1,0	1,0	-4,4	-18,7	-11,2	79,9	36,0	75,9	67,0
2000	609,0	11,1	1,1	1,3	-5,9	-36,2	-15,2	67,4	34,5	75,9	64,1

Table5. Example of control parameters configuration

Case of consumption:

To manage to highlight the contribution of the second approach in the improvement of the prediction of consumption we consider another representation of the results in figure 12.
We note that for the first approach, the Kriging method could estimate with a good accuracy all the points which are close to the cloud used for the adjustment. The prediction of the points which are far from the cloud was bad (as it is explained in section 5.1).
The use of the second approach brought back an improvement for the estimate of these points. This gives a force of extrapolation to the Kriging method.



The first approach The second approach

Fig. 12. Comparison of consumption estimation for the two case approaches. (the + points are the experimental data and the red line is the model)

7. Conclusion

This paper deals with the problem of engine calibration, when the number of parameters of control is considerable. An effective process to resolve such problems contains generally, three successive stages: design of experiments, statistical modeling and optimization. In this paper, we concentrate on the second stage. We discuss the important role of the experimental design on the quality of the prediction of the Kriging model in the case of consumption response. The Kriging model was adapted to allow an estimation of the response in the case of higher dimensions. It was applied to predict the two engine responses NOx and consumption through two approaches. The first approach gives acceptable results. These results were clearly improved in the second approach especially in the case of consumption. We demonstrate that the resulting model can be used to predict the different responses of engine. It is easy to generalize for various diesel engine configurations and is also suitable for real time simulations. In the future, this model will be coupled with the evolutionary algorithms for multi-objective constrained optimization of calibration.

8. References

Arnaud, M.; Emery, X. (2000). Estimation et interpolation spatiale. *Hermes Science Publications*, Paris.

Bates, R.A.; Buck, R.J.; Riccomagno, E. ; Wynn, H.P. (1996). Experimental Design and Observation for large Systems. *J. R. Statist. Soc. B*, vol. 58, (1996) pp. 77-94.

Baillargeon, S.; Pouliot, J.; Rivest, L.P.; Fortin, V. ; Fitzback, J. interpolation statistique multivariable de données de précipitations dans un cadre de modélisation hydrologique, Colloque Géomatique 2004: un choix stratégique, Montréal (2004)

Castric, S.; Talon, V.; Cherfi, Z.; Boudaoud, N.; Schimmerling, N. P. A, (2007) Diesel engine com-bustion model for tuning process and a calibration method. IMSM07 The

Third International Conference on Advances in Vehicul Control and Safety AVCS'07, Buenos Aires, Argentine (2007).

Castric, S. (2007) Readjusting methods for models and application for diesel emissions, PhD thesis, University of Technology of Compiègne, 2007.

Christakos, G. (1984). On the problem of permissible covariance and variogram models. *Water Resources Research*, 20(2):251-265.

Cochran, W. G.; Cox, G. M. (1957). Experimental Designs. Second edition. New York : Wiley. p 611.

Cressie, N. A. C. (1993) Statistics for spatial data. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons Inc., New York. Revised reprint of the 1991 edition. A Wiley-Interscience Publication.

Davis, J.C. Statistics and Data Analysis in Geology, second edition John Wiley and Sons. New York (1986).

Edwards,S.P.; A.D.P.; Michon, S.; Fournier, G. The optimization of common rail FIE equipped engines through the use of statistical experimental design, mathematical modelling and genetic algorithms, *S.A.E paper,* vol. 106, n°3, (1997), pp. 505-523.

Goers, A.; Mosher, L.; Higgins, B. (2003). Calibration of an aftermarket EFI conversion system for increased performance and fuel economy with reduced emissions, *S.A.E. paper,* vol. 112, n°3, March 2003, pp. 1390-1407, 2003-01-1051.

Heywood,J. (1988) Internal combustion engine fundamentals, London : Mac Graw-Hill (1988)

Koehler J.R.; Owen A.B.(1996) Computer Experiments. In Ghosh, S., Rao, C.R.,(Eds.), Handbook of Statistics, 13 : Designs and Analysis of Experiments, North- Holland, Amsterdam, p.261-308. (1996)

Krige, D.G. (1951) A statistical approach to some basic mine valuation problems on the Witwatersrand, *J. of Chem. Metal. and Mining Soc. of South Africa*. Vol. 52 pp 119-139 (1951).

McKay M.D., Beckman R.J., Conover W.J. Comparison of three methods for selecting values input variables in the analysis of output from a computer code, *Technometrics*, Vol. 42, n°1, (February 2000) pp. 55 – 61, 239-245

Matheron, G. (1963) Principles of Geostatistics, *Economic Geology*, v. 58, n° 8, (December 1963) pp. 1246-12688.

Pierpont D. A.; Montgomery D. T.; Reitz R. D. Reducing particulate and NOx using multiple injection and EGR in a D.I. diesel, *S.A.E paper*, vol. 104, n°4 March(1995) , pp. 171-183 950217.

Pilley, A.D.; A.J.B.; Robinson, D.; Mowll, D. (1994) Design of experiments for optimization of engines to meet future emissions target, International Symposium on Advanced Transportation Applications (1994).

Sacks J., Schiller S.B., Welch W.J. (1989) Designs for Computer Experiments. *Technometrics*, vol. 31,41-47.

Schimmerling, P.; J.C.S. ; Zaidi, A. (1998) *Use of design of experiments*. Lavoisier.

Stein, M. Large sample properties of simulations using Latin hypercube sampling, *Technometrics*, vol. 29, n°2, (1987) pp. 143-151, 0040-1706.

4

An approach to obtain a PLC program from a DEVS model

Hyeong T. Park, Kil Y. Seong, Suraj Dangol,
Gi N. Wang and Sang C. Park
Department of Industrial Information & System Engineering, Ajou University
Republic of Korea

1. Introduction

To survive and prosper in the modern manufacturing era, a manufacturing company should be capable of adapting reduced life cycle of products in a continuously changing market place. Simulation is a useful tool for manufacturers to adapt this kind of rapidly changing market to design and analyze complex systems that are difficult to model analytically or mathematically (Choi, 2000). Manufacturers who are using simulation can reduce time to reach stable state of automated manufacturing process by utilizing statistics, finding bottlenecks, pointing out scheduling error etc. For the simulation of manufacturing systems, manufacturers have been using various simulation languages, simulation software for example ARENA, AutoMod. Most of traditional simulation languages and softwares focus on the representation of independent entity flows between processes; their method is commonly referenced to as a transaction-oriented approach. In this paper, we propose an object-oriented approach that is based on the set of object classes capable of modeling a behavior of existing system components.

The object-oriented modeling (OOM) is a modeling paradigm, that uses real world objects for modeling and builds language independent design organized around those objects (Rumbaugh, 1991). Even though OOM has been widely known to be an effective method for modeling complicated software systems, very few researchers tried to apply the OOM to design and simulate manufacturing system software models. Based on the OOM paradigm, different researchers have proposed various modeling approaches despite the fact that they express them in different ways with different notations. For example, Choi et al. presented the JR-net framework for modeling which is based on the OOM paradigm of Rumbaugh et al., which is made of three sub-models(an object model, functional model, and dynamic model). Chen and Lu proposed an object-oriented modeling methodology to model production systems in terms of the Petri-nets, the entity relationship diagram (ERD) and the IDEF0 (Chen, 1994). Virtual factory (VF) is also very important concept to be considered in today's simulation environment. By using the OOM paradigm, VF concept can be implemented efficiently (Onosato, 1993).

Recently, Park (Park, 2005) proposed a 'three-phase-modeling framework' for creating a virtual model for an automated manufacturing system. This paper employs the three-phase-

modeling framework of creating a virtual model, and the Discrete Event System Specification(DEVS) (Zeigler, 1984) for process modeling. The proposed virtual model consists of four types of objects. The virtual device model represents the static layout of devices. This can be decomposed into the shell and core, which encourages the reusability making possible to adapt different system configurations. For the fidelity of the virtual model, The Transfer handler model handles a set of device-level command that mimics the physical mechanism of a transfer. The Flow controller model decides the firable transfers based on decision variables that are determined by the State manager model. The State manager model and Flow controller model can be converted to PLC part. After finishing the process modeling by employing the three-phase-modeling framework, those two models will be the control information for the converting to PLC.
The overall structure of the paper is as follows. Section 2 represents the brief explanation about the PLC, and Section 3 is about the DEVS. The overall approach to create manufacturing system model for generation PLC code is described in Section 4. Section 5 gives as example cell, which is observed to find correlation between the PLC code and the DEVS model in Section 6. Finally, Conclusion and discussion is addressed in Section 7.

2. Programmable Logic Controller(PLC)

The Programmable Logic Controller (PLC) is an industrial computer used to control automated processes in manufacturing (Parr, 1999). PLC is designed for multiple inputs and outputs arrangements, it detects process state data through the sensing devices such as limit sensors, proximity sensors or signals from the robots executes logics in its memory and triggers the next command through the actuator such as motor, solenoid valve or command signal for the robots etc. PLC executes the control logic programmed in different types of languages. IEC published IEC 61131-3 to standardize PLC languages including Ladder diagram, Sequential Function Chart, Structured Text and Function Block Diagram (Maslar, 1996).

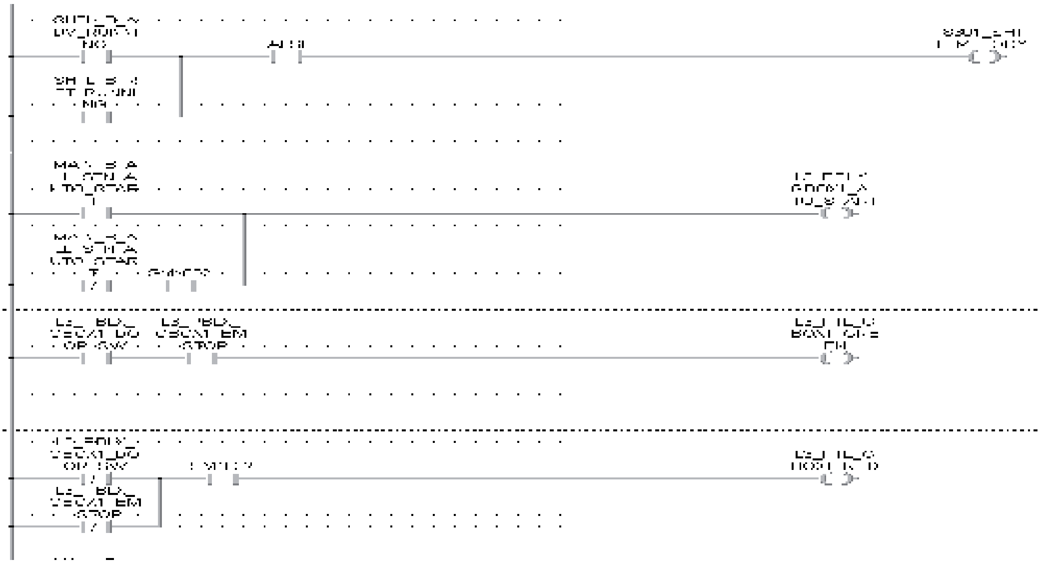


Fig. 1. The PLC code in the form of Ladder diagram

3. Discrete Event System Specification(DEVS)

DEVS formalism is introduced by Zeigler, which is a theoretic formalism and it supplies a means of modeling discrete event system in a modular, hierarchical way. With this DEVS formalism, we can perform modeling more easily and correctly by dividing large system into segment models and define the coupling between them. Formally, an atomic model **M** is specified by a 7-tuple:

$M = < X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a >$

X : input events set;

S : sequential states set;

Y : output events set;

δ_{int} : S→S : internal transition function;

δ_{ext}: Q x X → S : external transition function

Q = $\{ (s, e) \mid s \in S, \ 0 \leq e \leq t_a(s)\}$: total state of M;

λ: S->Y : output function;

t_a : S→$Real$: time advance function:

The second form of the model, called a coupled model, indicates how to couple several element models together to form a new and bigger model. Formally, a coupled model **DN** is defined as:

DN = < X, Y, M, EIC, EOC, IC, SELECT >

X : input events set;

Y : output events set;

M: set of all component models in DEVS;

EIC $\in$ DN.IN x M.IN : external input coupling relation;

EOC $\in$ M.OUT x DN.OUT : external output coupling relation;

IC $\in$ M.OUT x M.IN : internal coupling relation;
SELECT : 2^M - ø-> M : tie-breaking selector,
Where the extension .IN and .OUT represent the input ports set and the output ports set of each DEVS models.

4. Approach to create manufacturing system model to generate PLC code

To construct the automated process, the factory designers have to consider the overall process layout. After deciding skeletal layout, the process cycle time is simulated by the discrete event system software like ARENA or AutoMod. In this stage, including the process cycle time and production capability, the physical validity and efficiency of co-working machines are also described. Simulation and modeling software QUEST or IGRIP are used for this purpose (Breuss, 2005).

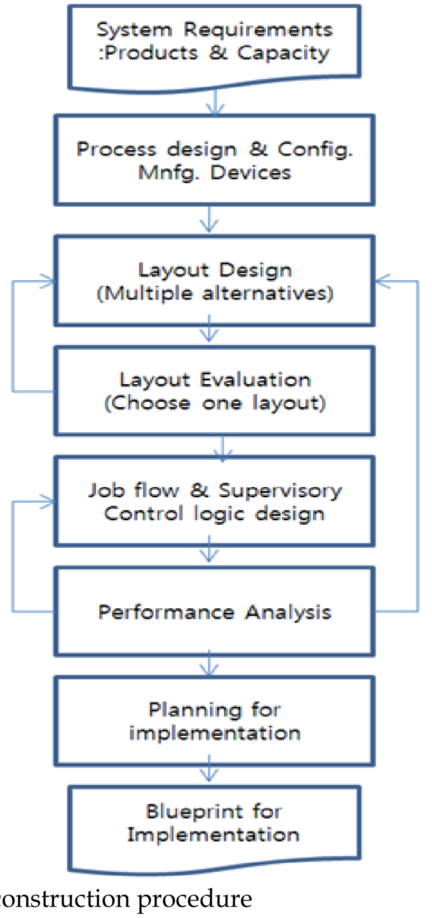


Fig. 2. Automated factory construction procedure

On the next step, the PLC code programming for logical functioning is done without utilizing information from previous discrete event systems modeling. The gap between the high level simulation of discrete event system and the low level physical process control logic need to be bridged for the utilization of process modeling and practical simulation in terms of physical automated device movement. This paper tries to find the bridge between these two different simulation levels and further describes automatic generation of PLC code from the DEVS model.

In developing the DEVS model, the first thing we have to do is to model the manufacturing system by the three-phase-modeling framework (Park, 2005). The framework describes manufacturing system modeling with 4 components; the Virtual device model, the Transfer handler model, the State manager model and the Flow controller model as shown in Figure 3.

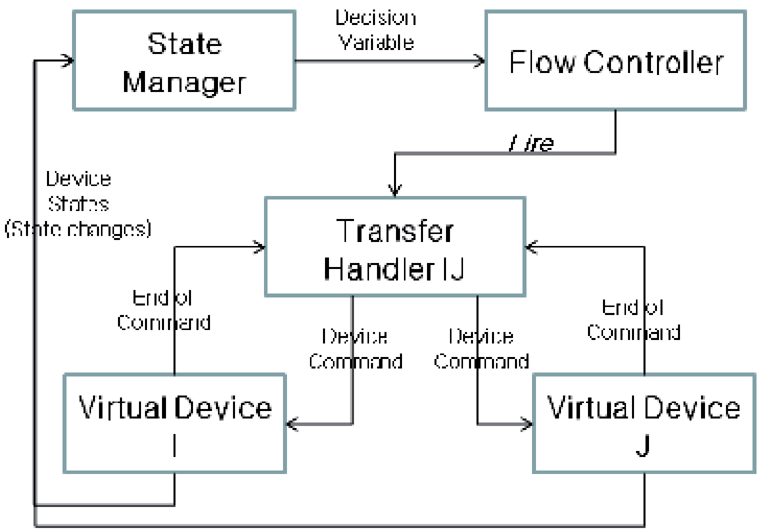


Fig. 3. Outline of the virtual manufacturing model

The Virtual device model shows the manufacturing devices. It has input port to receive the action signal and output port to send the work done signal. The Transfer handler model handles the parts stream and assisting resources (tools and pallets) between devices. This approach focused on the physical mechanism enabling the transfer than conventional approaches. In reality, a transfer happens by the combination of device-level command between co-working devices (giving and taking devices). The State manager model collects the state data of every device. Whenever there is a state change of devices, it will update the device states. Then, this information will be delivered to the Flow controller model as a decision variable. After getting the state information from the State manager model, the Flow controller model will decide firable transfer based on the system state (decision variables).

For the implementation of the virtual manufacturing system model, this paper employs the Discrete Event Systems Specification (DEVS) formalism, which supports the specification of discrete event models in a hierarchical modular manner. The formalism is highly compatible with OOM for simulation. Under the DEVS formalism, we need to specify two types of sub-models: (1) the atomic model, the basic models, from which larger ones are built and (2) the coupled model, how atomic models are related in a hierarchical manner.

When the DEVS model is developed, both the State manager atomic model for the process monitoring and the Flow controller atomic model for the actual control can be replaced the PLC part. Namely, control part for the manufacturing cell. Here is the goal of this paper.

5. DEVS modelling of a simple cell based on the three-phase-modeling framework

In this Chapter, we will observe a small work cell example. The work cell is modeled according to the three-phase-modeling framework and converted to the DEVS model like mentioned above. Finally, we will compare the DEVS model and the PLC code to find some meaningful bridge.
Figure 4 shows the small cell example. At first, an entity is generated from the Stack, which will lay on the AGV machine in P1, then AGV senses this raw part and moves to the P2 for machining. When machine detects the part arrival by the AGV, the machine starts to operate.

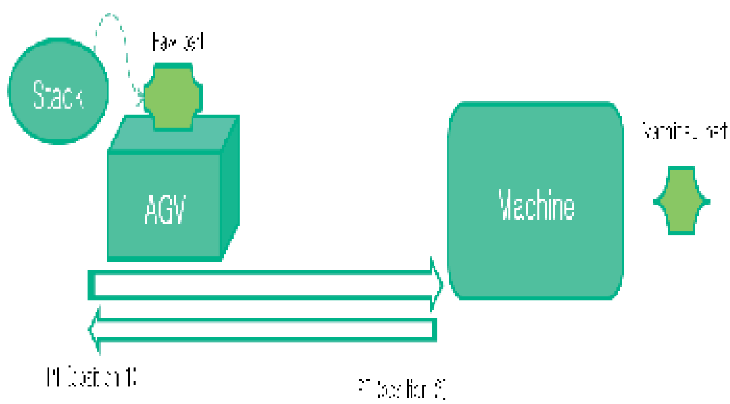


Fig. 4. Example cell

When we consider this example cell in terms of the three-phase-modeling framework, there are three virtual device models; the stack model, the AGV model and the machine model. The stack model generates the raw part entity and places it on the AGV for transfer. Until this point, the entity transfer process is between the stack and the AGV virtual device model as a result the transfer handler model is created between the stack the AGV model. Similarly, entity transferring between the AGV model and the Machine happens. This transfer handling model can be represented as *THam.* If there is any state change among the virtual devices, the changes are supposed to be reported to the State manager model. The State manager model maintains the decision variables in compliance with the reported state changes of the virtual devices and the Flow controller model will make a decision on firable transfer based on the decision variables. Figure 5 represents the constructed model about the example cell.

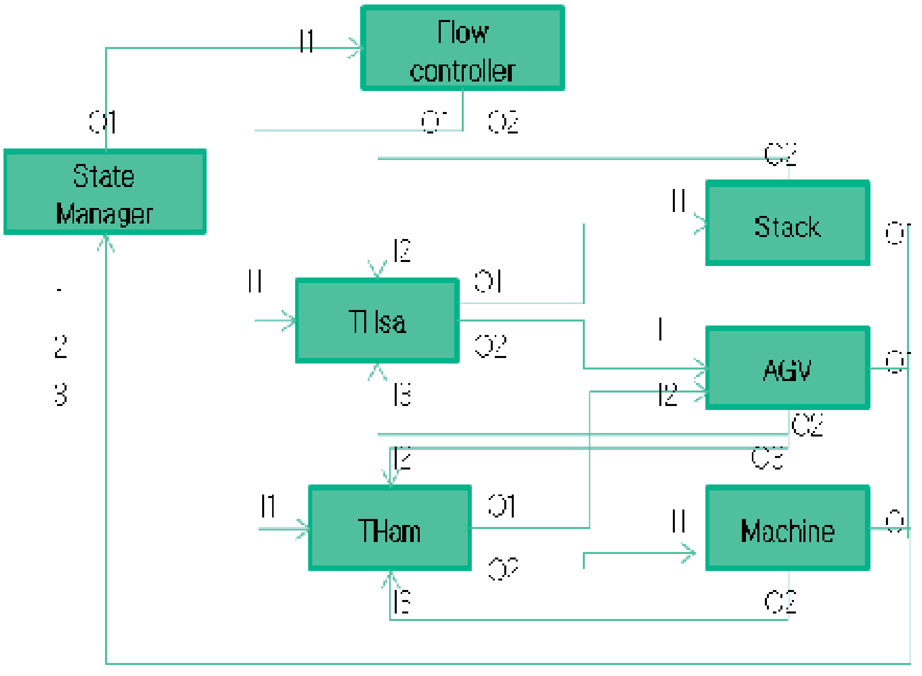


Fig. 5. Modeling of the example cell in the Park's methodology

Once the modeling by means of the three-phase-modeling framework is finished, second step is to convert the model to the DEVS formalism. In this example, every model is converted to the atomic model and entire cell will be the coupled model that is consist of all atomic models. Figure 6 is the converted DEVS model example of AGV. In the traditional implementation of discrete event system simulation using DEVS, DEVSIM++ is a simulation framework which realizes the DEVS formalism for modeling and related abstract simulator concepts for simulation, all in C++ (Kim, 1994). Through this open source frame, we can develop the discrete event system simulation engine easily. Once, both the DEVS implementation and the simulation with PLC control logic is done, we can achieve the overall physical control simulator for automated process.

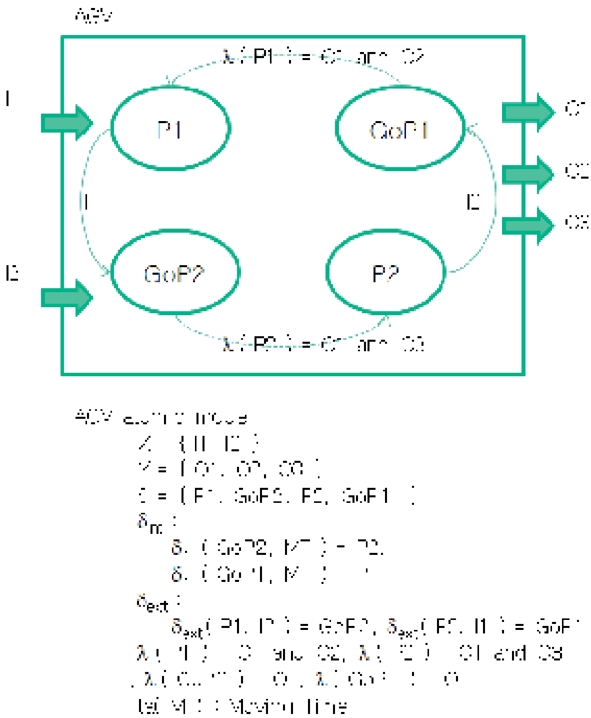


Fig. 6. DEVS model of the AGV

6. Correlation between the PLC code and the DEVS models

For the auto generation of PLC code from the DEVS model, we need to examine the PLC code of example cell and the DEVS models, especially the State manager and the Flow controller model.

In the manufacturing unit, PLC collects the process state information through the sensors. These sensor signals are referenced to decide next command or operation. This task is done by the state manager model in the modeled frame. The State manager model detects every change in state of the virtual device and then updates the decision variables. Similar to PLC code, the Flow controller model is supposed to have running logic that is kind of combination of decision variables. As a result, PLC code from the DEVS model can be divided into two parts. One part is for updating the decision variable from the signal of input port in the State manager model. Another is for actual logic composed of decision variables to fulfill the intended process control.

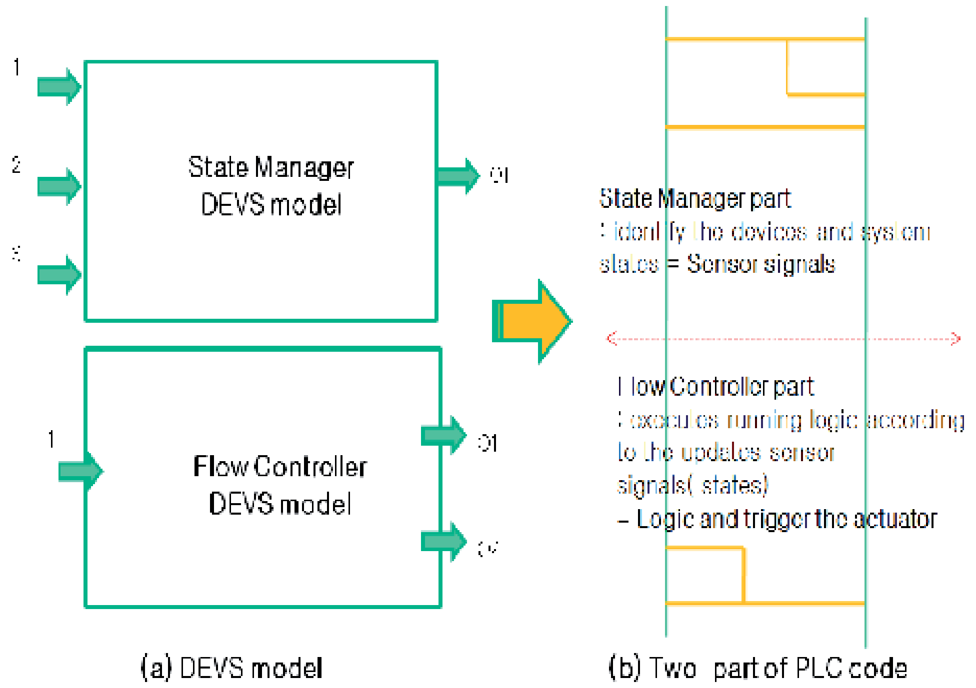


Fig. 7. Two part of PLC code

In the front part, the State manager model collects every state changes through the input port. The one input port of example cell has different kind of signal depend on the state. For example, the input port I2 is the signal from the AGV and it has 4 different kinds of state signals. With the same way, each input port of the State manager model has multiple input signals like shown in Table 1.

	Atomic models	States	Input Signals
I1	Stack	Idle, Release	STACK_IDLE STACK_RELEASE
I2	AGV	P1, GoP2, P2, GoP1	AGV_P1 AGV_GOP2 AGV_P2 AGV_GOP1
I3	Machine	Idle, Run	MACHINE_IDLE MACHINE_RUN

Table 1. The States of Atomic models

The memory structure in the PLC code can be classified into three groups. The first group is input memory which consists of input signal names and the second group is the output memory consisting output signal names and the last is the internal memory which is used to maintain the signal information of input or output and for temporary numerical calculation.

The name of input signal can be determined with combination between the input port and its state name. In this way, we can give a name to all input signals.
As mentioned before, the flow controller model reads the decision variables to execute next command. Thus, we have to make decision variables representing the process state as the internal memory. As we did in the input variable for naming, we can give decision variables' name by putting the 'On' between the port name and the state name. Then, this decision variable shows the port's current state is active condition. Once decision variables are set, the Flow controller detects the firable output signals from the set variables. Figure 8 show the decision variables of each input of AGV model and moving condition. To the AGV, the possible condition to move from P1 to P2 is when the raw part is on the AGV, AGV's state is 'GoP2', and the machine state is 'Idle' at the same time.

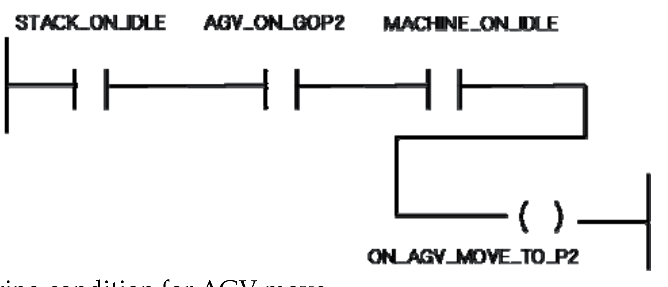
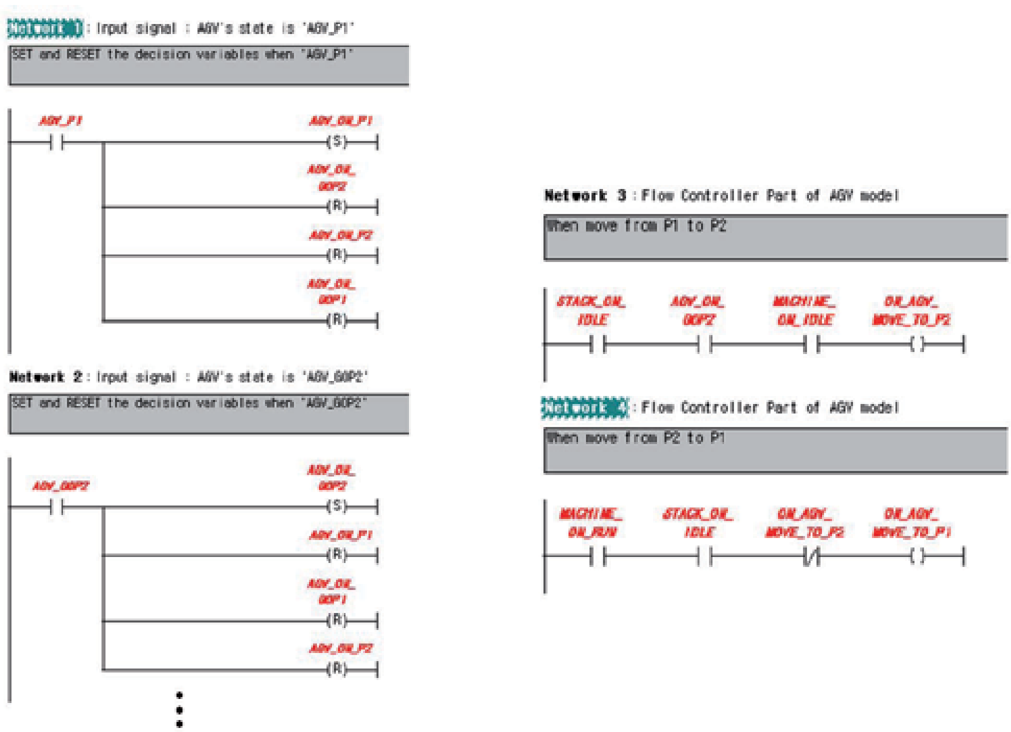


Fig. 8. The triggering condition for AGV move

As we have noticed for the case of the AGV model, the other devices' executing condition can be derived. While the PLC code for the State manager model part can be generated automatically with a combination of decision variables, the flow controller part is sometimes rather ambiguous. That is because unlike the flow controller, DEVS model is quite abstract and high level, the PLC part is very specific control area. Even though, process system designer can construct the DEVS model including low level of PLC, normally DEVS modeling is not fulfilled in this way. This aspect will be limitation or designer's choice in reference to PLC code auto generation. The DEVS modeling here is done specifically in mind of the PLC code generation of the Flow Controller model part. Figure 9 illustrates the two part of PLC code about the AGV from the State manager and the Flow controller model. And the Flow controller DEVS model for PLC code auto generation with the simple work cell is shown in Fig. 10.

7. Discussion and conclusion

This paper presents the PLC code auto generation methodology from the DEVS model. The PLC level control logic is rather closed and unopened engineering area while discrete event system modeling and simulation is widely used to measure the process capacity. By using the discrete event system simulation technique, the process or overall cycle time and throughput can be calculated.



(a) PLC Code from the State Manager model (b) PLC Code from the Flow Controller model

Fig. 9. PLC code from the State Manager and the Flow Controller model

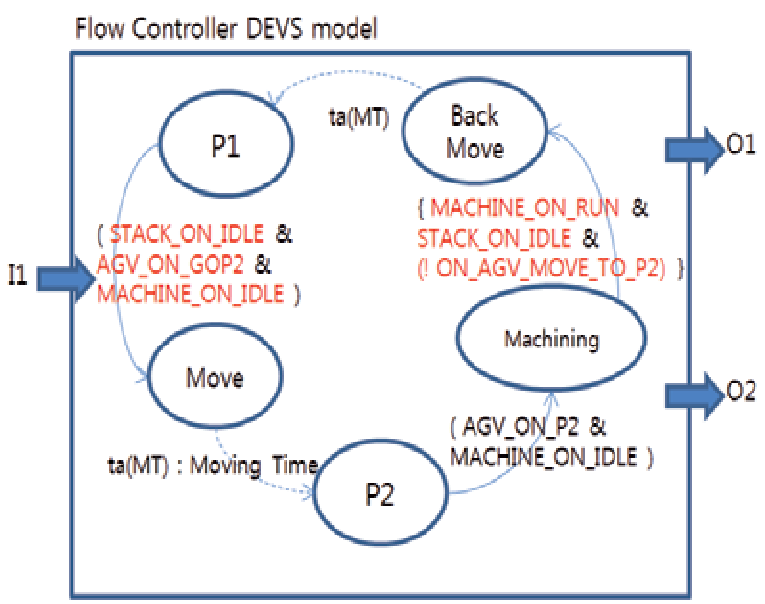


Fig. 10. The Flow Controller DEVS model

However, there is a big gap between the PLC code and the discrete event system simulation. This gap causes the repetition of process analysis work for the PLC programmer and the time delay to implement automated processing system in a manufacturing unit.
The overall procedure for proposed approach has three steps. Modeling the real system according to the three-phase-modeling framework is first step. And this model is converted to the DEVS formalism in second step. Among the 4 kind of models, the State manger and the Flow controller model is going to be replaced to the PLC part.
The generated PLC code from our approach can be categorized into two parts, one is from the state manager and another is from the flow controller. The first part is created from the input signals and the decision variable. And the latter part is from the control part which is from combination of decision variables.
The latter part generation is not achieved perfectly because the DEVS modeling level is more abstracted than the PLC level. However, this approach offers the overall framework for the PLC code generation from DEVS model. In the following future, the direction mentioned above will be the inevitable stream for the more physical process simulation, for the time saving toward the mass production condition and for better competitiveness to the company.

8. References

B. K. Choi, B. H. Kim, 2000. Paper templates, In *Current Advances in Mechanical Design and Production Seventh Cairo University International MDP Conference.* New Trend in CIM: virtual manufacturing systems for next generation manufacturing.
J. Rumbaugh, M. Blaha, W. Premerlani. 1991. Paper templates, In *Prentice Hall Inc*. Object-Oriented Modeling and Design.
B. K. Choi, H. Kwan, T. Y. Park, 1996. Paper templates, In *The International journal of Flexible Manufacturing Systems.* Object-Oriendted graphical modelling of FMSs.
K. Y. Chen, S. S. Lu, 1997. Paper templates, In *International journal of Computer Integrated Manufacturing.* A Petri-net and entity-relationship diagram based object oriented design method for manufacturing systems control.
M. Onosato, K. Iwata, 1993. Paper templates, In *CIRP*. Development of a virtual manufacturing system by integrating product models and factory models.
Sang C. Park, 2005. Paper templates, In *Computers in Industry*. A methodology for creating a virtual model for a flexible manufacturing system.
B. P. Zeigler, 1984. Paper templates, In *Academic Press*. Multifacetted Modeling and Discrete Event Simulation.
E. A. Parr, 1999. *The book*, Programmable Controllers : An Engineer's Guide 3rd ed.
M. Maslar, 1996. Paper templates, In *IEEE Pulp and Paper Industry Technical Conference*. PLC standard programming language: IEC61131-3
F. Breuss, W. Roeger, 2005. Paper templates, In Journal of Policy Modeling. The SGP fiscal rule in the case of sluggish growth: Simulations with the QUEST
T. G. Kim, 1994. *The Book*. DEVS++ User's Manual

A framework for simulating home control networks

Rafael J. Valdivieso-Sarabia, Jorge Azorín-López,
Andrés Fuster-Guilló and Juan M. García-Chamizo
University of Alicante
Spain

1. Introduction

Trasgu[1] is a control networks design environment valid for digital home or other places. The introduction of services provided by information society technologies, especially control networks, is growing at business, buildings, houses… There are a high number of protocols and technologies available in control networks. The set of control technologies that makes the applications viable are diverse and each follows his own rules. For example, there are different standard for control technologies like X10 (Fuster & Azorín, 2005), KNX/EIB (Haenselmann et al., 2007), LonWorks (Ming et al., 2007) , CAN (Jung et al., 2005), Zigbee (Pan & Tseng, 2007), etc and owned technologies like Bticino, Vantage, X2D,... In spite of standardization attempt, the design and implementation of control facilities is complex. Every technology presents a few limitations. We find among them the own configuration tool. It is proprietary software that allows the network design and configuration. Proprietary software, provided by the supplier, is the main design and configuration tool for control networks. Tools used by technologies considered as automation networks standard are: European Installation Bus Tool Software (ETS) for Konnex (KNX), and LonMaker Integration Tool (LonMaker) for Lonworks. Both tools have the same purpose, but they have different design and configuration methodology. A design realized with any tool is not compatible with other one and in many cases they cannot be linked. This has repercussions on increase of time and cost when the design needs several technologies that must coexist to offer a higher service. Technology choice depends of user requirements, because it might not be solved as well with all technologies. Even there might be project whose requirements are unsolved with a single technology, so we will need some technologies integration. In many situations, it turns out complex to integrate them in a common system. In spite of it, we need to communicate them to provide higher services, even if they belong to different networks: control networks, information networks, multimedia networks and security networks. There are residential gateways based on middleware and discovery protocols to make integration task easier.

---

[1] Trasgu is Asturian (Spanish) mythology goblin. It realizes household chores, but if it gets angry, he will break and hide objects, he will shout, etc.

Residential gateway middleware is connectivity software that allows communication among different technologies, in this case control technologies. Common middleware in ICT are: CORBA, J2EE y .Net. J2EE and .Net are the most used, the first one is based on Java and the second one is based on different Microsoft programming languages: C#, Java#, Visual Basic .Net,...

Discovery protocols facilitate devices connection to networks and services negotiation: Universal Plug and Play (UPnP) (Rhee et al., 2004), Jini Network Technology (Sun, 1999), Home Audio/Video Interoperability (Havi), Open Service Gateway initiative (OSGi) (Kawamura & Maeomichi, 2004), Services and Network for Domotics Applications (SENDA) (Moya & López, 2002)... UPnP is an open architecture distributed on a network, which is independent from technology. The goal is to get an unattended connection among different devices technology. Jini gets infrastructure to federate services in a distributed system. It is based on Java programming language. Jini has not been successful because there are not many devices supporting Jini. Havi is architecture oriented to electrodomestic appliance. The objective is to get a services set to make interoperability and distribuited networks development in home easier. OSGi defines a open and scalable framework to execute services in a safe way. The goal is to join devices in a heterogeneus network at aplication layer in order to compone services. SENDA framework is a device networking that uses CORBA. The aim of SENDA is similar to OSGi philosophy although SENDA is trying to improve it. There are some technologies to integrate control network, but none of them is being clearly succesful in all contexts, although some of them have a little market share.

Independently from integration there is another problem in actual control networks. Once we have designed a control network according to user requiments, we must realize the network installation to validate the correct operation. This fact introduces high temporal and economical costs in the installation, because if designer detects a fault, he should to solve it in the real facilities. This situation can be avoided through a simulation task after designing and before network installing, but control networks owned tools do not allow realizing simulations. They manage to realize validations to low level. For example, ETS realizes validations with physical and group addresses assigned to the devices. There is a tool associated with the ETS called EIB Interworking Test Tool, EITT. It is specialized in analysis of devices, moreover offers the possibility of simulating the network protocols. In spite of the low level validations that these tools realize, none of them manages to realize a simulation of the control network behaviour. A consequence is that designer cannot be able to verify the correct functioning, until control network has been implemented and installed.

Simulation brings advantages in the design of control installation. Simulation as a tool in the development techno-scientist allows detect errors prematurely in the design phase (Denning, 1989). It is able to verify and validate the control network design (Balci, 1998) in order to reduce costs. Simulation has the same place in design phase, that testing at implementation phase, since in both cases checks are made on the work performed. Therefore simulation is seen as a test case (Norton & Suppe, 2001), where checks are made to a higher level of abstraction. Besides the professional tools, in the literature there are some control network simulators like VINT project (Breslau, 2000), DOMOSIM (Bravo et al., 2000), (Bravo et al., 2006), VISIR (González et al. 2001) and (Conte, 2007). The VINT project presents a common simulator containing a large set of networks model. It is composed by a simulator base and a visualization tool. It is focused for simulation of TCP/IP network

protocols for research in TCP behaviour, multicast transport, multimedia, protocols response to topology changes, application level protocols, etc.
DOMOSIM and VISIR are orientated to educational area, so his principal use is teaching methodology of design of facilities. The negative aspect is the disability to join with tools of design that are used in the professional environment. (Conte, 2007) presents a study of home automation systems through simulation/emulation environment. It modelling home automation system like agents, where each agent action is characterized with cost, duration and quality. The agent behaviour is modelled with a set of states and a set of rules. The simulator is a software environment and agents are implemented in LabView and LabWindows CVI and are executed at same time. To run a simulation, user has to define the virtual environment and execute it. This is a powerful simulation environment but is not integrated in designing environment, so we must design the control network in the owned tool and later we must re-design at simulation environment.
Control networks carry a high temporary and economic cost. High cost due to three factors principally: Requirements of integration different technologies, inappropriate designing methodologies and design validation by means of experimentation. The proposal gathered in this chapter is to provide an environment to provide control network architectures in the digital home. The objective is that these architectures can be valid for any technology, paying special attention to network simulation task.

2. Modelling control systems

Modelling systems are based principally in two types of methodologies: bottom-up methodologies and top-down methodologies (Sommerville, 2004).
The technologies and methods used for the modelling of home automated systems are few developed. They are based on the use of very low level technologies (Muñoz et al., 2004). The great diversity of control technologies causes that designing control networks following bottom-up methodologies might turn out a complex task and might generate systems inadequate to requirements.
The top-down methodologies are characterized essentially abstract. They require that ingenuities are conceived before any consideration. Ingenuities are conceived, therefore, free from any condition imposed by the technologies. These methodologies match perfectly with our philosophy of design independent implementation technologies.
Model Driven Architecture (MDA) (Mellor et al., 2004) and Services Oriented Architectures (SOA) (Newcomer & Lomow, 2005) are within independent implementation philosophy. In one hand MDA allows transitions from conceptual models of systems using automatic code generation is possible to obtain fast and efficient middleware solutions. In the other hand SOA provides a methodology and framework for documenting business skills and power to support the consolidation and integration activities. The network nodes make its resources available to other participants in the network as independent services that have access to a standardized way. In contrast to object-oriented architectures, SOAs are formed by application services weakly coupled and highly interoperable. The communication between these services is based on a formal definition platform independent and programming language. The interface definition's encapsulates the peculiarities of an implementation, which makes it independent of the technology, the programming language or developer.
Designing control networks, with this methodology, require answer the following questions:

What functionalities I want to offer? E.g. for following functionalities: safety, comfort and automation, we have to provide the following services: intrusion detection, access control, alarms, lighting and temperature control.
How should behave services? That is, how relate them. In the case of the above examples: we should indicate how want to regulate temperature or the relationships with other devices: intrusion detection, alarms.
What technologies I'm going to use for implement the system? Once we have clear functionalities that network is going to offer and the behaviour, we have to decide among available technologies to choose the best for this situation.
We propose a model based on the three above questions. The model consists in three layers called: functional, structural and technological.
Functional is the most abstraction layer. It describes installation functionalities. It avoids thinking about how to do this and technology implementation. So we got translate user requirements to the functionalities that the installation will provide. In this layer, the control installation, *CI*, is defined by a set of services, *Si*, which are demanded by users:

$$CI = \{S_1,S_2,...,S_n\} \tag{1}$$

Each service, *Si*, needs to satisfy a set of tasks, *ti:*

$$S_i = \{t_{i1},t_{i2},...,t_{in}\} \tag{2}$$

The next level of abstraction is called structural. It is focused on the structure and behaviour of the generic devices. Since the structural layer, the control installation, *CI*, is composed of a set of generic resources, *Rs*, and a wide range of connections, *C*, which are established between resources:

$$\begin{aligned} CI &= CI(Rs,C), \\ Rs &= \{Rs_1, Rs_2,...,Rs_n\}, \\ C &= \{C_1,C_2,...,C_m\} \end{aligned} \tag{3}$$

A resource represents the entity that may be physical or logical and provides some tasks, *tij*. At this level, both the resources and connections are independent of technology, because it only represents the installation structure. Resources are represented as a set of tasks, *tij*, offered to other entities:

$$RS_i = RS_i\{t_{i1},t_{i2},...,t_{im}\} \tag{4}$$

Connections are represented as associations between two resources. They are formed by an input resource, *Rsi*, and an output resource, *Rso*:

$$C_i = C_i(Rs_i,Rs_o) \tag{5}$$

The lower abstraction layer is technological. It is an instance of structural layer using a specific technology. It takes into account more realistic aspects of the implementation. Resources viewed from the technological layer are defined by the set of tasks, *Ti,* and by his set of physical characteristics, *CAi*, like position, size, weight, etc:

$$\begin{aligned} R_i &= R_i(T_i, CA_i), \\ CA_i &= \{ca_{i1},ca_{i2},...,ca_{ip}\} \end{aligned} \tag{6}$$

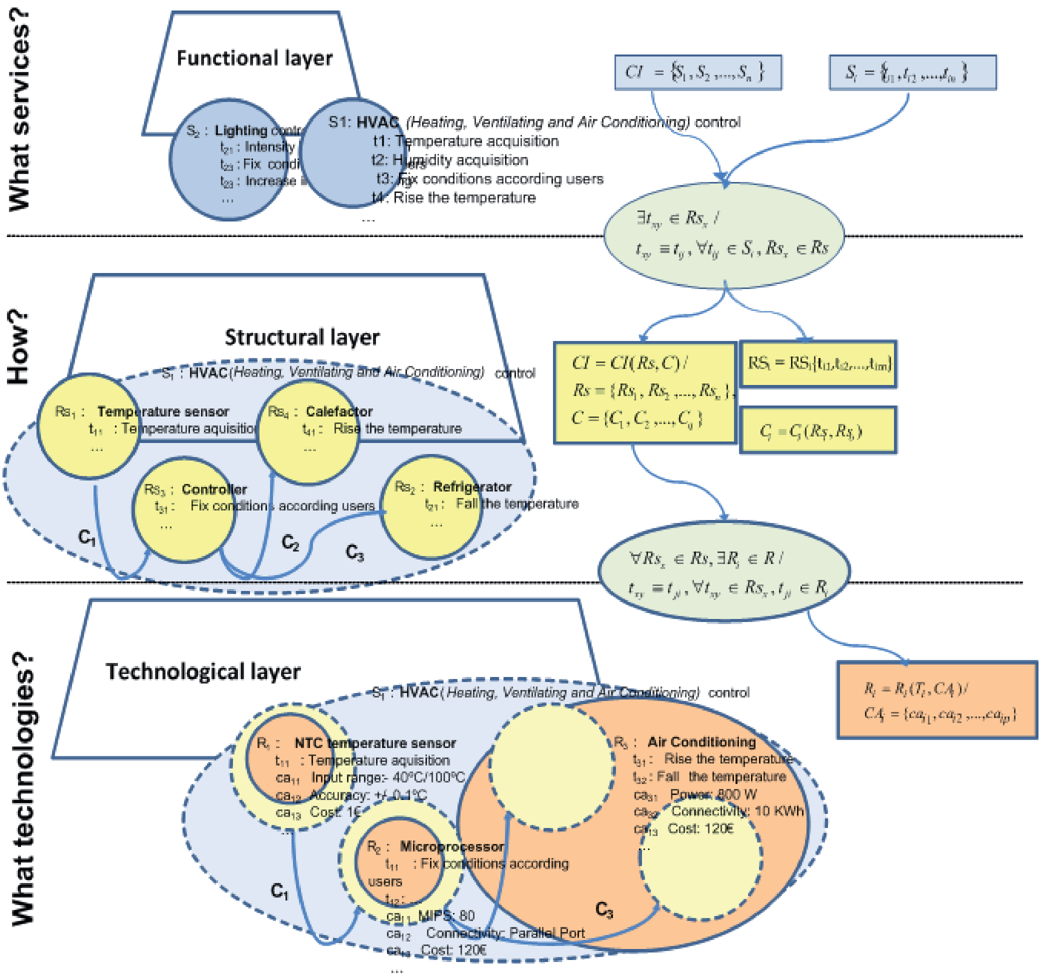


Fig. 1. Correspondence between layers and model equations

At this layer, resources, *Ri*, tasks, *Ti*, and characteristics, *CAi*, are determined by implementation technologies. This representation reflects reality faithfully.

As the level of abstraction is reduced, it is necessary to make transitions from functional to structural and, finally to technological level. To achieve the first transition, services should have all their tasks paired with some of tasks provided by resources of the technological level. Therefore task, *txy*, of any resource, *Rsx*, should be equivalent to task, tij, required by service *Si*. This must be satisfied for all tasks, *tij*, required by the service, *Si*:

$$\exists\ t_{xy} \in Rs_x \ /\ t_{xy} \equiv t_{ij},\ \forall\ t_{ij} \in S_i,\ Rs_x \in Rs \tag{7}$$

Structural to technological level transition needs to match generic resources *Rs* with technological resources *Ri*. Therefore all tasks, *txy*, at all generic resources, *Rsx*, should be matched with task, *tij*, of technological resources *Ri*. This matching must be between equivalent tasks:

$$\forall\ Rs_x \in Rs,\ \exists R_i \in R\ /\ t_{xy} \equiv t_{ji},\ \forall t_{xy} \in Rs_x,\ t_{ji} \in R_i \tag{8}$$

When these transitions have been performed successfully, the control network will be defined in three abstraction levels: functional, structural and technological. Figure 1 shows the correspondence between layers and model equations using a Heating Ventilating and Air Conditioning (HVAC) service as example. It defines the functional level a set of tasks like: temperature acquisition, humidity acquisition, fixing conditions according users, rises the temperature, falls the temperature etc. The structural level is defined by resources that implements the previous tasks like: temperature sensor, which implements temperature acquisition, controller, which implements fixing conditions according users, calefactory which rises the temperature and refrigerator which falls the temperature. Finally the technological level defines what technology implements the resources defined at the structural level. In this case a NTC temperature sensor has been chosen as temperature sensor. The temperature acquisition task and can be defined by characteristics like: input range, accuracy and cost. Microprocessor was chosen as controller. It has fixing conditions according user task and it is defined by characteristics like: MIPS, connectivity and cost. Finally, the air-conditioning implements two tasks: rise the temperature and fall the temperature and it is defined by power, connectivity, cost, etc.

3. Trasgu: Design environment

Trasgu is a prototyping environment based on the previously defined model. It can be used from three different abstraction levels: functional, structural and technological. The installation can be defined from a higher level of abstraction and choose specific implementation aspects in the last steps. It provides robustness design and reduction of development time, because transitions from the highest abstraction level to the lowest are progressive and simple. It facilitates finishing all tasks that are needed throughout the life cycle of control network. These tasks will be used by their respective actors who are involved in a control installation: designers, integrators, property developers, installation engineer...

The model has power enough for developing features that the environment offers. Environment provides the following features: designing a control installation independently any technology; perform simulations to verify and validate that our design meets users specifications; create architecture in agreement to the chosen technology for implementation; make a budget with each technology in order to determine the cost of each one; develop different types of reports, including a report showing the wiring installation. Nowadays tasks implemented by trasgu are: designing control network and simulation task. Figure 2 shows possible users for each environment feature, e. g. Prescriber will be in charge of designing the control network and simulate it in order to validate initial decisions, programmers will be in charge of modelling architecture with implementation technology and simulate it. On the other hand, promoters will be in charge of budgets and installer will be in charge of installation reports.

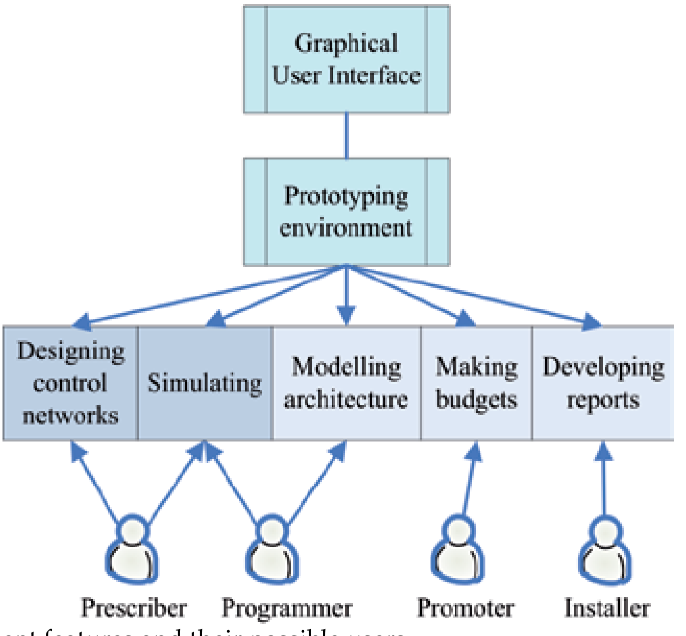


Fig. 2. Environment features and their possible users

The control network design is the first task. This task is characterized by adding some resources into a building plan. Resources have been classified into three types: input, Eq. (4), services, Eq. (2) and output, Eq. (4). First of all, services must have been added to the building floor. Then we have to add the input and output resources and connect with their corresponding services. All resources are connected to one or more services that are controlling their network behaviour. Representation based on services has been chosen to represent control network because this representation reflects that input and output resources can be used by several services. Figure 3 shows an example of three services: security, lighting and gas detection. The presence sensor and binary light are shared with security and lighting services. Gas detection service uses a gas sensor and acoustic alarm. And finally acoustic alarm is shared with security service.

The simulation task is based on design of the control network. Control network can be simulated specifying simulation parameters. This task probes that design is satisfied with specification.

The architecture modelling task allows generating the architecture that implements the design in a real installation. The architecture is middleware based (Valdivieso et al., 2007), (Fuster et al, 2005). This middleware provides communication among different control technologies and protocols.

The budget creation task allows generate budgets from the technological design. It will provide an estimate cost for the costumer. This task requires an automatic communication with the devices dealer for database maintenance.

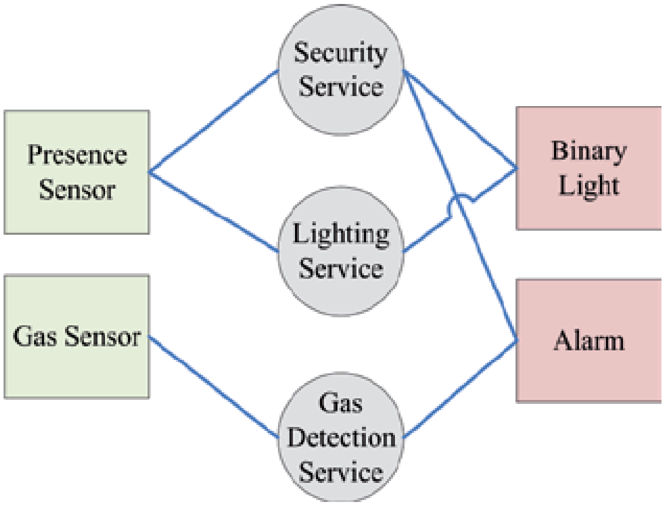


Fig. 3. Representation based on services

The reports development task allows generating some types of reports. Each of them is focused to show an installation feature like: wired connections, devices situation, device configuration, services…

Environment features can be used from different abstraction layers. Figure 4 shows relationships between environment features and abstraction layer, in which are implemented. The installation design could be used from functional, structural and technological layers. Installation simulation can be used from functional, structural and technological layers. Architecture modelling only supports the structural and technological layer. It is technological because it is implemented over real devices and it is structural because some architecture layers could be common. Making budgets is useful only from technological layer because the budgets are for customers. Instead reports can be created from the functional, structural and technological layers, because reports can be useful for different applications. The customer could need a services report, a developer could need a structural report and an installation engineer could need a wired reports.

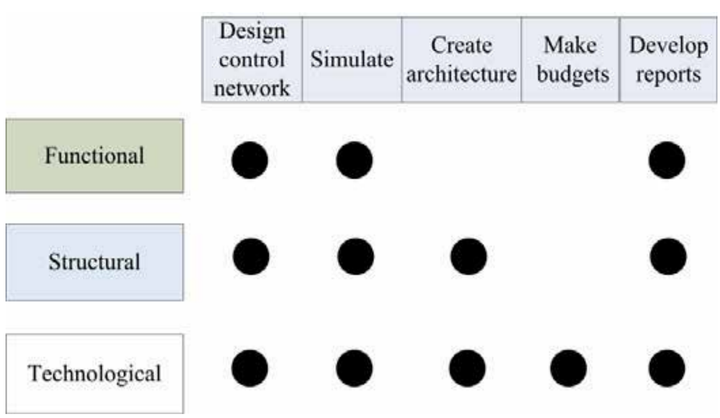


Fig. 4. Abstraction layers supported by each feature

3.1 Resources definition

Resources of functional level are classified according to input and output resources. At functional level resources are classified in order to determine subcategories inside input and output resources.

Input resources are also subdivided to continuous and discrete sensors. Each family sensor is associated to a single magnitude to measure. Continuous sensors are those which measures continuous magnitudes like temperature, humidity, lighting, etc. Discrete sensors are those which detect events like: gas escape, presence sensors, broken windows, etc. These sensors indicate that an event has happened.

Output resources are subdivided into two groups: continuous and discrete actuators. Continuous actuators are characterized because they feedback continuous sensors and discrete do not feedback sensors. The first type is e.g. HVAC device that modifies the magnitude measured previously by a continuous sensor. Second type is e.g. acoustic alarm which does not invoke any event. Continuous actuators interact and modify the magnitude measured by sensors. Continuous actuators are determined by the direct magnitude that they modify directly and indirect magnitude list that might be modifies in any way by the actuator, e.g. HVAC modifies directly temperature and indirectly humidity. Other aspect to be explained is the capacity of some actuators to modify the physical structure where they are installed, e.g. blind engine or window engine that modify the wall structure allowing or prevent lightening and rise or fall the temperature.

3.2 Implementation

The environment has been implemented using Java and uses an information system implemented based on XML. The XML files reflect information of resources definition that can be used in the installation and the information about the current network. This information is reflected from abstraction layers defined at the model. The resources reflected in the XML files represent a subset of the features offered by market. The environment could expand the information system adding new resources in XML files. Adding new resources is easy because we only have to define the XML files that define the new resource and interactions with magnitudes.

The prototyping environment presents four different areas, see figure 5. On the top of the window, we can see the services that can be added to the network. On the left side there are all the generic resources that can be used in the design. In the middle, there is the building floor map. In this area services and generic resources can be added for designing the control network. On the right side, there is hierarchical list of all the resources added in the network classified by resource type.

The control network design starts up opening a map of the building where user can drag and drop resources and services in order to design the network. The next action is adding services, input and output resources into the map, after doing that, user has to connect resources and services in order to establish relationships. Moreover, user has to configure parameters of input and output resources and, finally, user must determine the behaviour of each service through logical operators between input and output resources. It is not possible link inputs and outputs resources directly because it is necessary a service that link them. Figure 5 shows a control network designed at the environment. The control network contains two services: HVAC and technical security. HVAC is formed by a temperature sensor and an air-conditioning and heating. The technical security is composed by a gas sensor and an acoustic alarm. Although this design does not share any resource, it is possible share resources into some services as we have show at figure 3.

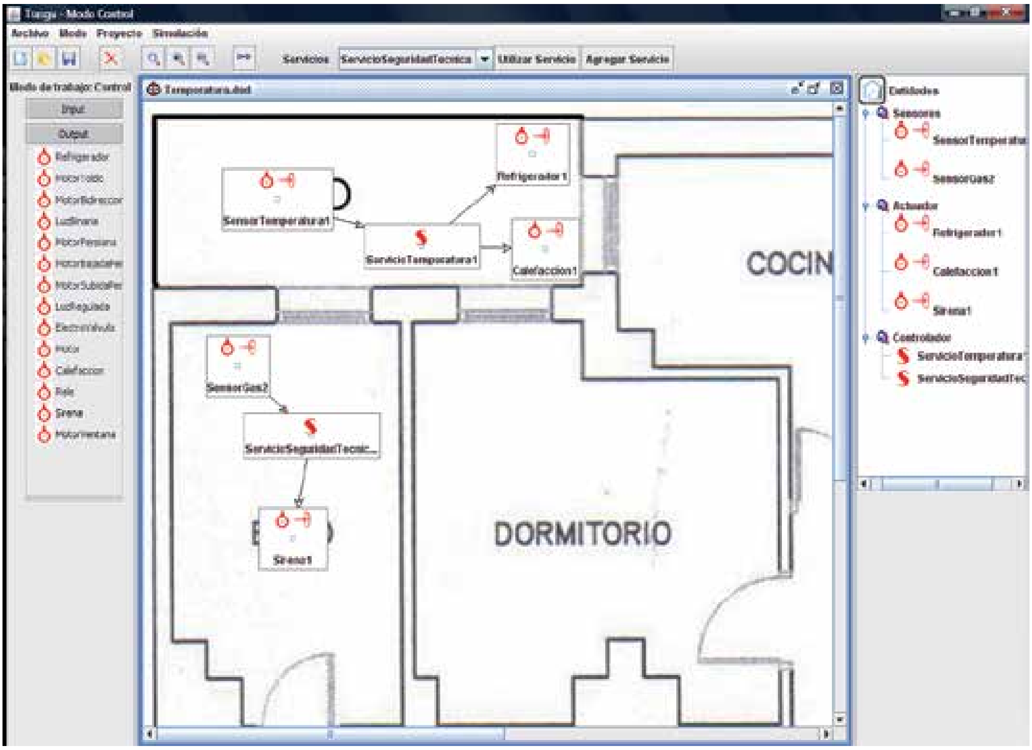


Fig. 5. Environment aspect and control network design. It contains HVAC and technical security services

Each service defines their behaviour based on some functions. These functions receive inputs from input resources and provide outputs to output resources. Functions are composed by logical, arithmetical or comparing operators. This specification contains inputs, and outputs that allows interconnect them using operators. Figure 6 shows services definition window. On the middle of the figure we can see inputs resources on the left and output resources on the right. Operators are situated on the left side of the window and it could be drag and drop into the previous area in order to link inputs and outputs. Operators should be configured in order to define the behaviour of the service. This figure contains the HVAC service behaviour. Refrigerator will be turned on when temperature rises above 25 Celsius degrees and it will be turned off when temperature falls under 25 Celsius degrees. Heating will be turn on when temperature reaches 15 Celsius degrees and it will be turn off when temperature rises above 15 Celsius degrees. According to this parameters temperature will be between 15 and 25 Celsius degrees.

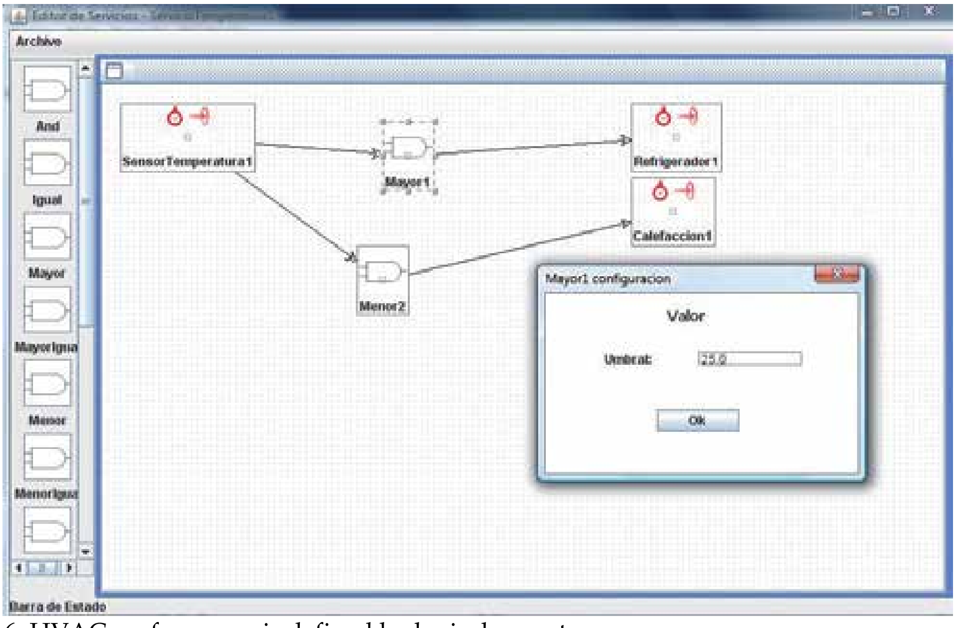


Fig. 6. HVAC performance is defined by logical operators

4. Simulation

The environment is designed to perform simulations from different abstraction levels. Environment simulates from functional layer Eq. (1), Eq. (2), structural layer Eq. (3), Eq. (4) and Eq. (5) and technological layer Eq. (6). The functional simulation is responsible for simulating the behaviour of generic control network. With this validation we can be sure that generic device configuration and connections are correct. The structural simulation includes the functional, but adds new features, like the real position of the resources and installation regulation. The technological simulation determines the real behaviour that the implemented control network is going to provide.

Functional layer simulating it is explained. This simulation is determined by two parameters: simulation time and frequency. The first one represents the interval we want to simulate. The second is the number of values from each resource that we will take in a time unit.

Simulation task needs a flag for continuous and discrete sensors. This action is carried out by event generators. We have designed two event generator types, discrete event generator and continuous event generator. The first type gives up data for discrete sensors, and is based on probability percentage, that is defined for each discrete sensor in the network. The second type represents input data for continuous sensors and is based on medium value and variation value throughout simulation time. Obtaining a sinusoidal wave which represents continuous magnitudes like temperature, humidity, etc...

The results are represented in two views. The first one shows building map, control network and resources values are drawn under their own resource. Resources values are changing at a given frequency in order to view the evolution of the control network in simulation time. This view is showed in figure 7.

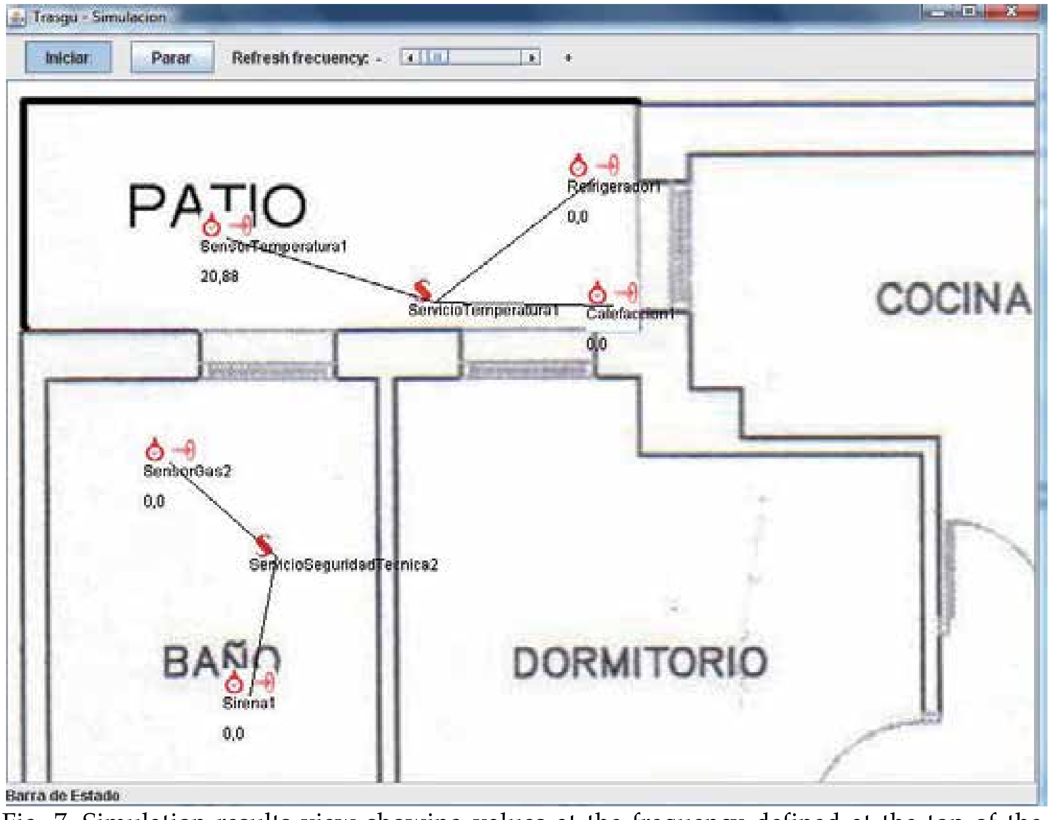


Fig. 7. Simulation results view showing values at the frequency defined at the top of the window

The second view represents a graph. A graph is showed for each resource in the previous view. The graph represents the resource behaviour in simulation time. There are three types of graphs.

The first graph type shows services. This graph shows all signals of the resources linked in the service, representing all the information in one graph, although sometimes data range are different so we have to view graphs individually at each resource. Thinking in our HVAC service example, the graph shows temperature sensor, air-conditioning and heating values. The second graph type represents sensors and shows event generators values and values given by the sensor. In HVAC example, the temperature sensor graph shows external temperature and the temperature sensor measured, which might have a little error. The third graph is reserved for actuators, and shows actuator values drawn as a single signal representing the actuator performance. In HVAC example, the air-conditioning and heating will be show in two graphs, each one shows a signal indicating when the air-conditioning is on or off. Figure 8 shows three graphs, temperature sensor at left, air-conditioning at right on top and heating at right on bottom. Temperature sensor graph shows the external temperature in blue, which is generated as a variation event, and sensor temperature in red. The red signal has peaks because the sensor has been defined with measure error. Moreover the red signal is ever between 25 and 15 Celsius degrees because the air-conditioning starts when temperature rises and the heating starts when temperature falls. Air-conditioning and heating graph shows when they start to work.

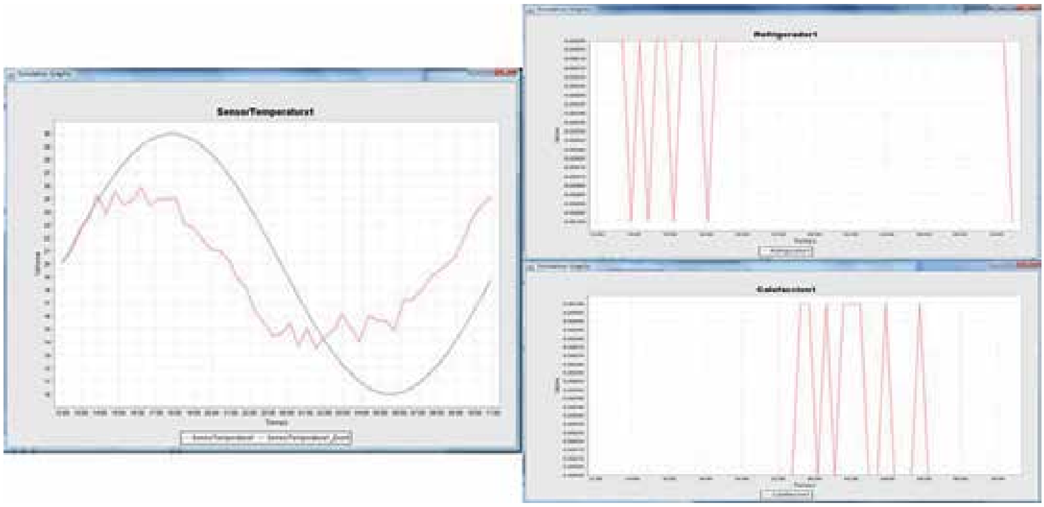


Fig. 8. Simulation results graph of temperature sensor, air-conditioning and heating

4.1 Implementation

Simulation is realized by an external module. It is implemented in external way to leave us introduce future changes in the simulation module esaily, without impact for the environment. It has been developed in Matlab / Simulink because allows high precision and efficiency.

The communication between the simulation module and environment has been done through a Java library called jMatLink (Müller & Waller, 1999). It is responsible for sending commands to Matlab. This action requires knowing Matlab commands perfectly. We have developed a library called jSCA. It is responsible for encapsulating Matlab commands to facilitate the communication. The features of jSCA are implemented as method which allows: create instances of Matlab; create, configure, and remove blocks in Simulink; configure parameters simulation, and obtain the results of the simulation. This library is a level above jMatLink. The environment make calls to jSCA library and this library calls to jMatlink library.

The architecture that achieves the communication between Java application and Simulink is showed at figure 9. The first three layers are inside the environment, but the latter two are external environment. The first layer is the own prototyping environment. The second one is the jSCA library. The third one is jMatlink library. The fourth is Matlab engine. It is a daemon that is listening commands thrown by jMatlink. We use Matlab Engine as middleware between jMatLink and Simulink. So we launch Simulink and throws commands through Matlab Engine.

The interaction between the user and the architecture layers defined previously is showed by a sequence diagram. This diagram shows temporal sequence of main calls. Figure 10 shows the sequence diagram, which corresponds to a specification language: Unified Modelling Language (UML). Temporal sequences of calls begin when user designs the control network. The next action is to simulate it. If user starts simulation trasgu will call jMatlink layer to createMDL file. This layer starts Matlab server and calls Matlab engine to create the mdl file. Finally, Matlab engine calls Simulink and it creates the mdl file. Once mdl has been created Matlab engine calls Simulink for start simulation. When simulation finish, simulation values are returned layer after layer until trasgu received it and then trasgu visualizes it.

Fig. 9. Architecture that achieves communication among Java application and Simulink

Functional simulation requires a parallel model from the functional layer in Simulink. This model is equivalent to resources XML files defined in the information system used by the environment. Each resource defined in the environment is corresponded by another defined in Simulink. The Simulink model consists on a block set formed by input blocks, controller blocks, output-blocks and input-output blocks. There are also two block types: discrete and continuous. The discrete blocks are dedicated to digital devices, and continuous are for analogical devices.

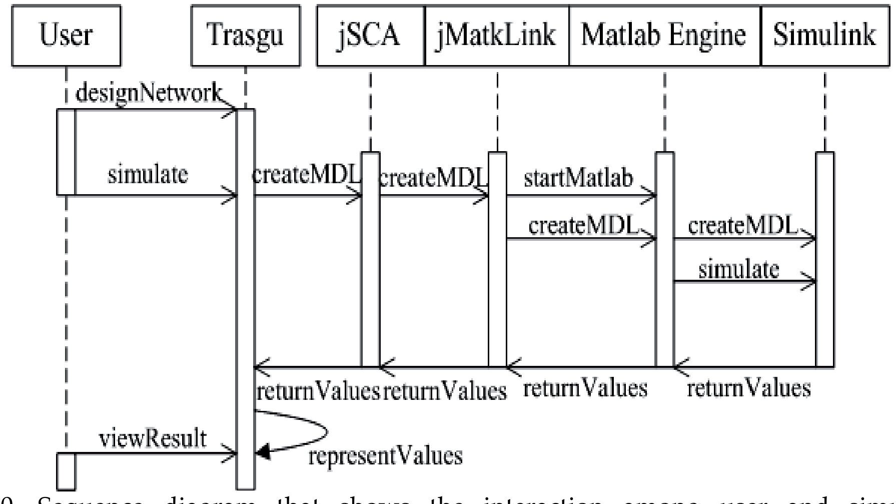


Fig. 10. Sequence diagram that shows the interaction among user and simulation architecture

The simulation starts creating an equivalent control network in Simulink dynamically. The Simulink control network is saved into mdl file. To create the mdl file, we add the corresponding network blocks. Parameters of each block are configured and connections among blocks are created. The next step is to add probability or variation events for each sensor. Through these events the input resources are activated and we can see how the network reacts. The last step is to feedback the signal from actuators to sensors, adding actuators outputs and original event, obtaining real value.
An example of this Simulink network is showed on figure 11. It shows, at section I, events generators and adding block that represent feedback. Each input resource has a correspondent event generator: discrete or continuous. Section II has input resources. Section III shows services which join input and output resources. Services definitions are made recursively by others mdl files. These mdl files are equivalent to the services defined at environment. At section IV we have output resources. Finally, section V shows feedback generated by actuators, which modify measured magnitudes. The feedback is adjusted in function of how we have configured the actuator at the environment. Each section shows a white box connected to each resource. These boxes store a value list obtained from simulation. This list contains discrete values taken at given frequency.

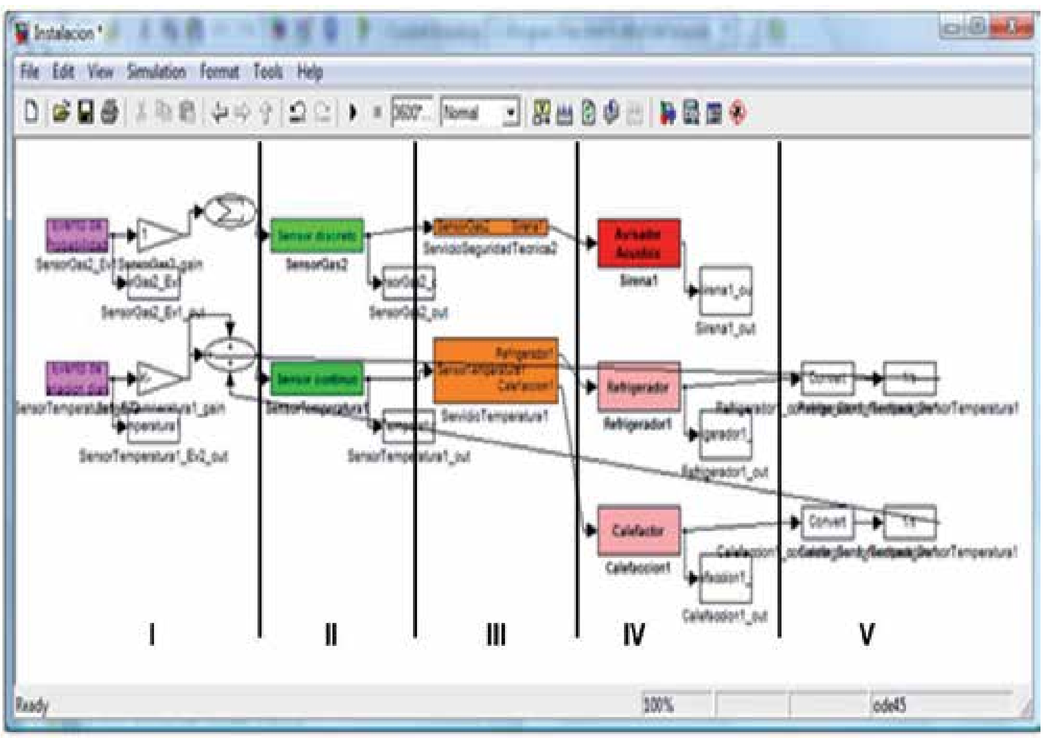


Fig. 11. Simulink control network generated dynamically from a previous design made at the environment

When the mdl file is created, Simulink executes the simulation and the environment reads all variable lists from Simulink and represents them at the given frequency in the view defined previously at environment. This way, the simulation is not interactive, when the simulation process is launched, the user has to wait until Simulink returns the results of the simulation.

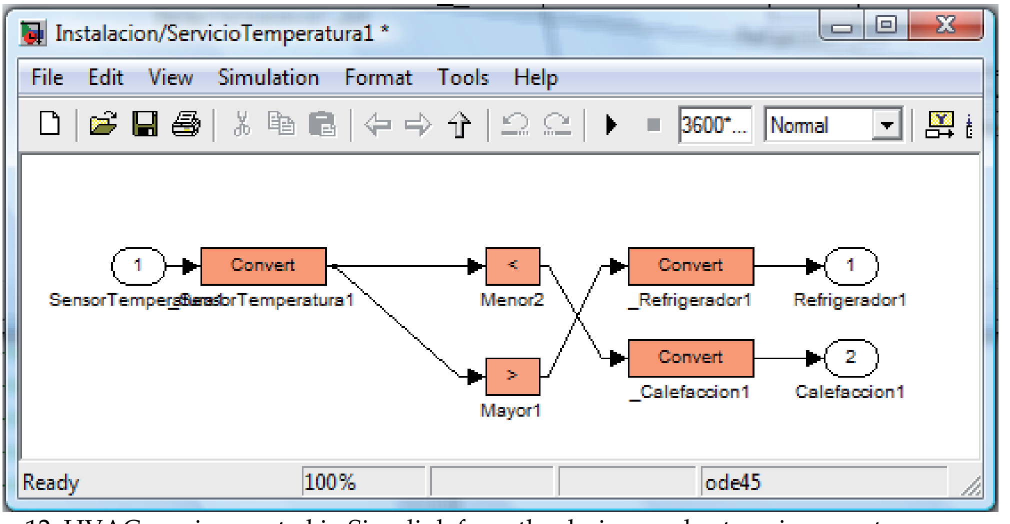


Fig. 12. HVAC service created in Simulink from the design made at environment

Figure 12 shows the mdl file that defines a HVAC service designed previously at environment on figure 6. At left side, input resources connection can be viewed as white circles, in this case temperature sensor. In the middle, we there are conversion data type blocks for avoiding Simulink data type errors and comparison operators that defines when the actuators will start to work. Actuators are drawn as white circle on the right.

5. Conclusions

This chapter presents a prototyping environment for the design and simulation of control networks in some areas: digital home, intelligent building. The objective is to facilitate the tasks of designing and validating control networks. We propose a top-down methodology, where technology implementation choice is left to the last phase of designing process. The environment kernel is the control network model around different abstraction layers: functional, structural and technological.

The model is the basis on which the environment gets applications by particularization: network design, simulation, specifying architectures, developing reports, budgets, and so on.

Simulation is presented as an intermediate phase in methodology of design control network. It reduces costs and helps us to design effective and efficient control networks.

Future work is aimed to provide competitive solutions in the design of control networks. The nature of the problem requires addressing increasingly scientific and technological aspects of the problem. Therefore, in short term, we deepening in aspects of generalization of the results: new models for structural and technological layers, an interactive simulation module and simulation of technological layer… In long term we are going to generalize to others context like industrial buildings, office buildings…

6. References

Balci, O., 1998. Verification, validation, and testing. In The Handbook of Simulation. John Wiley & Sons.

Bravo, C., Redondo, M.A., Bravo, J., and Ortega, M., 2000. DOMOSIMCOL: A Simulation Collaborative Environment for the Learning of Domotic Design. ACM SIGCSE Bulletin, Vol. 32, No. 2, 65-67.

Bravo, C., Redondo, M., Ortega, M.., and Verdejo, M.F., 2006. Collaborative environments for the learning of design: a model and a case study in Domotics. Computers & Education, Vol. 46, No. 2, 152-173.

Breslau, L. Estrin, D. Fall, K. Floyd, S. Heidemann, J. Helmy, A. Huang, P. McCanne, S. Varadhan, K. Ya Xu Haobo Yu. 2000. Advances in network simulation. Computer, Vol. 33, Issue: 5, pp: 59-67. ISSN: 0018-9162.

Conte, G. Scaradozzi, D. Perdon, A. Cesaretti, M. Morganti, G. 2007. A simulation environment for the analysis of home automation systems. Control & Automation, 2007. MED '07. Mediterranean Conference on, pp: 1-8, ISBN: 978-1-4244-1282-2

Denning, P. J., Comer, D. E., Gries, D., Michael Mulder, C., Tucker, A., Turner, A. J., Young, P. R., 1989. Computing as a discipline. Communications of the ACM, Vol.32 No.1, 9-23

Fuster, A. and Azorín, J., 2005. Arquitectura de Sistemas para el hogar digital. Hogar digital. El camino de la domótica a los ambientes inteligentesI Encuentro Interdisciplinar de Domótica 2005, pp 35-44

Fuster, A., de Miguel, G. and Azorín, J., 2005. Tecnología Middleware para pasarelas residenciales. Hogar digital. El camino de la domótica a los ambientes inteligentes.I Encuentro Interdisciplinar de Domótica 2005, 87-102.

González, V. M., Mateos, F., López, A.M., Enguita, J.M., García M.and Olaiz, R., 2001. Visir, a simulation software for domotics installations to improve laboratory training, Frontiers in Education Conference, 31st Annual Vol. 3, F4C-6-11.

Haenselmann, T., King, T., Busse, B., Effelsberg, W. and Markus Fuchs, 2007. Scriptable Sensor Network Based Home-Automation. Emerging Directions in Embedded and Ubiquitous Computing, 579-591.

Jung, J., Park, K. and Cha J., 2005. Implementation of a Network-Based Distributed System Using the CAN Protocol. Knowledge-Based Intelligent Information and Engineering Systems. Vol. 3681.

Kawamura, R. and Maeomichi, 2004. Standardization Activity of OSGi (Open Services Gateway Initiative), NTT Technical Review, Vol. 2, No. 1, 94-97.

Mellor, S., Scott K., Uhl, A. and Weise, D., 2004. MDA Distilled, Principles of Model Driven Architecture, Addison-Wesley Professional.

Min, W., Hong, Z., Guo-ping, L. and Jin-hua, S., 2007. Networked control and supervision system based on LonWorks fieldbus and Intranet/Internet. Journal of Central South University of Technology, Vol. 14, No.2, 260-265.

Moya, F. and López, J.C., 2002 .SENDA: An Alternative to OSGi for Large Scale Domotics. Networks, pp. 165-176.

Müller, S. and Waller, H., 1999. Efficient Integration of Real-Time Hardware and Web Based Services Into MATLAB. 11th European Simulation Symposium and Exhibition, 26-28.

Muñoz, J., Fons, J., Pelechano, V. and Pastor, O., 2003. Hacia el Modelado Conceptual de Sistemas Domóticos, Actas de las VIII Jornadas de Ingeniería del Software y Bases de Datos. Universidad de Alicante, 369-378

Newcomer, E. and Lomow, G., 2005. Understanding SOA with Web Services. Addison Wesley.

Norton S. and Suppe F., 2001. Why atmospheric modeling is good science. MIT Press. p. 88-133.

Pan, M. and Tseng, Y. 2007. ZigBee and Their Applications. Sensor Networks and Configuration. Ed. Springer Berlin Heidelberg, pp:349-368 ISBN:978-3-540-37364-3

Rhee, S., Yang, S., Park, S., Chun, J.and Park, J., 2004. UPnP Home Networking-Based IEEE1394 Digital Home Appliances Control. Advanced Web Technologies and Applications, Vol. 3007, 457-466.

Sommerville, I. 2004. Software Engineering, 7th ed. Boston: Addison-Wesley.

Sun Microsystems Inc, 1999. JINI Architectural Overview, Technical White Paper, 1999

Valdivieso, R.J., Sánchez, J.G., Azorín, J. and Fuster, A., 2007. Entorno para el desarrollo y simulación de arquitecturas de redes de control en el hogar. II Internacional Simposium Ubiquitous Computing and Ambient Intelligence.

Comparison of Defuzzification Methods: Automatic Control of Temperature and Flow inHeat Exchanger

Alvaro J. Rey Amaya[†], Omar Lengerke[†*], Carlos A. Cosenza[*], Max Suell Dutra[*] and Magda J.M. Tavera[*]

[†]*Autonomous University of Bucaramanga – UNAB, Calle 48 # 39 -234, Bucaramanga - Colombia*

[*]*Federal University of Rio de Janeiro – COPPE-UFRJ Postal Box 68.503 – CEP 21.945-970 – Rio de Janeiro, RJ, Brazil*

1. Introduction

The purpose of this work is to analyze the behavior of traditional and fuzzy control, applying in flow and temperature control to load current of a heat exchanger, and the analysis of different methods of defuzzification, utilized just as itself this carrying out the fuzzy control. Acting on the fuzzy controller structure, some changes of form are carried out such that this tune in to be able to obtain optimal response. Proceeds on the traditional controller and comparisons techniques on these two types of controls are established. Inside the changes that are carried out on the fuzzy controller this form of information defuzzification, that is to say the methods are exchanged defuzzification in order then to realize comparisons on the behavior of each one of methods.

In many sectors of the industry where include thermal processes, is important the presence of heat exchangers (Shah & Sekulic, 2003, Kuppan, 2000). Said processes are components of everyday life of an engineer that has as action field the control systems, therefore is considered interesting to realize a control to this type of systems. This work studies two significant aspects: A comparison between traditional and fuzzy control, and an analysis between several defuzzification methods utilized in the fuzzy logic (Kovacic & Bogdan, 2006, Harris, 2006), development an analysis on each one of methods taking into consideration, contribute that other authors have done and leaving always in allow, that the alone results obtained will be applicable at the time of execute control on an heat exchanger (Xia et al., 1991, Fischer et al., 1998, Alotaibi et al., 2004,). The test system this composed for two heat exchangers, one of concentric pipes and other of hull and pipes, to which implemented them a temperature and flow automatic control to the load current of heating (Fig. 1).

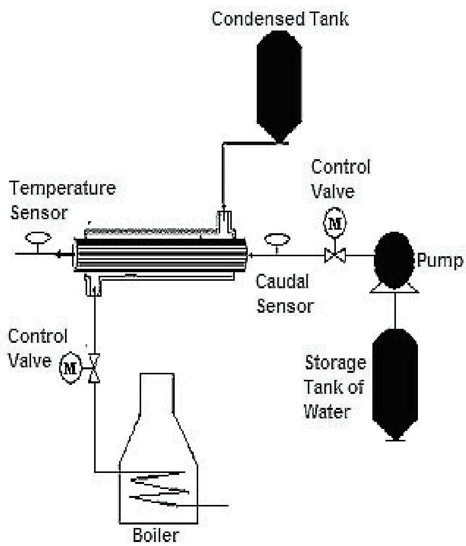


Fig. 1. Thermal pilot plant scheme

The control is realized through two proportional valves, one on input of the water, responsible for keep the value of order of the water and other installed in the line of input of the vapor (source of heat), responsible of keep the quantity necessary of vapor to obtain the target temperature. Physical experimentation typically attaches the notions of uncertainty to measurements of any physical property state (Youden, 1998). The measurement of the flow is realized by a sensor of rotary palette and the measurement of the temperature by thermocouples (Eckert & Goldstein, 1970, McMillan & Considine, 1999, Morris, 2001). The signals supplied by sensors are acquired by National Instruments® FieldPoint module (Kehtarnavaz & Kim, 2005, Archila et al., 2006, Blume, 2007) that takes charge and send signal to the control valves, after to be processed the data by controller.

2. Fuzzy and Classic Control Software

The control software designed uses two sections, the fuzzy and PID (Proportional/Integral/Derivative) control program. These controllers are created on environment Labwindows/CVI by National Instruments Company®, which permits to realize the pertinent operations with the data captured through FieldPoint modules, utilized in systems control. The fuzzy control interface, is the responsible for receiving data of sensors, so much of temperature for the temperature control case, as of the flow sensor for control of the same one, to process, and according to an order established, to determine an response sent to the actuators. Basically, this program is responsible of to schematize the fuzzy sets, according to established by the user, defuzzification of the inputs, to realize the inference of these inputs in rules, to realize aggregation in the outputs sets, and to execute the defuzzification process, to determine the response that assist to the system to obtain the established state. The PID control classic interface is similar of fuzzy control interface, but the difference is in entrusted of to execute the three control actions, proportional, derivative and integral to determine the responses that assist to the system to obtain its target state.

The PID control system general is represented in figure 2, where R(s), is the signal target or set point, U(s), is the output of the PID controller that goes in the direction of the plant G(s), and Y(s), is the value in use of the variable to control, which reduces to the reference and the error is determined (controller input).

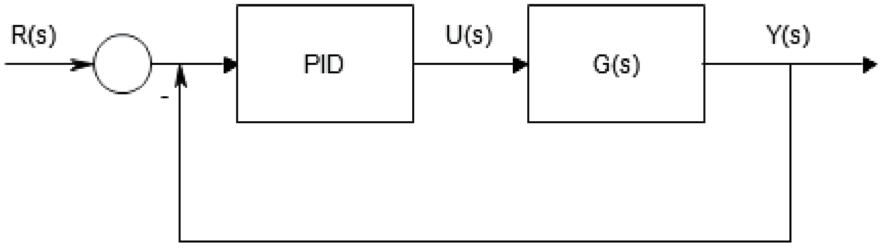


Fig. 2. General PID Control System

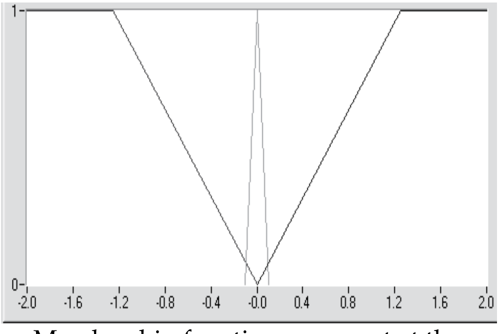
The purpose of temperature control is to achieve that water that the heat exchanger overtakes the value of target temperature and to keep it in the value even with external disruptions. On operate control valve is supplies the quantity of vapor that heats water. The input to system control is the temperature error, obtained since the thermocouple placed on the exit of the exchanger, and the exit control the quantity of necessary current to open or to close the proportional valve (Plant). This control is realized through a PID controller.
The flow control has as purpose to obtain that water mass flow that enters to heat exchanger, achieve the target value, and can to keep it during its operation, and even with disruptions. This means that should operate on the valve of control, who is the one that restrain the water quantity that enters to the system. The system input will be the error obtained through the flow sensor installed in the input of the system, and the PID controller will control the quantity of necessary current to manipulate the proportional valve. Both processes begin, calculating the difference between the measured temperature and the temperature desired or flow measured and flow desired. In this form, identify the error. The values of control parameters are taken, and the output is calculated that goes in the direction of the plant. This output obtains values since 0 to 20 mA, they will represent aperture angles of proportional valve.

3. Fuzzy Control System

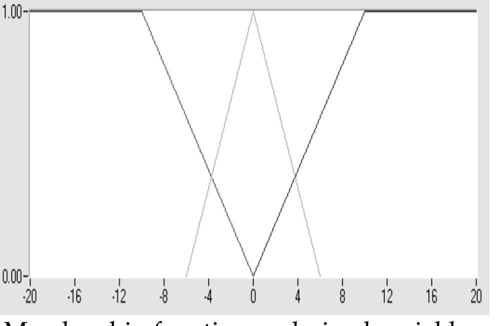
The inputs to control system are temperature error and gradient, obtained since the sensor placed on the way out of exchanger, and the exit control the quantity current necessary to open the proportional valve. The rules and membership function system are obtained in table 1 and figure 3, respectively.

Error / **Δ Error**	**Negative**	**Zero**	**Positive**
Negative	Open	Open	Not operation
Zero	Open	Not operation	Close
Positive	Not operation	Close	Close

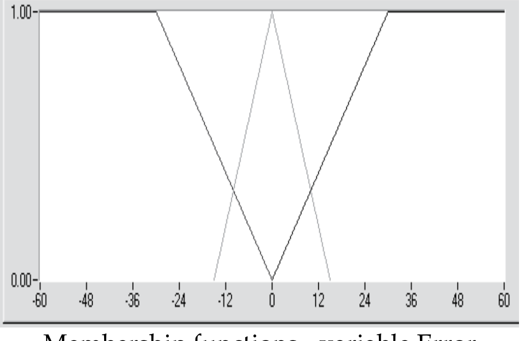
Table 1. Temperature control rules assembly



Membership functions - current at the outset



Membership functions – derived variable from Error



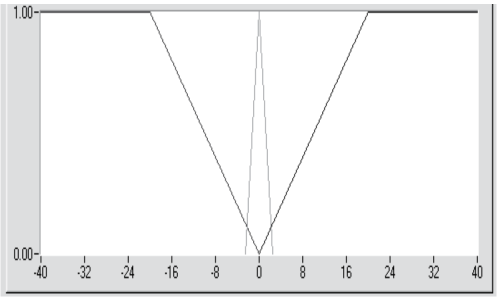
Membership functions - variable Error

Fig. 3. Membership Functions - temperature control

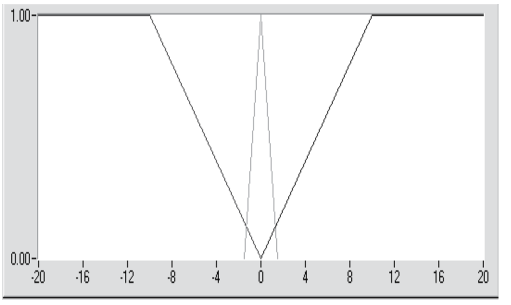
The flow control has as purpose to obtain that water mass flow that enters to heat exchanger, achieve the order value and can to maintain it during its operation, and even before disruptions. This means that should act on the control valve is the one that restrain the water quantity that enters to system. The system input, they will be the error obtained through the flow sensor installed to system entrance, and change of error in the time, and the output will quantity control necessary of current to manipulate the proportional valve. Both processes begin, calculating the difference between measured temperature and desired temperature, or measured flow and desired flow. In this form identify the Error. Calculate the gradient, reducing the error new of previous one. Once known these variables, that constitute the inputs of fuzzy logic controller, proceeds to realize the fuzzification, inference and defuzzification, to obtain the controller output. This output obtains values since 0 to 20 mA; represent aperture angles of proportional valve. The system rules and membership function are obtained in table 2 and figure 4, respectively.

Error / **Δ Error**	**Negative**	**Zero**	**Positive**
Negative	Close	Close	Not operation
Zero	Close	Not operation	Open
Positive	Not operation	Open	Open

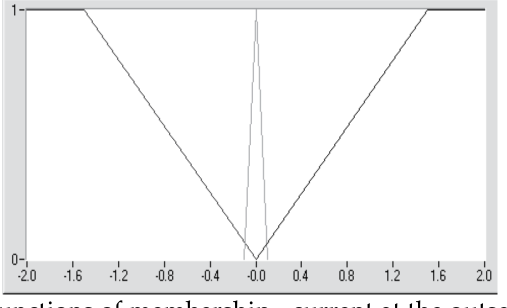
Table 2. Flow control rules assembly



Functions of membership - variable error



Functions of membership – derived variable from error



Functions of membership - current at the outset

Fig. 4. Functions of membership - flow control

3.1 Comparative Results Relating Defuzzification Methods Implemented

The defuzzification methods selects were five; identify in the control area by center of gravity weighted by height, center gravity weighted by area, average of centers, points of maximum criterion weighted by height and points of maximum criterion weighted by area (Zhang & Edmunds, 1991, Guo et al., 1996, Saade & Diab, 2000). In systems control, the main term is the stability that can offer the system, for this is necessary the delayed time that the system in being stabilized, error margin between value desired (V_c), and system stabilization values (V_e) and inertial influence of system. For temperature and flow control tests, is defined a set point 25 [Lts/min] and 40 [ºC]. The parameters and equations used for different responses in each one of the methods are shows in table 3, table 4 and table 5, according the parameters established in table 3.

METHODS	EQUATIONS	
1. Center of gravity weighted by height	$\bar{x} = \dfrac{\sum_{i=1}^{n} h_i * w_i}{\sum_{i=1}^{n} h_i}$	Where, w is gravity center of resultant assembly after fuzzy operation select, and h is the height of the same assembly.
2. Center of gravity weighted by area.	$\bar{x} = \dfrac{\sum_{i=1}^{n} s_i * w_i}{\sum_{i=1}^{n} s_i}$	Where, w is gravity center of resultant assembly after fuzzy operation select, and s is the area of the same assembly.
3. Points of maximum criterion weighted by area.	$\bar{x} = \dfrac{\sum_{i=1}^{n} s_i * G_i}{\sum_{i=1}^{n} s_i}$	Where, G is the point of maximum criterion of resultant set after to realize fuzzy operation select and s is the area of the same set.
4. Points of maximum criterion weighted by height.	$\bar{x} = \dfrac{\sum_{i=1}^{n} h_i * G_i}{\sum_{i=1}^{n} h_i}$	Where, G is the point of maximum criterion of resultant set after to realize fuzzy operation select and h is height of the same set.
5. Average of centers	$y = \dfrac{\sum_{l=1}^{M} y^{-l}(\mu_{B^{`}}(y^{-l}))}{\sum_{l=1}^{M}\left(\mu_{B^{`}}(y^{-l})\right)}$	Where y^{-l} represents the center of fuzzy set G^l (defined as the point V in which $\mu_{G^l}(y)$ reaches its value maximum), and $\mu_B(y)$ defined for the degrees of membership resultant by fuzzy inference.

Table 3. Methods and models defuzzification

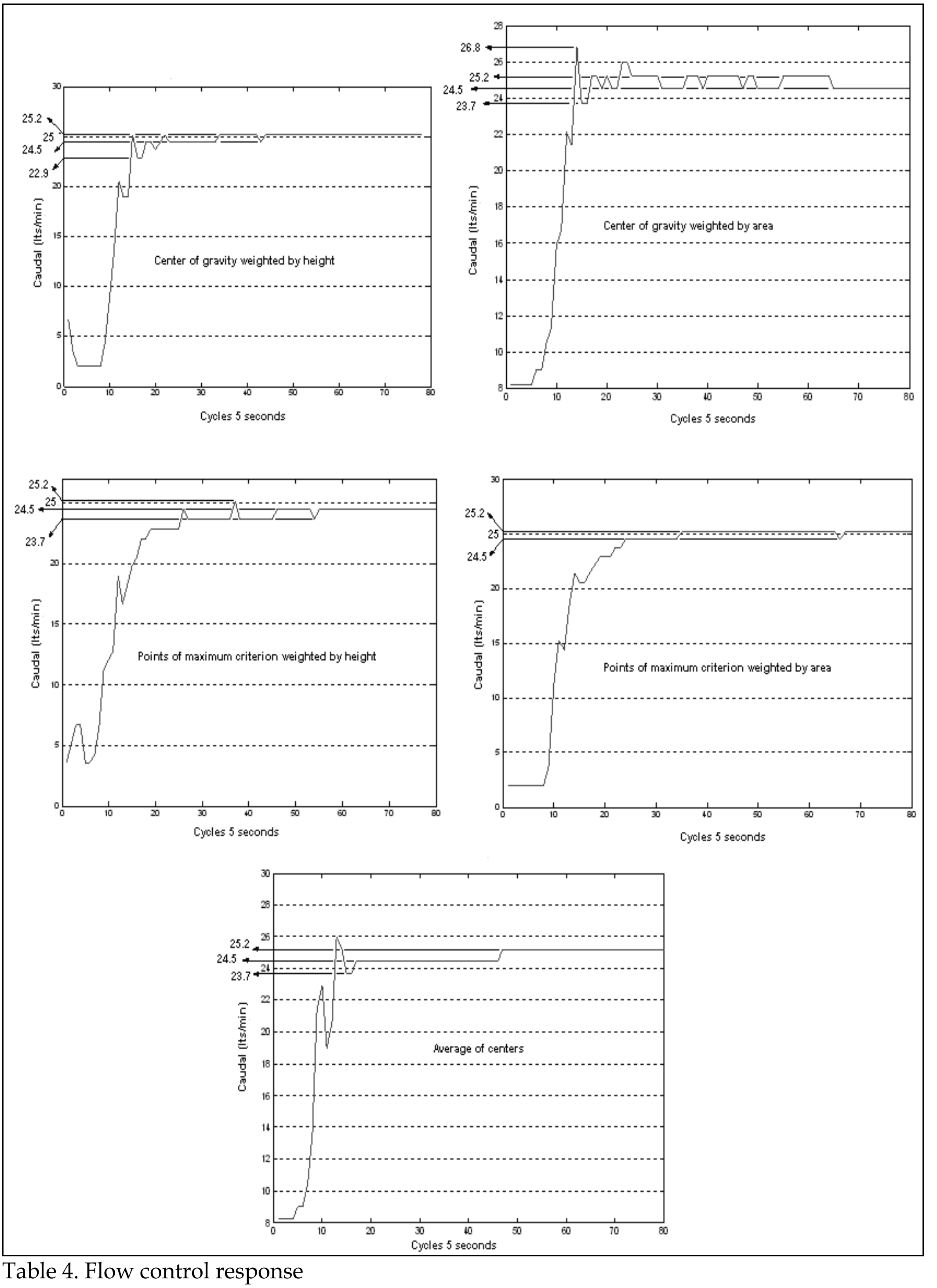


Table 4. Flow control response

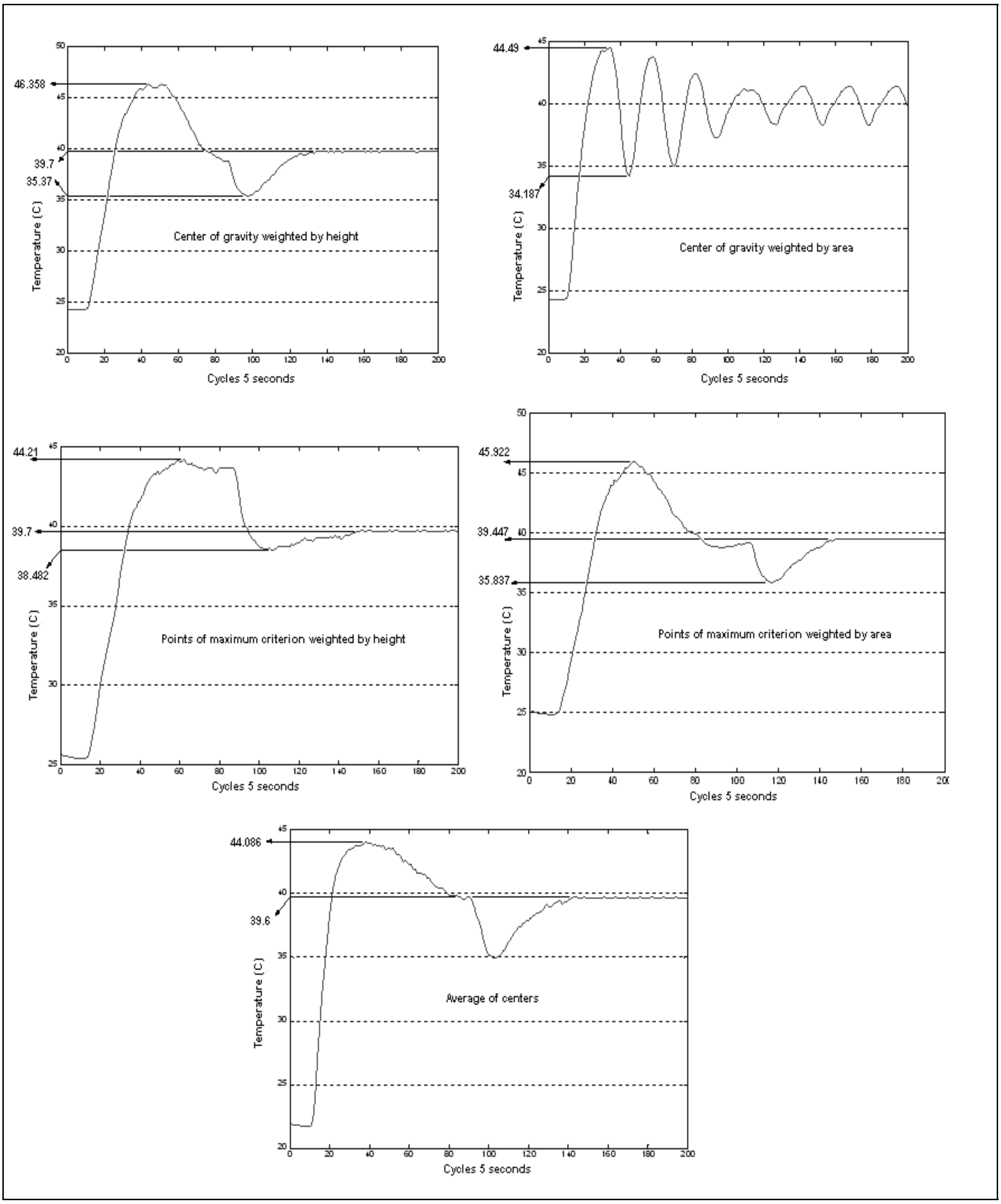


Table 5. Temperature control response

A summary of results obtained on different methods is shown in table 6 for flow control and table 7 for temperature control.

Defuzzification method	Stability time [sec]	Error margin (V_c - V_e)	Inertial influence of system
Center gravity weighted by height	105	0.8% above of the set point 2% below of the set point	0.8% above of the set point 8.4% below of the set point
Center gravity weighted by area	125	0.8% above of the set point 2% below of the set point	7.2% above of the set point 5.2% below of the set point
Average of centers	85	0.8% above of the set point 2% below of the set point	4% above of the set point 5.2% below of the set point
Points of maximum criterion weighted by height	230	2% below of the set point	0.8% above of the set point 5.2% below of the set point
Points of maximum criterion weighted by area	120	0.8% above of the set point 2% below of the set point	0.8% above of the set point 2% below of the set point

Table 6. Response of defuzzification methods - flow control

Defuzzification method	Stability time [sec]	Error margin (V_c - V_e)	Inertial influence of system
Center gravity weighted by height	670	0.75% below of the set point	40.89% above of the set point 11.57% below of the set point
Center gravity weighted by area	Not stabilized	Not stabilized	11.25% above of the set point 14.53% below of the set point
Average of centers	710	1% below of the set point	10.21% above of the set point 12.5% below of the set point
Points of maximum criterion weighted by height	745	0.75% below of the set point	10.52% above of the set point 3.79% below of the set point
Points of maximum criterion weighted by area	735	1.38% below of the set point	14.80% above of the set point 10.40% below of the set point

Table 7. Response of defuzzification methods - temperature control

3.2 Comparative Analysis between Classic and Fuzzy Controller

To be able to realize this analysis should make use of fundamentals concepts at the moment of to evaluate the controller efficiency. The concepts in this case are: systems delayed time in being stabilized, error margin between order value (V_c) and stabilization values (V_e) and inertial Influence of system. For the comparative analysis between fuzzy controller and PID controller, in the flow control use of tests realized to each one of these controllers with set point 25 [Lts/min] and 40 [ºC]. The results obtained are shown in the table 8.

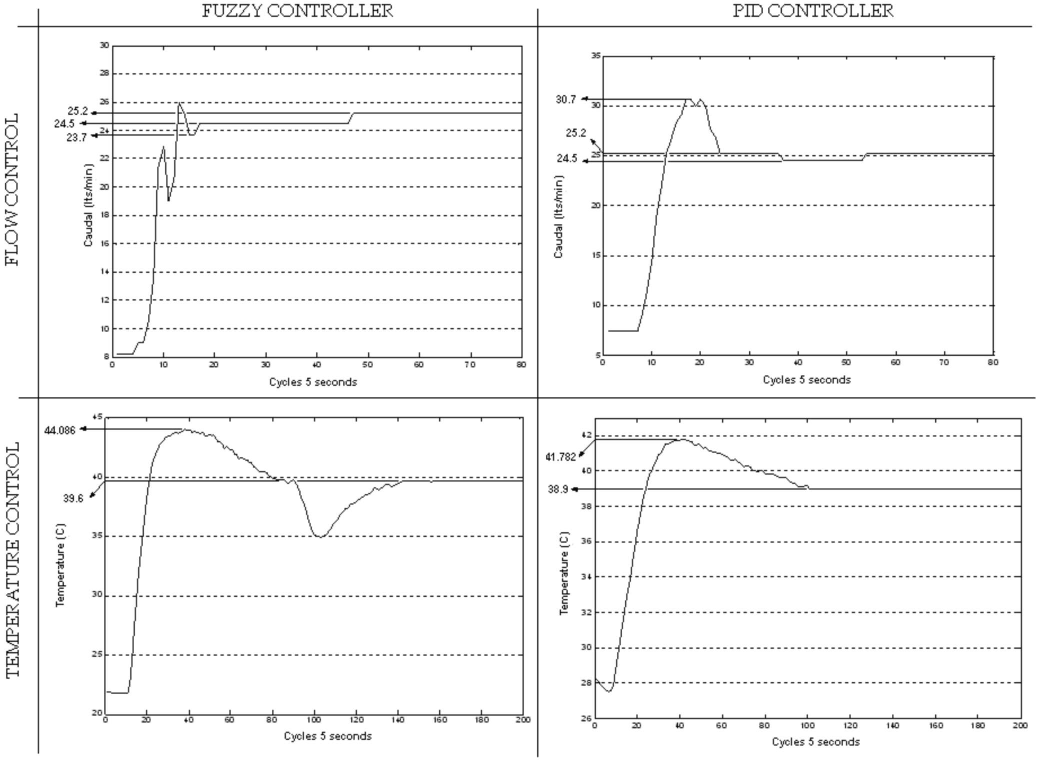


Table 8. Controllers response

A summary of results obtained on different methods is shown for flow control (table 9) and temperature control (table 10).

Controller	Stability time [sec]	Error margin (V_c - V_e)	Inertial influence of system
FUZZY CONTROL	85	0.8% below of the set point 2% above of the set point	4% below of the set point 5.2% above of the set point
PID CONTROL	115	0.8% below of the set point 2% above of the set point	22.8% below of the set point 2% above of the set point

Table 9. Response of controllers - flow control

Controller	Stability time [sec]	Error margin (V_c - V_e)	Inertial influence of system
FUZZY CONTROL	710	1% below of the set point	10.21% above of the set point 12.5% below of the set point
PID CONTROL	505	2.75% below of set point	4.45% above of set point 2.75% below of set point

Table 10. Response of controllers - temperature control

7. Conclusion

The results obtained in this work show the technical viability of the utilization fuzzy logic in the flow and temperature control to the warming-up current input of heat exchanger. The implementation the flow and temperature control with fuzzy logic possesses the advantages of not requires a precision mathematical model for control system. Some disadvantage is the design should be realized generally with test and error method. Is possible to control through fuzzy techniques industrial process with greater facility and errors minimum and sufficient with to identify its general behavior to structure a series of fuzzy sets and its respective rules. The fuzzy controller tune, depending on the rules matrix, also, depends on the size of variable sets, already itself of input or output. This depends on the same behavior system. For the implementation of fuzzy control, is necessary, the establishment of methods and alternatives utilized in each one of the blocks that conform it. In this form, can be obtained optimal results, at the moment of tuning the system. The respond of the fuzzy controller does not depend on the defuzzification method utilized, if not of the adequate utilization of the membership functions, and of numbers of linguistic variables utilized for each one of the variables of input and output of the system. Also, depends on type and size of the sets utilized.

8. References

Alotaibi, S.; Sen, M.; Goodwine, B. & Yang, K.T. (2004). Controllability of Cross-flow Heat Exchanger. *International Journal of Heat and Mass Transfer*, Vol. 4, No. 5, February 2004, pp. 913–924.

Archila, J.F.; Dutra, M.S.; Lengerke, O. & Vega, T.J. (2006). Design and implementation of a control system in closed-loop using the communication system FIELD POINT for a hydrostatic transmission. *RESET: Journal Specialized in Informatics and Electronics Systems of Telecommunications Systems*. Vol. 1, pp. 48–53.

Blume, P.A. (2007). *The LabVIEW Style Book (National Instruments Virtual Instrumentation Series)*. Prentice Hall PTR, ISBN-10/ASIN: 0-1314-5835-3, Upper Saddle River, New Jersey, USA.

Eckert, E.R.G. & Goldstein, R.J. (1970). *Measurements in Heat Transfer*. Technivision Services, ISBN-10/ASIN: 0-8510-2026-7, Slough, England.

Fischer, M.; Nelles, O. & Isermann, R. (1998). Adaptive Predictive Control of a Heat Exchanger Based on a Fuzzy Model. *Control Engineering Practice*, Vol. 6, No. 2, pp. 259-269.

Guo, S.; Peters, L. & Surmann, H. (1996). Design and application of an analog fuzzy logic controller. *IEEE Transactions on Fuzzy Systems*, Vol. 4, No. 4, November 1996, pp. 429-438 ISSN: 1063-6706.

Harris, J. (2006). *Fuzzy Logic Applications in Engineering Science*, Springer, ISBN-10 1-4020-4078-4, Netherland.

Kehtarnavaz, N. & Kim, N. (2005). *Digital Signal Processing System-Level Design Using LabVIEW*. Newnes Elsevier, ISBN-10/ASIN: 0-7506-7914-X, Burlington, USA.

Kovacic, Z. & Bogdan, S. (2006). *Fuzzy Controller Design: Theory and Applications*, Control Engineering Series, CRC Press, Taylor & Francis Group, ISBN 0-8493-3747-X, Boca Raton.

Kuppan, T., (2000). *Heat Exchanger Design Handbook*, Marcel Dekker, Inc., ISBN-10: 0-8247-9787-6, New York.

McMillan, G.K. & Considine, D.M. (1999). *Process/Industrial Instruments and Controls Handbook*. 5th Edition, McGraw-Hill, ISBN-10/ASIN: 0-0701-2582-1, New York, USA.

Morris, A.S. (2001). *Measurement and Instrumentation Principles*. Third Edition, Butterworth-Heinemann, ISBN-10: 0-7506-5081-8, Great Britain.

Saade, J.J. & Diab, H.B. (2000). Defuzzification Techniques for Fuzzy Controllers. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, Vol. 30, No. 1, February 2000, pp. 223-229,

Shah, R.K. & Sekulic, D.P. (2003). *Fundamentals of Heat Exchanger Design*, John Wiley & Sons, ISBN-10/ASIN: 0-4713-2171-0, New Jersey.

Xia, L.; De Abreu-Garcia, J.A. & Hartley, T.T. (1991). Modeling and Simulation of a Heat Exchanger. *IEEE International Conference on Systems Engineering*, August 1991, pp. 453–456, ISBN: 0-7803-0173-0.

Youden, W. J. (1998). *Experimentation and Measurement*. Dover Publications, Mineola, New York.

Zhang, B.S. & Edmunds, J.M. (1991). On Fuzzy Logic Controllers. *Proceedings of International Conference Control'91*, Vol. 2, pp. 961-965, ISBN: 0-85296-509-5.

7

Nonlinear Analysis and Design of Phase-Locked Loops

G.A. Leonov, N.V. Kuznetsov and S.M. Seledzhi
Saint-Petersburg State University
Russia

1. Introduction

Phase-locked loops (PLLs) are widely used in telecommunication and computer architectures. They were invented in the 1930s-1940s (De Bellescize, 1932; Wendt & Fredentall, 1943) and then intensive studies of the theory and practice of PLLs were carried out (Viterbi, 1966; Lindsey, 1972; Gardner, 1979; Lindsey and Chie, 1981; Leonov et al., 1992; Encinas, 1993; Nash, 1994; Brennan, 1996; Stensby, 1997).

One of the first applications of phase-locked loop (PLL) is related to the problems of data transfer by radio signal. In radio engineering PLL is applied to a carrier synchronization, carrier recovery, demodulation, and frequency synthesis (see, e.g., (Stephens, 2002; Ugrumov, 2000)).

After the appearance of an architecture with chips, operating on different frequencies, the phase-locked loops are used to generate internal frequencies of chips and synchronization of operation of different devices and data buses (Young et al., 1992; Egan, 2000; Kroupa, 2003; Razavi, 2003; Shu & Sanchez-Sinencio, 2005; Manassewitsch, 2005). For example, the modern computer motherboards contain different devices and data buses operating on different frequencies, which are often in the need for synchronization (Wainner & Richmond, 2003; Buchanan & Wilson, 2001).

Another actual application of PLL is the problem of saving energy. One of the solutions of this problem for processors is a decreasing of kernel frequency with processor load. The independent phase-locked loops permit one to distribute more uniformly a kernel load to save the energy and to diminish a heat generation on account of that each kernel operates on its own frequency. Now the phase-locked loops are widely used for the solution of the problems of clock skew and synchronization for the sets of chips of computer architectures and chip microarchitecture. For example, a clock skew is very important characteristic of processors (see, e.g., (Xanthopoulos, 2001; Bindal, 2003)).

Various methods for analysis of phase-locked loops are well developed by engineers and are considered in many publications (see, e.g., (Banerjee, 2006; Best, 2003; Kroupa, 2003; Bianchi, 2005; Egan, 2007)), but the problems of construction of adequate nonlinear models and nonlinear analysis of such models are still far from being resolved and require using special methods of qualitative theory of differential, difference, integral, and integro-differential equations (Gelig et al., 1978; Leonov et al., 1996a; Leonov et al., 1996b; Leonov & Smirnova, 2000; Abramovitch, 2002; Suarez & Quere, 2003; Margaris, 2004; Kudrewicz & Wasowicz, 2007; Kuznetsov, 2008; Leonov, 2006). We could not list here all references in the area of design and analysis of PLL, so readers should see mentioned papers and books and the references cited therein.

2. Mathematical model of PLL

In this work three levels of PLL description are suggested:
1) the level of electronic realizations,
2) the level of phase and frequency relations between inputs and outputs in block diagrams,
3) the level of difference, differential and integro-differential equations.
The second level, involving the asymptotical analysis of high-frequency oscillations, is necessary for the well-formed derivation of equations and for the passage to the third level of description.
Consider a PLL on the first level (Fig. 1)

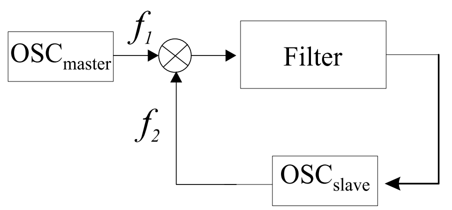


Fig. 1. Block diagram of PLL on the level of electronic realizations.

Here $\text{OSC}_{\text{master}}$ is a master oscillator, $\text{OSC}_{\text{slave}}$ is a slave (tunable) oscillator, which generate high-frequency "almost harmonic oscillations"

$$f_j(t) = A_j \sin(\omega_j(t)t + \psi_j) \quad j = 1, 2, \tag{1}$$

where A_j and ψ_j are some numbers, $\omega_j(t)$ are differentiable functions. Block $\otimes$ is a multiplier of oscillations of $f_1(t)$ and $f_2(t)$ and the signal $f_1(t)f_2(t)$ is its output. The relations between the input $\xi(t)$ and the output $\sigma(t)$ of linear filter have the form

$$\sigma(t) = \alpha_0(t) + \int_0^t \gamma(t - \tau)\xi(\tau)\, d\tau. \tag{2}$$

Here $\gamma(t)$ is an impulse transient function of filter, $\alpha_0(t)$ is an exponentially damped function, depending on the initial data of filter at the moment $t = 0$. The electronic realizations of generators, multipliers, and filters can be found in (Wolaver, 1991; Best, 2003; Chen, 2003; Giannini & Leuzzi, 2004; Goldman, 2007; Razavi, 2001; Aleksenko, 2004). In the simplest case it is assumed that the filter removes from the input the upper sideband with frequency $\omega_1(t) + \omega_2(t)$ but leaves the lower sideband $\omega_1(t) - \omega_2(t)$ without change.
Now we reformulate the high-frequency property of oscillations $f_j(t)$ and essential assumption that $\gamma(t)$ and $\omega_j(t)$ are functions of "finite growth". For this purpose we consider the great fixed time interval $[0, T]$, which can be partitioned into small intervals of the form $[\tau, \tau + \delta], (\tau \in [0, T])$ such that the following relations

$$\begin{aligned} &|\gamma(t) - \gamma(\tau)| \le C\delta, \ \ |\omega_j(t) - \omega_j(\tau)| \le C\delta, \\ &\forall\, t \in [\tau, \tau + \delta], \ \ \forall\, \tau \in [0, T], \end{aligned} \tag{3}$$

$$|\omega_1(\tau) - \omega_2(\tau)| \leq C_1, \ \ \forall\, \tau \in [0, T], \tag{4}$$

$$\omega_j(t) \geq R, \ \ \forall\, t \in [0, T] \tag{5}$$

are satisfied. Here we assume that the quantity δ is sufficiently small with respect to the fixed numbers T, C, C_1, the number R is sufficiently great with respect to the number δ. The latter means that on the small intervals $[\tau, \tau+\delta]$ the functions $\gamma(t)$ and $\omega_j(t)$ are "almost constants" and the functions $f_j(t)$ rapidly oscillate as harmonic functions.
Consider two block diagrams shown in Fig. 2 and Fig. 3.

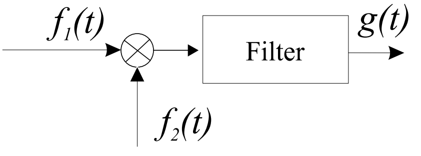


Fig. 2. Multiplier and filter.

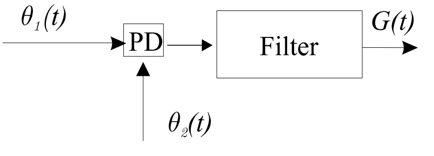


Fig. 3. Phase detector and filter.

Here $\theta_j(t) = \omega_j(t)t + \psi_j$ are phases of the oscillations $f_j(t)$, PD is a nonlinear block with the characteristic $\varphi(\theta)$ (being called a phase detector or discriminator). The phases $\theta_j(t)$ are the inputs of PD block and the output is the function $\varphi(\theta_1(t) - \theta_2(t))$. The shape of the phase detector characteristic is based on the shape of input signals.
The signals $f_1(t)f_2(t)$ and $\varphi(\theta_1(t) - \theta_2(t))$ are inputs of the same filters with the same impulse transient function $\gamma(t)$. The filter outputs are the functions $g(t)$ and $G(t)$, respectively.
A classical PLL synthesis is based on the following result:

Theorem 1. *(Viterbi, 1966) If conditions (3)–(5) are satisfied and we have*

$$\varphi(\theta) = \frac{1}{2} A_1 A_2 \cos\theta,$$

then for the same initial data of filter, the following relation

$$|G(t) - g(t)| \leq C_2\delta, \ \forall\, t \in [0, T]$$

is satisfied. Here C_2 is a certain number being independent of δ.

Proof of Theorem 1 (Leonov, 2006)
For $t \in [0, T]$ we obviously have

$$\begin{aligned}
&g(t) - G(t) = \\
&= \int_0^t \gamma(t-s)\Big[A_1 A_2\Big(\sin\big(\omega_1(s)s + \psi_1\big)\sin\big(\omega_2(s)s + \psi_2\big)\Big) - \\
&- \varphi\Big(\omega_1(s)s - \omega_2(s)s + \psi_1 - \psi_2\Big)\Big] ds = \\
&= -\frac{A_1 A_2}{2}\int_0^t \gamma(t-s)\left[\cos\Big(\big(\omega_1(s) + \omega_2(s)\big)s + \psi_1 + \psi_2\Big)\right] ds.
\end{aligned}$$

Consider the intervals $[k\delta, (k+1)\delta]$, where $k = 0, \ldots, m$ and the number m is such that $t \in [m\delta, (m+1)\delta]$. From conditions (3)–(5) it follows that for any $s \in [k\delta, (k+1)\delta]$ the relations

$$\gamma(t-s) = \gamma(t-k\delta) + O(\delta) \tag{6}$$

$$\omega_1(s) + \omega_2(s) = \omega_1(k\delta) + \omega_2(k\delta) + O(\delta) \tag{7}$$

are valid on each interval $[k\delta, (k+1)\delta]$. Then by (7) for any $s \in [k\delta, (k+1)\delta]$ the estimate

$$\cos\Big(\big(\omega_1(s) + \omega_2(s)\big)s + \psi_1 + \psi_2\Big) = \cos\Big(\big(\omega_1(k\delta) + \omega_2(k\delta)\big)s + \psi_1 + \psi_2\Big) + O(\delta) \tag{8}$$

is valid. Relations (6) and (8) imply that

$$\begin{aligned}
&\int_0^t \gamma(t-s)\left[\cos\Big(\big(\omega_1(s) + \omega_2(s)\big)s + \psi_1 + \psi_2\Big)\right] ds = \\
&= \sum_{k=0}^{m} \gamma(t-k\delta)\int_{k\delta}^{(k+1)\delta}\left[\cos\Big(\big(\omega_1(k\delta) + \omega_2(k\delta)\big)s + \psi_1 + \psi_2\Big)\right] ds + O(\delta).
\end{aligned} \tag{9}$$

From (5) we have the estimate

$$\int_{k\delta}^{(k+1)\delta}\left[\cos\Big(\big(\omega_1(k\delta) + \omega_2(k\delta)\big)s + \psi_1 + \psi_2\Big)\right] ds = O(\delta^2)$$

and the fact that R is sufficiently great as compared with δ. Then

$$\int_0^t \gamma(t-s)\left[\cos\Big(\big(\omega_1(s) + \omega_2(s)\big)s + \psi_1 + \psi_2\Big)\right] ds = O(\delta).$$

Theorem 1 is completely proved. ■

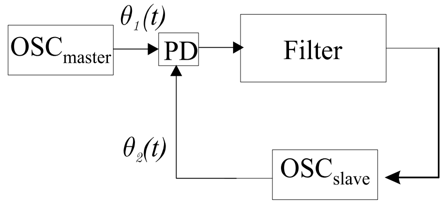


Fig. 4. Block diagram of PLL on the level of phase relations

Thus, the outputs $g(t)$ and $G(t)$ of two block diagrams in Fig. 2 and Fig. 3, respectively, differ little from each other and we can pass (from a standpoint of the asymptotic with respect to δ) to the following description level, namely to the second level of phase relations.
In this case a block diagram in Fig. 1 becomes the following block diagram (Fig. 4).
Consider now the high-frequency impulse oscillators, connected as in diagram in Fig. 1. Here

$$f_j(t) = A_j \text{sign}\,(\sin(\omega_j(t)t + \psi_j)). \tag{10}$$

We assume, as before, that conditions (3)– (5) are satisfied.
Consider 2π-periodic function $\varphi(\theta)$ of the form

$$\varphi(\theta) = \begin{cases} A_1 A_2 (1 + 2\theta/\pi) \text{ for } \theta \in [-\pi, 0], \\ A_1 A_2 (1 - 2\theta/\pi) \text{ for } \theta \in [0, \pi]. \end{cases} \tag{11}$$

and block diagrams in Fig. 2 and Fig. 3.

Theorem 2. *(Leonov, 2006)*
If conditions (3)–(5) are satisfied and the characteristic of phase detector $\varphi(\theta)$ has the form (11), then for the same initial data of filter the following relation

$$|G(t) - g(t)| \le C_3 \delta, \ \forall\, t \in [0, T]$$

is satisfied. Here C_3 is a certain number being independent of δ.

Proof of Theorem 2
In this case we have

$$\begin{aligned} & g(t) - G(t) = \\ & = \int_0^t \gamma(t-s) \Big[A_1 A_2 \text{sign}\Big(\sin\big(\omega_1(s)s + \psi_1\big) \sin\big(\omega_2(s)s + \psi_2\big) \Big) - \\ & - \varphi\big(\omega_1(s)s - \omega_2(s)s + \psi_1 - \psi_2\big) \Big] \, ds. \end{aligned}$$

Partitioning the interval $[0, t]$ into the intervals $[k\delta, (k+1)\delta]$ and making use of assumptions (5) and (10), we replace the above integral with the following sum

$$\sum_{k=0}^{m} \gamma(t-k\delta)\Big[\int\limits_{k\delta}^{(k+1)\delta} A_1A_2\text{sign}\Big[\cos\Big(\big(\omega_1(k\delta)\omega_2(k\delta)\big)k\delta+\psi_1-\psi_2\Big)-$$
$$-\cos\Big(\big(\omega_1(k\delta)+\omega_2(k\delta)\big)s+\psi_1+\psi_2\Big)\Big]\,ds-$$
$$-\varphi\Big(\big(\omega_1(k\delta)-\omega_2(k\delta)\big)k\delta+\psi_1-\psi_2\Big)\delta\Big].$$

The number m is chosen in such a way that $t \in [m\delta, (m+1)\delta]$. Since $(\omega_1(k\delta)+\omega_2(k\delta))\delta \gg 1$, the relation

$$\begin{aligned}\int\limits_{k\delta}^{(k+1)\delta} A_1A_2\text{sign}\Big[\cos\Big(\big(\omega_1(k\delta)-\omega_2(k\delta)\big)k\delta+\psi_1-\psi_2\Big)-&\\ -\cos\Big(\big(\omega_1(k\delta)+\omega_2(k\delta)\big)s+\psi_1+\psi_2\Big)\Big]\,ds \approx&\\ \approx \varphi\Big(\big(\omega_1(k\delta)-\omega_2(k\delta)\big)k\delta+\psi_1-\psi_2\Big)\delta,&\end{aligned} \tag{12}$$

is satisfied. Here we use the relation

$$A_1A_2\int\limits_{k\delta}^{(k+1)\delta} \text{sign}\big[\cos\alpha-\cos\left(\omega s+\psi_0\right)\big]\,ds \approx \varphi(\alpha)\delta$$

for $\omega\delta \gg 1$, $\alpha \in [-\pi, \pi]$, $\psi_0 \in R^1$.
Thus, Theorem 2 is completely proved. ■
Theorem 2 is a base for the synthesis of PLL with impulse oscillators. For the impulse clock oscillators it permits one to consider two block diagrams simultaneously: on the level of electronic realization (Fig. 1) and on the level of phase relations (Fig. 4), where general principles of the theory of phase synchronization can be used (Leonov & Seledzhi, 2005b; Kuznetsov et al., 2006; Kuznetsov et al., 2007; Kuznetsov et al., 2008; Leonov, 2008).

3. Differential equations of PLL

Let us make a remark necessary for derivation of differential equations of PLL.
Consider a quantity

$$\dot{\theta}_j(t) = \omega_j(t) + \dot{\omega}_j(t)t.$$

For the well-synthesized PLL such that it possesses the property of global stability, we have exponential damping of the quantity $\dot{\omega}_j(t)$:

$$|\dot{\omega}_j(t)| \leq Ce^{-\alpha t}.$$

Here C and α are certain positive numbers being independent of t. Therefore, the quantity $\dot{\omega}_j(t)t$ is, as a rule, sufficiently small with respect to the number R (see conditions (3)– (5)). From the above we can conclude that the following approximate relation $\dot{\theta}_j(t) \approx \omega_j(t)$ is valid. In deriving the differential equations of this PLL, we make use of a block diagram in Fig. 4 and exact equality

$$\dot{\theta}_j(t) = \omega_j(t). \tag{13}$$

Note that, by assumption, the control law of tunable oscillators is linear:

$$\omega_2(t) = \omega_2(0) + LG(t). \tag{14}$$

Here $\omega_2(0)$ is the initial frequency of tunable oscillator, L is a certain number, and $G(t)$ is a control signal, which is a filter output (Fig. 4). Thus, the equation of PLL is as follows

$$\dot{\theta}_2(t) = \omega_2(0) + L\Big(\alpha_0(t) + \int\limits_0^t \gamma(t-\tau)\varphi\big(\theta_1(\tau) - \theta_2(\tau)\big)d\tau\Big).$$

Assuming that the master oscillator is such that $\omega_1(t) \equiv \omega_1(0)$, we obtain the following relations for PLL

$$\big(\theta_1(t) - \theta_2(t)\big)' + L\Big(\alpha_0(t) + \int\limits_0^t \gamma(t-\tau)\varphi\big(\theta_1(\tau) - \theta_2(\tau)\big)d\tau\Big) = \omega_1(0) - \omega_2(0). \tag{15}$$

This is an equation of standard PLL. Note, that if the filter (2) is integrated with the transfer function $W(p) = (p+\alpha)^{-1}$

$$\dot{\sigma} + \alpha\sigma = \varphi(\theta)$$

then for $\phi(\theta) = \cos(\theta)$ instead of equation (15) from (13) and (14) we have

$$\ddot{\tilde{\theta}} + \alpha\dot{\tilde{\theta}} + L\sin\tilde{\theta} = \alpha\big(\omega_1(0) - \omega_2(0)\big) \tag{16}$$

with $\tilde{\theta} = \theta_1 - \theta_2 + \dfrac{\pi}{2}$. So, if here phases of the input and output signals mutually shifted by $\pi/2$ then the control signal $G(t)$ equals zero.

Arguing as above, we can conclude that in PLL it can be used the filters with transfer functions of more general form

$$K(p) = a + W(p),$$

where a is a certain number, $W(p)$ is a proper fractional rational function. In this case in place of equation (15) we have

$$\begin{aligned}\big(\theta_1(t) - \theta_2(t)\big)' + L\Big[a\varphi\big(\theta_1(t) - \theta_2(t)\big) + \alpha_0(t) + \int\limits_0^t \gamma(t-\tau)\varphi\big(\theta_1(\tau) - \theta_2(\tau)\big)d\tau\Big] = \\ = \omega_1(0) - \omega_2(0).\end{aligned} \tag{17}$$

In the case when the transfer function of the filter $a + W(p)$ is non-degenerate, i.e. its numerator and denominator do not have common roots, equation (17) is equivalent to the following system of differential equations

$$\begin{aligned}\dot{z} &= Az + b\psi(\sigma) \\ \dot{\sigma} &= c^*z + \rho\psi(\sigma).\end{aligned} \tag{18}$$

Here $\sigma = \theta_1 - \theta_2$, A is a constant $(n \times n)$-matrix, b and c are constant (n)-vectors, ρ is a number, and $\psi(\sigma)$ is 2π-periodic function, satisfying the relations:

$$\begin{aligned}
&\rho = -aL, \\
&W(p) = L^{-1}c^*(A - pI)^{-1}b, \\
&\psi(\sigma) = \varphi(\sigma) - \frac{\omega_1(0) - \omega_2(0)}{L(a + W(0))}.
\end{aligned}$$

The discrete phase-locked loops obey similar equations

$$\begin{aligned}
&z(t+1) = Az(t) + b\psi(\sigma(t)) \\
&\sigma(t+1) = \sigma(t) + c^*z(t) + \rho\psi(\sigma(t)),
\end{aligned} \tag{19}$$

where $t \in Z$, Z is a set of integers. Equations (18) and (19) describe the so-called standard PLLs (Shakhgil'dyan & Lyakhovkin, 1972; Leonov, 2001). Note that there exist many other modifications of PLLs and some of them are considered below.

4. Mathematical analysis methods of PLL

The theory of phase synchronization was developed in the second half of the last century on the basis of three applied theories: theory of synchronous and induction electrical motors, theory of auto-synchronization of the unbalanced rotors, theory of phase-locked loops. Its main principle is in consideration of the problem of phase synchronization at three levels: (i) at the level of mechanical, electromechanical, or electronic models, (ii) at the level of phase relations, and (iii) at the level of differential, difference, integral, and integro-differential equations. In this case the difference of oscillation phases is transformed into the control action, realizing synchronization. These general principles gave impetus to creation of universal methods for studying the phase synchronization systems. Modification of the direct Lyapunov method with the construction of periodic Lyapunov-like functions, the method of positively invariant cone grids, and the method of nonlocal reduction turned out to be most effective. The last method, which combines the elements of the direct Lyapunov method and the bifurcation theory, allows one to extend the classical results of F. Tricomi and his progenies to the multidimensional dynamical systems.

4.1 Method of periodic Lyapunov functions

Here we formulate the extension of the Barbashin–Krasovskii theorem to dynamical systems with a cylindrical phase space (Barbashin & Krasovskii, 1952). Consider a differential inclusion

$$\dot{x} \in f(x), \;\; x \in R^n, \;\; t \in R^1, \tag{20}$$

where $f(x)$ is a semicontinuous vector function whose values are the bounded closed convex set $f(x) \subset R^n$. Here R^n is an n-dimensional Euclidean space. Recall the basic definitions of the theory of differential inclusions.

Definition 1. *We say that $U_\varepsilon(\Omega)$ is an ε-neighbourhood of the set Ω if*

$$U_\varepsilon(\Omega) = \{x \mid \inf_{y \in \Omega} |x - y| < \varepsilon\},$$

where $|\cdot|$ is an Euclidean norm in R^n.

Definition 2. *A function $f(x)$ is called semicontinuous at a point x if for any $\varepsilon > 0$ there exists a number $\delta(x, \varepsilon) > 0$ such that the following containment holds:*

$$f(y) \in U_\varepsilon(f(x)), \ \forall y \in U_\delta(x).$$

Definition 3. *A vector function $x(t)$ is called a solution of differential inclusion if it is absolutely continuous and for the values of t, at which the derivative $\dot{x}(t)$ exists, the inclusion*

$$\dot{x}(t) \in f(x(t))$$

holds.

Under the above assumptions on the function $f(x)$, the theorem on the existence and continuability of solution of differential inclusion (20) is valid (Yakubovich et al., 2004). Now we assume that the linearly independent vectors $d_1, \dots, d_m$ satisfy the following relations:

$$f(x + d_j) = f(x), \ \forall x \in R^n. \tag{21}$$

Usually, $d_j^* x$ is called the phase or angular coordinate of system (20). Since property (21) allows us to introduce a cylindrical phase space (Yakubovich et al., 2004), system (20) with property (21) is often called a system with cylindrical phase space.
The following theorem is an extension of the well–known Barbashin–Krasovskii theorem to differential inclusions with a cylindrical phase space.

Theorem 3. *Suppose that there exists a continuous function $V(x) : R^n \to R^1$ such that the following conditions hold:*
1) $V(x + d_j) = V(x), \ \forall x \in R^n, \ \forall j = 1, \dots, m$;
2) $V(x) + \sum_{j=1}^{m} (d_j^* x)^2 \to \infty$ as $|x| \to \infty$;
3) for any solution $x(t)$ of inclusion (20) the function $V(x(t))$ is nonincreasing;
4) if $V(x(t)) \equiv V(x(0))$, then $x(t)$ is an equilibrium state.
Then any solution of inclusion (20) tends to stationary set as $t \to +\infty$.

Recall that the tendency of solution to the stationary set Λ as t means that

$$\lim_{t \to +\infty} \inf_{z \in \Lambda} |z - x(t)| = 0.$$

A proof of Theorem 3 can be found in (Yakubovich et al., 2004).

4.2 Method of positively invariant cone grids. An analog of circular criterion

This method was proposed independently in the works (Leonov, 1974; Noldus, 1977). It is sufficiently universal and "fine" in the sense that here only two properties of system are used such as the availability of positively invariant one-dimensional quadratic cone and the invariance of field of system (20) under shifts by the vector d_j (see (21)).
Here we consider this method for more general nonautonomous case

$$\dot{x} = F(t, x), \quad x \in \mathbb{R}^n, \ t \in \mathbb{R}^1,$$

where the identities $F(t, x + d_j) = F(t, x)$ are valid $\forall x \in \mathbb{R}^n, \forall t \in \mathbb{R}^1$ for the linearly independent vectors $d_j \in \mathbb{R}^n$ $(j = 1, ..., m)$. Let $x(t) = x(t, t_0, x_0)$ is a solution of the system such that $x(t_0, t_0, x_0) = x_0$.

We assume that such a cone of the form $\Omega = \{x^* H x \leq 0\}$, where H is a symmetrical matrix such that one eigenvalue is negative and all the rest are positive, is positively invariant. The latter means that on the boundary of cone $\partial\Omega = \{xHx = 0\}$ the relation

$$\dot{V}(x(t)) < 0$$

is satisfied for all $x(t)$ such that $\{x(t) \neq 0, x(t) \in \partial\Omega\}$ (Fig. 5).

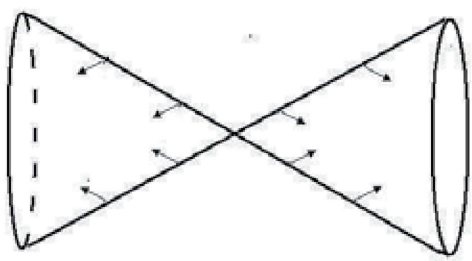


Fig. 5. Positively invariant cone.

By the second property, namely the invariance of vector field under shift by the vectors kd_j, $k \in Z$, we multiply the cone in the following way

$$\Omega_k = \{(x - kd_j)H(x - kd_j) \leq 0\}.$$

Since it is evident that for the cones Ω_k the property of positive invariance holds true, we obtain a positively invariant cone grid shown in Fig. 6. As can be seen from this figure, all the

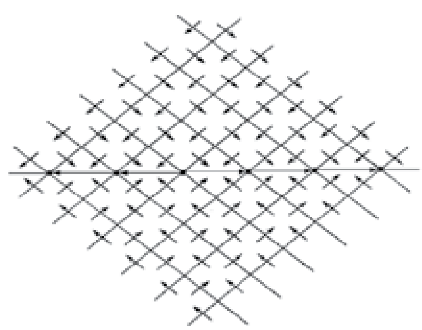


Fig. 6. Positively invariant cone grid.

solutions $x(t, t_0, x_0)$ of system, having these two properties, are bounded on $[t_0, +\infty)$.
If the cone Ω has only one point of intersection with the hyperplane $\{d_j^* x = 0\}$ and all solutions $x(t)$, for which at the time t the inequality

$$x(t)^* H x(t) \geq 0$$

is satisfied, have property $\dot{V}(x(t)) \leq -\varepsilon |x(t)|^2$ (here ε is a positive number), then from Fig. 6 it is clear that the system is Lagrange stable (all solutions are bounded on the interval $[0, +\infty)$).

Thus, the proposed method is simple and universal. By the Yakubovich–Kalman frequency theorem it becomes practically efficient (Gelig et al., 1978; Yakubovich et al., 2004).
Consider, for example, the system

$$\dot{x} = Px + q\varphi(t,\sigma), \ \ \sigma = r^* x, \tag{22}$$

where P is a constant singular $n \times n$-matrix, q and r are constant n-dimensional vectors, and $\varphi(t,\sigma)$ is a continuous 2π-periodic in σ function $R^1 \times R^1 \to R^1$, satisfying the relations

$$\mu_1 \leq \frac{\varphi(t,\sigma)}{\sigma} \leq \mu_2, \ \forall\, t \in R^1, \ \forall \sigma \neq 0, \quad \varphi(t,0) = 0.$$

Here μ_1 and μ_2 are some numbers, which by virtue of periodicity of $\varphi(t,\sigma)$ in σ, without loss of generality, can be assumed to be negative, $\mu_1 < 0$, and positive, $\mu_2 > 0$, respectively.
We introduce the transfer function of system (22)

$$\chi(p) = r^*(P - pI)^{-1} q,$$

which is assumed to be nondegenerate. Consider now quadratic forms $V(x) = x^* H x$ and

$$G(x,\xi) = 2x^* H[(P + \lambda I)x + q\xi] + (\mu_2^{-1}\xi - r^* x)(\mu_1^{-1}\xi - r^* x),$$

where λ is a positive number.
By the Yakubovich–Kalman theorem, for the existence of the symmetrical matrix H with one negative and $n-1$ positive eigenvalues and such that the inequality $G(x,\xi) < 0, \forall x \in R^n, \xi \in R^1$, $x \neq 0$ is satisfied, it is sufficient that
(C1) the matrix $(P + \lambda I)$ has $(n-1)$ eigenvalues with negative real part and
(C2) the frequency inequality

$$\mu_1^{-1}\mu_2^{-1} + (\mu_1^{-1} + \mu_2^{-1})\mathrm{Re}\chi(i\omega - \lambda) + |\chi(i\omega - \lambda)|^2 < 0, \ \forall \omega \in R^1$$

is satisfied.
It is easy to see that the condition $G(x,\xi) < 0, \forall x \neq 0, \forall \xi$ implies the relation

$$\dot{V}\big(x(t)\big) + 2\lambda V\big(x(t)\big) < 0, \ \forall x(t) \neq 0.$$

This inequality assures the positive invariance of the considered cone Ω.
Thus, we obtain the following analog of the well-known circular criterion.

Theorem 4. *(Leonov, 1974; Gelig et al., 1978; Yakubovich et al., 2004)*
If there exists a positive number λ such that the above conditions (C1) and (C2) are satisfied, then any solution $x(t,t_0,x_0)$ of system (22) is bounded on the interval $(t_0,+\infty)$.

A more detailed proof of this fact can be found in (Leonov & Smirnova 2000; Gelig et al., 1978; Yakubovich et al., 2004). We note that this theorem is also true under the condition of nonstrict inequality in (C2) and in the cases when $\mu_1 = -\infty$ or $\mu_2 = +\infty$ (Leonov & Smirnova 2000; Gelig et al., 1978; Yakubovich et al., 2004).
We apply now an analog of the circular criterion, formulated with provision for the above remark, to the simplest case of the second-order equation

$$\ddot{\theta} + \alpha\dot{\theta} + \varphi(t,\theta) = 0, \tag{23}$$

where α is a positive parameter (equation (16) can be transformed into (23) by $\tilde{\theta} = \theta + \arcsin\left[\alpha\left(\omega_1(0) - \omega_2(0)\right)/L\right]$). This equation can be represented as system (22) with $n = 2$ and the transfer function

$$\chi(p) = \frac{1}{p(p+\alpha)}.$$

Obviously, condition (C1) of theorem takes the form $\lambda \in (0, \alpha)$ and for $\mu_1 = -\infty$ and $\mu_2 = \alpha^2/4$ condition (C2) is equivalent to the inequality

$$-\omega^2 + \lambda^2 - \alpha\lambda + \alpha^2/4 \leq 0, \ \forall \omega \in R^1.$$

This inequality is satisfied for $\lambda = \alpha/2$. Thus, if in equation (23) the function $\varphi(t, \theta)$ is periodic with respect to θ and satisfies the inequality

$$\frac{\varphi(t,\theta)}{\theta} \leq \frac{\alpha^2}{4}, \tag{24}$$

then any its solution $\theta(t)$ is bounded on $(t_0, +\infty)$.
It is easily seen that for $\varphi(t, \theta) \equiv \varphi(\theta)$ (i.e. $\varphi(t, \theta)$ is independent of t) equation (23) is dichotomic. It follows that in the autonomous case if relation (24) is satisfied, then any solution of (23) tends to certain equilibrium state as $t \to +\infty$.
Here we have interesting analog of notion of absolute stability for phase synchronization systems. If system (22) is absolutely stable under the condition that for any nonlinearity φ from the sector $[\mu_1, \mu_2]$ any its solution tends to certain equilibrium state, then for equation (23) with $\varphi(t, \theta) \equiv \varphi(\theta)$ this sector is $(-\infty, \alpha^2/4]$.
At the same time, in the classical theory of absolute stability (without the assumption that φ is periodic), for $\varphi(t, \theta) \equiv \varphi(\theta)$ we have two sectors: the sector of absolute stability $(0, +\infty)$ and the sector of absolute instability $(-\infty, 0)$.
Thus, the periodicity alone of φ allows one to cover a part of sector of absolute stability and a complete sector of absolute instability: $(-\infty, \alpha^2/4] \supset (-\infty, 0) \cup (0, \alpha^2/4]$ (see Fig. 7).

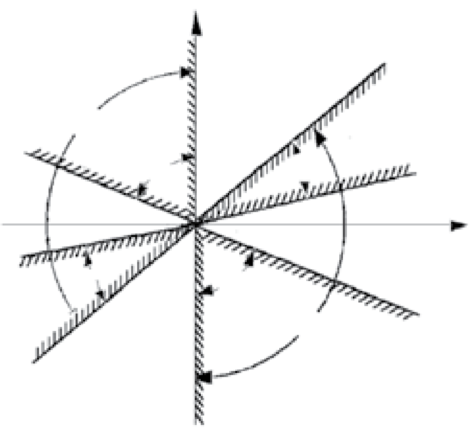


Fig. 7. Sectors of stability and instability.

More complex examples of using the analog of circular criterion can be found in (Leonov & Smirnova 2000; Gelig et al., 1978; Yakubovich et al., 2004).

4.3 Method of nonlocal reduction

We describe the main stages of extending the theorems of Tricomi and his progenies, obtained for the equation

$$\ddot{\theta} + \alpha\dot{\theta} + \psi(\theta) = 0, \tag{25}$$

to systems of higher dimensions.
Consider first the system

$$\begin{aligned} \dot{z} &= Az + b\psi(\sigma) \\ \dot{\sigma} &= c^*z + \rho\psi(\sigma), \end{aligned} \tag{26}$$

describing a standard PLL. We assume, as usual, that $\psi(\sigma)$ is 2π-periodic, A is a stable $n \times n$-matrix, b and c are constant n-vectors, and ρ is a number.
Consider the case when any solution of equation (25) or its equivalent system

$$\begin{aligned} \dot{\eta} &= -\alpha\eta - \psi(\theta) \\ \dot{\theta} &= \eta \end{aligned} \tag{27}$$

tends to the equilibrium state as $t \to +\infty$. In this case it is possible to demonstrate (Barbashin & Tabueva, 1969) that for the equation

$$\frac{d\eta}{d\theta} = \frac{-\alpha\eta - \psi(\theta)}{\eta} \tag{28}$$

equivalent to (27) there exists a solution $\eta(\theta)$ such that $\eta(\theta_0) = 0$, $\eta(\theta) \neq 0, \forall\theta \neq \theta_0$,

$$\lim_{\theta \to +\infty} \eta(\theta) = -\infty, \quad \lim_{\theta \to -\infty} \eta(\theta) = +\infty. \tag{29}$$

Here θ_0 is a number such that $\psi(\theta_0) = 0$, $\psi'(\theta_0) < 0$.
We consider now the function

$$V(z,\sigma) = z^*Hz - \frac{1}{2}\eta(\sigma)^2,$$

which induces the cone $\Omega = \{V(z,\sigma) \leq 0\}$ in the phase space $\{z,\sigma\}$. This is a generalization of quadratic cone shown in Fig. 5. We prove that under certain conditions this cone is positively invariant. Consider the expression

$$\begin{aligned} \frac{dV}{dt} + 2\lambda V &= 2z^*H\left[(A+\lambda I)z + b\psi(\sigma)\right] - \lambda\eta(\sigma)^2 - \eta(\sigma)\frac{d\eta(\sigma)}{d\sigma}(c^*z + \rho\psi(\sigma)) = \\ &= 2z^*H\left[(A+\lambda I)z + b\psi(\sigma)\right] - \lambda\eta(\sigma)^2 + \psi(\sigma)(c^*z + \rho\psi(\sigma)) + \alpha\eta(\sigma)(c^*z + \rho\psi(\sigma)). \end{aligned}$$

Here we make use of the fact that $\eta(\sigma)$ satisfies equation (28).
We note that if the frequency inequalities

$$\begin{aligned} &\operatorname{Re} W(i\omega - \lambda) - \varepsilon|K(i\omega - \lambda)|^2 > 0, \\ &\lim_{\omega \to \infty} \omega^2(\operatorname{Re} K(i\omega - \lambda) - \varepsilon|K(i\omega - \lambda)|^2) > 0, \end{aligned} \tag{30}$$

where $K(p) = c^*(A - pI)^{-1}b - \rho$, are satisfied, then by the Yakubovich–Kalman frequency theorem there exists H such that for ξ and all $z \neq 0$ the following relation

$$2z^*H[(A+\lambda I)z + b\xi] + \xi(c^*z + \rho\xi) + \varepsilon|(c^*z + \rho\xi)|^2 < 0$$

is valid. Here ε is a positive number. If $A + \lambda I$ is a stable matrix, then $H > 0$.
Thus, if $(A + \lambda I)$ is stable, (30) and $\alpha^2 \leq 4\lambda\varepsilon$ are satisfied, then we have

$$\frac{dV}{dt} + 2\lambda V < 0, \ \ \forall z(t) \neq 0$$

and, therefore, Ω is a positively invariant cone.
We can make a breeding of the cones $\Omega_k = \{z^* Hz - \frac{1}{2}\eta_k(\sigma)^2 \leq 0\}$ in the same way as in the last section and construct a cone grid (Fig. 6), using these cones. Here $\eta_k(\sigma)$ is the solution $\eta(\sigma)$, shifted along the axis σ by the quantity $2k\pi$. The cone grid is a proof of boundedness of solutions of system (26) on the interval $(0, +\infty)$. Under these assumptions there occurs a dichotomy. This is easily proved by using the Lyapunov function

$$z^* Hz + \int_0^{\sigma} \psi(\sigma) d\sigma.$$

Thus we prove the following

Theorem 5. *If for certain $\lambda > 0$ and $\varepsilon > 0$ the matrix $A + \lambda I$ is stable, conditions (30) are satisfied, and system (27) with $\alpha = 2\sqrt{\lambda\varepsilon}$ is a globally stable system (all solutions tend to stationary set as $t \to +\infty$), then system (26) is also a a globally stable system.*

Various generalizations of this theorem and numerous examples of applying the method of nonlocal reduction, including the applying to synchronous machines, can be found in the works (Leonov, 1975; Leonov, 1976; Gelig et al., 1978; Leonov et al., 1992; Leonov et al., 1996a). Various criteria for the existence of circular solutions and second-kind cycles were also obtained within the framework of this method (Gelig et al., 1978; Leonov et al., 1992; Leonov et al., 1996a; Yakubovich et al., 2004).

5. Floating PLL

The main requirement to PLLs for digital signal processors is that they must be floating in phase. This means that the system must eliminate the clock skew completely. Let us clarify the problem of eliminating the clock skew in multiprocessor systems when parallel algorithms are applied. Consider a clock C transmitting clock pulses through a bus to processors P_k operating in parallel (Fig. 8).

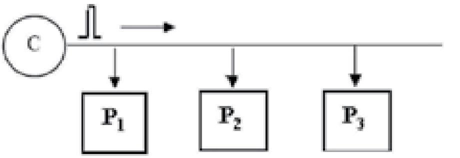


Fig. 8. Clock C that transmits clock pulses through a bus to processors P_k working in parallel.

In realizing parallel algorithms, the processors must perform a certain sequence of operations simultaneously. These operations are to be started at the moments of arrival of clock pulses to processors. Since the paths along which the pulses run from the clock to every processor are of different length, a mistiming between processors arises. This phenomenon is called a clock skew.

The elimination of the clock skew is one of the most important problems in parallel computing and information processing (as well as in the design of array processors (Kung, 1988)).

Several approaches to the solution of the problem of eliminating the clock skew have been devised for the last thirty years.

In developing the design of multiprocessor systems, a way was suggested (Kung, 1988; Saint-Laurent et al., 2001) for joining the processors in the form of an *H*-tree, in which (Fig. 9) the lengths of the paths from the clock to every processor are the same.

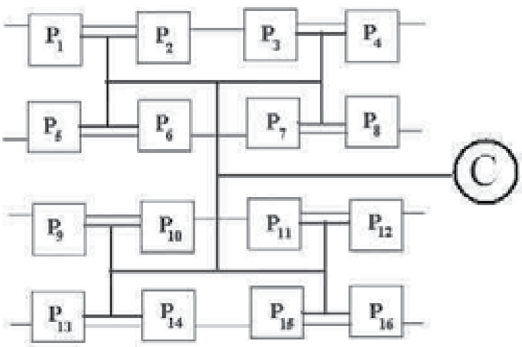


Fig. 9. Join of processors in the form of an *H*-tree.

However, in this case the clock skew is not eliminated completely because of heterogeneity of the wires (Kung, 1988). Moreover, for a great number of processors, the configuration of communication wires is very complicated. This leads to difficult technological problems.

The solution of the clock skew problem at a hard- and software levels has resulted in the invention of asynchronous communication protocols, which can correct the asynchronism of operations by waiting modes (Kung, 1988). In other words, the creation of these protocols permits one not to distort the final results by delaying information at some stages of the execution of a parallel algorithm. As an advantage of this approach, we may mention the fact that we need not develop a special complicated hardware support system. Among the disadvantages we note the deceleration of performance of parallel algorithms.

In addition to the problem of eliminating the clock skew, one more important problem arose. An increase in the number of processors in multiprocessor systems required an increase in the power of the clock. But powerful clocks lead to produce significant electromagnetic noise.

About ten years ago, a new method for eliminating the clock skew and reducing the generator's power was suggested. It consists of introducing a special distributed system of clocks controlled by PLLs. An advantage of this method, in comparison with asynchronous communication protocols, is the lack of special delays in the performance of parallel algorithms. This approach allows one to reduce significantly the power of clocks.

Consider the general scheme of a distributed system of oscillators (Fig. 10).

By Theorem 2, we can make the design of a block diagram of floating PLL, which plays a role of the function of frequency synthesizer and the function of correction of the clock-skew (see parameter τ in Fig. 11). Its block diagram differs from that shown in Fig. 4 with the phase

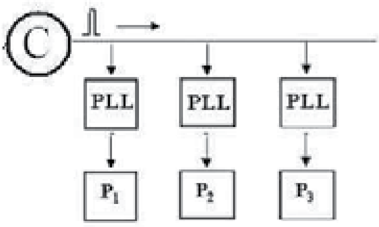


Fig. 10. General scheme of a distributed system of oscillators with PLL.

detector characteristic (11) only in that a relay element with characteristic sign G is inserted after the filter.
Such a block diagram is shown in Fig. 11.

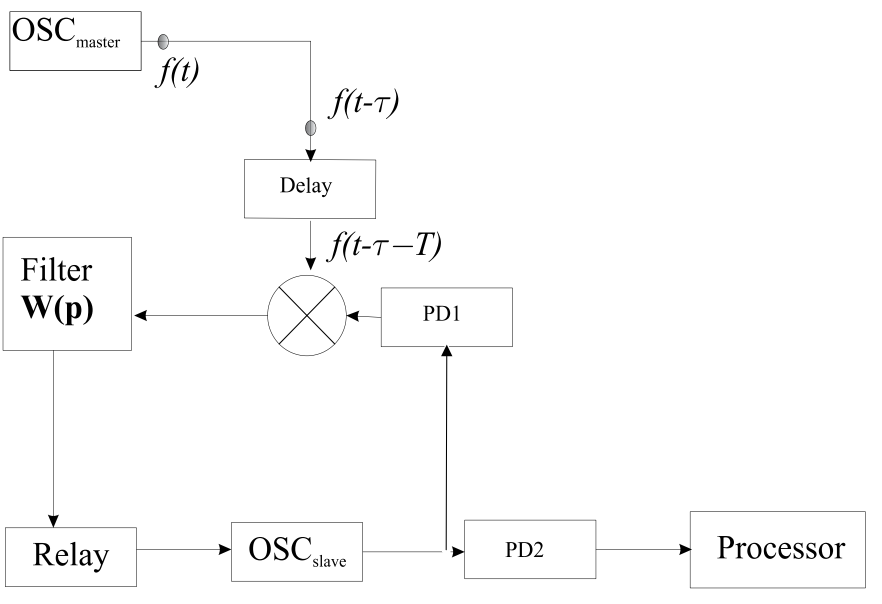


Fig. 11. Block diagram of floating PLL

Here OSC_{master} is a master oscillator, *Delay* is a time-delay element, *Filter* is a filter with transfer function

$$W(p) = \frac{\beta}{p + \alpha},$$

OSC_{slave} is a slave oscillator, PD1 and PD2 are programmable dividers of frequencies, and *Processor* is a processor.
The *Relay* element plays a role of a floating correcting block. The inclusion of it allows us to null a residual clock skew, which arises for the nonzero initial difference of frequencies of master and slave oscillators.

We recall that it is assumed here that the master oscillator $\dot{\theta}_1(t) \equiv \omega_1(t) \equiv \omega_1(0) = \omega_1$ is highly stable. The parameter of delay line T is chosen in such a way that $\omega_1(0)(T+\tau) = 2\pi k + 3\pi/2$. Here k is a certain natural number, $\omega_1(0)\tau$ is a clock skew.
By Theorem 2 and the choice of T the block diagram, shown in Fig. 11, can be changed by the close block diagram, shown in Fig. 12. Here $\varphi(\theta)$ is a 2π-periodic characteristic of phase detector. It has the form

$$\varphi(\theta) = \begin{cases} 2A_1A_2\theta/\pi & \text{for}\, \theta \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ \\ 2A_1A_2(1-\theta/\pi) & \text{for}\, \theta \in [\frac{\pi}{2}, \frac{3\pi}{2}], \end{cases} \tag{31}$$

$\theta_2(t) = \dfrac{\theta_3(t)}{M}$, $\theta_4(t) = \dfrac{\theta_3(t)}{N}$, where the natural numbers M and N are parameters of programmable divisions PD1 and PD2, respectively.

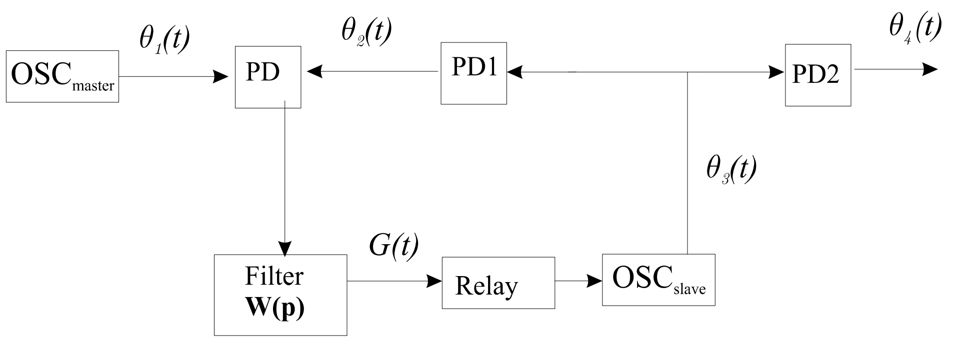


Fig. 12. Equivalent block diagram of PLL

For a transient process (a capture mode) the following conditions

$$\lim_{t\to+\infty}\left(\theta_4(t) - \frac{M}{N}\theta_1(t)\right) = \frac{2\pi k M}{N} \tag{32}$$

(a phase capture) and

$$\lim_{t\to+\infty}\left(\dot{\theta}_4(t) - \frac{M}{N}\dot{\theta}_1(t)\right) = 0 \tag{33}$$

(a frequency capture), must be satisfied.
Relations (32) and (33) are the main requirements to PLL for array processors. The time of transient processors depends on the initial data and is sufficiently large for multiprocessor system (Kung, 1988; Leonov & Seledzhi, 2002).
Assuming that the characteristic of relay is of the form $\Psi(G) = \text{sign}G$ and the actuating element of slave oscillator is linear, we have

$$\dot{\theta}_3(t) = R\text{sign}G(t) + \omega_3(0), \tag{34}$$

where R is a certain number, $\omega_3(0)$ is the initial frequency, and $\theta_3(t)$ is a phase of slave oscillator.

Taking into account relations (34), (1), (31) and the block diagram in Fig. 12, we have the following differential equations of PLL

$$\begin{aligned} &\dot{G} + \alpha G = \beta\varphi(\theta), \\ &\dot{\theta} = -\frac{R}{M}\text{sign}G + \left(\omega_1 - \frac{\omega_3(0)}{M}\right). \end{aligned} \tag{35}$$

Here $\theta(t) = \theta_1(t) - \theta_2(t)$. In general case, we get the following PLL equations:

$$\begin{aligned} &\dot{z} = Az + b\varphi(\sigma) \\ &\dot{\sigma} = g(c^*z), \end{aligned} \tag{36}$$

where $\sigma = \theta_1 - \theta_2$, the matrix A and the vectors b and c are such that

$$W(p) = c^*(A - pI)^{-1}b,$$

$$g(G) = -L(\text{sign}\,G) + \big(\omega_1(0) - \omega_2(0)\big).$$

Rewrite system (35) as follows

$$\begin{aligned} &\dot{G} = -\alpha G + \beta\varphi(\theta), \\ &\dot{\theta} = -F(G), \end{aligned} \tag{37}$$

where

$$F(G) = \frac{R}{M}\text{sign}G - \left(\omega_1 - \frac{\omega_3(0)}{M}\right).$$

Theorem 6. *If the inequality*

$$|R| > |M\omega_1 - \omega_3(0)| \tag{38}$$

is valid, then any solution of system (37) tends to a certain equilibrium state as $t \to +\infty$. If the inequality

$$|R| < |M\omega_1 - \omega_3(0)| \tag{39}$$

is valid, then all the solutions of system (37) tend to infinity as $t \to +\infty$.

Consider equilibrium states for system (37). For any equilibrium state we have

$$\dot{\theta}(t) \equiv 0, \quad G(t) \equiv 0, \quad \theta(t) \equiv \pi k.$$

Theorem 7. *We assume that relation (38) is valid. In this case if $R > 0$, then the following equilibria*

$$G(t) \equiv 0, \quad \theta(t) \equiv 2k\pi \tag{40}$$

are locally asymptotically stable and the following equilibria

$$G(t) \equiv 0, \quad \theta(t) \equiv (2k+1)\pi \tag{41}$$

are locally unstable. If $R < 0$, then equilibria (41) are locally asymptotically stable and equilibria (40) are locally unstable.

Thus, for relations (32) and (33) to be satisfied it is necessary to choose the parameters of system in such a way that the inequality holds

$$R > |M\omega_1 - \omega_3(0)|. \tag{42}$$

Proofs of Theorems 6 and 7. Let $R > |M\omega_1 - \omega_3(0)|$. Consider the Lyapunov function

$$V(G,\theta) = \int_0^G \Phi(u)du + \beta \int_0^\theta \varphi(u)du,$$

where $\Phi(G)$ is a single-valued function coinciding with $F(G)$ for $G \neq 0$. For $G = 0$, the function $\Phi(G)$ can be defined arbitrary. At points t such that $G(t) \neq 0$, we have

$$\dot{V}\big(G(t),\theta(t)\big) = -\alpha G(t)F(G(t)). \tag{43}$$

Note that, for $G(t) = 0$, the first equation of system (35) yields

$$\dot{G}(t) \neq 0 \text{ for } \theta(t) \neq k\pi.$$

It follows that there are no sliding solutions of system (35). Then, relation (43) and the inequality $F(G)G > 0, \ \forall G \neq 0$ imply that conditions 3) and 4) of Theorem 3 are satisfied. Moreover, $V(G,\theta + 2\pi) \equiv V(G,\theta)$ and $V(G,\theta) \to +\infty$ as $G \to +\infty$. Therefore, conditions (1) and (2) of Theorem 3 are satisfied. Hence, any solution of system (35) tends to the stationary set as $t \to +\infty$. Since the stationary set of system (35) consists of isolated points, any solution to system (35) tends to equilibrium state as $t \to +\infty$.
If the inequality

$$-R > |M\omega_1 - \omega_3(0)|, \tag{44}$$

is valid, then, in place of the function $V(G,\theta)$, one should consider the Lyapunov function $W(G,\theta) = -V(G,\theta)$ and repeat the above considerations.
Under inequality (39), we have the relation $F(G) \neq 0$, $\forall G \in R^1$. Together with the second equation of system (37), this implies that

$$\lim_{t\to+\infty} \theta(t) = \infty.$$

Thus, Theorem 6 is completely proved.
To prove Theorem 7, we note that if condition (42) holds in a neighbourhood of points $G = 0$, $\theta = 2\pi k$, then the function $V(G,\theta)$ has the property

$$V(G,\theta) > 0 \quad \text{for} \quad |G| + |\theta - 2k\pi| \neq 0.$$

Together with equality (43), this implies the asymptotic stability of these equilibrium states.
In a neighbourhood of points $G = 0$, $\theta = (2k+1)\pi k$, the function $V(G,\theta)$ has the property $V(0,\theta) < 0 \quad \text{for} \quad \theta \neq (2k+1)\pi$. Together with equality (43), this implies the instability of these equilibrium states.
If inequality (44) holds, then, in place of the function $V(G,\theta)$, we can consider the function $W(G,\theta) = -V(G,\theta)$ and repeat the considerations. ■

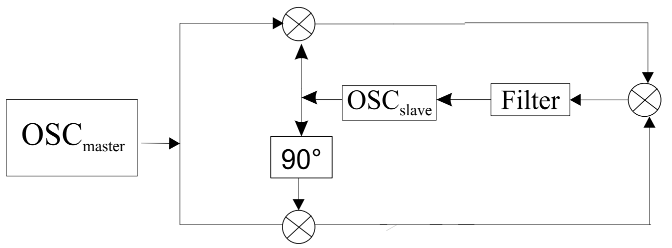


Fig. 13. Costas loop

6. Costas loop

Consider now a block diagram of the Costas loop (Fig. 13). Here all denotations are the same as in Fig. 1, 90^o is a quadrature component. As before, we consider here the case of the high-frequency harmonic and impulse signals $f_j(t)$.
However together with the assumption that conditions (3) and (5) are valid we assume also that (4) holds for the signal of the type (1) and the relation

$$|\omega_1(\tau) - 2\omega_2(\tau)| \leq C_1, \quad \forall\, \tau \in [0, T], \tag{45}$$

is valid for the signal of the type (10). Applying the similar approach we can obtain differential equation for the Costas loop, where

$$\begin{aligned} \dot{z} &= Az + b\Psi(\sigma), \\ \dot{\sigma} &= c^* z + \rho\Psi(\sigma). \end{aligned} \tag{46}$$

Here A is a constant $n \times n$-matrix, b and c are constant n-vectors, ρ is a number, and $\Psi(\sigma)$ is a 2π- periodic function, satisfying the following relations

$$\begin{aligned} &\psi(\sigma) = \frac{1}{8}A_1^2 A_2^2 \sin\sigma - \frac{\omega_1(0) - \omega_2(0)}{L(a + W(0))}, \\ &\sigma = 2\theta_1 - 2\theta_2 \end{aligned}$$

in the case of harmonic oscillations (1) and

$$\begin{aligned} &\psi(\sigma) = P(\sigma) - \frac{\omega_1(0) - 2\omega_2(0)}{2L(a + W(0))}, \\ &P(\sigma) = \begin{cases} -2A_1^2 A_2^2 \left(1 + \dfrac{2\sigma}{\pi}\right), & \sigma \in [0, \pi] \\ -2A_1^2 A_2^2 \left(1 - \dfrac{2\sigma}{\pi}\right), & \sigma \in [-\pi, 0] \end{cases} \\ &\sigma = \theta_1 - 2\theta_2 \end{aligned}$$

in the case of impulse oscillations (10), where $\rho = -2aL, \quad W(p) = (2L)^{-1}c^*(A - pI)^{-1}b$. This implies that for deterministic (when the noise is lacking) description of the Costas loops the conventional introduction of additional filters turns out unnecessary. Here a central filter plays their role.

7. Bifurcations in digital PLL

For the study of stability of discrete systems, just as in the case of continuous ones, a discrete analog of direct Lyapunov method can be applied. By the frequency theorem similar to that of Yakubovich – Kalman for discrete systems (Leonov & Seledzhi, 2002), the results of applying the direct Lyapunov method can be formulated in the form of frequency inequalities. In the same way as in the continuous case, the phase systems have certain specific properties and for these systems it is necessary to use the direct Lyapunov method together with another research methods. For discrete systems there exist analogs of the method of cone grids and the reduction procedure of Bakaev–Guzh (Bakaev, 1959; Leonov & Smirnova, 2000).
Here we consider bifurcation effects, arising in discrete models of PLL (Osborne, 1980; Leonov & Seledzhi, 2002).
Discrete phase-locked loops (Simpson, 1994; Lapsley et al., 1997; Smith, 1999; Solonina et al., 2000; Solonina et al., 2001; Aleksenko, 2002; Aleksenko, 2004) with sinusoidal characteristic of phase discriminator are described in details in (Banerjee & Sarkar, 2008). Here a description of bifurcations of a filter-free PLL with a sine-shaped characteristic of phase detector (see (Belykh & Lebedeva, 1982; Leonov & Seledzhi, 2002; Lindsey & Chie, 1981; Osborne, 1980)) is considered. If the initial frequencies of the master and slave oscillators coincide, then the equation of the PLL is of the form

$$\sigma(x+1) = \sigma(x) - r \sin\big(\sigma(x)\big), \tag{47}$$

where r is a positive number. It is easy to see (Abramovich et al., 2005; Leonov & Seledzhi, 2002; Leonov & Seledzhi, 2005a) that this system is globally asymptotically stable for $r \in (0, 2)$. Now we study equation (47) for $r > 2$. Let $r \in (2, r_1)$, where r_1 is a root of the equation $\sqrt{r^2 - 1} = \pi + arccos\frac{1}{r}$. Then (47) maps $[-\pi, \pi]$ into itself, that is $\sigma(t) \in [-\pi, \pi]$ for $\sigma(0) \in [-\pi, \pi]$ $t = 1, 2, \dots$.
In the system (47) there is transition to chaos via the sequence of period doubling bifurcations (Fig. 14). Equation (47) is not unimodal, so we can not directly apply the usual Renorm-Group

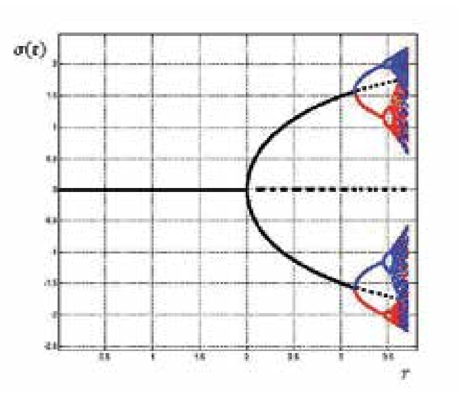


Fig. 14. Bifurcation tree.

method for its analytical investigation. Some first bifurcation parameters can be calculated analytically (Osborne, 1980), the others can be found only by means of numerical calculations (Abramovich et al., 2005; Leonov & Seledzhi, 2005a).

The first 13 calculated bifurcation parameters of period doubling bifurcations of (47) are the following

$$
\begin{array}{rr}
r_1 = 2 & r_2 = \pi \\
r_3 = 3.445229223301312 & r_4 = 3.512892457411257 \\
r_5 = 3.527525366711579 & r_6 = 3.530665376391086 \\
r_7 = 3.531338162105000 & r_8 = 3.531482265584890 \\
r_9 = 3.531513128976555 & r_{10} = 3.531519739097210 \\
r_{11} = 3.531521154835959 & r_{12} = 3.531521458080261 \\
r_{13} = 3.531521523045159 &
\end{array}
$$

Here r_2 is bifurcation of splitting global stable cycle of period 2 into two local stable cycles of period 2. The other r_j correspond to period doubling bifurcations.

We have here following Feigenbaum's effect of convergence for $\delta_{n+1} = \dfrac{r_{n+1} - r_n}{r_{n+2} - r_{n+1}}$:

$$
\begin{array}{rr}
\delta_2 = 3.759733732581654 & \delta_3 = 4.487467584214882 \\
\delta_4 = 4.624045206680584 & \delta_5 = 4.660147831971297 \\
\delta_6 = 4.667176508904449 & \delta_7 = 4.668767988303247 \\
\delta_8 = 4.669074658227896 & \delta_9 = 4.669111696537520 \\
\delta_{10} = 4.669025736544542 & \delta_{11} = 4.668640891299296 \\
\delta_{12} = 4.667817727564633. &
\end{array}
$$

8. Conclusion

The theory of phase synchronization was developed in the second half of the last century on the basis of three applied theories: theory of synchronous and induction electrical motors, theory of auto-synchronization of the unbalanced rotors, theory of phase-locked loops. Its main principle is consideration of the problems of phase synchronization at the three levels:
(i) at the level of mechanical, electromechanical, or electronic model,
(ii) at the level of phase relations,
(iii) at the level of difference, differential, and integro-differential equations.
In this case the difference of oscillation phases is transformed in the control action, realizing synchronization. These general principles gave impetus to creation of universal methods for studying the phase synchronization systems. Modification of the direct Lyapunov method with the construction of periodic Lyapunov-like functions, the method of positive invariant cone grids, and the method of nonlocal reduction turned out to be most effective. The last method, which combines the elements of the direct Lyapunov method and the bifurcation theory, allows one to extend the classical results of F. Tricomi and his progenies to the multidimensional dynamical systems.

9. Acknowledgment

This work partly supported by projects of Ministry of education and science of RF, Grants board of President RF, RFBR and CONACYT.

10. References

Abramovitch, D. (2002). Phase-Locked Loops: A control Centric Tutorial, *Proceedings of the American Control Conference 2002*. vol. 1, pp. 1–15

Abramovich, S., Kudryashova, E., Leonov, G.A. & Sugden, S. (2005). Discrete Phase-Locked Loop Systems and Spreadsheets, *Spreadsheets in Education (eJSiE)*,vol. 2, Issue 1
Aleksenko, A.G. (2004). *Digital engineering*, Unimedstyle, Moscow (in Russian)
Alekseenko, A.G. (2002). Fundamentals of Microcurcuit, Yunimediastail, Moscow (in Russian)
Bakaev, Yu.N. (1959). Some Questions of the Nonlinear Theory of Phase Systems, *Proc. Zhukovskii VVIA*, vol. 800
Banerjee, D. (2006). *Pll Performance, Simulation and Design*, Ed. 4. Dog Ear Publishing
Banerjee, T. & Sarkar, B.C. (2008). Chaos and bifurcation in a third-order digital phase-locked loop, *Int. J. Electron. Commun.*, 62, pp. 86–91
Barbashin, E.A. & Krasovskii, N.N. (1952). On Motion Stability in the Large, *Dokl. Akad. Nauk SSSR*, vol. 86, no. 3, pp. 453–459
Barbashin, E.A. & Tabueva, V.A. (1969). *Dynamic Systems with Cylindrical Phase Space*, Nauka, Moscow (in Russian)
De Bellescize, H. (1932). *La Reseption synchrone*, Onde Electrique
Belykh, V.N. & Lebedeva, L.N. (1982). Studying Map of Circle, *Prikl. Mat. Mekh.*, no. 5, pp. 611–615
Best Ronald, E. (2003). *Phase-Lock Loops: Design, Simulation and Application* 5^{ed}, McGraw Hill
Bianchi, G. (2005). *Phase-locked Loop Synthesizer Simulation*, McGraw-Hill
Bindal, N., Kelly, T., Velastegui, N. & Wong, K.L. (2003). Scalable sub-10ps Skew Global Clock Distribution for a 90nm Multi-GHz IA Microprocessor, *Proceedings of IEEE International Solid-State Circuits Conference, Digest of Technical Papers. ISSCC.* vol. 1, pp. 346–498
Brennan, P.V. (1996). *Phase-locked loops: principles and practice*, Macmillan Press
Buchanan, W. & Wilson, A. (2001). *Advanced PC architecture*, Addison-Wesley
Chen, Wai-Kai. (2003). *The circuits and filters handbook*, CRC Press
Gardner, F. (1979). *Phase–lock techniques*, 2^{ed}, John Wiley & Sons
Giannini, F. & Leuzzi, G. (2004). *Nonlinear microwave circuit design*, John Wiley & Sons
Gelig, A.Kh., Leonov, G.A. & and Yakubovich, V.A. (1978). *Stability of Nonlinear Systems with Nonunique Equilibrium State*, Nauka, Moscow (in Russian)
Goldman, S.J. (2007). *Phase-locked Loop Engineering Handbook for Integrated Circuits*, Artech House
Egan, W. (2007). *Phase-Lock Basics*, Wiley-IEEE
Egan, W. (2000). *Frequency Synthesis by Phase Lock*, 2nd ed., John Wiley & Sons
Encinas, J. (1993). *Phase Locked Loops*, Springer
Kroupa, V. (2003). *Phase Lock Loops and Frequency Synthesis*, John Wiley & Sons
Kudrewicz, J. & Wasowicz S. (2007). *Equations of Phase-Locked Loops: Dynamics on the Circle, Torus and Cylinder*, World Scientific
Kung, S. (1988). *VLSI Array Processors*, Prentice Hall
Kuznetsov, N.V. (2008). *Stability and Oscillations of Dynamical Systems: Theory and Applications*, Jyväskylä University Printing House, Jyväskylä
Kuznetsov, N.V., Leonov, G.A. & Seledzhi, S.M. (2008). Phase Locked Loops Design And Analysis, *Int. Conference on Informatics in Control, Automation and Robotics, Proceedings*, pp. 114–118
Kuznetsov, N.V., Leonov, G.A. & Seledzhi S.M. (2007). Global stability of phase-locked loops. *3rd Int. IEEE conference on Physics and Control* (http://lib.physcon.ru/?item=1192)

Kuznetsov, N.V., Leonov, G.A. & Seledzhi, S.M. (2006). Analysis of phase-locked systems with discontinuous characteristics of the phase detectors. *Preprints of 1st IFAC conference on Analysis and control of chaotic systems*, pp. 127–132

Lapsley, P., Bier, J., Shoham, A. & Lee, E. (1997). *DSP Processor Fundamentals Architecture and Features*, IEE Press

Leonov, G.A. (1974). On Boundedness of the Trajectories of the Phase Systems, *Sib. Mat. Zh.*, vol. 15, no. 3, pp. 687–692

Leonov, G.A. (1975). Stability and Oscillaitons of the Phase Systems, *Sib. Mat. Zh.*, vol. 16, no. 5, pp. 1031–1052

Leonov, G.A. (1976). Second Lyapunov Method in the Theory of Phase Synchronization, Prikl. Mat. Mekh., 1976, vol. 40, no. 2, pp. 238–244

Leonov, G., Reitmann, V. & Smirnova, V. (1992). *Nonlocal Methods for Pendulum-Like Feedback Systems*, Teubner Verlagsgesselschaft, Stuttgart–Leipzig

Leonov, G., Ponomarenko, D. & Smirnova, V. (1996a). *Frequency-Domain Methods for Non-linear Analysis. Theory and Applications*, World Scientific

Leonov, G.A., Burkin, I.M. & Shepeljavy, A.I. (1996b). *Frequency Methods in Oscillation Theory*, Kluver, Dordrecht

Leonov, G.A. & Smirnova, V.B. (2000). *Mathematical Problems of the Phase Synchronization Theory*, Nauka, St. Petersburg

Leonov, G.A. (2001). *Mathematical Problems of Control Theory*, World Scientific

Leonov, G. A. & Seledzhi, S.M. (2002). *Phase-locked loops in array processors*, Nevsky dialekt, St.Petersburg (in Russian)

Leonov, G.A. & Seledghi, S.M. (2005a). Stability and bifurcations of phase-locked loops for digital signal processors, *International journal of bifurcation and chaos*, 15(4), pp. 1347–1360

Leonov, G.A. & Seledghi, S.M. (2005). Design of phase-locked loops for digital signal processors, *International Journal of Innovative Computing, Information Control*, 1(4), pp. 779–789

Leonov, G.A. (2006). Phase-Locked Loops. Theory and Application, *Automation and remote control*, 10, pp. 47–55

Leonov, G.A. (2008). Computation of phase detector characteristics in phase-locked loops for clock synchronization, *Doklady Mathematics*, vol. 78, no. 1, pp. 643–645

Lindsey, W. (1972). *Sinchronization systems in communication and control*, Prentice-Hall

Lindsey, W. & Chie, C. (1981). A Survey of Digital Phase Locked Loops. *Proceedings of the IEEE*, vol. 69, pp. 410–431

Manassewitsch, V. (2005). *Frequency synthesizers: theory and design*, Wiley

Margaris, N.I. (2004). *Theory of the Non-Linear Analog Phase Locked Loop*, Springer Verlag

Nash, G. (1994). *Phase Locked Loop, Design Fundamentals*, Motorola, Phoenix

Noldus, E., (1977). New Direct Lyapunov-type Method for Studying Synchronization Problems, Automatika, 1977, vol. 13, no. 2, pp. 139–151

Osborne, H.C. (1980). Stability Analysis of an N-th Power Digital Phase Locked Loop. Part. I: First-order DPLL, *IEEE Trans. Commun.*, vol. 28, pp. 1343–1354

Razavi, B. (2003). *Phase-Locking in High-Performance Systems: From Devices to Architectures*, John Wiley & Sons

Razavi, B. (2001). *Design of Analog CMOS Integrated Circuits*, McGraw Hill

Saint-Laurent, M., Swaminathan, M. & Meindl, J.D. (2001). On the Micro-Architectural Impact of Clock Distribution Using Multiple PLLs, In *Proceedings of International Conference on Computer Design*, pp. 214–220

Shakhgildyan, V.V. & Lyakhovkin, A.A. (1972). *Sistemy fazovoi avtopodstroiki chastoty (Phase Locked Loops)*, Svyaz, Moscow (in Russian)

Shu, K. & Sanchez-Sinencio, E. (2005). *CMOS PLL Synthesizers: Analysis and Design*, Springer

Simpson, R. (1994). *Digital Signal Processing Using The Motorola DSP Family*, Prentice Hall

Smith, S.W. (1999). *The Scientist and Engineers Guide to Digital Signal Processing*, California Technical Publishing, San Diego

Solonina, A., Ulakhovich, D. & Yakovlev, L. (2001). *Algorithms and Processors for Digital Signal Processing*, BHV, St. Petersburg (in Russian)

Solonina, A., Ulahovich, D. & Jakovlev, L. (2000). *The Motorola Digital Signal Processors*, BHV, St. Petersburg (in Russian)

Stensby, J.L. (1997). *Phase-locked Loops: Theory and Applications*. CRC Press

Stephens, D.R. (2002). *Phase-Locked Loops For Wireless Communications (Digital, Analog and Optical Implementations)*, Second Edition, Kluwer Academic Publishers

Suarez, A. & Quere, R. (2003). *Stability Analysis of Nonlinear Microwave Circuits*, Artech House

Ugrumov, E. (2000). *Digital engineering*, BHV, St.Petersburg (in Russian)

Viterbi, A. (1966). *Principles of coherent communications*, McGraw-Hill

Wainner, S. & Richmond, R. (2003). *The book of overclocking: tweak your PC to unleash its power*. No Starch Press

Wendt, K. & Fredentall, G. (1943). Automatic frequency and phase control of synchronization in TV receivers, *Proc. IRE*, vol. 31, no. 1, pp. 1–15

Wolaver, D.H. (1991). *Phase-locked Loop Circuit Design*, Prentice Hall

Xanthopoulos, T., et al. (2001). The Design and Analysis of the Clock. Distribution. Network for a 1.2GHz Alpha Microprocessor, *IEEE ISSCC Tech. Dig.*, pp. 232–233

Yakubovich, V., Leonov, G. & Gelig, A. (2004). *Stability of Systems with Discontinuous Nonlinearities*, World Scientisic

Young, I.A., Greason, J. & Wong, K. (1992). A PLL clock generator with 5 to 110MHz of lock range for microprocessors. *IEEE J. Solid-State Circuits*, vol. 27, no. 11, pp. 1599–1607

8

Methods for parameter estimation and frequency control of piezoelectric transducers

Constantin Volosencu
"Politehnica" University of Timisoara
Romania

1. Introduction

This chapter presents some considerations related to parameter identification and frequency control at piezoelectric transducers used in high power ultrasonic applications as: metal or plastic welding, ultrasonic cleaning and other. Piezoelectric transducers are used as actuators, which convert electric power in acoustic power at ultrasonic frequencies. In practical design it is important to have information about their equivalent electrical circuit: parameters and resonance frequency. The main purpose of research is to develop control systems for piezoelectric transducers with a higher efficiency of the energy conversion and greater frequency stability. In the world literature there are many publications treating the domain of transducer measurement and frequency control. Some of these references, related to the chapter theme, are presented as follows.

Piezoelectric transducers (Gallego-Juarez, 2009) have proved their huge viability in the high power ultrasonic applications as cleaning, welding, chemical or biological activations and other for many years (Hulst, 1972), (Neppiras, 1972). And these applications continue to be of a large necessity. The power ultrasonic transducers are fed with power inverters, using transistors working in commutation at high frequency (Bose, 1992). A large scale of electronic equipments, based on analogue or digital technology, is used for control in the practical applications (Marchesoni, 1992). A good efficiency of the energy conversion in the power ultrasonic equipments is very important to be assured. Different control methods are used in practice to control the signal frequency in the power inverters (Senchenkov, 1991), (Khemel & all, 2001) and many other. The high power ultrasonic piezoelectric transducers are analysed with complex structures by using equivalent circuits, starting from Mason's model, implemented on circuit analysis programs (Morris, 1986). Impedance measurement is done using different methods (Chen & all, 2008). The wave guides are analysed in the assemble of ultrasonic systems: transducer- wave guide - process (Mori, 1989). Books on ultrasonic are presenting modelling and design methods (Prokic, 2004), (Radmanovici & all, 2004). There are many technical solutions reported in practice, for example (Fuichi & Nose, 1981), (Khmelev & all, 2001), (Kirsh & Berens, 2006) or (Sulivan, 1983).

In the beginning of this chapter some general consideration related to the equivalent electrical circuit of the piezoelectric transducers and their characteristics useful in practice are presented. In the second paragraph a parameter estimation method based on the transformation from the Nyquist hodograph from the frequency domain to a parameter model - the transfer function of the transducer's impedance, is presented. In the third paragraph a second parameter estimation method is based on an automatic measurement of piezoelectric transducer impedance using a deterministic convergence scheme with a gradient method with continuous adjustment. In the end the chapter provides a method for frequency control at ultrasonic high power piezoelectric transducers, using a feedback control systems based on the first derivative of the movement current.

2. Ultrasonic piezoelectric transducers

2.1 Constructive and functional characteristics

The ultrasonic piezoelectric transducers are made in a large domain of power from ten to thousand watts, in a frequency range of 20 kHz – 2 MHz. Example of characteristics of some commercial transducers are given in Tab. 1.

Transducer type	P [W]	f_s [KHz]	f_p [KHz]	m [Kg]	I [mA]	Constr. type	C_0 [nF]
TGUS 100-020-2	100	20±1	22±2	0,65	300	2	4,2±0,6
TGUS 100-025-2	100	25±1	27±2	0,6	300	2	4,2±0,6
TGUS 150-040-1	150	40±2	43±2	0,26	300	1, 2	4,1±0,6
TGUS 500-020-1	500	20±1	22±2	1,1	500	1	5,8±0,6

Table 1. Characteristics of some piezoelectric transducers made at I.F.T.M. Bucharest

The 1st type is for general applications and the 2nd type is for ultrasonic cleaning to be mounted on membranes. Two examples of piezoelectric transducers TGUS 150-040-1 and TGUS 500-25-1 are presented in Fig. 1.

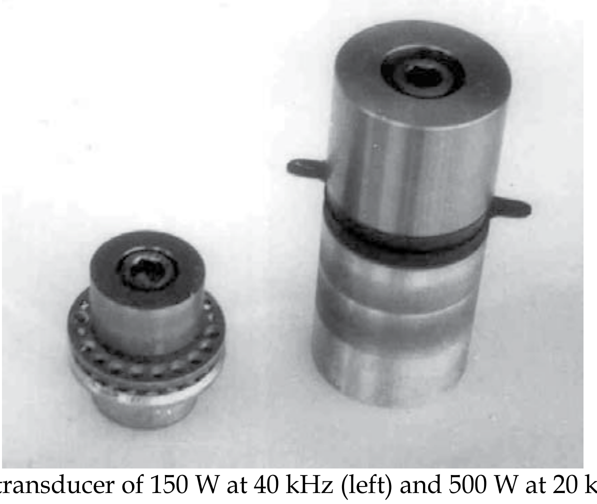


Fig. 1. Piezoelectric transducer of 150 W at 40 kHz (left) and 500 W at 20 kHz (right)

They have small losses, a good coupling coefficient k_{ef}, a good quality mechanical coefficient Q_{m0} and a high efficiency η_0:

$$k_{ef} = \sqrt{1 - \left(\frac{f_s}{f_p}\right)^2}\ ,\ Q_{m0} = \frac{f_p}{\Delta f_p}\ ,\ \eta_0 = 1 - \frac{2}{k_{ef}}\sqrt{\frac{tg\delta}{Q_{m0}}} \tag{1}$$

in normal operating conditions of temperature, humidity and atmospheric pressure.

3. Electrical characteristics

The high power ultrasonic installations have as components ultrasonic generator piezoelectric transducers, which are accomplish some technical conditions. They have the electrical equivalent circuit from Fig. 2.

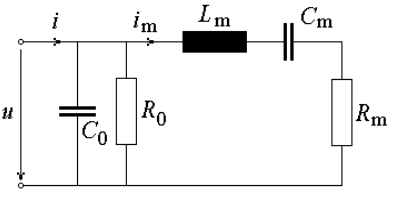


Fig. 2. The simplified linear equivalent electrical circuit

Their magnitude-frequency characteristic is presented in Fig. 3.

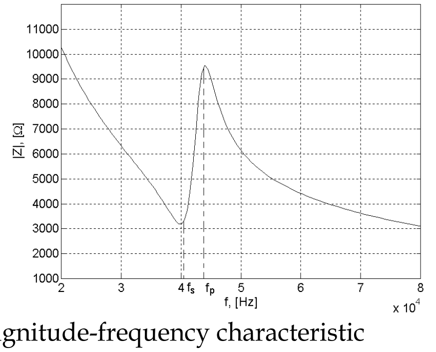


Fig. 3. The impedance magnitude-frequency characteristic

We may notice on this characteristic a series resonant frequency f_s and a parallel resonant frequency f_p, placed at the right. The magnitude has the minimum value Z_m at the series frequency and the maximum value Z_M at the parallel resonant frequency, on bounded domain of frequencies. The piezoelectric transducer is used in the practical applications working at the series resonant frequency.

The most important aspect of this magnitude characteristic is the fact that the frequency characteristic is modifying permanently in the transient regimes, being affected by the load applied to the transducer, in the following manner: the minimum impedance Z_m is increasing, the maximum impedance Z_M is decreasing and also the frequency bandwidth [f_s, f_p] is modifying in specific ways according to the load types. So, when at the transducer a concentrator is coupled, as in Fig. 4, the frequency bandwidth [f_s, f_p] became very narrow, as f_p- $f_s \approx$ 1-2 Hz.

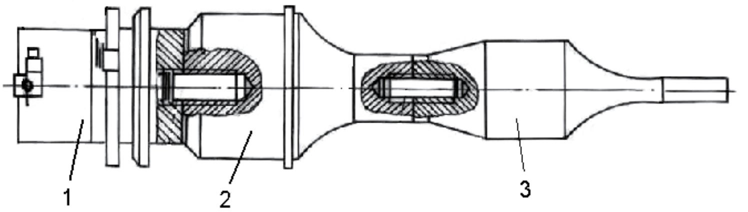


Fig. 4. A transducer 1 with a concentrator 2 and a welding tool 3

This is a great impediment because in this case a very précised and stable frequency control circuit is necessary at the electronic ultrasonic power generator for the feeding voltage of the transducer.
When at the transducer a horn or a membrane is mounted, as in Fig. 5, the frequency bandwidth [f_s, f_p] increases for 10 times, f_p- $f_s \approx n$ kHz.

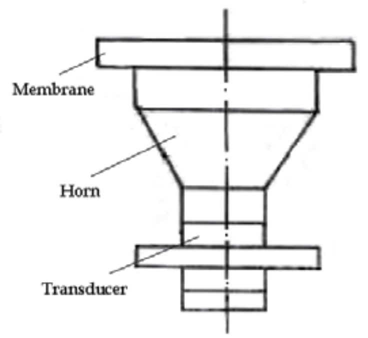


Fig. 5. A transducer with a horn and a membrane

The resonance frequencies are also modifying by the coupling of a concentrator on the transducer. In this case, to obtain the initial resonance frequency of the transducer the user must adjust mechanically the concentrator at the transducer own resonance frequency. At the ultrasonic blocks with three components (Fig. 4) a transducer 1, a mechanical concentrator 2 and a processing tool 3, the resonance frequency is given by the entire assembled block (1, 2, 3) and in the ultimate instance by the processing tool 3. At cleaning equipments the series resonance frequency is decreasing with 3÷4 KHz.
The transducers are characterised by a Nyquist hodograph of the impedance present in Fig. 6, which has the theoretical form of a circle. In reality, due to the non-linear character of the transducer, especially at high power, this circle is deformed.
The movement current i_m of piezoelectric transducer is important information related to the maximum power conversion efficiency at resonance frequency. It is the current passing through the equivalent RLC series circuit, which represents the mechanical branch of the equivalent circuit. It is obtained as the difference:

$$i_m = i - i_{C0} \tag{2}$$

An example of the measured movement current is presented in Fig. 7.

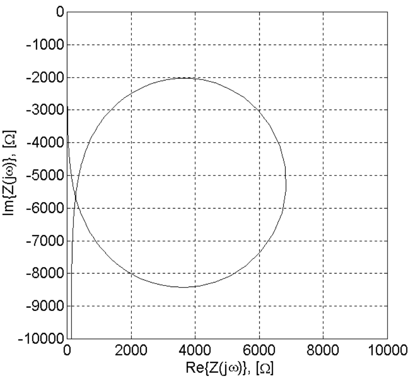


Fig. 6. Impedance hodograph around the resonant frequency

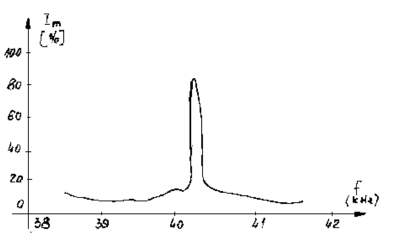


Fig. 7. Movement current frequency characteristic

4. Identification with frequency characteristics

4.1 Generalities

A good design of ultrasonic equipment requests a good knowledge of the equivalent models of ultrasonic components, when the primary piece is the transducer, as an electromechanical power generator of mechanical oscillations of ultrasonic frequency. The model is theoretical demonstrated and practical estimated with a relative accuracy. In practice the estimation consists in the selection of a model that assures a behaviour simulation most closed to the real effective measurements. The identification is taking in consideration some aspects as: model type, test signal type and the evaluation criterion of the error between the model and the studied transducer. Starting from a desired model we are adjusting the parameters until the difference between the behaviour of the transducer and the model is minimized. For the transducer its structure is presumed known, and it is the equivalent circuit from Fig. 2. The purpose of the identification is to find the equivalent parameters of this electrical circuit. The model is estimated from experimental data. One of the parametric models is the complex impedance of the transducer, given in a Laplace transformation. Other model, but in the frequency domain, is the Nyquist hodograph of impedance from Fig. 6. The frequency model is given by a finite set of measured independent values. For the piezoelectric transducer a method that converts the frequency model into a parameter model – the complex impedance, is recommended (Tertisco & Stoica, 1980). A major disadvantage of this method is that the requests for complex estimation equipment and we must know the transducer model – the complex impedance of the equivalent electrical circuit. The frequency characteristic may be determinate easily testing the transducer with sinusoidal test signal with variable frequency. The passing from a frequency model to the parameter model is reduced to the determination of the parameters of the transfer impedance. The steps in such identification procedure are: organization and obtaining of experimental data on the transducer, interpretation of measured data, model deduction with its structure definition and model validation.

4.2 Identification method

Frequency representation of a transducer was presented before. The frequency characteristics may be obtained applying a sinusoidal test voltage signal to the transducer and obtaining a current with the same frequency, but with other magnitude and phase, variables with the applied frequency. The theoretic complex impedance is:

$$Z(j\omega) = |Z(j\omega)| e^{\varphi(\omega)} = \mathrm{Re}\{Z(j\omega)\} + j\,\mathrm{Im}\{Z(j\omega)\} \tag{3}$$

Its parameter representation is:

$$Z(s) = \frac{B(s)}{A(s)} = \frac{\sum_{i=0}^{m} b_i s^i}{1 + \sum_{i=1}^{n} a_i s^i} = \frac{b_0 + b_1 s + b_2 s^2 + \dots + b_m s^m}{1 + a_1 s + a_2 s^2 + \dots + a_n s^n} \tag{4}$$

A general dimensional structure for identification with the orders $\{n, m\}$ is considered, where n and m follow to be estimated.
The model that must be obtained by identification is given by:

$$Z_M(j\omega_k) = \frac{b_0 + b_1(j\omega_k) + \dots + b_m(j\omega_k)^m}{1 + a_1(j\omega_k) + \dots + a_n(j\omega_k)^n} = \frac{\alpha + j\omega_k\beta}{\delta + j\omega_k\gamma} = \frac{B(j\omega_k)}{A(j\omega_k)} \tag{5}$$

We presume the existence of the experimental frequency characteristic, as samples:

$$\begin{gathered} Z_e(j\omega_k) = \mathrm{Re}\{Z_e(j\omega_k)\} + j\,\mathrm{Im}\{Z_e(j\omega_k)\} \\ R_k = \mathrm{Re}\{Z_e(j\omega_k)\} \\ I_k = \mathrm{Im}\{Z_e(j\omega_k)\},\ k = 1,2,3,\dots,n_p \end{gathered} \tag{6}$$

For any particular value ω_k the error $\varepsilon(\omega_k)$ is defined as:

$$\varepsilon(\omega_k) = | Z_e(j\omega_k) - Z_M(j\omega_k) | = \left| Z_e(j\omega_k) - \frac{B(j\omega_k)}{A(j\omega_k)} \right| \tag{7}$$

The error criterion is defined as:

$$E=\sum_{k=0}^{n_p}\left|\varepsilon(\omega_k)\right|^2 \tag{8}$$

The estimation of orders $\{n, m\}$ and parameters is formulated as a parametric optimisation:

$$p=\begin{bmatrix}a_0 & a_1 & \dots & a_n & b_1 & b_2 & \dots & b_m\end{bmatrix}=\arg\min_{p}\sum_{k=0}^{n_p}\left|\varepsilon(\omega_k)\right|^2 \tag{9}$$

The error criterion is non-linear in parameters and the direct has practical difficulties: a huge computational effort, local minima, instability and so on. To simplify the algorithm, the error $\varepsilon(\omega_k)$ is weighted with $A(j\omega_k)$. A new error function is obtain:

$$B(j\,\omega_k)\varepsilon(j\omega_k)=A(j\,\omega_k).Z(j\,\omega_k)-B(j\,\omega_k)=X(\omega_k)+jY(\omega_k) \tag{10}$$

The weighted error function $e(\omega_k)$ is given by

$$e(\omega_k)=A(j\,\omega_k).\varepsilon(j\omega_k) \tag{11}$$

The new approximation error, corresponding to the weighted error is:

$$E=\sum_{k=1}^{n_p}\left|e(\omega_k)\right|^2=\sum_{k=1}^{n_p}\left|A(j\,\omega_k).\varepsilon(j\omega_k)\right|^2=\sum_{k=1}^{n_p}\left[X^2(\omega_k)+Y^2(\omega_k)\right] \tag{12}$$

The minimization of E is done based on the weighted least squares criterion, in which the weighting function $[A(j\omega_k)]^2$ was chosen so E to be square in model parameters:

$$p=\arg\min_{p}\sum_{k=1}^{n_p}\left|e(\omega_k)\right|^2 \tag{13}$$

$$E^i=\sum_{k=1}^{n_p}\left|\frac{A_i(j\omega_k)}{A_{i-1}(j\omega_k)}\right|\varepsilon_i^2(\omega_k) \tag{14}$$

But, also this method is not good in practice. The frequency characteristic must be approximated on the all frequency domain. The low frequencies are not good weighted, so the circuit gain will be wrong approximated. To eliminate this disadvantage the criterion is modified in the following way:

$$p^i=\arg\min_{p}\sum_{k=1}^{n_p}\left|\frac{A_i(j\omega_k)}{A_{i-1}(j\omega_k)}\right|^2\varepsilon_i^2(\omega_k) \tag{15}$$

where i represents the iteration number, p^i is the vector of the parameters at the iteration i. The error $\varepsilon_i(\omega_k)$ is given by:

$$\varepsilon_i(\omega_k)=\left|Z_e(j\omega_k)-\frac{B_i(j\omega_k)}{A_i(j\omega_k)}\right| \tag{16}$$

At the algorithm initiation:

$$B_0(j\omega_k)=1 \tag{17}$$

The criterion is quadratic in p^i, so the parameter vector at the iteration i may be analytically determinate.

In the same time the method converges, because there is the condition:

$$\lim_{i\to\infty}\frac{A_i(j\omega_k)}{A_{i-1}(j\omega_k)}=1 \tag{18}$$

The estimation accuracy will have the same value on the entire frequency spectre.
The procedure is an iterative variant of the least weighted squares method. At each iteration the criterion is minimized and the linear equation system is obtained:

$$\begin{aligned}\left(\frac{\partial E_i}{\partial a_k}\right)_i &= 0\\ \left(\frac{\partial E_i}{\partial b_k}\right)_i &= 0\end{aligned} \tag{19}$$

To obtain an explicit relation for p^i we notice that:

$$\begin{aligned}\mathrm{Re}\{A_i(j\omega_k)\} &= \sum_{i=0}^{r_1}(-1)^i a_{2i}\omega_k^{2i}\\ \mathrm{Im}\{A_i(j\omega_k)\} &= \sum_{i=0}^{r_2}(-1)^i a_{2i+1}\omega_k^{2i+1}\end{aligned} \tag{20}$$

where $r_1 = n/2$ and $r_2=n/2-1$, if n is odd and $r_1 = (n-1)/2$ şi $r_2=(n-1)/2$, if n is even. By analogy $\mathrm{Re}\{B(j\omega_k)\}$ and $\mathrm{Im}\{B(j\omega_k)\}$ may be represented in the same way, for r_3 and r_4, function of m.
From the linear relations the following linear system is obtained:

$$E\times p_i = F \tag{21}$$

where the matrix E, p_i, F are given by the relations (24), in which k takes the values from 1, 0, 0 and 0 until $r_{1,2,3,4}$ for rows, from up to down, and j takes values from 1, 0, 0 and 0 until $r_{1,2,3,4}$ for columns from the left to the right.

$$E=\begin{bmatrix}(-1)^{j+1}\Gamma_{2(j+k)} & 0 & (-1)^j\Delta_{2(j+k)} & (-1)^j\Phi_{2(j+k)+1}\\ 0 & (-1)^j U_{2(j+k+1)} & (-1)^j\Phi_{2(j+k)+1} & (-1)^{j+1}\Delta_{2(j+k+1)}\\ (-1)^{j+1}\Delta_{2(j+k)} & (-1)^j\Phi_{2(j+k)+1} & (-1)^j\Pi_{2(j+k)+1} & 0\\ (-1)^{j+1}\Phi_{2(j+k)+1} & (-1)^{j+1}\Delta_{2(j+k+1)} & 0 & (-1)^j\Gamma_{2(j+k+1)}\end{bmatrix} \tag{22}$$

$$\begin{aligned}p_i &= \begin{bmatrix}\hat{a}_2 & \dots & \hat{a}_{2r1} & \hat{a}_1 & \dots & \hat{a}_{2r2+1} & \hat{b}_0 & \dots & \hat{b}_{2r3} & \hat{b}_1 & \dots & \hat{b}_{2r4+1}\end{bmatrix}^T\\ F &= \begin{bmatrix}\Gamma_2 & \dots & \Gamma_{2r1} & 0 & \Delta_0 & \dots & \Delta_{2r3} & \Phi_1 & \dots & \Phi_{2r4+1}\end{bmatrix}^T\end{aligned} \tag{23}$$

$$\Gamma_i=\sum_{k=1}^{n_p}\frac{(R_k{}^2+I_k{}^2)\omega_k{}^i}{\left|A(j\omega_k)\right|_{i-1}^2},\ \Delta_i=\sum_{k=1}^{n_p}\frac{R_k\cdot\omega_k{}^i}{\left|A(j\,\omega_k)\right|_{i-1}{}^2}, \tag{24}$$

$$\Phi_i=\sum_{k=1}^{n_p}\frac{I_k\cdot\omega_k{}^i}{\left|A(j\,\omega_k)\right|_{i-1}{}^2},\ \Pi_i=\sum_{k=1}^{n_p}\frac{\omega_k{}^i}{\left|A(j\,\omega_k)\right|_{i-1}{}^2} \tag{25}$$

The values of n and m are determinate after iterative modifications and iterative estimations.

The block diagram of the estimation procedure is given in Fig. 8.

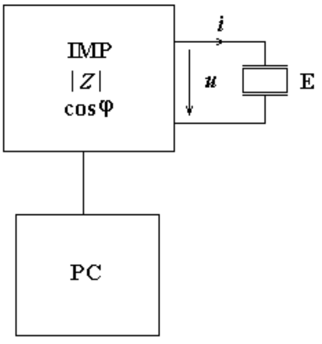


Fig. 8. Estimation equipment

The frequency characteristic of the piezoelectric transducer E is measured with a digital impedance meter IMP. An estimation program on a personal computer PC processes measured data. In practical application estimated parameter are obtain with a relative tolerance of 10 %.

5. Automatic parameter estimation

The method estimates the parameters of the equivalent circuit from Fig. 2: the mechanical inductance L_m, the mechanical capacitor C_m, the resistance corresponding to acoustic dissipation R_m, the input capacitor C_0 and other characteristics as: the mechanical resonance frequency f_m, the movement current i_m or the efficiency η. The estimation is done in a unitary and complete manner, for the functioning of the transducer loaded and unloaded, mounted on different equipments. By reducing the ultrasonic process at the transducer we may determine by the above parameters and variables the global characteristics of the ultrasonic assembling block transducer-process.

The identification is made based on a method of automatic measuring of complex impedances from the theory of system identification (Eyikoff, 1974), by implementation of the generalized model of piezoelectric transducer, and the instantaneous minimization of an imposed error criterion, with a gradient method – the deepest descent method.

In the structure of industrial ultrasonic equipments there are used piezoelectric transducers, placed between the electronic generators and the adapter mechanical elements. Over the transducer a lot of forces of electrical and mechanical origin are working and stressing. The knowledge of electrical characteristics is important to assure a good process control and to increase the efficiency of ultrasonic process.

Based on the equivalent circuit, considered as a physical model for the transducer, we may determine a mathematic model, the integral-differential equation:

$$L_m \frac{di}{dt} + R_m i + \frac{1}{C_m}\int idt = L_m C_0 \frac{d^2u}{dt^2} + L_m C_0 \frac{R_m}{R_p}\frac{du}{dt} + \frac{R_m}{R_0}u + \frac{1}{C_m R_0}\int udt \tag{26}$$

This model represents a relation between the voltage u applied at the input, as an acting force and the current i through transducer. The model is in continuous time. We do not know the parameters and the state variables of the model. This model assures a good representation. A complex one will make a heavier identification. The classical theory of identification is using different methods as: frequency methods, stochastic methods and other. This method has the disadvantage that it determines only the global transfer function.
Starting from equation (28) we obtain the linear equation in parameters:

$$\sum_{0}^{2} \alpha_i i_i + \sum_{0}^{3} \beta_i u_i = 0 \tag{27}$$

$$\begin{gathered} i_0 = i,\ i_1 = di/dt,\ i_2 = \int i dt, \\ u_0 = -u,\ u_1 = -du/dt,\ u_2 = -d^2u/dt^2,\ u_3 = -\int u dt \end{gathered} \tag{28}$$

$$\begin{gathered} \alpha_0 = R_m,\ \alpha_1 = L_m,\ \alpha_2 = 1/C_m, \\ \beta_0 = R_m/R_0,\ \beta_1 = L_m/R_d + C_0 R_m,\ \beta_2 = L_m C_0,\ \beta_3 = 1/(C_m R_0) \end{gathered} \tag{29}$$

The relation gives the transducer generalized model, with the generalized error:

$$e = \sum_{0}^{2} \alpha_i i_i + \sum_{0}^{3} \beta_i u_i \tag{32}$$

The estimation is doing using a signal continuous in time, sampled, sinusoidal, with variable frequency. For an accurate determination of parameters there are necessary the following knowledge: the magnitude order of the parameters and some known values of them.
The error criterion is imposed as a quadratic one:

$$E = e^2 \tag{30}$$

which influences in a positive sense at negative and positive variations of error. To minimize this error criterion we may adopt, for example a gradient method in a scheme of continuous adjustment of parameters, with the deepest descent method. In this case the model is driven to a tangential trajectory, what for a certain adjusted speed it gives the fastest error decreasing. The trajectory is normally to the curves with E=ct. The parameters are adjusted with the relation:

$$\begin{bmatrix} \dot{\alpha}_i \\ \dot{\beta}_i \end{bmatrix} = -\Gamma \begin{bmatrix} -\dfrac{\partial E}{\partial \alpha_i} \\ -\dfrac{\partial E}{\partial \beta_i} \end{bmatrix} = -\Gamma \begin{bmatrix} 2e\dfrac{\partial e}{\partial \alpha_i} \\ 2e\dfrac{\partial e}{\partial \beta_i} \end{bmatrix} = -2\Gamma \begin{bmatrix} i_i \\ u_i \end{bmatrix} \tag{31}$$

where Γ is a constant matrix, which together with the partial derivatives determines parameter variation speed. Derivative measuring is not instantaneously, so a variation speed limitation must be maintained. To determine the constant Γ we may apply Lyapunov stability method. Based on the generalized model and of equation (34) the estimation algorithm may be implemented digitally. The block diagram of the estimator is presented in Fig. 9.

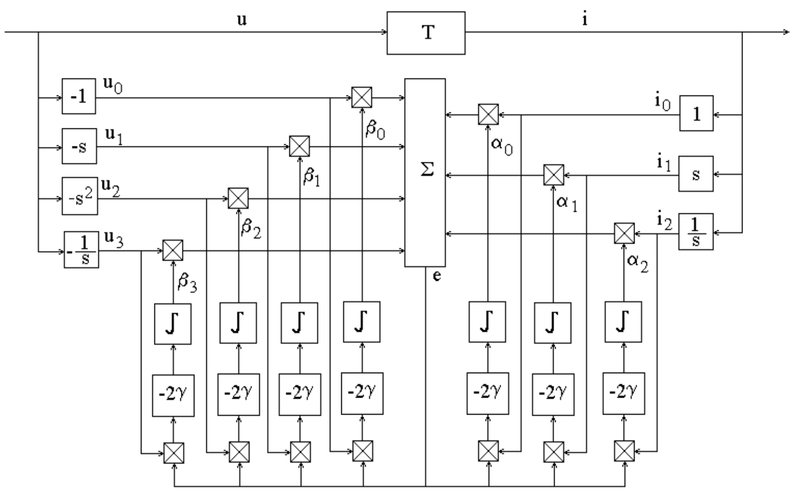


Fig. 9. The block diagram of parameter estimator

Shannon condition must be accomplished in sampling. We may notice some identical blocks from the diagram are repeating themselves, so they may be implemented using the same procedures. Based on differential equation:

$$u = L_m \frac{di_m}{dt} + R_m i_m + \frac{1}{C_m}\int i_m dt = \alpha_1 i_1 + \alpha_2 i_2 + \alpha_0 i_0 \tag{32}$$

which is characterising the mechanical branch of transducer with the parameters obtained with the above scheme, we may determine the movement current with the principle block diagram from Fig. 10.

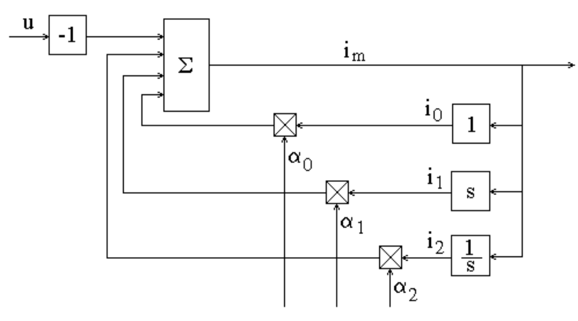


Fig. 10. The block diagram of movement current estimation

The variation of the error criterion E in practical tests is presented in Fig. 11, for 1000 samples.

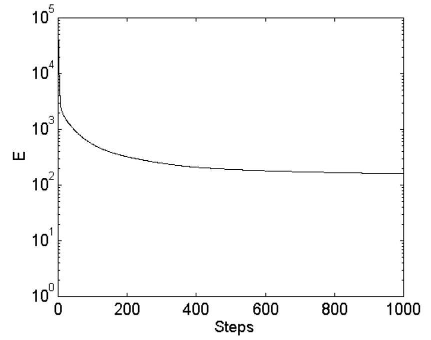


Fig. 11. Error criterion variation

Using the model parameters we may compute the mechanical resonance frequency with the relation:

$$f_m = \frac{1}{2\pi\sqrt{L_m C_m}} \tag{33}$$

The efficiency of conversion as a rapport from the acoustic power P_m and total power P_t is

$$\eta = \frac{P_m}{P_t} \tag{34}$$

where P_m is the power on the resistance R_m:

$$P_m = I_m{}^2 R_m \tag{35}$$

and the total power is the power consumed from the source:

$$P_t = P_m + P_0 \tag{36}$$

where P_0 is the power consumed by the unloaded transducer:

$$P_0 = I_{mo}{}^2 R_p \tag{37}$$

where I_{m0} is the movement current through the unloaded transducer and R_p is the resistance corresponding to mechanical circuit unloaded.

Using the estimator from Fig. 9 and 10 we may do an identification of mechanical adapters. A mechanical adaptor coupled to the transducer influences the equivalent electrical circuit, modifying the equivalent parameters, the resonance frequency and the movement current. We may do the same measuring several times over the unloaded and then over the loaded transducer. Knowing the characteristics of the unloaded transducer we may find the way how the adapter influences the equivalent circuit. So, we may determine the parameters of the assemble transducer – adapter, reduced to the transducer: resonance frequency, movement current and efficiency. To determine efficiency we must take in consideration the power of the unloaded transducer and the power of the unloaded adapter.

Also, the process may be identified using the same estimator. Considering the transducer coupled with an adapter and introduced into a ultrasonic process, as welding, cleaning and other, we may determine by an identification for the loaded functioning the way that the process influences the equivalent parameters. We may determine the resonant frequency of the

ultrasonic process and the global acoustic efficiency of ultrasonic system transducer-adapter-process. We may determine the mechanical resonant frequency of the entire assemble, which is the frequency at what the electronic power generator must functioning to obtain maximum efficiency, the movement current of the loaded transducer and total efficiency, including the power given to the ultrasonic process.
This estimation method has the following advantages: easy to be treated mathematically; easy to implement; generally applicable to all the transducers which have the same equivalent circuit; it assures an optimal estimation with a know error; it offers a good convergence speed,
The method may be implemented digitally, on DSPs, or on PCs, for example using Simulink and dSpace, or using LabView. We present an example of a simple virtual instrument in Fig. capable to be developed to implement the block diagram from Fig. 12.

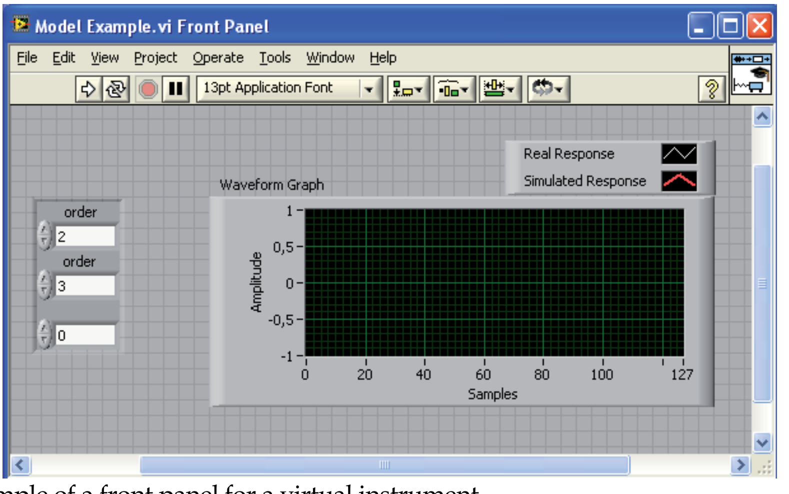


Fig. 12. Example of a front panel for a virtual instrument

The instantaneous variation of parameters and variables of the equivalent circuit may be presented on waveform graphs, data values may be introduced using input controllers. Behind the panel a LabView block diagram similar may be developed using existent virtual instruments from the LabView toolboxes.

6. Frequency control

6.1 Control principle

To perform an effective function of an ultrasonic device for intensification of different technological processes a generator should have a system for an automatic frequency searching and tuning in terms of changes of the oscillation system resonance frequency. The present method is based on a feedback made using the estimated movement current from the transducer. The following presentation has at its basic the paper (Volosencu, 2008).
In the general case the ultrasonic piezoelectric transducers have a non-linear equivalent electric circuit from Fig. 13.

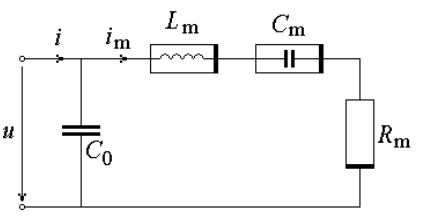


Fig. 13. The non-linear equivalent circuit

In this circuit there is emphasized the mechanical part, seen as a series RLC circuit, with the equivalent parameters R_m, L_m and C_m, which are non-linear, depending on transducer load. The current through mechanical part i_m is the movement current. The input capacitor C_0 of the transducer is consider as a constant parameter. The equations (14) are describing the time variation of the signals and the mechanical parameters, where φ is the magnetic flux through the mechanical inductance L_m and q is the electric load over the mechanical capacitor C_m:

$$
\begin{aligned}
u_{Lm} &= \frac{d\varphi_{Lm}}{dt} = \frac{dL_m}{dt} i_m + L_m \frac{di_m}{dt} \\
i_{Cm} &= \frac{dq_{Cm}}{dt} == \frac{dC_m}{dt} u_{Cm} + C_m \frac{du_{Cm}}{dt} \\
R_m &= \frac{du_{Rm}}{di_m} \\
i &= i_0 + i_m \\
u &= u_{Lm} + u_{Cm} + u_{Rm} \\
C_0 \frac{du}{dt} &= i_{C0}
\end{aligned}
\tag{38}
$$

The piezoelectric traducer has a frequency characteristic of its impedance Z with a series and a parallel resonance, as it is presented in Fig. 14. The movement current i_m has the frequency characteristic from Fig. 15.

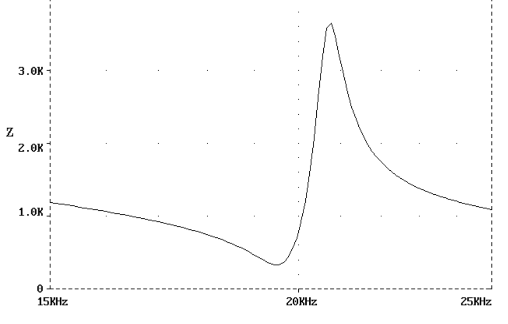


Fig. 14. The magnitude-frequency characteristic of transducer impedance

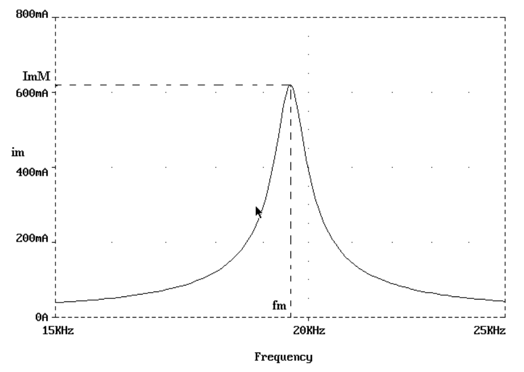


Fig. 15. The frequency characteristic of the transducer movement current

The maximum mechanical power developed by the transducer is obtained when it is fed at the frequency f_m, were the maximum movement current $i_m=I_{mM}$ is obtained. Of course, the maximum of the movement current i_m is obtained when the movement current derivative *dim* is zero:

$$\dim(t) = \frac{di_m}{dt} = 0 \tag{39}$$

So, a frequency control system, functioning after the error of the derivative of movement current may be developed, is using a PI frequency controller, to assure a zero value for this error in the permanent regime.

6.2 Control system

The block diagram of the frequency control system based on the above assumption is presented in Fig. 16.

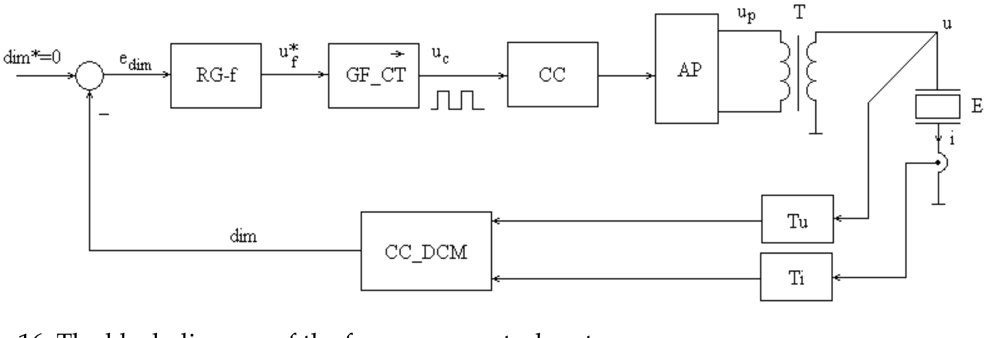


Fig. 16. The block diagram of the frequency control system

A power amplifier AP, working in commutation, at high frequency, feeds a piezoelectric transducer E, with a rectangular high voltage *u*, with the frequency *f*. An output transformer T assures the high voltage *u* for the ultrasonic transducer E. A command circuit CC assures the command signals for the power amplifier AP. The command signal u_c is a rectangular signal, generated by a voltage controlled frequency generator GF_CT. The rectangular command signal u_c has the frequency f and equal durations of the pulses. The frequency of the signal u_c is controlled with the voltage u_f*. The frequency control system from Fig. 16 is based on the derivative movement current error $e_{\dim}$ as the difference between the reference value *dim**=0 and the computed value of the derivative *dim*:

$$e_{\dim} = \dim^* - \dim \tag{40}$$

A PI controller RG-f is used to control the frequency, with the following transfer function:

$$u_f^*(s) = K_R\left(1 + \frac{1}{T_R s}\right)e_{\dim}(s) \tag{41}$$

The frequency controller is working after the error of the derivative of the movement current $e_{\dim}$. The derivative of the movement current *dim* is computed using a circuit CC_DCM, based on the following relation with Laplace transformations:

$$\dim(s) = s[i(s) - sC_0 u(s)] \tag{42}$$

where C_0 is the known constant value of the capacitor from transducer input and u and i are the measured values of the transducer voltage and current. The voltage upon the transducer u and the current i through the transducer are measured using a voltage sensor Tu and respectively a current sensor Ti.

6.3 Modelling and simulation

Two models for the transducer and for the block diagram from Fig. 16 were developed to test the control principle by simulation. In the first model the parameters of the mechanical part are considered with a fix static value and a dynamical variation. In the second model the electromechanical transducer is considered coupled with a mechanical concentrator and the equivalent circuits are coupled in series. Approximating the relations (41), the following relations are used to model the behaviour of the piezoelectric transducer:

$$\begin{gathered} u_{Lm}(s) = sL_m(s)i_m(s) + sL_m(s)i_m(s) \\ i_{Cm}(s) = sC_m(s)u_{Cm}(s) + sC_m(s)su_{Cm}(s) \\ u_{Rm}(s) = \frac{1}{s}R_m[si_m(s)] \\ u(s) = u_{Lm}(s) + u_{Cm}(s) + u_{Rm}(s) \end{gathered} \tag{43}$$

The mechanic parameters from the above relations have the following variations, in the vicinity of the stationary points R_{m0}, L_{m0} and C_{m0}:

$$R_m = R_{m0} + \Delta R_m,\ L_m = L_{m0} + \Delta L_m,\ C_m = C_{m0} + \Delta C_m \tag{44}$$

The movement current i_m (s) is modelled, based on the above relations, with the following relation:

$$i_m(s) = \frac{1}{s}\left[\frac{1}{L_m} \times \left(u(s) - \frac{1}{s}.\frac{1}{C_m} \times i_m(s) - R_m \times i_m(s)\right)\right] \tag{45}$$

The block diagram of the movement current model is presented in Fig. 17.

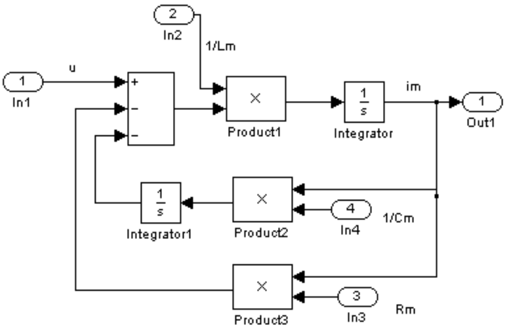


Fig. 17. Simulation model for mechanical part

A second model is taken in consideration. The transducer is considered coupled with the concentrator and the equivalent circuit is presented in Fig. 18.

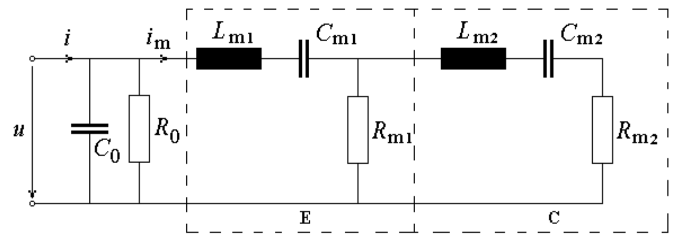


Fig. 18. Equivalent circuit of the transducer with concentrator

In this model there is a series RLC circuit with the parameters L_{m1}, C_{m1} and R_{m1} for the transducer T and a series RLC circuit with the parameters L_{m2}, C_{m2} and R_{m2} for the concentrator C, coupled in cascade. The parts of the control block diagram are modelled using Simulink blocks. A transient characteristic of the frequency control system is presented in Fig. 19.

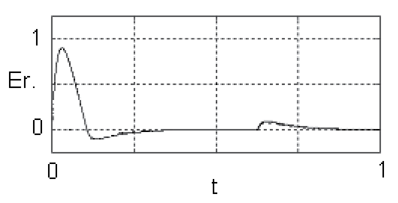


Fig. 19. Transient characteristic for the error, obtained by simulation

The simulation is made considering for the first model the variation with 10 % at the transducer parameters. The deviation in frequency is eliminated fast. The frequency response has a small overshoot.

6.4 Implementation and test results

The frequency control system is developed to be implemented using analogue, high and low power circuits, for general usage. The power amplifier AP is built using four power IGBT transistors, in a complete bridge, working in commutation at high frequency. The electric circuit of the power inverter is presented in Fig. 20.

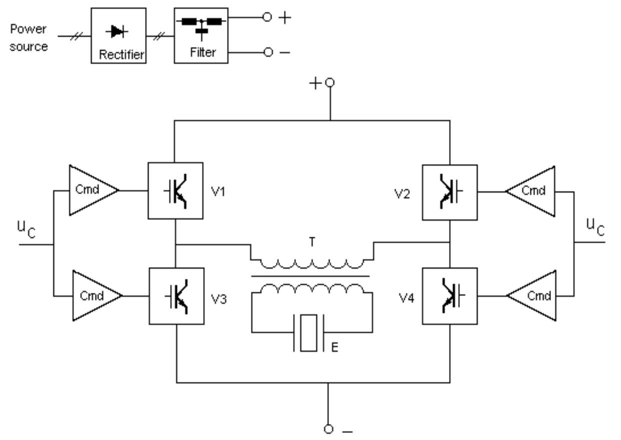


Fig. 20. The power inverter

The four IGBT transistors V1÷4 have the voltage and current protection circuits. The power transistors are commanded with 4 circuits Cmd. The Power inverter is fed from the power system with a rectifier and a filter. The command circuits are receiving the command voltage u_c from the VC_FG circuit. The voltage controlled frequency generator GF_CT is made using a phase lock loop PLL circuit and a comparator. The computing circuit CC_DCM, which implements the relations and the frequency controller RG-f are realized using analogue operational amplifiers. The transformer T is realized using ferrite cores, working at high frequency. The electronic generator is presented in Fig. 21.

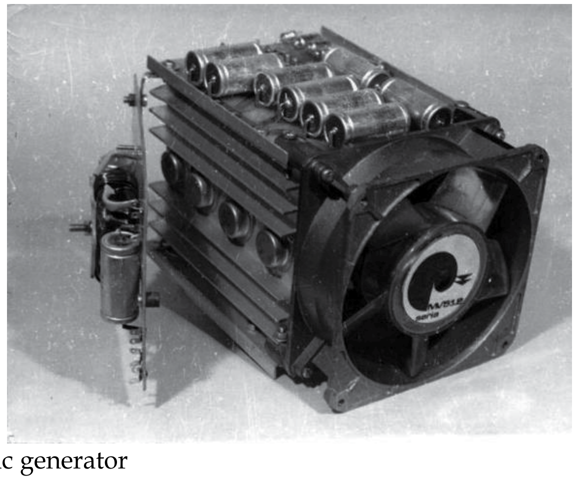


Fig. 21. The electronic generator

Some transient signal variations are presented as follows. The pulse train of the command voltage u_c is presented in Fig. 22.

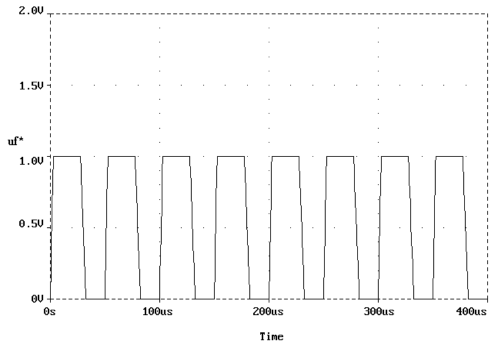


Fig. 22. Examples of sensor impulse trains.

The output voltage of the power amplifier is presented in Fig. 23.

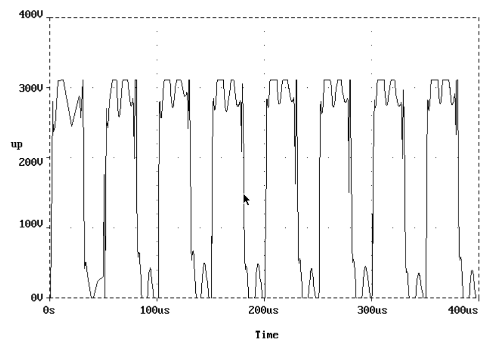


Fig. 23. The output voltage

The voltage u over the piezoelectric transducer is presented in Fig. 24.

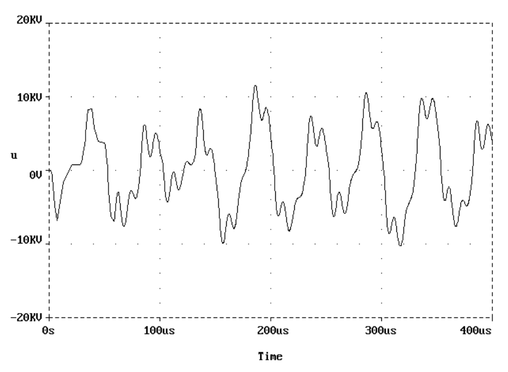


Fig. 24. The transducer voltage

The measured movement current i_m is presented in Fig. 25.

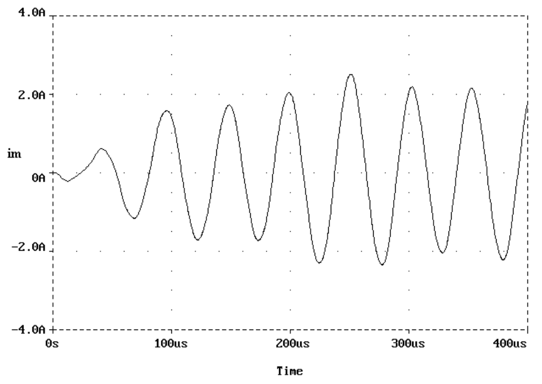


Fig. 25. The movement current

The control system was developed for a plastic welding machine in the range of 3000 W at 40 kHz.

7. Conclusion

This chapter presented two methods for parameter identification at the piezoelectric transducers used in high power ultrasonic applications as welding, cleaning and other. The methods are offering information about the equivalent electrical circuit: parameters and resonance frequency, efficiency and other. The first parameter estimation method is using the transformation from non-parameter model – the Nyquist hodograph - to a parameter model - the transfer function of the transducer's impedance, testing the piezoelectric transducers with a sinusoidal signal with variable frequency. The second parameter estimation method is based on an automatic measurement of piezoelectric transducer impedance using a deterministic convergence scheme with a gradient method with continuous adjustment – the method of deepest descent with a maximum slope. Some practical results are given. Some indications to implement the method using LabView are given. In the end, the paper provides a method for frequency control at ultrasonic high power piezoelectric transducers, using a feedback control systems based on the first derivative of the movement current. This method assures a higher efficiency of the energy conversion and greater frequency stability. A simulation for two kinds of transducer model is made. The control principle is implanted on a power electronic generator. Some transient characteristics are presented. The frequency control system was modelled and simulated using Matlab and Simulink. Two models for the mechanical part of the transducer are chosen. Two different regimes for the time variations of the mechanical parameters of the transducer was chosen and tested. A Simulink model and a simulation result are presented. The simulation results have proven that the control principle developed in this paper gives good quality criteria for the output frequency control. The control system is implemented using a power inverter with transistors working in commutation at high frequencies and analogue circuits for command. Transient characteristics of the control systems are presented. The frequency control system may be developed for piezoelectric transducers in a large scale of constructive types, powers and frequencies, using general usage analogue components, at a low price, with good control criteria.

8. References

Bose, B.K., (2000). Energy, Environment and Advances in Power Electronics, *IEEE Transactions on Power Electronics*, vol. 15, no. 4, July, 2000, pag. 688-701.

Chen, Y. C., Hung, L. C., Chaol, S.H. & Chien, T. H. (2008). PC-base impedance measurement system for piezoelectric transducers and its implementation on elements values extraction of lump circuit model, *WSEAS Transactions on Systems,* Volume 7, Issue 5, May 2008. Pages 521-526.

Eykhoff, P. (1974). *System Identification*, John Wiley and Sons, Chicester, U.K., 1974.

Furuichi, S. & Nose, T., (1981). Driving system for an ultrasonic piezoelectric transducer, *U.S. patent 4271371*.

Gallego-Juarez, J. A. (2009). Macrosonics: Sound as a Source of Energy, In: *Recent Advances in Accoustic and Music, Theory and Applications, Proceedings of the 10th WSEAS International Conference on Acoustics & Music Theory & Applications*, pag. 11-12, ISBN: 978-960-474-061-1, ISSN 1790-5095, Prague, Czech Rep., March 23-25, 2009.

Hulst, A.P., (1972). Macrosonics in industry 2. Ultrasonic welding of metals, *Ultrasonics*, Nov., 1972.

Khmelev, V.N., Barsukov, R.V., Barsukov, V., Slivin, A.N. and Tchyganok, S.N., (2001). System of phase-locked-loop frequency control of ultrasonic generators, *Proceedings of the 2nd Annual Siberian Russian Student Workshop on Electron Devices and Materials*, 2001. pag. 56-57.

Kirsch, M. & Berens, F., (2006). Automatic frequency control circuit, *U. S. Patent 6571088*.

Marchesoni, M., (1992). High-Performance Current Control Techniques for Applications to Multilevel High-Power Voltage Source Inverters, In *IEEE Trans. on Power Electronics*, Jan.

Mori, E., 1989. High Power Ultrasonic Wave Transmission System, In *J. Inst. Electron. Inf. Commun. Eng.*, vol. 72, no. 4, April.

Morris, A.S., (1986). Implementation of Mason's model on circuit analysis programs, In *IEEE Transactions on ultrasonics, ferroelectric and frequency control*, vol. UFFC-33, no. 3.

Neppiras, E.A., (1972). Macrosonics in industry, 1. Introduction. *Ultrasonics,* Jan., 1972.

Prokic, M., (2004). *Piezoelectric Converters Modelling and Characterization*, MPI Interconsulting, Le Locle, Switzerland.

Radmanovic, M. D. & Mancic, D. D., (2004). *Design and Modelling of Power Ultrasonic Transducers*, University of Nis, 2004, MPI Interconsulting.

Senchenkov, I.K., (1991). Resonance vibrations of an electromechanical rod system with automatic frequency control, *International Applied Mechanics*, Vol. 27, No. 9, Sept., Springer, N. Y.

Sulivan, R.A. (1983). Power supply having automatic frequency control for ultrasonic bonding, *U. S. Patent 4389601*.

Tertisco, M. & Stoica, P. (1980), *Identificarea si estimarea parametrilor sistemelor*, Editura Academiei Romaniei, Bucuresti, 1980.

Volosencu, C. (2008). Frequency Control of the Pieozoelectric Transducers, Based on the Movement Current, *Proceedings of ICINCO 2008 – Fifth International Conference on Informatics in Control, Automation and Robotics*, ISBN 978-989-8111-35-7, Funchal, Madeira, Portugal, May 11-15, 2008.

Design of the Wave Digital Filters

Bohumil Psenicka, Francisco García Ugalde and Andrés Romero M.
Universidad Nacional Autónoma de México
México

1. Introduction

In control automation and robotics filters are used in image processing, in the automatics of alarm system, adaptive signal processing and control vibration, as well as in the control of mobile robots, system identification, speech system recognition, etc.

In this chapter "Design of the Wave Digital filters" we shall propose a very simple procedure for designing, analysis and realization of low-pass, high-pass, band-pass and band-stop wave digital filters from reference LC filters given in the lattice configuration and will be introduced tables for simple design of the wave digital filters.

Wave digital filters are derived from LC-filters. The reference filter consists of parallel and serial connections of several elements. Since the load resistance R is not arbitrary but dependent on the element or source to which the port belongs, we cannot simply interconnect the elements to a network. The elements of the filters are connected with the assistance of serial and parallel adaptors. This adaptors in the discrete form are connected in one port by delay elements. The possibility of changing the port resistance can be achieved using parallel and serial adaptors. These adaptors contain the necessary adders, multipliers and inverters.

2. Serial and parallel adaptors.

In this chapter, we use adaptors with three ports. Blocks the serial and parallel reflection-free adapters and theirs signal-flow diagram are shown in figure 1.

The coefficient of the 3-port reflection-free serial adaptor in figure 1A) is calculated from the port resistances R_i i=1,2 by (1).

$$B = \frac{R_1}{R_1 + R_2} \tag{1}$$

The coefficient of the reflection-free parallel adaptor in figure 1B) can be calculated from the port conductance G_i i=1,2 by (2)

$$A = \frac{G_1}{G_1 + G_2} \tag{2}$$

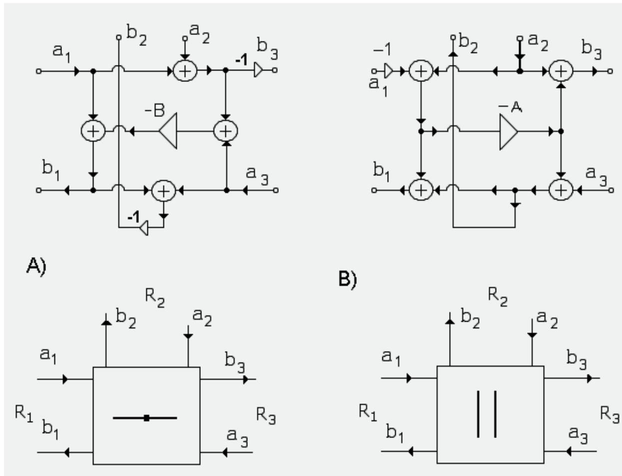


Fig. 1. A) Three port serial adaptor whose port 3 is reflection-free and its signal flow-graph, B) Three port parallel adaptor whose port 3 is reflection-free and its signal flow-graph.

The coefficients of the dependent parallel adaptor in the figure 2B) can be get from port conductances G_i i=1,2,3 by (3)

$$A_1 = \frac{2G_1}{G_1 + G_2 + G_3} \qquad A_2 = \frac{2G_2}{G_1 + G_2 + G_3} \tag{3}$$

The coefficients of the dependent serial adaptor in the figure 2A) can be obtained from the port resistances R_i i=1,2,3 by (4)

$$B_1 = \frac{2R_1}{R_1 + R_2 + R_3} \qquad B_2 = \frac{2R_2}{R_1 + R_2 + R_3} \tag{4}$$

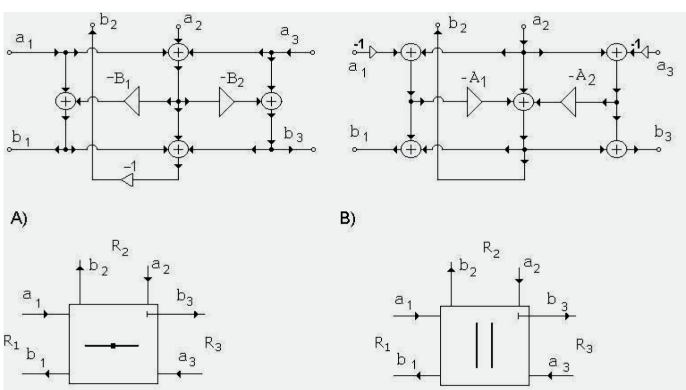


Fig. 2. A) Three-port serial dependent adaptor and its signal-flow graph, B)Three-port parallel dependent adaptor and its signal flow graph.

When connecting adaptors, the network must not contain any feedback loops without a delay element in order to guarantee that the structure is realizable. This means that we cannot connect the dependent adaptors from figure 2A) and 2B). The free-port dependent parallel is reflection free at port 3 if $G_3 = G_1+G_2$ and three-port dependent serial adaptor is reflection free if $R_3=R_1+R_2$.

3. Examples

By means of examples we shall demonstrate calculation of the low-pass and high-pass wave digital filters. The most important components for the realization of wave digital filters according to Fettweis procedure (Fettweis 1972) are the ladder LC filters. The tables for wave digital structures was designed for the corner frequency f_1=1/(2pi)=0.159155 and sampling frequency f_s=0.5

4. Realization of the low-pas filter.

In the first example we shall realize a Butterworth low-pass of the 5th order and A_{max}=3 dB. In the figure 3 we show the structure of a 5th order ladder LC reference Butterworth filter and its corresponding block connection in the digital form.

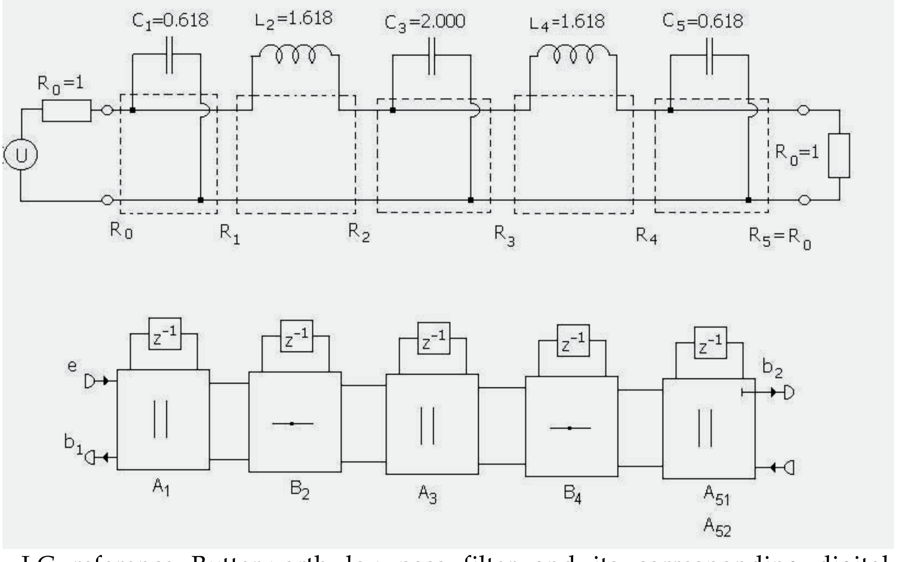


Fig. 3. LC reference Butterworth low-pass filter and its corresponding digital block connection.

First we must calculate from figure 3 (G_0=1) wave port resistances R_1,R_2,R_3,R_4 then the coefficients of the parallel and serial adaptors A_1,B_2,A_3,B_4 and finally the coefficients of the dependent parallel adaptor A_{51},A_{52} according to (1)-(3).

$$G_1 = G_0 + C_1 = 1.618 \qquad G_3 = G_2 + C_3 = 2.447$$

$$R_1 = 1/G_1 = 0.618 \qquad R_3 = 1/G_3 = 0.408$$

$$R_2 = R_1 + L_2 = 2.236 \qquad R_4 = R_3 + L_4 = 2.026$$

$$G_2 = 1/R_2 = 0.447 \qquad G_4 = 1/R_4 = 0.493$$

$$A_1 = \frac{G_0}{G_0 + C_1} = 0.618 \qquad B_2 = \frac{R_1}{R_1 + L_2} = 0.276$$

$$A_3 = \frac{G_2}{G_2 + C_3} = 0.182 \qquad B_4 = \frac{R_3}{R_3 + L_4} = 0.201$$

$$A_{51} = \frac{2G_4}{G_4 + C_5 + G_5} = 0.467 \qquad A_{52} = \frac{2G_5}{G_4 + C_5 + G_5} = 0.947$$

The program for the analysis of the wave digital filter of the 5th order written in MATLAB follows, and the frequency response obtained is given in figure 4. The equation in the program for computing XN1-XN4, BN4-BN1,N1-N9 and YN(i) was obtained from the structure in the figure 4.

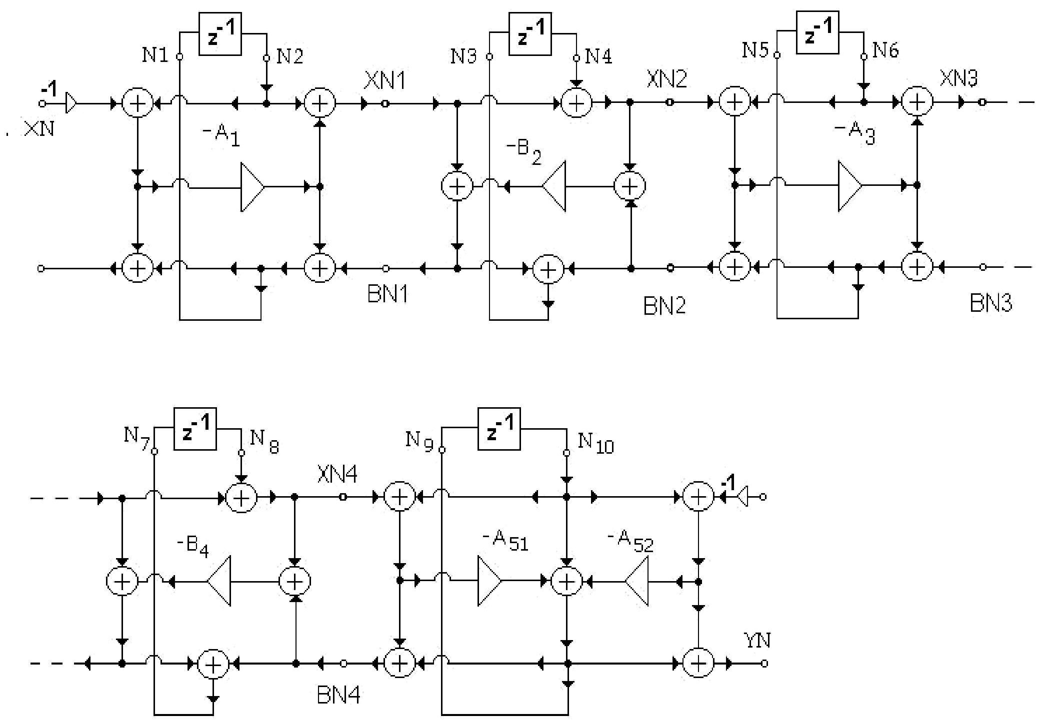


Fig. 4. Butterworth wave digital low-pass filter of the 5th order.

A1=0.618146;B2=0.276524;A3=0.182858;B4=0.201756;A51=0.467353;A52=0.947276;
N2=0; N4=0; N6=0; N8=0; N10=0; XN=1

```
for  i=1:1:200
     XN1=A1*XN-A1*N2+N2;        XN2=XN1+N4;
       XN3=N6-A3*XN2-A3*N6;       XN4=XN3+N8;
       BN4=XN4-A51*XN4+2*N10-A51*N10-A52*N10;
       BN3=XN3-B4*XN4-B4*BN4;
       BN2=XN2-A3*XN2+BN3+N6-A3*N6;
       BN1=XN1-BN2*XN2-B2*BN2;
       N1=XN*A1-A1*N2+BN1;      N3=BN1+BN2;
       N5=BN3-A3*XN2-A3*N6;     N7=BN3+BN4;
       N9=N10-A51*XN4-A51*N10-A52*N10;
       YN(i)=2*N10-A51*XN4-A51*N10-A52*N10;
     N2=N1;N4=N3;N6=N5;N8=N7;N10=N9;XN=0;
end
 [h,w]=freqz(YN,1,200)
Plot(w,20*log10(abs(h)))
```

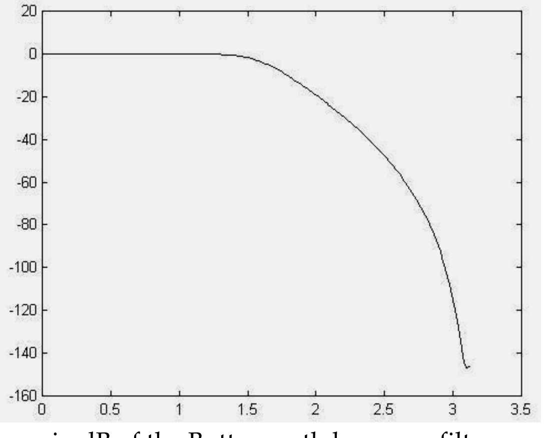


Fig. 5. Frequency response in dB of the Butterworth low-pass filter.

5. Design of the low-pass and high-pass Chebychev filter

As a second example we shall propose low-pass and high-pass Chebychev wave digital filter for n=5, A_{max}=3 dB. From table 7 we get the values of the WDF A_1=0.223, B_2=0.226, A_3=0.182, B_4=0.192, A_{51}=0.383 and A_{52}=0.360. Using previous MATLAB program and changing the values of the coefficients we get the attenuation of Chebychev low-pass and high-pass filter in the figure 6. Additionally to get the high-pass filter from the previous program we also have to change N2= -N1; N4= -N3; N6= -N5; N8= -N7; N10= -N9.

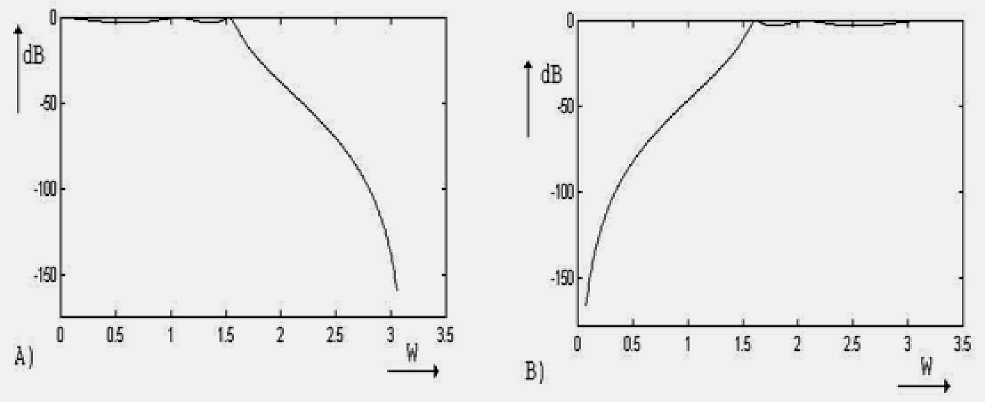


Fig. 6. Attenuation of the Chebychev low-pass and high-pass filter

6. Realization of the low-pass Cauer WDF.

In figure 7 the structure of the 5th order ladder LC reference Cauer low-pass filter is shown. The values of the LC filter was obtained from table (Rudolf Saal, p. 114, 1979) (C 0550 for Θ=30°), A_{max}=1.2494 dB, A_{min}=70.5 dB and Ω_{s}=2.0000. (Saal 1979).

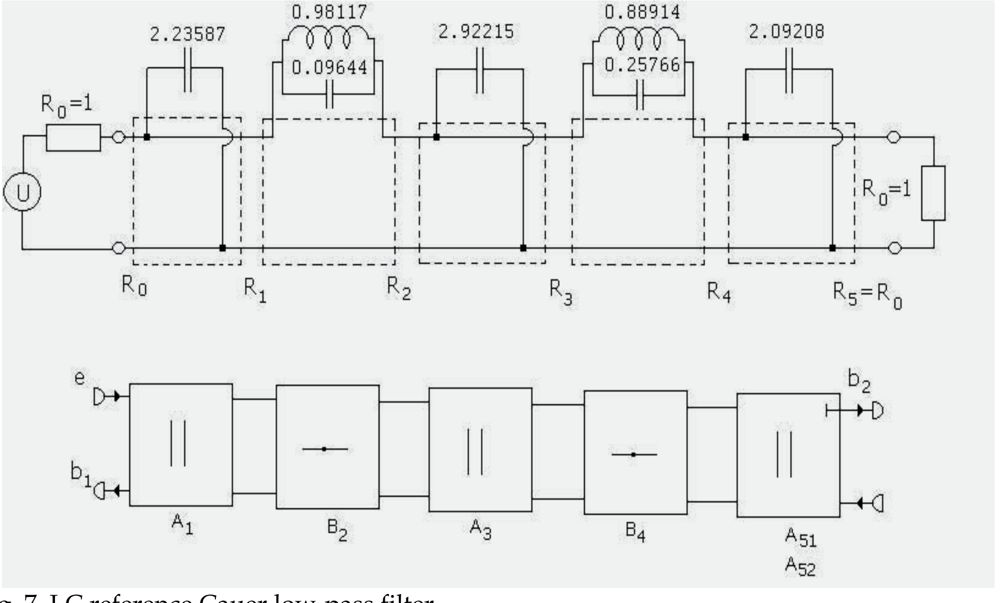


Fig. 7. LC reference Cauer low-pass filter

Parallel resonant LC circuit in low-pass filter figure 7 wil be realized by digital structure according to Figure 8. Structure of the wave digital Cauer low-pass filter of the 5th order is presented in the Figure 9.

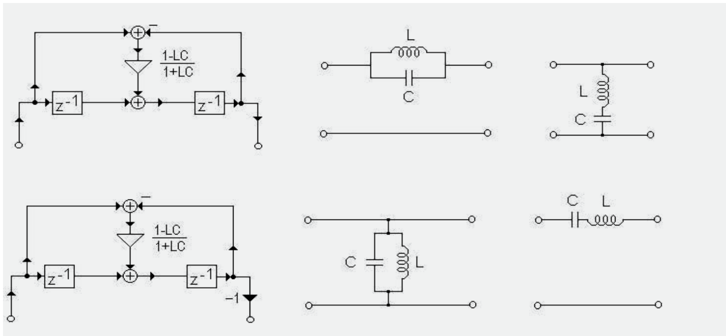


Fig. 8. Discrete realization of parallel and serial LC circuit.

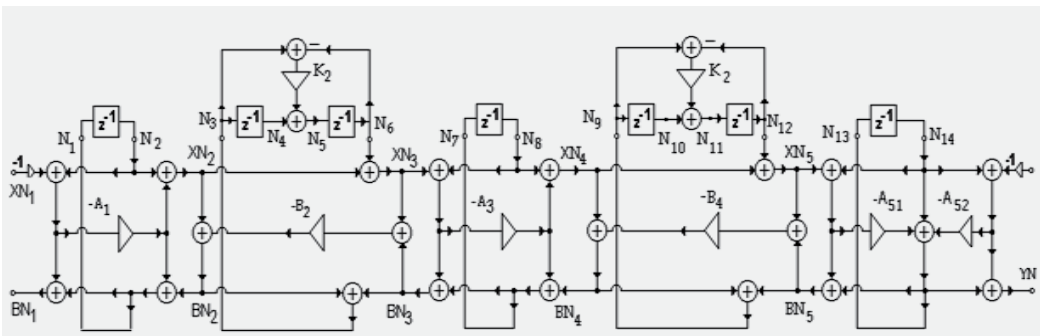


Fig. 9. Structure of the Cauer low-pass filter n=5.

With the assistance of the following MATLAB program we can calculate coefficients of the wave digital filter A_1,B_2,A_3,B_4,A_{51} and A_{52}. The attenuation of the low-pass Cauer filter is presented in Figure 10. The program was obtained from the structure in the Figure 9. The input data of the LC filter was obtained from the catalogue of the Cauer filter (Saal 1979.)

```
C(1)=2.235878; C(3)=2.922148; C(5)=2.092084; L(2)=0.981174; L(4)=0.889139;
C(2)=0.096443; C(4)=0.257662; XN1=1; N2=0; N4=0; N6=0; N8=0; N10=0; N12=0;
N14=0; G(1)=1; G(2)=G(1)+C(1); R2=1/G(2);
R(3)=R(2)+1/(C(2)+1/L(2));        G(3)=1/R(3);      G(4)=G(3)+C(3);    R(4)=1/G(4);
R(5)=R(4)+1/(C(4)+1/L(4));        G(5)=1/R(5);      K(2)=(L(2)*C(2)-1)/(L(2)*C(2)+1);
K(4)=(L(4)*C(4)-1)/(L(4)*C(4)+1);  A(1)=G(1)/G(2);  B(2)=R(2)/R(3);
A(3)=G(3)/(G(3)+C(3));            B(4)=R(4)/R(5);
A(51)=2*G(5)/(G(5)+C(5)+1);       A(52)=2/(G(5)+C(5)+1);
for i=1:1:200
   XN2=A(1)*XN1+N2-A(1)*N(2);       XN3=XN2+N6;
   XN4=N8-A(3)*XN3-A(3)*N8;       XN5=XN4+N12;
   BN5=XN5-A(51)*XN5+2*N14-A(51)*N14-A(52)*N14*;
   BN4 =XN4-B(4)*BN5-B(4)*XN5;
   BN3 =XN3-A(3)*XN3+BN4+N8-A(3)*N8;
   BN2 =XN2-B(2)*XN3-B(2)*BN3;
   N1 =A(1)*XN1-A(1)*N2+BN2;      N3=BN2*BN3;
   N5=-K(2)*N3+K(2)*N6+N4:       N7=BN4-A(3)*XN3-A(3)*N8;
   N9=BN4+BN5;                   N11=N10-N9*K(4)+N12*K(4);
```

```
    N13=N14-A(51)*XN5-A(51)*N14-A(52)*N14;
    YN(i)=2*N14-A(51)*XN5-A(52)*N14-A(51)*N14;
    N2=N1;N4=N3;N6=N5;N8=N7;N10=N9;N12=N11;N14=N13;XN1=0;
end
[h,w]=freqz(YN,1,200);    plot(w,20*log10(abs(h)))
```

High-pass can be obtained by changing in the program N2=-N1; N4=-N3; N6=-N5; N8=-N7; N10=-N9; N12=-N11; N14=-N13. In Figure 10 the attenuation of low-pass and high-pass filter are presented.

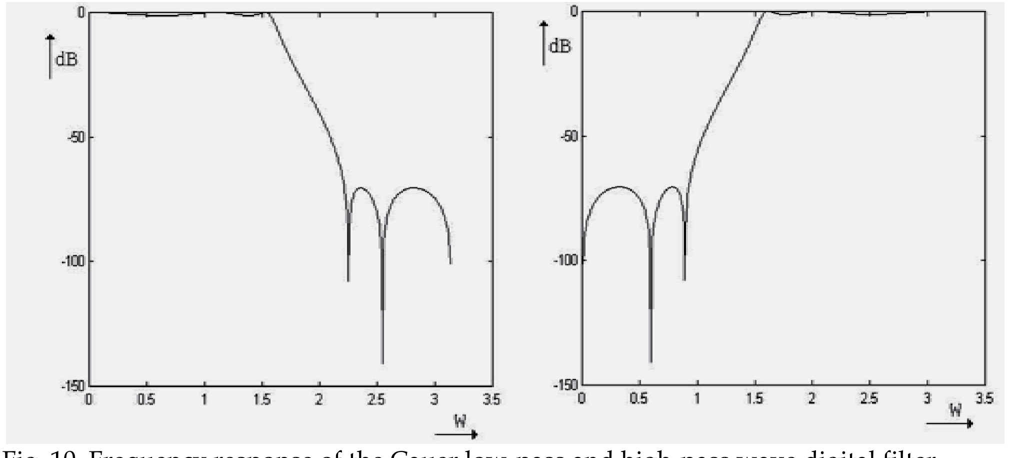


Fig. 10. Frequency response of the Cauer low-pass and high-pass wave digital filter

7. Design of the Cauer band-pass filter

We shall realize band-pass filter that accomplish the tolerance scheme given in Figure 11. The values of the normalized low-pass filter for C0350 θ=53 can be obtained from the table of Cauer filter (Saal 1979). The values of the normalized low-pass filter are shown in figure 12A. With the transformation of the low-pass filter to the band-pass filter can be obtained values of the band-pass filter presented in Figure 12B.

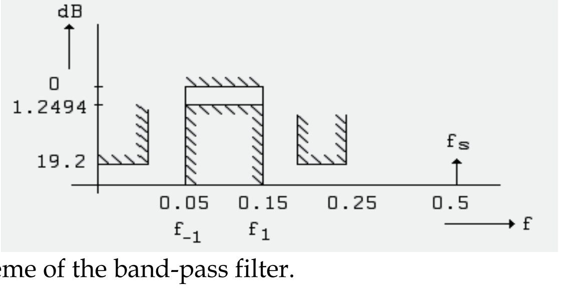


Fig. 11.Tolerance scheme of the band-pass filter.

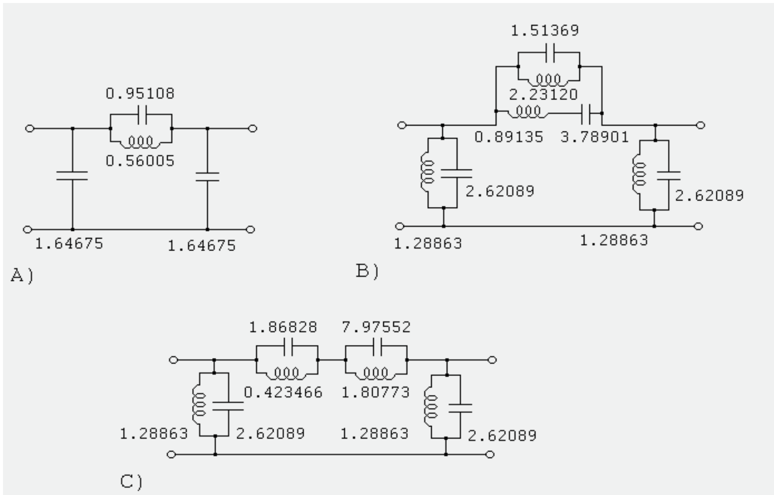


Fig. 12. A) LC low-pass Cauer filter, B) LC band-pass Cauer filter, C) LC band-pass filter after impedance transformation.

Serial-parallel resonant circuit in the longitudinal branch in the figure 12B) can be transformed into parallel-parallel resonant circuit, figure 13, by equations (5) up to (11).

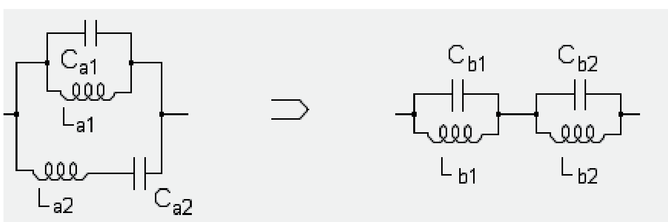


Fig. 13. Impedance transformation of the serial-parallel resonant LC circuit to the parallel-parallel resonant circuit

$$a = \left(1 + \frac{C_{A1}}{C_{A2}} + \frac{L_{A2}}{L_{A1}}\right)^2 - 4\frac{C_{A1}L_{A2}}{C_{A2}L_{A1}} \tag{5}$$

$$u = \sqrt{1 - \frac{4L_{A2}}{aL_{A1}}} \tag{6}$$

$$v = \sqrt{1 - \frac{4C_{A1}}{aC_{A2}}} \tag{7}$$

$$L_{B2} = L_{A1}\frac{1+u}{2} \tag{8}$$

$$L_{B1} = L_{A1}\frac{1-u}{2} \tag{9}$$

$$C_{B2} = aC_{A2}\frac{1+v}{2} \tag{10}$$

$$C_{B1} = aC_{A2}\frac{1-v}{2} \tag{11}$$

From the circuit in the figure 14 can be obtained the values $Y_i=C_i+1/L_i$ (i=1, 2, 3, 4) of the parallel resonant circuit and the resistance R_1, R_2, R_3 and R_4 at the output port of each adaptors. Parallel resonant circuit can be realized by the discrete structure presented in figure 8. The coefficients K_i i=(1, 2, 3, 4) of the wave digital band-pass filter presented in figure 8 can be obtained by following equation

$$K_i = \frac{L_iC_i - 1}{L_iC_i + 1}$$

$K_1=0.543105$ $K_2=-0.11659$

$K_3=0.870278$ $K_4=0.543105$

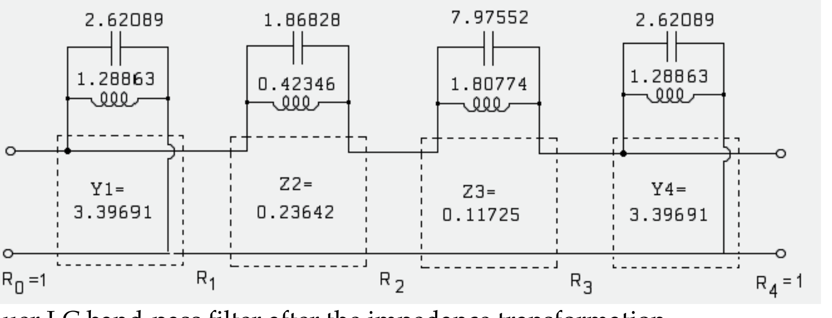


Fig. 14. Cauer LC band-pass filter after the impedance transformation

From the port admittances Y_i and impedances R_i we get the following values,

$G_1=G_{0-}+Y_1=4.39611$ $R_1=0.227432$

$R_2=R_1+1/Y_2=0.463853$

$R_3=R_2+1/Y_3=0.581104$ $G_3=1.720862$

and the coefficients of the parallel and serial adaptors of the wave digital band-pass in the figure 15. At the output of the band-pass filter the parallel dependent adaptor must be used in order to realize the load resistor $R_4=1$.

$$A_1 = \frac{G_0}{G_0+G_1} = \frac{1}{1+3.3969} = 0.2274$$

$$B_2 = \frac{R_1}{R_1+Z_2} = \frac{0.2274}{0.2274+0.2364} = 0.4903$$

$$B_3 = \frac{R_2}{R_2+Z_3} = \frac{0.4638}{0.4638+0.1172} = 0.7982$$

$$A_{41} = \frac{2G_3}{G_3+Y_4+G_4} = \frac{2\cdot 1.7208}{1.7208+3.3969+1} = 0.5625$$

$$A_{42} = \frac{2\cdot G_4}{G_3+Y_4+G_4} = \frac{2}{1.7208+3.3969+1} = 0.3269$$

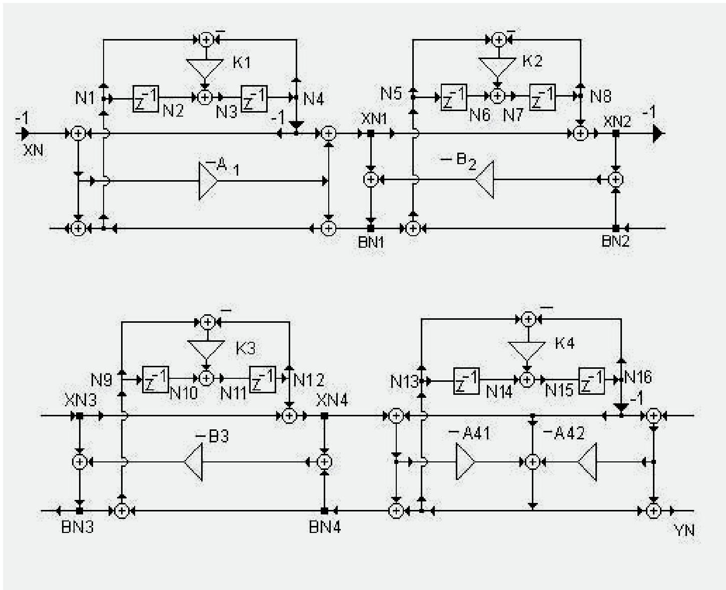


Fig. 15. Cauer wave digital band-pass filter.

8. Tables of the Wave digital filters.

In the figure 16 there are frequency response and structure of the wave digital filter. The structure of Butterworth WDF is created by connection of the parallel and serial adaptors terminated at the port 3 with delay elements. At the end of the structure must be connected parallel dependent adaptor to realize for n odd load resistance R_L=1. In case of n even at the end of structure must be connected serial dependent adaptor.

9. Tables of the Butterworth wave digital filter

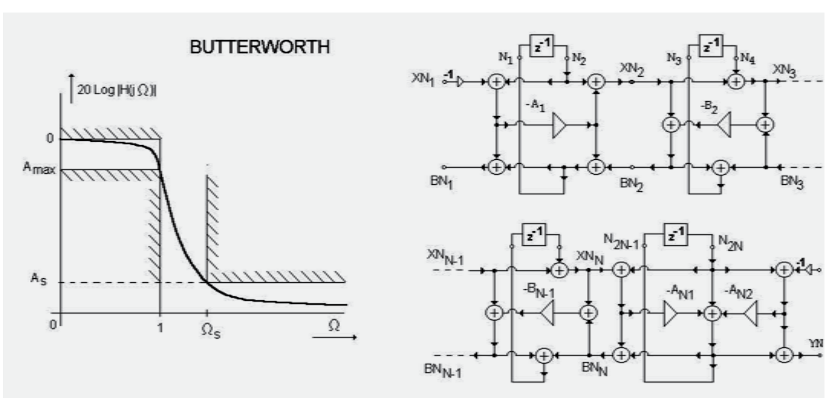


Fig. 16. Frequency response and structure of the Butterworth wave digital filter.

In tables 1-5 are the elements of the Butterworth wave digital filters for various attenuation A_{max} in the pass-band and n=3, 4, 5, 6 and 7. The tables was designed for sampling frequency f_s=0.5.

A_{max}	A_1	B_2	A_{31}	A_{32}
0.1	0.6517	0.3788	0.5494	0.9453
0.2	0.6246	0.3419	0.5096	0.9309
0.3	0.6082	0.3206	0.4856	0.9210
0.4	0.5962	0.3056	0.4682	0.9133
0.5	0.5867	0.2941	0.4545	0.9068
0.6	0.5789	0.2846	0.4331	0.9012
0.7	0.5721	0.2767	0.4334	0.8963
0.8	0.5662	0.2698	0.4249	0.8918
0.9	0.5609	0.2637	0.4174	0.8876
1.0	0.5561	0.2583	0.4105	0.8838

Table 1. Elements of the Butterworth WDF N=3, A_{max} in dB.

A_{max}	A_1	B_2	A_3	B_{41}	B_{42}
0.1	0.6764	0.3694	0.3210	0.5690	0.9679
0.2	0.6568	0.3424	0.2924	0.5385	0.9599
0.3	0.6449	0.3268	0.2760	0.5200	0.9545
0.4	0.6364	0.3157	0.2645	0.5067	0.9503
0.5	0.6295	0.3071	0.2556	0.4961	0.9468
0.6	0.6239	0.3000	0.2484	0.4874	0.9437
0.7	0.6190	0.2941	0.2423	0.4798	0.9409
0.8	0.6147	0.2889	0.2369	0.4732	0.9385
0.9	0.6109	0.2843	0.2326	0.4673	0.9362
1.0	0.6074	0.2802	0.2281	0.4620	0.9431

Table 2. Elements of the Butterworth WDF N=4, A_{max} in dB.

A_{max}	A_1	B_2	A_3	B_4	A_{51}	A_{52}
0.1	0.7022	0.3784	0.2876	0.3188	0.6021	0.9815
0.2	0.6872	0.3658	0.2655	0.2951	0.5781	0.9722
0.3	0.6782	0.3531	0.2532	0.2813	0.5636	0.9742
0.4	0.6716	0.3441	0.2446	0.2717	0.5530	0.9718
0.5	0.6664	0.3371	0.2380	0.2642	0.5446	0.9699
0.6	0.6621	0.3313	0.2325	0.2580	0.5376	0.9682
0.7	0.6584	0.3264	0.2280	0.2529	0.5317	0.9667
0.8	0.6551	0.3222	0.2240	0.2484	0.5264	0.9653
0.9	0.6522	0.3184	0.2205	0.2444	0.5217	0.9641
1.0	0.6495	0.3149	0.2173	0.2408	0.5174	0.9629

Table 3. Elements of the Butterworth WDF N=5, A_{max} in dB.

A_{max}	A_1	B_2	A_3	B_4	A_5	B_{61}	B_{62}
0.1	0.7256	0.4125	0.2871	0.2635	0.3355	0.6363	0.9895
0.2	0.7131	0.3944	0.2696	0.2458	0.3149	0.6170	0.9870
0.3	0.7066	0.3837	0.2595	0.2356	0.3029	0.6053	0.9854
0.4	0.7014	0.3761	0.2523	0.2284	0.2948	0.5968	0.9841

0.5	0.6072	0.3702	0.2468	0.2229	0.2879	0.5901	0.9830
0.6	0.6938	0.3653	0.2422	0.2184	0.2824	0.5845	0.9821
0.7	0.6909	0.3611	0.2384	0.2145	0.2779	0.5797	0.9831
0.8	0.6883	0.3574	0.2350	0.2112	0.2739	0.5754	0.9805
0.9	0.6859	0.3542	0.2321	0.2083	0.2703	0.5716	0.9798
1.0	0.6838	0.3512	0.2294	0.2056	0.2671	0.5681	0.9792

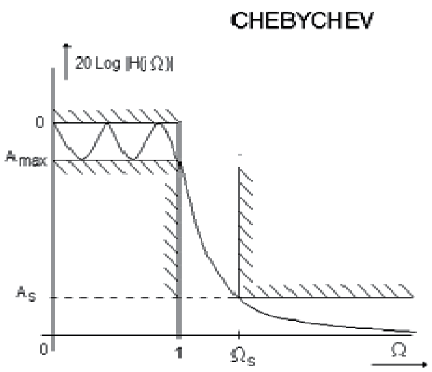
Table 4. Elements of the Butterworth WDF N=6, A_{max} in dB.

A_{max}	A_1	B_2	A_3	B_4	A_5	B_6	A_{71}	A_{72}
0.1	0.7462	0.4392	0.2994	0.2497	0.2628	0.3597	0.6679	0.9940
0.2	0.7635	0.4236	0.2842	0.2352	0.2473	0.3415	0.6522	0.9927
0.3	0.7307	0.4144	0.2754	0.2268	0.2384	0.3308	0.6428	0.9917
0.4	0.7265	0.4078	0.2692	0.2208	0.2320	0.3232	0.6358	0.9911
0.5	0.7231	0.4027	0.2643	0.2162	0.2271	0.3173	0.6304	0.9904
0.6	0.7204	0.3984	0.2603	0.2125	0.2231	0.3125	0.6258	0.9899
0.7	0.7280	0.3948	0.2569	0.2093	0.2197	0.3083	0.6218	0.9895
0.8	0.7159	0.3916	0.2539	0.2065	0.2168	9.3047	0.6184	0.9890
0.9	0.7140	0.3887	0.2513	0.2041	0.2142	0.3015	0.6153	0.9887
1.0	0.7123	0.3862	0.2490	0.2019	0.2118	0.2986	0.6124	0.9883

Table 5. Elements of the Butterworth WDF N=7, A_{max} in dB.

10. Tables of the Chebychev wave digital filter

In tables 6-9 are the elements of Chebychev wave digital filters for various attenuation A_{max} in the pass-band and n=3, 5, 7 and 9. The tables was designed for sampling frequency f_s=0.5.



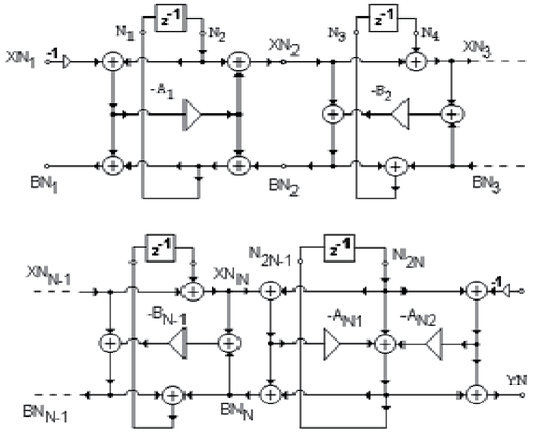


Fig. 17. Frequency response and structure of Chebychev wave digital filter

A_{max}	A_1	B_2	A_{31}	A_{32}
0.10	0.4922	0.3022	0.4620	0.7574
0.25	0.4341	0.2774	0.4310	0.6812
0.50	0.3852	0.2599	0.4126	0.6114
1.00	0.3307	0.2496	0.3995	0.5293
2.00	0.2695	0.2445	0.3929	0.4331
3.00	0.2300	0.2442	0.3925	0.3696

Table 6. Elements of Chebychev WDF N=3, A_{max} in dB.

A_{max}	A_1	B_2	A_3	B_4	A_{51}	A_{52}
0.10	0.4658	0.2536	0.2161	0.2245	0.4179	0.7374
0.25	0.4197	0.2404	0.2059	0.2132	0.3987	0.6721
0.50	0.3696	0.2311	0.1975	0.2044	0.3869	0.5965
1.00	0.3190	0.2262	0.1911	0.1981	0.3798	0.5168
2.00	0.2610	0.2251	0.1857	0.1933	0.3797	0.4229
3.00	0.2231	0.2265	0.1828	0.1922	0.3830	0.3608

Table 7. Elements of Chebychev WDF N=5, A_{max} in dB.

A_{max}	A_1	B_2	A_3	B_4	A_5	B_6	A_{71}	A_{72}
0.10	*0.4585*	*0.2437*	*0.2022*	*0.1947*	*0.1962*	*0.2123*	*0.4049*	*0.7313*
0.25	*0.4087*	*0.2316*	*0.1944*	*0.1884*	*0.1903*	*0.2028*	*0.3875*	*0.6550*
0.50.	*0.3653*	*0.2250*	*0.1893*	*0.1861*	*0.1867*	*0.1968*	*0.3872*	*0.5925*
1.00	*0.3158*	*0.2213*	*0.1847*	*0.1834*	*0.1836*	*0.1919*	*0.3735*	*0.4206*
2.00	*0.2587*	*0.2210*	*0.1805*	*0.1814*	*0.1812*	*0.1880*	*0.3734*	*0.3921*
3.00	*0.2213*	*0.2227*	*0.1783*	*0.1806*	*0.1802*	*0.1862*	*0.3782*	*0.3589*

Table 8.Elements of Chebychev WDF N=7, A_{max} in dB

A_{max}	A_1	B_2	A_3	B_4	A_5	B_6	A_7	B_8	A_{91}	A_{92}
0.10	0.4554	0.2399	0.1970	0.1886	0.1854	0.1860	0.1908	0.2081	0.4000	0.7287
0.25	0.4064	0.2287	0.1912	0.1847	0.1821	0.1825	0.1863	0.1996	0.3836	0.6569
0.50	0.3636	0.2227	0.1867	0.1823	0.1800	0.1844	0.1834	0.1943	0.3751	0.5908
1.00	0.3145	0.2193	0.1826	0.1804	0.1783	9.1787	0.1811	0.1899	0.3708	0.5123
2.00	0.2578	0.2194	0.1789	0.1790	0.1769	0.1773	0.1793	0.1864	0.3273	0.4196
3.00	0.2205	0.2213	0.1769	0.1784	0.1763	0.1604	0.1647	0.1873	0.3753	0.3583

Table 9.Elements of Chebychev WDF N=9, A_{max} in dB

11. Tables of Cauer wave digital filter

In tables 10-12 are the elements of Cauer wave digital filters for various attenuation A_{max} in the pass-band and N=3, 5 and 7. The tables was designed for sampling frequency f_s=0.5.

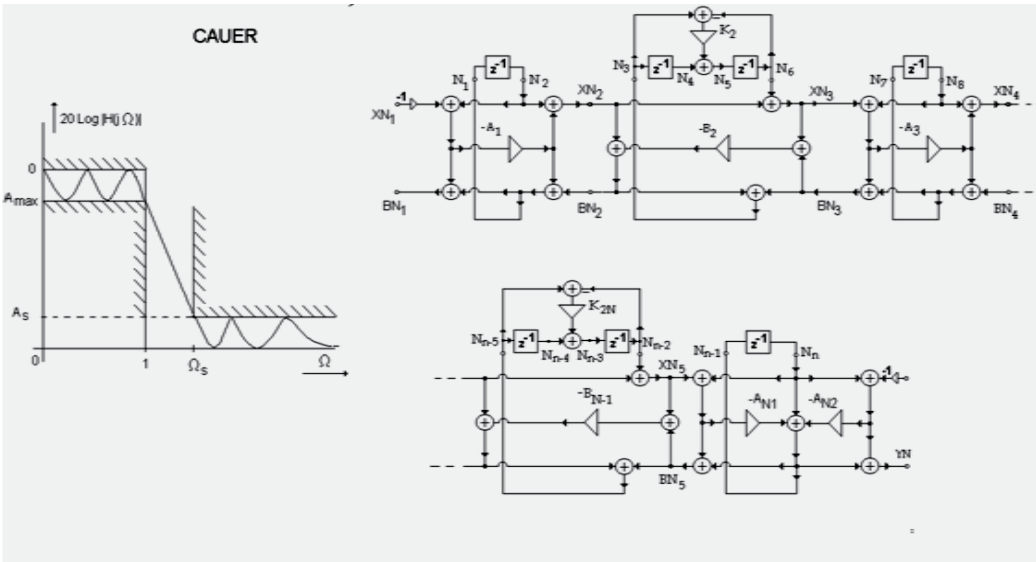


Fig. 18. Frequency response and structure of Cauer filter

C	A_s	A_1	B_2	A_{31}	A_{32}
0.0004	24.7	0.7504	0.5766	0.7314	0.9629
0.0017	30.7	0.7209	0.4907	0.6664	0.9374
0.0039	34.3	0.6673	0.4569	0.6272	0.9160
0.0109	38.7	0.6190	0.4063	0.5778	0.8803
0.0279	42.8	0.5706	0.3612	0.5339	0.8365
0.0436	44.8	0.5461	0.3459	0.5140	0.8114
0.0988	48.3	0.4983	0.3161	0.4804	0.7573
0.1773	50.9	0.4614	0.2979	0.4590	0.7110
0.2803	53.0	0.4306	0.2855	0.4442	0.6699
1.2494	60.0	0.3147	0.2593	0.4118	0.4999

Table 10. Elements of Cauer WDF N=3, Ω_s=4.8097, K_2=-0.93686

A_{max}	A_s	A_1	B_2	A_3	B_4	A_{51}	A_{52}
0.0017	41.3	0.657	0.399	0.330	0.436	0.734	0.915
0.0039	44.8	0.627	0.373	0.311	0.404	0.690	0.893
0.0109	49.2	0.585	0.343	0.290	0.369	0.639	0.856
0.0279	53.3	0.543	0.318	0.272	0.341	0.596	0.813
0.0436	55.3	0.522	0.306	0.264	0.323	0.577	0.788
0.0988	58.8	0.479	0.288	0.251	0.310	0.546	0.736
0.1773	61.4	0.446	0.277	0.243	0.298	0.527	0.691
0.2803	63.5	0.418	0.270	0.237	0.289	0.514	0.652
1.2494	70.5	0.309	0.256	0.221	0.269	0.492	0.487

Table 11. Elements of Cauer WDF N=5, Ω_s=2.000, K2=-0.827, K_4=-0.627

A_{max}	A_s	A_1	B_2	A_3	B_4	A_5	B_6	A_{71}	A_{72}
0.0004	43.5	0.6901	0.4211	0.3521	0.4838	0.4446	0.4972	0.8781	0.9364
0.0017	49.5	0.6543	0.3827	0.3240	0.4464	0.4166	0.4471	0.7973	0.9071
0.0039	53.0	0.6167	0.3611	0.3093	0.4274	0.4015	0.4209	0.7502	0.8844
0.0109	57.5	0.5776	0.3353	0.2924	0.4064	0.3841	0.3912	0.6972	0.8482

0.0279	61.6	0.5375	0.3134	0.2785	0.3897	0.3698	0.3672	0.6527	0.8056
0.0436	63.5	0.5168	0.3038	0.2725	0.3826	0.3636	0.3569	0.6293	0.7766
0.0988	67.1	0.4758	0.2880	0.2624	0.3712	0.3534	0.3401	0.6015	0.7300
0.1773	69.7	0.4434	0.2784	0.2558	0.3642	0.3470	0.3295	0.5823	0.6862
0.2803	71.7	0.4158	0.2720	0.2511	0.3593	0.3425	0.3221	0.5697	0.6472
1.2494	78.7	0.3084	0.2606	0.2378	0.3475	0.3310	0.3035	0.5484	0.4845

Table 12. Elements of Cauer WDF N=7, Ω_s=1.41421, K_2=-0.7815, K_4=-0.3491, K_6=-0.4867

12. Realization of wave digital filters with DSP C6711 by SIMULINK

The simulink models of the parallel and serial adaptors figure 19 was added in the simulink browse library between common used block.

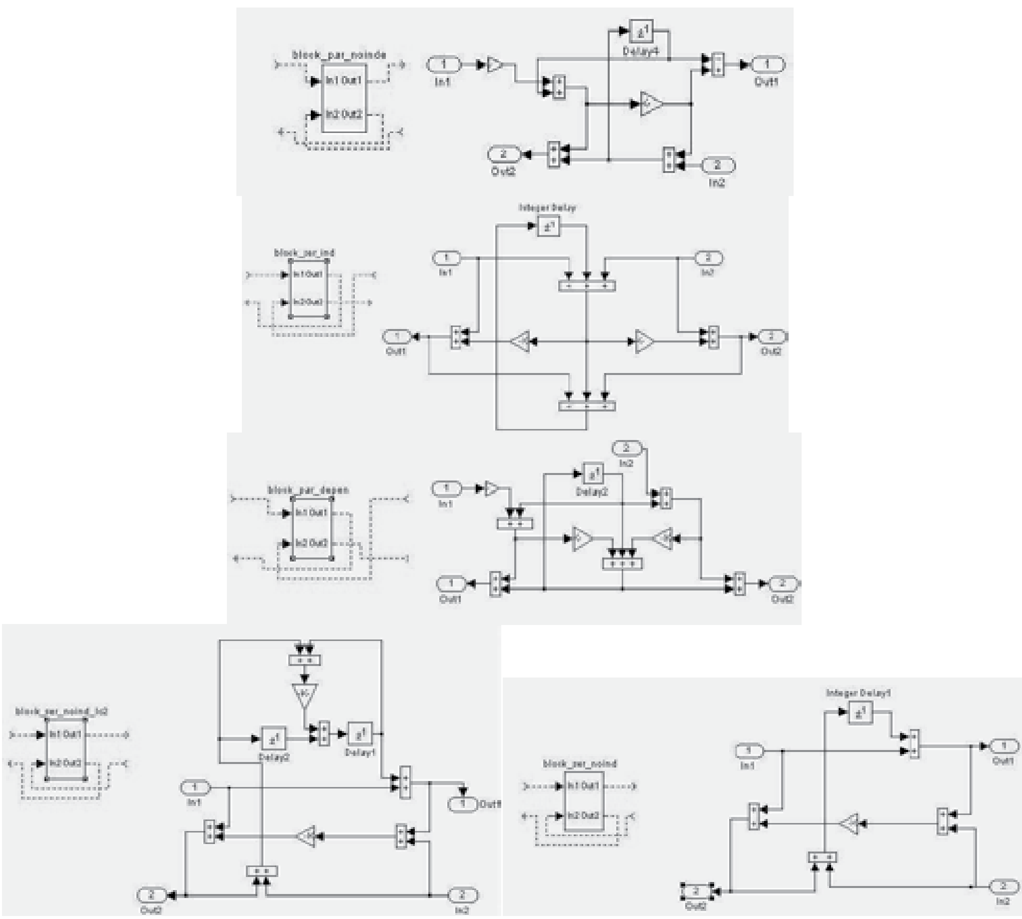


Fig. 19. Models of the serial and parallel adaptors deposited in simulink browse library.

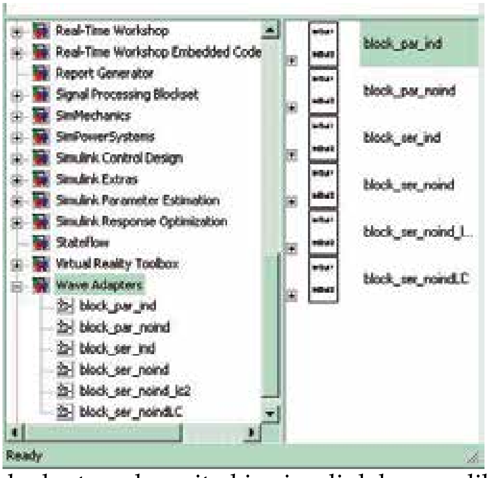


Fig. 20. Parallel and serial adaptors deposited in simulink browse library.

The simulink model of the fifth order filter in the figure 4 corresponds to the realization of a wave digital filter application on TMS320C6711 DSK using Embedded Target for Texas Instruments TMS320C6000 DSP Platform. The model of the filter, figure 21, was created by means of serial and parallel block that were added to the window simulink library browser. In the output and input of WDF were added ADC and DAC convertors of the TMS320C6711 that are in the simulink library browser, Embedded Target for TI C6000 DSP and C6711DSK board support.

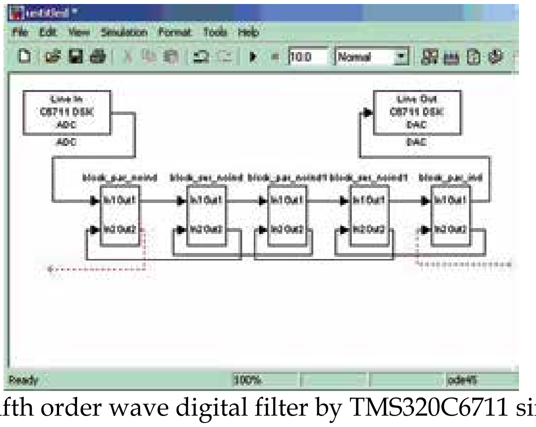


Fig. 21. Realization of fifth order wave digital filter by TMS320C6711 simulink.

This project created in code composer studio can be seen in figure 22 and can run on the DSPC6711.

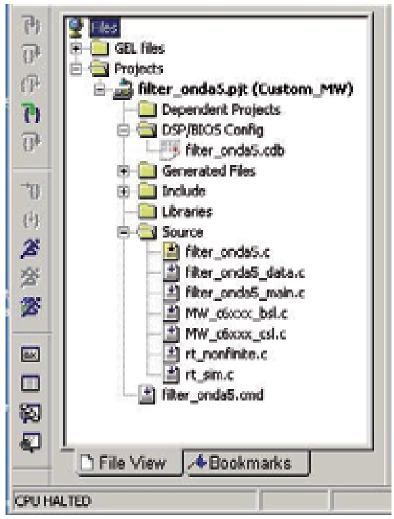


Fig. 22. Projects of the realization of Wave digital filter by TMS320C6711

13. Bank of filters with Chebychev and inverse Chebychev wave digital filters

In this example we shall design bank of filters with wave digital filters. We use Chebychev filter for order N=3 and A_{max}=0.1 dB. Transfer function of the Chebychev low-pass we obtain by matlab using the commands [a,b]=cheby1(3,0.1,0.5,'s')

$$H_0(s) = \frac{1}{2.0353s^3 + 2.0116s^2 + 2.5206s + 1.0000} = \frac{1}{h_0(s)}$$

$$Fl(s) = \frac{1}{2.0353s^3 + 1.5265s} = \frac{1}{f(s)}$$

From these equations we can obtain input impedance $Z_e(s)$ of the circuit and the structure in figure 23.

$$Z_e(s) = \frac{h0(s)+f(s)}{h0(s)-f(s)} = \frac{4.0707s^3 + 2.0116s^2 + 4.0472\,s + 1}{2.0116s^3 + 0.9941s^2 + 1}$$

$$Z_e(s) = \cfrac{1}{2.0235s + \cfrac{1}{0.9941s + \cfrac{1}{2.0235s + 1}}}$$

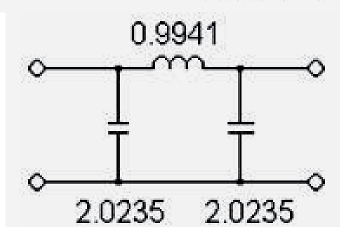


Fig. 23. LC prototype of Chebychef low-pass filter.

From the LC prototype of the Chebychev filter we obtain discrete realization of the wave digital filter that is demonstrate in figure 24, where

A_1=0.3307 B_2=0.2496 A_{31}=0.3995 A_{32}=0.5293

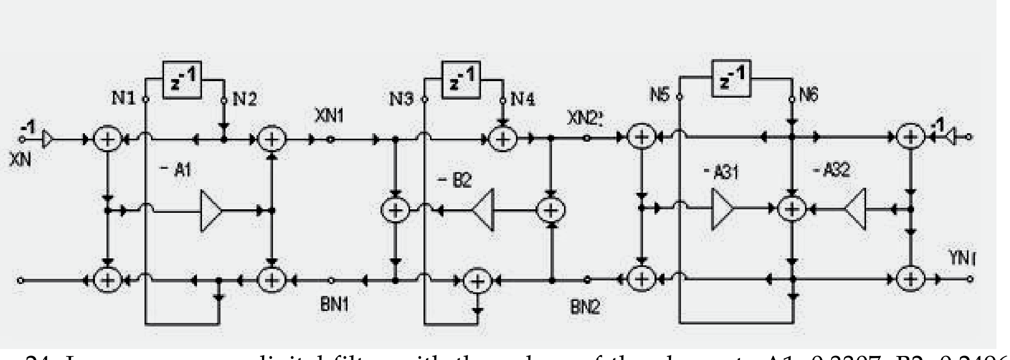


Fig. 24. Low-pass wave digital filter with the values of the elements A1=0.3307, B2=0.2496, A31=0.3995, A32=0.5293.

In the parte synthesis of the bank of filters we choose low-pass Chebychev inverse filter that is inverse to the low-pass in the part of analysis bank.

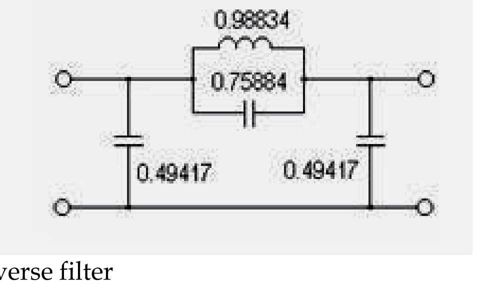


Fig. 25. LC Chebychev inverse filter

From the Chebychev inverse filter figure 25, the coefficients of the parallel and serial adaptor are

A_1=0.6693 B_2=0.2496 A_{31}=0.7032 A_{32}=0.5293 K=0.1428

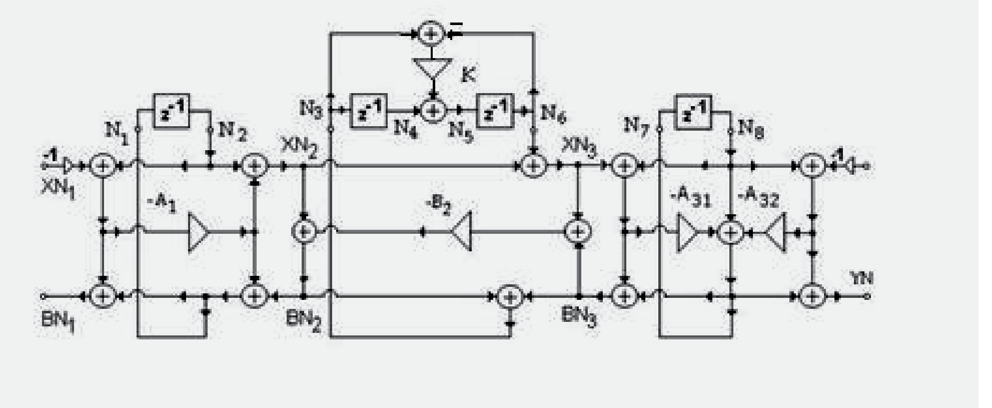


Fig. 26. Low-pass wave digital filter Chebychev inverse with coefficients A_1=0.6693 B_2=0.2496 A_{31}=0.7032 A_{32}=0.5293 K=0.1428

Normalized attenuation of the low-pass and high-pass filter is demonstrated in the figure 27.

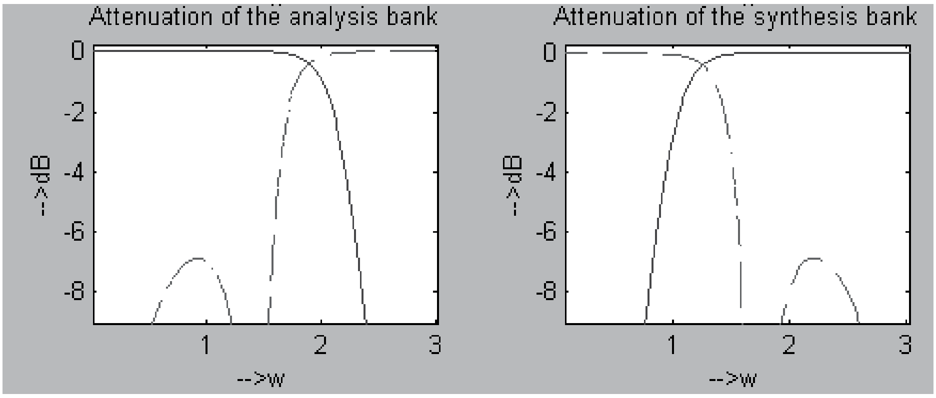


Fig. 27. Normalized attenuation of the filter bank.

In the figure 28 is presented analysis part of the filter bank. This part is designed by Chebychev low-pass filter and inverse Chebychev high-pass filter. The coefficients of the chebychev low-pass filter are A1=0.3307, B2=0.2496, A31=0.3995 and A32=0.5293. The coefficients of inverse Chebychev high pass filter are A1=0.6692, B2=0.2496, A31=0.7032, A32=0.5293 and K=0.1428. In the adders connected in the output of the delay elements must be changed the signature because must be fulfill the condition $H_1(z)=G_0(-z)$, where $G_0(z)$ is the transfer function of the inverse Chebychev low-pass transfer function from the Synthesis filter bank from figure 29.

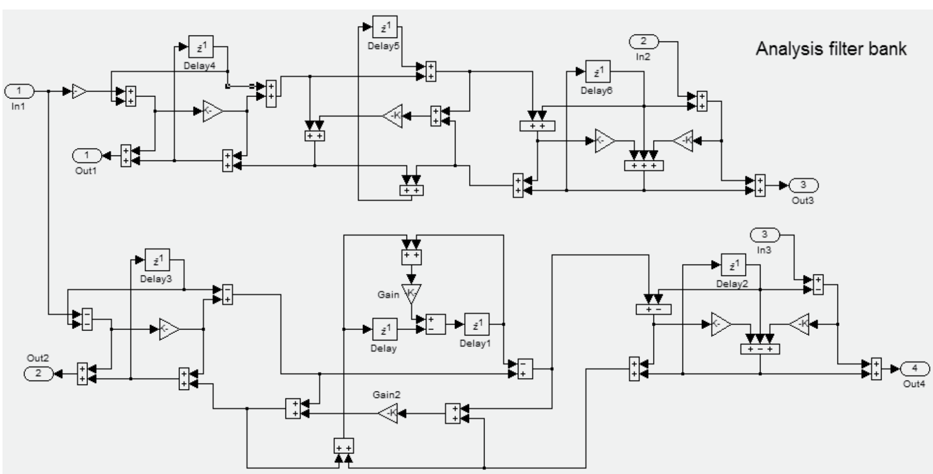


Fig. 28. Analysis wave digital filter bank

In the figure 29 is presented synthesis part of the filter bank. This part is designed by Chebychev inverse low-pass filter and Chebychev high-pass filter. The coefficients of the chebychev low-pass filter are A1=0.3307, B2=0.2496, A31=0.3995 and A32=0.5293. The coefficients of inverse Chebychev high pass filter are A1=0.6692, B2=0.2496, A31=0.7032, A32=0.5293 and K=0.1428. In the adders, connected in the output of the delay elements of the Chebychev high-pass wave digital filter, must be changed the signature because must be fulfill the condition $G_1(z)= -H_0(-z)$, where $H_0(z)$ is the transfer function of the Chebychev low-pass transfer function from the Analysis filter bank, figure 28.

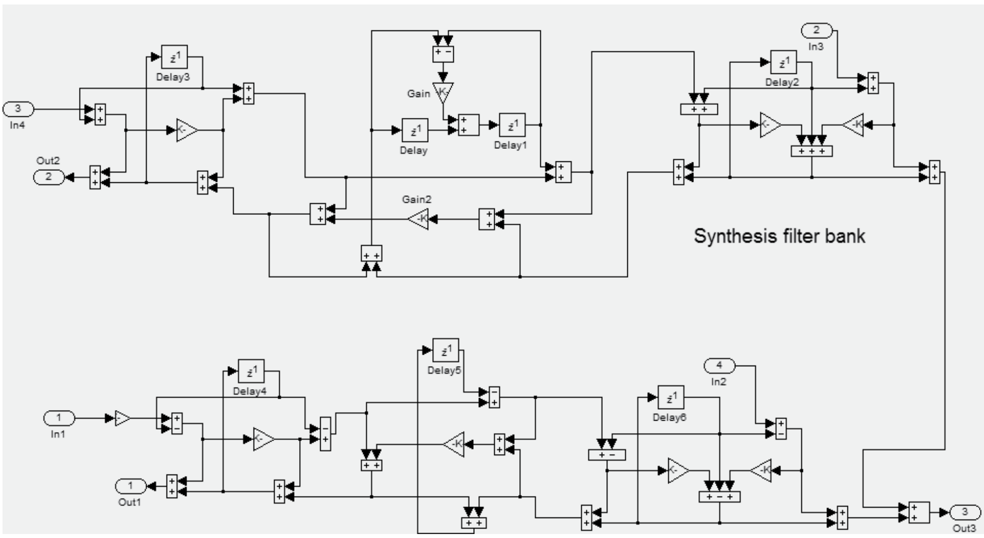


Fig. 29. Synthesis wave digital filter bank

In figure 30 is presented the model of the filter bank composed from analysis and synthesis parts and from down-sampler and sub-sampler. The input is connected to a speech signal saved in Workspace1. The output of the filter-bank was connected with oscilloscope. Figure 31 shows speech signal in the input and the output of the filter bank and spectrograms of a speech signal in the output and input of the filter bank. From the responses in the figure 31 we can deduce that filter bank composed from Chebychev and Inverse Chebychev filter has the good properties.

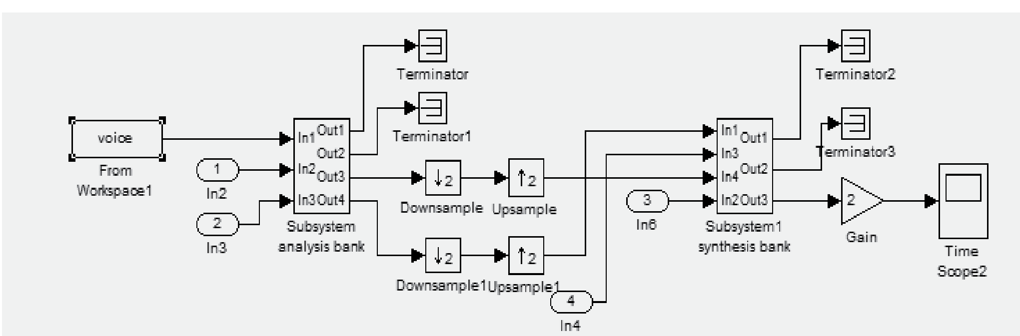


Fig. 30. Simulink model of the wave digital filter bank

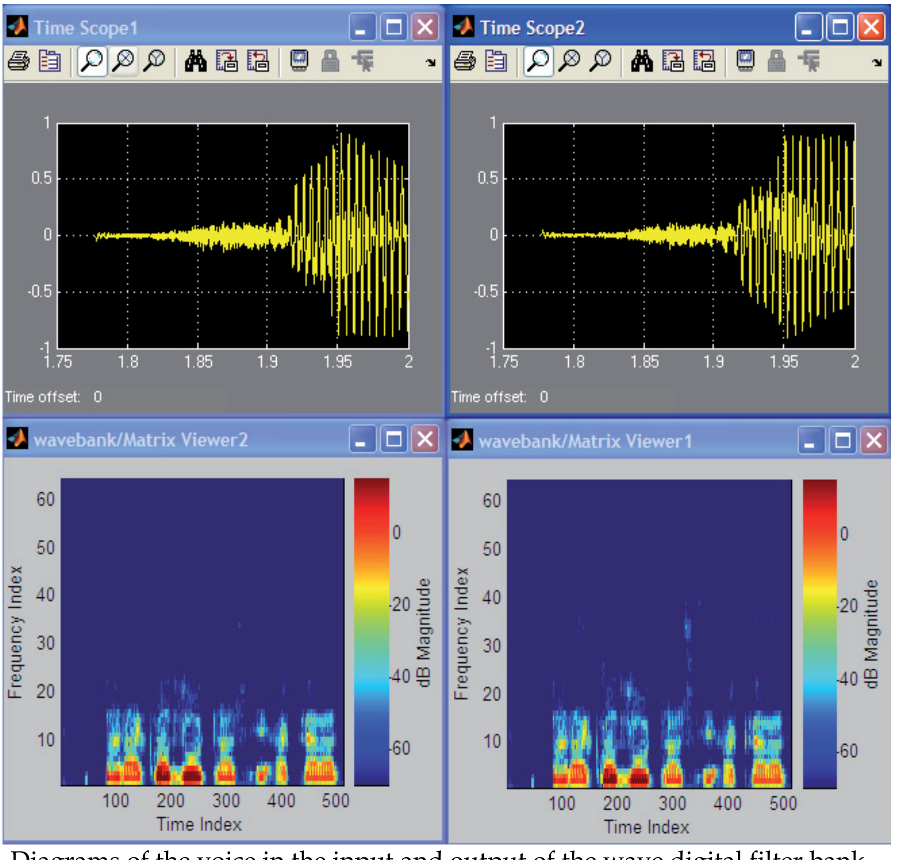


Fig. 31. Diagrams of the voice in the input and output of the wave digital filter bank.

14. Conclusion

Though the structure of the wave digital filter is more complicated than other structures, the algorithm for implementation on the DSP is very simple and it is very easy to propose the general algorithm for the arbitrary order of the wave digital filter. These structures are not too much sensitive to the error of quantization as other types of the filters. With small modification of the presented programs can be created another tables of the WDF. The parts of the programs can be utilized for implementing of the wave digital filters in digital signal processors DSP. In this chapter was constructed filter-bank from Chebychev and Inverse Chebychev filter. This filter bank was simulated by Signal Processing toolbox of MATLAB contains the speech signal saved in Worspace1. In the figure 31 we have illustrated results obtained by analysis of a speech signal using the Signal Processing Toolbox in MATLAB.

15. References

Fettweis.: *Digital filter structures related to classical filter networks*. Arch. Elektron Uebertragunstech., 1972,25, pp.78-89.

A.Sedlmayer and A. Fettweis.: *Digital filters with true ladder configurations* Int. J. Circuit Theory and Appl., 1973, 1, pp. 1-5.

F. García Ugalde, B Psenicka and M.A. Rodríguez.: *Implementation of Wave Digital Filters on a DSP Using Simulink.* IASTED, Circuits, Siognal and Systems, November 20-22, 2006, San Francisco, CA, USA

B. Psenicka, F. García-Ugalde and A. Romero Mier y Teran.: *Synthesis of the low-pass and high-pass Wave digital filters.ICINCO 2008. International conference on informatics in control and Robotics.* May 11-15, 2008, Madeira-Portugal.

Rudolf Saal.: Handbuch zum Filterentwurf AEG Telefunken, 1979.

Sanjit K. Mitra .: Digital Signal Processing. A computer-Based Approach. McGraw-Hill Companies, INC. New York, 1998.

10

Neural PDF Control Strategy for a Hydroelectric Station Simulator

German A. Munoz-Hernandez[1], Carlos A. Gracios-Marin[2], Alejandro Diaz-Sanchez[3], Saad P. Mansoor[4] and Dewi I. Jones[5]

[1]Instituto Tecnologico de Puebla, [2]Universidad Popular Autonoma de Puebla, [3]Instituto Nacional de Astrofisica, Optica y Electronica, [4]University of Bangor, School of Informatics, [5]GWEFR Cyf

[1,2,3]Mexico, [4,5]Wales, U.K.

1. Introduction

Dinorwig is the largest pumped storage hydroelectric scheme in Europe. Located in North Wales and operated by the First Hydro Company, Dinorwig is a power station situated in an environmentally sensitive region whose 16km of tunnels have been constructed deep inside Elidir Mountain. The station has six 300 MW rated turbines, driving synchronous generators which feed power into the British national grid. Dinorwig provides rapid response frequency control when peak demands occur; its reversible pump/turbines are capable of reaching maximum generation in less than 16 seconds. This hydroelectric station has a single tunnel drawing water from an upper reservoir (Lake Marchlyn) into a manifold, which splits the main flow into six penstocks. Each penstock feeds a turbine to generate power using a guide vane to regulate the flow. The electrical power is controlled by individual feedback loops on each unit. The reference input to the power loop is the grid frequency deviation from its 50 Hz set point, thus forming an outer frequency control loop. Using off-peak electricity (usually at night), Dinorwig's reversible Francis turbines are used to pump water from the lower reservoir, Lake Peris, back to Lake Marchlyn. Mansoor et al. have derived a multivariable nonlinear simulation model of this plant, which has provided an improved understanding of its characteristics (Mansoor, et al., 1999; Mansoor, et al., 2000). Its main features are non-minimum-phase dynamics, poorly damped poles (associated with water-hammer in the supply tunnel and electrical synchronization) and a nonlinear relationship between flow and power. It is also known (Kundur, 1994; Working group on prime mover energy supply, 1992) that there is a significant hydraulic coupling between the turbines because of the common supply. This makes the plant a good candidate for the application of auto-tuning control.

The chapter begins with a brief discussion of the nonlinear mathematical model of the power plant. Then neural network theory is reviewed, followed by a description of the application of neural pseudo-derivative control (PDF) to the model of Dinorwig (Kang, et al., 1991). This is followed by a brief review of some of the concepts of anti-wind-up PID, after which a discussion of the software platform based on Simulink© models is presented. Finally, results are presented which show the improved response provided by neural PDF.

2. Hydroelectric plant model

The hydroelectric plant model can be divided into three subsystems: guide vane, nonlinear hydraulics and turbine/generator (figure 1). Mansoor et al. (2000) developed a multivariable non-linear model that includes a rate limit and saturation in the guide vane dynamics, as shown in figure 2.

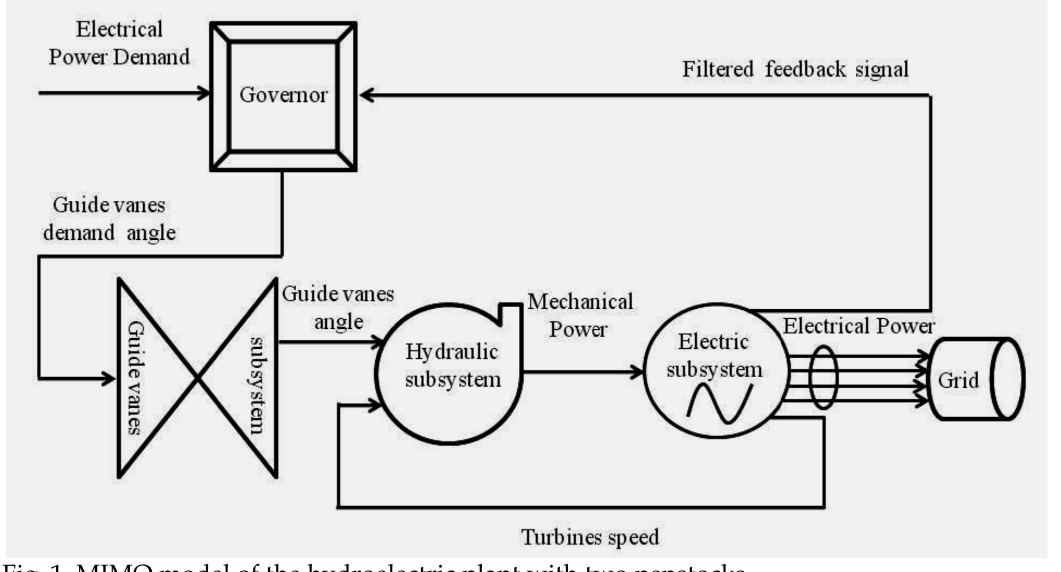


Fig. 1. MIMO model of the hydroelectric plant with two penstocks.

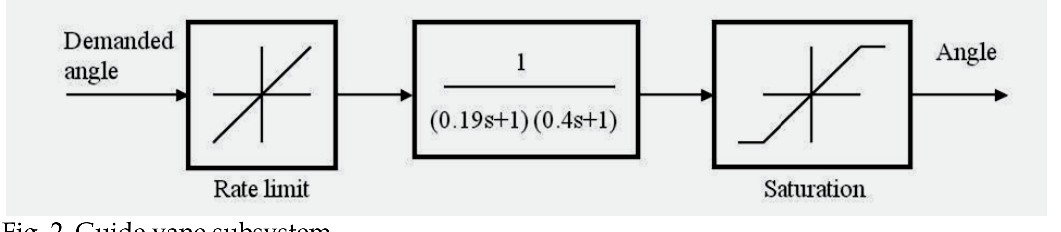


Fig. 2. Guide vane subsystem.

In this study a nonlinear model that takes into account the effects of the water column, including water compressibility and pipe wall elasticity, was employed (Working group on prime mover energy supply, 1992). Figure 3 shows the nonlinear elastic model of a single penstock. The coupling effect is represented with the inclusion of the main tunnel, which is modelled in the same form as a penstock. A_t is the turbine gain; its value depends directly on the turbine MW rating and inversely on the Generator MVA rating. f_p is the head loss coefficient for the penstock. Z_0 is the surge impedance of the conduit. T_e is the wave travel time, defined as the time taken for the pressure wave to travel the length of the penstock (*l*) to the open surface. *v* is the velocity of sound in water (Kundur, 1994).

$$T_e = \frac{l}{v} \tag{1}$$

$$Z_0 = \frac{T_W}{T_e} \tag{2}$$

T_w is the *water starting time* of the main tunnel and the penstocks. Kundur (1994) defines the *water starting time* as the time required for a head to accelerate the water in the penstock from standstill to a specific velocity. Its value depends directly on the constructional dimensions of the main tunnel and penstocks. In this nonlinear model, figure 3, G is the per unit gate opening and P_{mech} is the mechanical power produced by a single turbine. The value of T_e depends directly on the length of the penstock and inversely on the wave velocity (equation 1). Z_o depends directly on the flow rate and inversely on the head of water and on the acceleration due to gravity (equation 2). The value of A_t depends directly on the turbine MW rating and inversely on the Generator MVA rating (Mansoor, 2000). The models are expressed in the per-unit system, normalized to 300 MW and 50 Hz.
The electrical subsystem is based on the 'swing' equations (Kundur, 1994) and includes the effect of synchronizing torque. For noise reduction a first order filter is included in the feedback loop (figure 4).

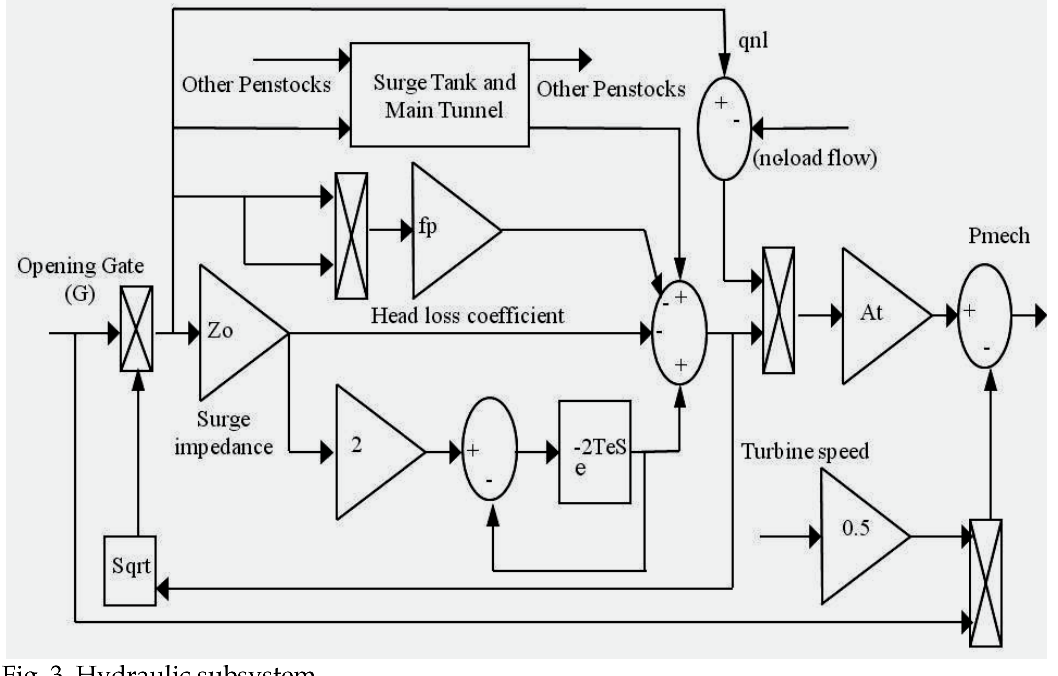


Fig. 3. Hydraulic subsystem

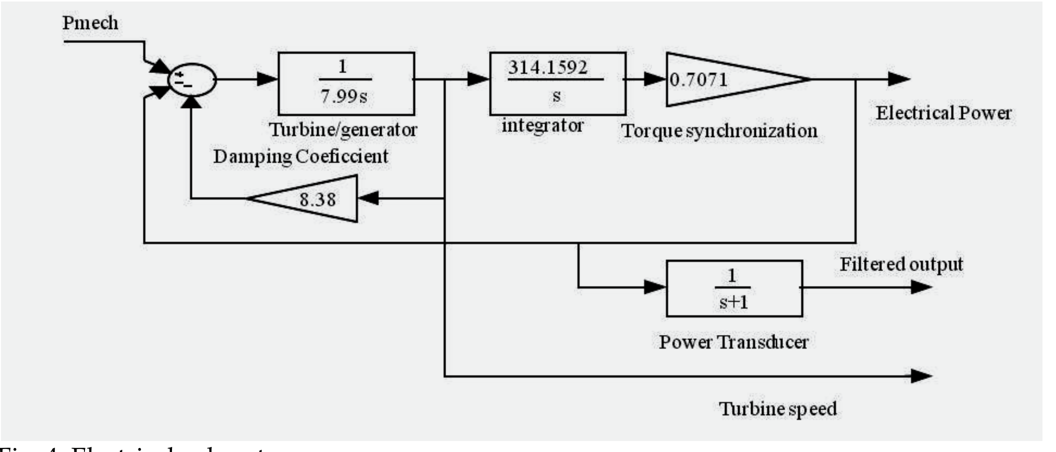


Fig. 4. Electrical subsystem.

3. Artificial Neural Networks in Adaptive Control

3.1 Basic theory

Artificial Neural Networks (ANN) has been a popular research theme within the scientific community since the early 1980's. The increasing range of application of ANNs in computational modelling and evolutionary structures has led to the development of diverse types of networks, which are loosely based on the structures of biological nervous systems. Munakata (2008) reports five basic types of ANN that are focused on the solution of control design problems. These are listed below:

1. **Supervised control**: This type of network tries to imitate a human or a computer program which already knows how to perform a task.
2. **Direct Inverse Control (DIC)**: The ANN consists of mapping the multivariable output values of the system under control versus the values of each independent variable. The desired behaviour is supplied by a human trainer or an 'expert' computer program in association with a database.
3. **Neural Adaptive Control**: Here it is necessary to achieve a pre-specified target because the ANN is substituted for the (linear) mappings used in conventional adaptive control. The two schemes most often used are the Self-Tuning Regulator (STR) and Model-Reference Adaptive Control (MRAC).
4. **Back-propagation Through Time (BTT):** In this case, the user specifies a utility function or performance measure to be maximized and a model of the external environment. Back-Propagation is used to calculate the derivative of the utility function *summed across all future times* with respect to *current* actions. These derivatives are then used to adapt the ANN which outputs the actions, or to adapt a schedule of actions. This type of applications can be found in pattern recognition and intelligent control of manufacturing processes (Miller, et al., 1991).
5. **Adaptive Critic Methods:** This scheme is similar to BTT but the ANN, now called a critic network, evaluates the progress that the system is making.

The five ANN strategies mentioned above have gained acceptance and are extensively used in the solution of actual complex, non-linear control problems. One engineering discipline that has been enriched by the properties of the ANN is adaptive control theory. This particular theory offers the possibility of adjusting the parameters of the regulator in order to reduce the difference between the set-point and the output of the process. Narendra & Mukhopadhyay (1996) address the application in adaptive control of Back Propagation (BP), the most common type of ANN because it computes rapidly and is easy to implement in an algorithm. A typical back-propagation neural network is formed as shown in figure 5. Back-Propagation is known as a *training method* for the neural network (ANN) and many authors have reported successful application of both the basic method and its more advanced variants to practical cases (Werbos, 1974; Rumelhart, et al., 1986; Albeto Aguado Behar, 2000).

The BP training method has the capability to develop as precise an approximation as required for any non-linear function y=f(x), starting on set of $x_{i,}y_i$ pairs of sample data. The method is in fact a generalization of the fastest descent learning law presented by Widrow and Hoff (1960). They analyzed a two layer ANN called Adaptive Linear Network (ADALINE) and developed a local descent method using the gradient direction. The method proposed by Widrow & Hoff was generalized to multiple layers and given the name "back-propagation algorithm" (Werbos, 1974; Rumelhart, et al., 1986). BP is the most widely used type of ANN for optimization problems. The user is allowed to pick any utility function, performance measure, or cost function, which can be maximized or minimized, and the method computes the impact on present or future actions. BP can be implemented using almost any Programmable Device (Gracios, et al., 2005).

The typical ADALINE structure is a linear combination of nodes interconnected to form several layers which may or may not have interactions between them. Typically, the layers are known (according to the flow of information through the Network) as the input, hidden and output layers. There is no limit to the number of internal layers contained within the hidden layer. Each layer is fully connected to the succeeding layer (figure 5). The arrows indicate the flow of information during the recall. Initially, the weights as well as the thresholds of the sigmoid functions are set with small random values. Training begins by presenting data to the input of the ANN. During training, information is propagated back through the network and used to adjust the connection weights. The training exercise will be considered satisfactory upon convergence of the weight adjustment, that is, the cost function that measures the I/O mapping of the ANN is minimized.

The number of layers in an ANN is very important. A two-layer NN can only perform well on a limited range of problems (Minsky & Papert, 1988). On the other hand, multilayer ANNs have a wide spectrum of applications and can deal with problems that are prohibitively difficult for the two-layer ANN. As discussed by Rumelhart et al (1986), the addition of internal layers allows the back propagation algorithm to develop an internal representation of dynamic processes and this feature can be crucial to finding a solution. The internal models formed in two-layer ANNs with back-propagation are linear ARX in form.

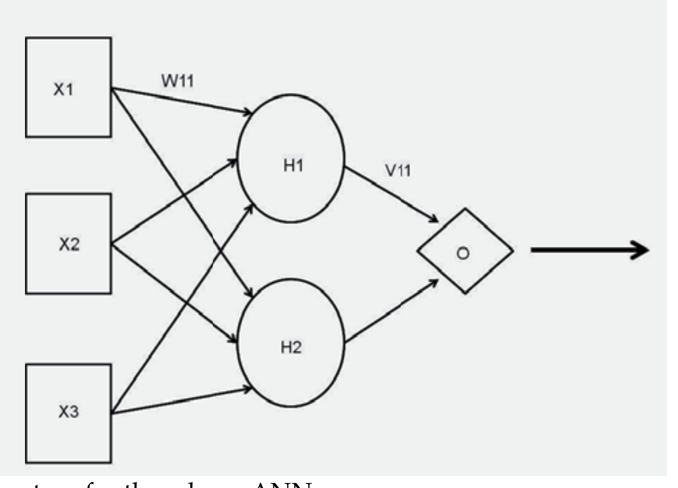


Fig. 5. Generic structure for three layer ANN.

The procedure to calculate the propagation values for each layer is as follows. The input layer receives a value X_i, which is propagated with a weight W_{ij} to the intermediate layer, according to equation (3) where S is called the activation coefficient.

$$S_j^{\ p} = \sum_{i=1}^{n} W_{ij} X_i^{\ p} + W_{n+1,j}^{\ \ p} \tag{3}$$

This activation coefficient S_j^p is propagated by an output function which represents the intermediate layer, in this case the sigmoid function, equation (4).

$$h(S_j^{\ p}) = \frac{1}{1+e^{-S_j^{\ p}}} \tag{4}$$

Propagation may be accomplished using alternative range-limited functions, such as $\tanh^{-1}$, - selection of an appropriate function depends on the final application. As shown in figure 5, it is also necessary to define an exciting function, in order to access the output layer, equation (5), and this is obtained when the weights V_{jk} have been calculated.

$$r_k^{\ p} = \sum_{j=1}^{l} V_{jk}^{\ p} h_j^{\ p} + V_{l+1,k}^{\ \ p} \tag{5}$$

Finally, the output function of the output neuron 'k' is obtained using the same sigmoid function that was used to connect the input and hidden layers, equation (6).

$$O_k^{\ p} = \frac{1}{1+e^{-r_k^{\ p}}} \tag{6}$$

3.2 Neural PDF

The principal reason that Artificial Neural Networks (ANNs) have earned their position in the analysis, design and implementation of control strategies is their flexibility. If it is possible to acquire any *a priori* knowledge of the structure of the model of a system to be controlled, this information can be used to improve the tuning of typical controllers, such as PD or PID and, in the case considered here, PDF. There are many contributions in the area of artificial neural networks aimed at defining fast and effective strategies to calculate and adjust the parameters of discrete PID control systems (Narendra & Mukhopadhyay, 1996; Garcez & Garcez, 1995). In this work a similar strategy is used to tune a discrete PDF.

There have been several works where ANN have been applied to hydroelectric systems. Garcez and Garcez (1995) applied PI neural control to a linear simulator of a 20 MW hydroelectric power plant. Djukanovic, et al. (1997) validated an adaptive-network based on fuzzy inference to control a low head hydropower system. Yin-Song, et al. (2000) have presented a self-learning control system using a PID fuzzy NN and applied it to a hydraulic turbine governor. Recently, Shu-Qing et al. (2005) have compared a PID controller with a hybridized controller based on genetic algorithms and fuzzy NN when the controllers are used as governors of a hydroelectric power plant model.

In this work a back-propagation strategy has been used to adjust the parameters of a discrete PDF regulator. This strategy was used to adjust a PID controller by Aguado Behar (2000) with great sucess in practical implementations. Figure 6 shows the scheme of the Neural-PDF scheme proposed. The regulation can be calculated by:

$$v_j(t+1) = v_j(t) + \eta sign(\frac{\partial e_y}{\partial e_u})\delta^1 h_j \tag{7}$$

$$w_{ji}(t+1) = w_{ji}(t) + \eta sign(\frac{\partial e_y}{\partial e_u})\delta^2{}_j x_i \tag{8}$$

$$\frac{\partial E(t)}{\partial v_j} = -\delta^1 h_j \frac{\partial e_y}{\partial e_u} \tag{9}$$

Equations 7 and 8 are expressed to recursively adjust the weights for each neuronal interconnection V_j and W_{ij}. Equation 9 is used to develop the minimization of the transfer function for the error. δ is included to calculate the gradient of the function and to express the change of sign in the evolution of the process.

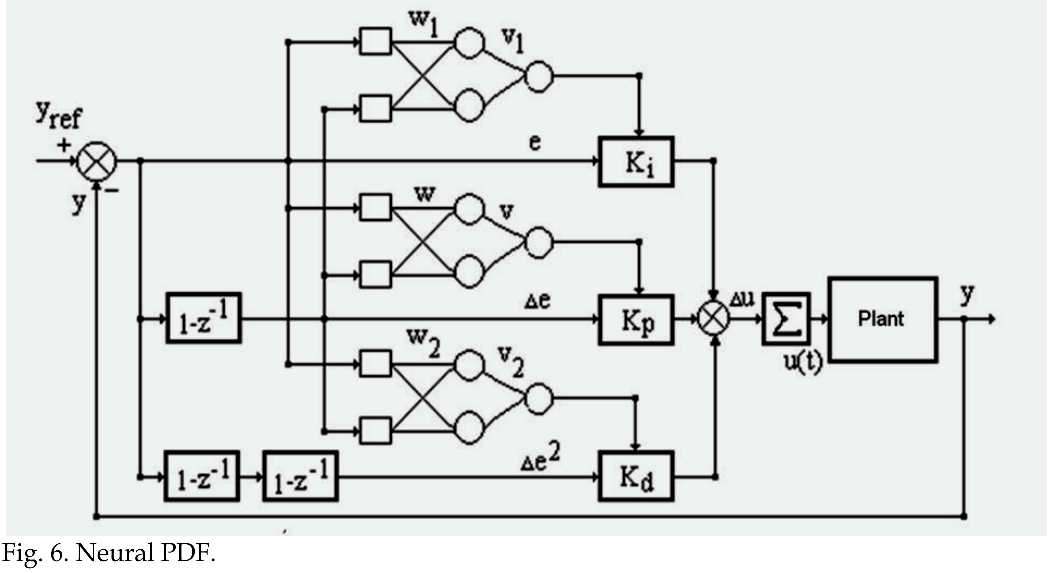


Fig. 6. Neural PDF.

4. Classic controllers for hydroelectric stations

4.1 Dinorwig Governor Configuration

Dinorwig has a digital Governor whose general configuration is shown in Figure 7. There are two control loops, for power and frequency (Mansoor, 2000). In the power control loop the turbine's guide vane is adjusted depending on the power deviation multiplied by the speed regulation droop. The Droop gain is used to change the speed reference of the governor. A PI configuration is used for this control. The frequency control loop provides a reference to the power control loop, which is proportional to the frequency error. There is also a derivative feed-forward loop that allows the system to respond to a rapidly-changing grid frequency.

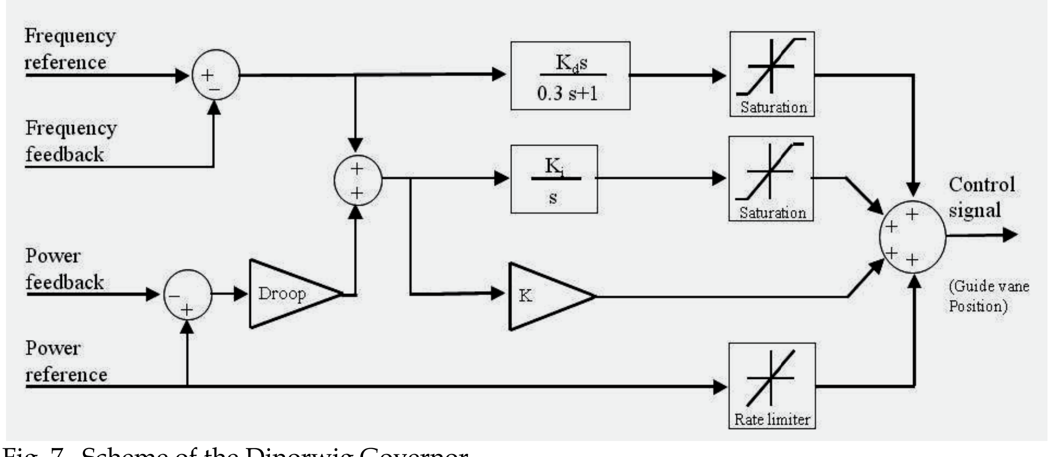


Fig. 7. Scheme of the Dinorwig Governor.

The generators must maintain the speed within an operational band defined by the regulatory authority. When the reference is raised the governor valve will open, thus increasing generation. On the other hand, when the output signal is lowered the reference signal to the governor valve will close, decreasing generation (Wright, 1989). At Dinorwig the governor operates with two droop settings; 1% for high regulation and 4% for low regulation (Mansoor, 2000). The power reference signal sets the reference position for guide vane opening and defines the operating point for the unit when it is working in frequency regulation mode (part load response). Changing the power reference, which also acts as a feed-forward signal, directly sets the guide vane position, in order to produce a rapid reaction when big changes in the power reference appear. The guide vane position reference signal (control signal) is produced by adding the output signals from the P, I and D parts to the feed-forward signal. The power feedback loop compensates the system for the non-linear relationship between guide vane opening and power.

4.2 Anti-windup PID

With careful tuning, PI control can offer good and robust performance. However, all Plant are subject to constraints and their behaviour changes when the constraints are activated. In these circumstances, the performance of a linear controller, such as PI, can deteriorate significantly (Peng, et. al., 1996). When the plant has actuator saturation the integrator value becomes excessively large compared to a linear response (an actuator without saturation), and it then "winds up". In addition, a higher integrator output and a longer settling time are caused by the saturation effect (Peng, et. al., 1996; Bohn & Atherton, 1995; Goodwin, et al., 2001). In other words, windup is produced when the control signal saturates the actuator, because increasing the control signal can no longer accelerate the response of the plant. If this behaviour persists the integrator value can become very large, without affecting the output of the plant. As a consequence, when recovering from saturation, bringing the system back to its correct steady-state value requires the control error to be of the opposite sign for a long time. The result is a large overshoot and a long settling time (Bohn & Atherton, 1995).

Figure 8 shows a general PI controller that includes a tracking anti-windup scheme (Bohn & Atherton, 1995). This controller has an internal feedback path, which drives the integrator to a negative value and forces the output of the system to be in the linear range. The internal saturation is used to reduce the integrator input. As can be seen from Figure 8 the signal to be integrated is modified by the proportional gain (K), therefore the values of the integral gain (K_i) are adjusted in order to maintain equivalence with the classic PI. The saturation limit and the dead zone depend on the constraints fixed by the operator; a value of 0.95 p. u. is commonly used. In this work, the responses of the plant when it is governed by a PI with tracking anti-windup structure will be used as a basis of comparison.

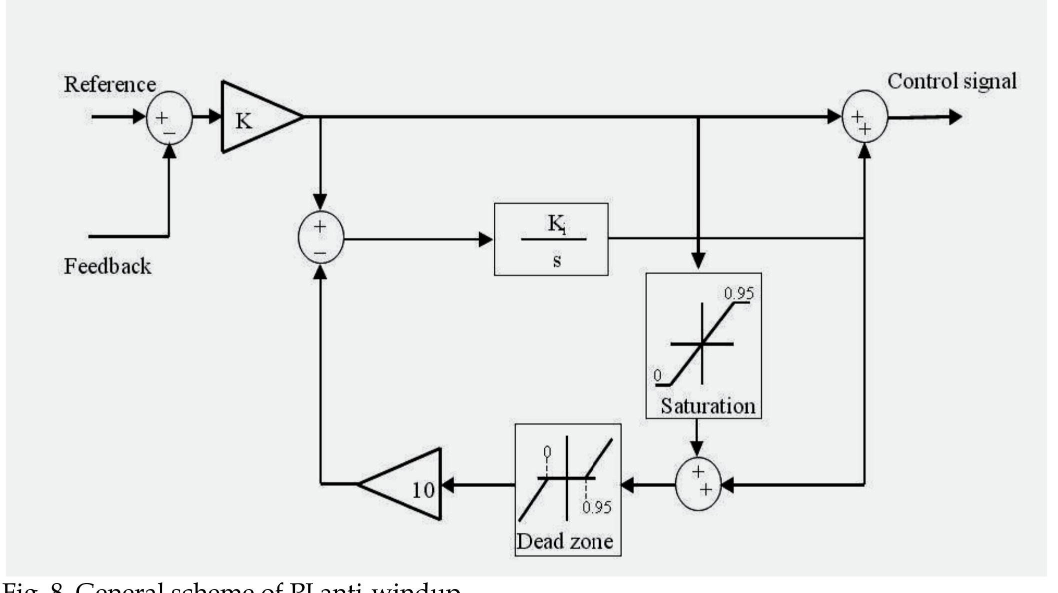


Fig. 8. General scheme of PI anti-windup.

5. Simulink© Model and Program

The Simulink© software tool was used to facilitate studies of the power plant under different governors. This tool has libraries of specific functions (blocks) and the power plant models were constructed using these standard Simulink© functions. Using a dialog box, the parameters of a specific block can be adjusted. For example, the operating point of linear models may be changed. These models can represent the power plant as a SISO or MIMO system and linear or nonlinear behaviour may be selected. Figure 9 shows a schematic of the Simulink© power plant model. The full hydroelectric station model is constructed by combining the four sub-systems: Guide vane dynamics, hydraulic subsystem, turbine/generator and sensor filters. Each block is part of the Simulink© library developed for this study; they can be selected to represent a diversity of modes of operation. For instance, there are three models available to simulate the hydraulic subsystem - linear, nonlinear non-elastic and nonlinear elastic. The guide vane dynamics can be selected with or without rate limitation and saturation. The sensor filters block is a fixed block. The grid model can be adjusted to represent different conditions of the national grid. The governor block has the option of classical and advanced controllers.

Customised Simulink© S-functions were developed for the neural PDF algorithms. These functions can be incorporated within Simulink© models. The neural PDF block accepts η (learning parameters) and sample time. Its inputs are the reference and the output signals of the plant and its output is the control signal. The versatility of Simulink© allows the plant model to be changed easily or the control algorithm to be modified and new results to be viewed and assessed. The neural algorithm calculates the optimal values of the control law parameters. The current criterion of optimality is quadratic error, where the error is the output deviation from the set-point; however this criterion can be changed if necessary. The algorithm takes around 10 iterations (the exact value depending on the magnitude of the change) to find the "best" range of parameter values (training time). When these ranges have been reached the parameters stay constant until the set-point or the plant model change.

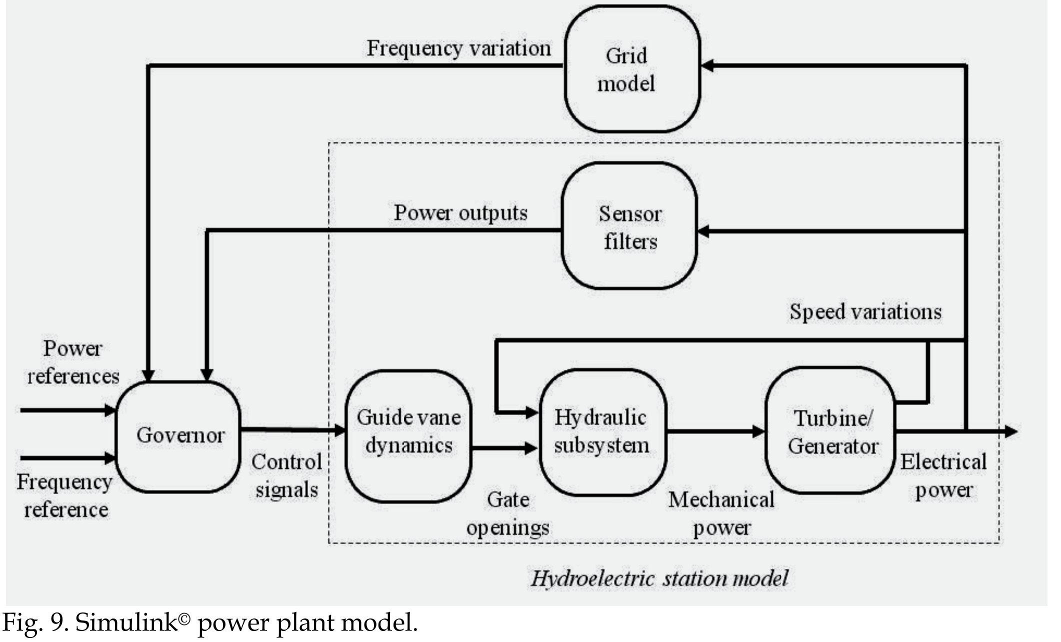


Fig. 9. Simulink© power plant model.

6. Simulation results

As discussed previously, the role of a hydroelectric station in frequency control mode is to provide timely and accurate supply of its demanded power contribution to the power system. The actual form of the power demand is related to Grid frequency variation but, for testing, it can be specified in terms of step, ramp and random input signals.

For all simulations, the model is expressed in the per-unit system, normalized to 300 MW and 50 Hz, and assumes a Grid system with infinite busbars. The neural PDF controller was connected to the nonlinear model of the hydroelectric power plant. A PI controller with parameters fixed at K=0.1 and T_i=0.12 (as currently implemented in practice) is used as a basis of comparison. Figure 10 shows the small step responses (0.05 p.u.) of the hydroelectric plant under anti-windup PI and neural PDF controllers for one unit operational. Figure 11 shows the small step responses (0.05 p.u.) of the power station when all six units are connected. In both cases, the hydroelectric plant performs better with the neural PDF controller, being respectively 10% and 30% faster in the one unit operational and six units operational cases. The undershoot is also reduced in both cases when a PDF controller is driving the process.

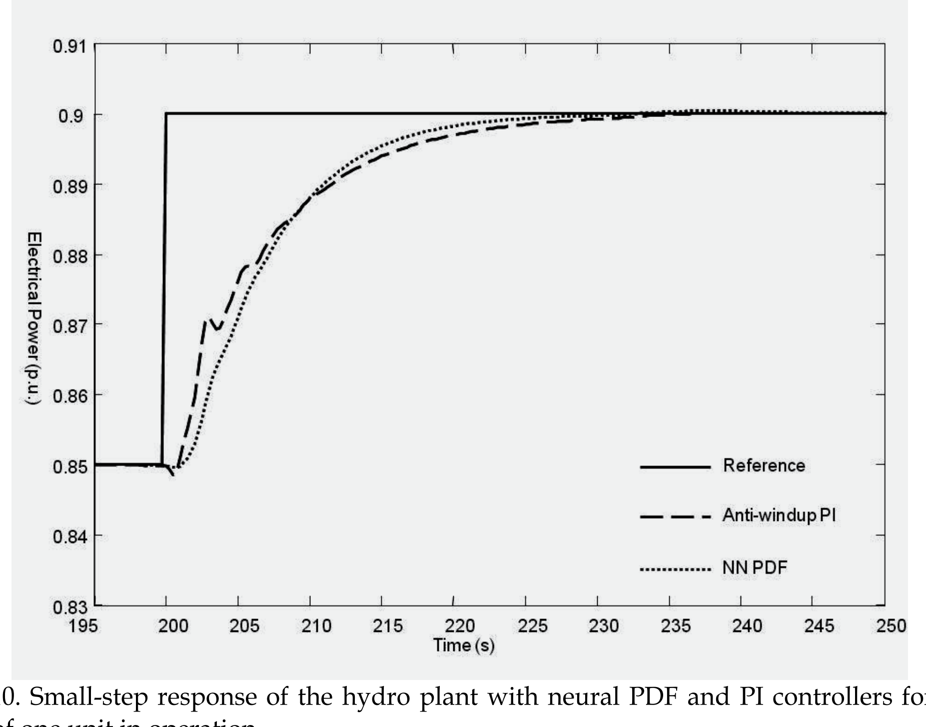


Fig. 10. Small-step response of the hydro plant with neural PDF and PI controllers for the case of one unit in operation.

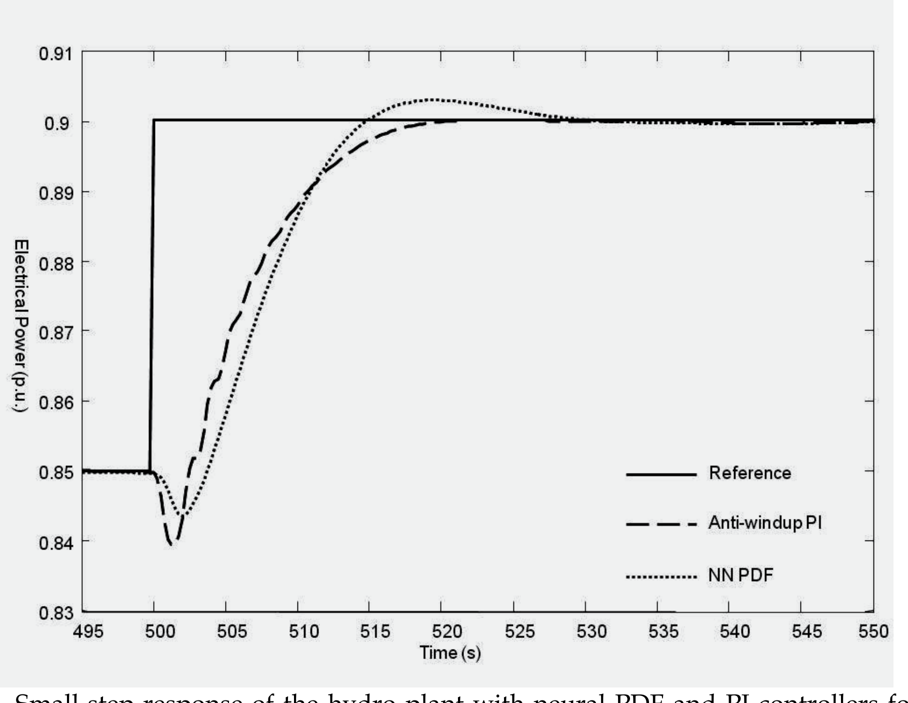


Fig. 11. Small-step response of the hydro plant with neural PDF and PI controllers for the case of six units in operation.

Figure 12 shows the large ramp responses (0.35 p.u.) of the hydroelectric plant with anti-windup PI and neural PDF controllers for one unit operational. Figure 13 shows large ramp responses (0.35 p.u.) of the power station when six units are generating. In both cases, the performance is better using the neural PDF controller, the response being 15% and 13% faster in, respectively, the one unit operational and six units operational cases. Again, the PDF controller reduces the undershoot.

To evaluate the cross coupling interaction a 0.8 p.u. step was applied simultaneously at t=300 to units 2-6 and the perturbation of unit 1 observed. Figure 14 shows that, although the neural PDF response has a higher overshoot, the PI response has a longer settling time and a higher undershoot.

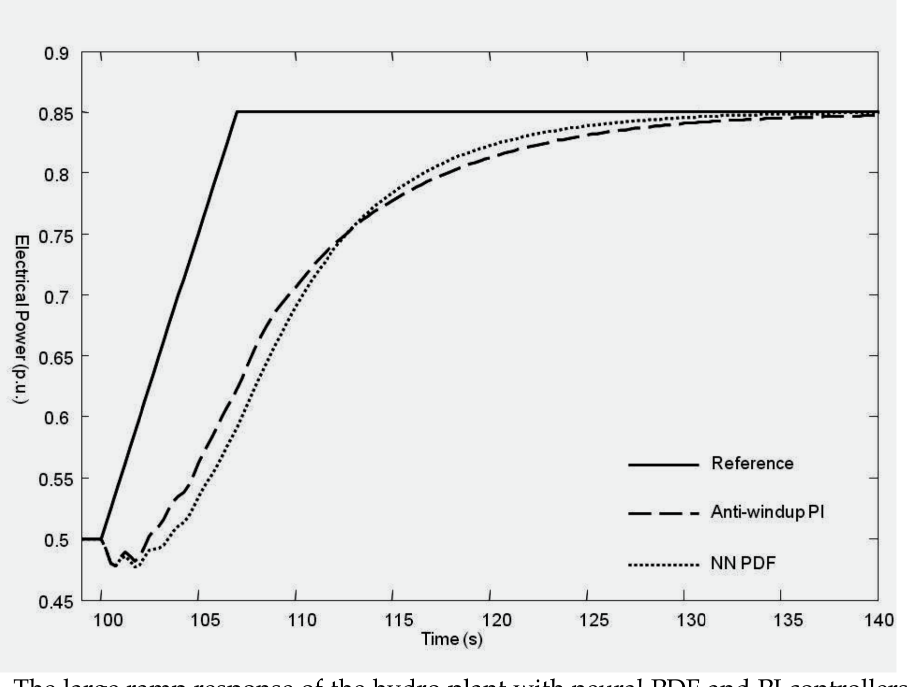


Fig. 12. The large ramp response of the hydro plant with neural PDF and PI controllers with one unit in operation.

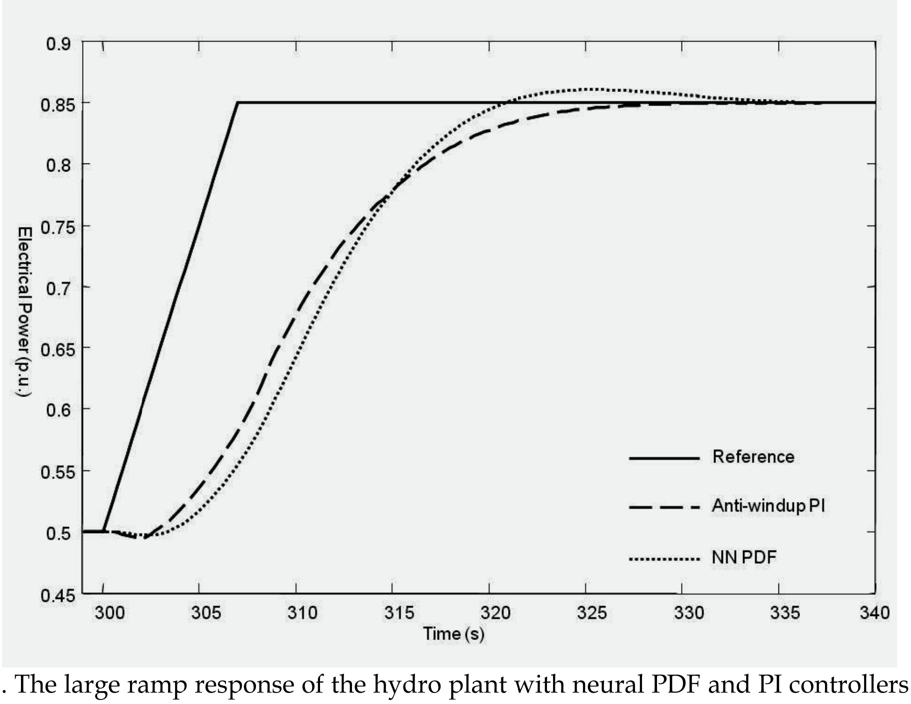


Fig. 13. The large ramp response of the hydro plant with neural PDF and PI controllers with six units in operation.

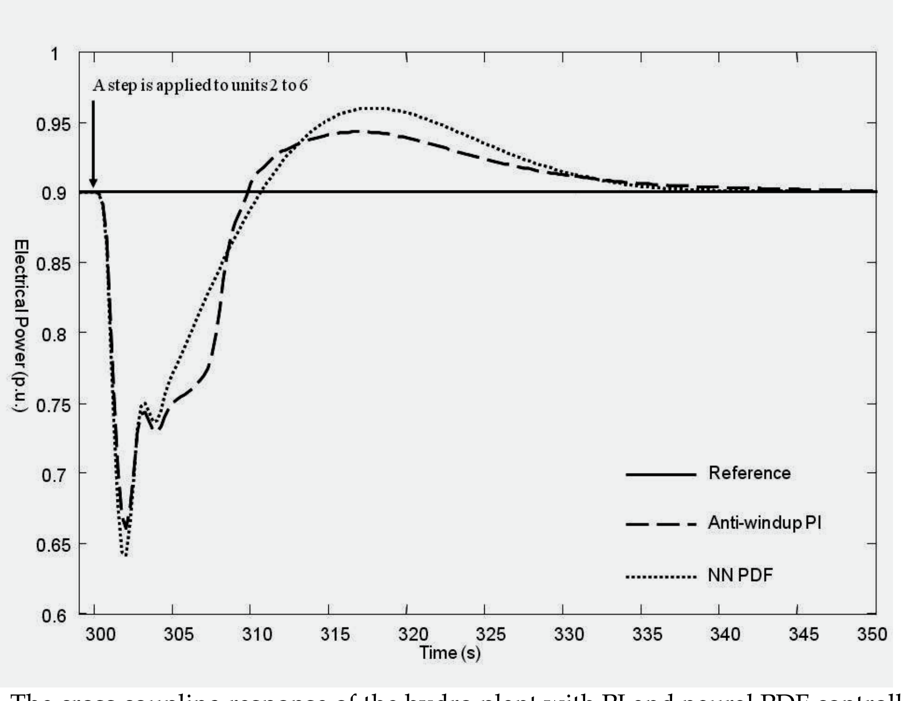


Fig. 14. The cross coupling response of the hydro plant with PI and neural PDF controllers.

7. Conclusions

In this chapter a software tool that models a hydropower plant and allows comparison of the performance of different controllers has been described. The modular nature of this platform has been important to gradually increasing the complexity of the simulations. The open architecture makes possible the rapid inclusion of other control methods and also the incremental improvement of the control approaches and models already included. The nonlinear model of pumped storage stations has been discussed. This model was applied to the Dinorwig power plant. The model includes representation of the guide vane, hydraulic and electrical subsystems and contains the principal features of the plant's dynamics.
The results have shown how the neural PDF can be applied to a hydroelectric pumped-storage station to improve its dynamic response. In particular, it has been shown that the step response of the system with neural PDF is improved. Multivariable effects have been taken into account to represent closely the real plant. The coupling between penstocks has been included in the nonlinear model. These are promising results for the use of neural PDF in this application and encourage us to address the issue of robustness of the response in future work.

8. Acknowledgments

The authors wish to thank First Hydro Company for their assistance.

9. References

Aguado-Behar, A., “Topics on identification and adaptive control” (in Spanish), Book edited by ICIMAF, La Habana, Cuba. 2000.

Bohn, C. and Atherton, D. P., "An analysis package comparing PID anti-windup strategies", IEEE Control Systems Magazine, vol. 15, p.p. 34-40. 1995.

Djukanovic, M. B., Calovic, M. S., Vesovic, B. V., and Sobajic, D. J., “Neuro-fuzzy controller of low head hydropower plants using adaptive-network based fuzzy inference system”, *IEEE Trans. on Energy Conversion* , 12, pp. 375-381. 1997.

Garcez, J. N., and Garcez, A. R., “A connectionist approach to hydroturbine speed control parameters tuning using artificial neural network”, Paper presented at *38th IEEE Midwest Symposium on Circuits and Systems*, pp. 641-644. 1995.

Goodwin, G. C., Graebe, S. F. and Salgado, M. E., "Control system design", Prentice Hall, USA. 2001.

Gracios, C., Vargas, E. & Diaz-Sanchez A., “Describing an IMS by a FNRTPN definition: a VHDL Approach”, *Elsevier Robotics and Computer-Integrated Manufacturing*, 21, pp. 241–247. 2005.

Kang, J. K., Lee, J. T., Kim, Y. M., Kwon, B. H., and Choi, K. S., “Speed controller design for induction motor drivers using a PDF control and load disturbance observer”, Paper presented at *IEEE IECON,* Kobe, Japan, pp. 799-803. 1991.

Kundur, P., Power System Stability and Control, New York, NY: Mc Graw Hill. 1994.

Mansoor, S. P., “Behaviour and Operation of Pumped Storage Hydro Plants”, Bangor, U.K.: PhD. Thesis University of Wales. 2000.

Mansoor, S. P., Jones, D. I., Bradley, D. A., and Aris, F. C., “Stability of a pumped storage hydropower station connected to a power system”, Paper presented at *IEEE Power Eng. Soc. Winter Meeting,* New York, USA, pp. 646-650. 1999.

Mansoor, S. P., Jones, D. I., Bradley, D. A., Aris, F. C., and Jones, G. R., “Reproducing oscillatory behaviour of a hydroelectric power station by computer simulation”, *Control Engineering Practice, 8*, pp. 1261-1272. 2000.

Miller T., Sutton S.R. and Werbos P., *Neural Networks for Control*, *Cambridge Massachusetts:* The MIT Press. 1991.

Minsky, M. L., and Papert, S. A., Perceptrons: Introduction to Computational Geometry. Cambridge, USA: MIT Press. 1988.

Munakata, T., Fundamentals of the New Artificial Intelligence: Neural, Evolutionary, Fuzzy and More. London, UK: Springer-Verlag. 2008.

Narendra, K. S., and Mukhopadhyay, S. “Adaptive control using neural networks and approximate models”, Paper presented at *American Control Conference,* Seattle, USA, pp. 355-359. 1996.

Peng, Y., Vrancic, D. and Hanus, R., "Anti-windup, bumpless, and conditioned transfer techniques for PID controllers", IEEE Control Systems Magazine, vol. 16, p.p. 48-57. 1996.

Rumelhart, D. E., McClelland, J. L., and Group, T. P., Parallel distributed processing: Explorations in the microstructure of cognition (Vol. 1). Cambridge, USA: MIT Press.1986.

Shu-Qing, W., Zhao-Hui, L., Zhi-Huai, X., and Zi-Peng, Z. “Application of GA-FNN hybrid control system for hydroelectric generating units”, Paper presented at *International Conference on Machine Learning and Cybernetics* 2, pp. 840-845. 2005.

Werbos, P. J., Beyond regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Cambridge, USA: PhD. Thesis Harvard University. 1974.

Widrow, B., and Hoff, M. E., "Adaptive switching circuits", Paper presented at *IRE WESCON Convention Record 4*, pp. 96-104. 1960.

Working group on prime mover energy supply, I, "Hydraulic turbine and turbine control model for system dynamic studies", *IEEE Trans.s on Power Systems , 7*, 167-179. 1992.

Wright, R. M., "Understanding modern generator control", IEEE Transactions on Energy Conversion, vol. 4, p.p. 453-458. 1989.

Yin-Song, W., Guo-Cai, S., & Ong-Xiang, "The PID-type fuzzy neural network control and it's application in the hydraulic turbine generators", Paper presented at *Power Engineering Society meeting. 1*, pp. 338-343. 2000.

11

Intelligent Network System for Process Control: Applications, Challenges, Approaches

Qurban A Memon,
UAE University,
United Arab Emirates

1. Introduction

The ever increasing size, sophistication and complexity of sorting and handling systems, as industries strive to increase efficiency and offer greater consumer choice, is a constant problem for system controllers and integrators who build such systems. In particular, the growing trend towards larger capacity systems, which increase throughput rates and provide greater flexibility to deal with product variation and changing supply demand, place increased pressure on system controllers to get things right on time and within budget, often in the face of loose specifications resulting in large and complex programs. In view of these demands it comes as no surprise that, when designing and integrating a control scheme, control engineers prefer to use past practice to minimize risk. Current practice includes distributed and modular systems like programmable logic controllers (PLC's) to break down complex control.

The need for modularity, real timeliness, integrated diagnostics, decentralized control, expanding physical setups, and functionality has resulted into various types of distributed control system (DCS). Presently, the DCS concept is not only applied to the process control and manufacturing automation but it has targeted many areas like communications engineering (Yang, 2006), physics and nuclear science (Cavineto et al., 2006; Kleines et al., 2004), power system engineering (Stewart, et al., 2003; Ioannides, 2004), a lighting system design (Alonso, et al., 2000) etc. The innovative ideas are being applied in field, for example in (Alonso, et al., 2000), the authors have made use of the power line as the communication medium for developing the DCS of the lighting system, based on a custom built large scale integrated Neuron chip incorporating both the control and communication features in a three microprocessors based system. Another domain of application is industrial plant automation to reduce operational cost, reduce product development time to market and improve flexibility of the plant in manufacturing a range of products. The decentralization in the control systems has brought communication and coordination amongst various sub-control systems to surface as an important aspect in building an optimal control system design. The research and development in the area of DCS network design is quite active because of developments in the communications and information technology field. As an example, the authors in (O'Hearn et al., 2002) discuss an approach for combination of

distributed control system with motor control center involving a number of remote terminal units within an industrial facility.
The chapter is organized as follows: In the next section, literature review of related work is presented to highlight existing approaches related to distributed process control and corresponding applications. The customary environment is discussed in detail along with performance specification. Additionally, a set of relevant standards are highlighted that set the direction for common approach using distributed control. In section 3, a set of approaches are described that maximize the performance of the central process controller. The section 4 presents recent advancements in communications and information technology that are proving to be better candidates to yield improved performance for process control. A set of approaches are described, which minimizes the communication delay, improve performance, project reconfigurability and interoperability amongst various process devices. The section 5 provides recent standardization efforts in this area. In section 6, conclusions are presented followed by references in section 7.

2. Process Control

The DCS consists of a network of small, compact and intelligent nodes placed throughout the environment and linked via a network to main process controller. Identical hardware simplifies installation and field replacement. The nodes interface with and control a variety of device types, while main controller oversees the operation. In DCS, multipoint and network based process control has replaced the traditional point-to-point architecture which had been used by the industry for decades. The new architecture has smaller volume of wiring and distributed processing resulting into quick and easy maintenance and low cost. These developments have initiated research in two main directions: (i) Network design, configuration and communication protocols for distributed process control (ii) DCS system hardware and software architectures. To abreast our selves of first direction of research, a research work in (Lian et al, 2001) is studied that carries out simulation and experimental evaluation of three commonly used control networks in the industry. The control networks evaluated were: Ethernet, ControlNet, and DeviceNet. The evaluation procedure considered how each control network is used as a communication backbone for a network controlled system connecting sensors, actuators, and controllers with a detailed discussion on how medium access control (MAC) protocol works for each network. The performance parameters addressed include access delay, transmission time, message delay, message collisions, message throughput, packet size, network utilization, and determinism boundaries. Timings analysis of each control network presented addresses blocking time, frame time and propagation time. Using some case studies, the authors concluded that DeviceNet outperforms the other two networks in case of shorter or prioritized messages, where as ControlNet is suitable in case of time critical and non-time-critical messages. Summarily, this research work provided insight in setting performance limits offered by respective networks.
The Profibus is the standard protocol for communication which is vendor independent so that the communication between devices from different manufacturers can be implemented without any special interface adjustment. Regarding the Profibus networks, a comprehensive study has been carried out in (Tovar & Francisco, 1999), where authors discuss how Profibus fieldbus networks can be used to support real time communication for

industrial systems. The major contribution noted in this research is that the real time behavior is guaranteed even in the absence of synchronous bandwidth allocation. The proposed approaches are based on an assumption based on which real time traffic is guaranteed. The author in (Hong, 2000) has performed experimental evaluation of Profibus-FMS. In this work the experimental model of a Profibus-based industrial application is developed. The experimental environment consists of devices connected in a Profibus network involving data exchange through the services provided by FMS. The model evaluates the delay characteristics of a message once it passes through FMS service. Based on the experimental results, the author suggests an optimal value of target token rotation and an appropriate segment size to reduce the delay due to variable messages and in domain management service of FMS respectively. This also concludes that Profibus-FMS service is greatly affected by conventional values of target rotation time and segment size.
Apart from consideration of network protocols or control networks with associated delay characteristics, a study has been carried out in (Lian et al, 2002) about the design considerations for a specific network like DCS. This research work highlights the impact of network architecture in Network Control System (NCS) class of DCS. Design considerations include network parameters, control parameters, and network controlled system performance. The main evaluation measures for the network quality of service (QoS) are: time delay statistics, network efficiency, network utilization, and the number of lost or unsent messages. The authors have used a simulation model and an experimental study to validate the performance characteristics shown by theoretical analysis. Based on results, the authors demonstrate that the performance characteristics are useful guidelines to choose the network and control parameters while designing an NCS. The authors suggest that the performance characteristics are dependent upon the sampling period. However, choosing a sampling period has not been addressed in the research.
The wireless Profibus integrating wireless and wired devices is also investigated for process control applications. A study has been carried out in (Willig, 2003), where the parameters of wireless Profibus protocol standard have been discussed. The author introduces two approaches to wireless Profibus and compares them with respect to real time performance in the presence of transmission errors. The work demonstrates that the performance of real time Profibus protocol in IEEE 802.11 type PHY environment or in presence of bursty errors is unsatisfactory. The author suggests a specifically tailored protocol for wireless stations in a situation where integration of wired and wireless stations is necessary in a common Profibus Local Area Network (LAN). The tailored protocol works on top of IEEE 802.11 PHY and offers the same link layer interface to upper layers as does the Profibus protocol. The author has shown that round robin protocol with three add-ons namely simple data relaying, simple poll relaying, and priority restriction outperforms Profibus protocol under bursty conditions, and a stable ring. However, it was also concluded that keeping logical ring stable in presence of transmission errors is difficult to achieve. In a similar effort using network based control, the authors (Almeida, et al., 2002) highlight the importance of operational flexibility as well as event and time triggered paradigms in fieldbus communication systems, and propose a protocol to support time-triggered communication in a flexible way. The authors also discuss an efficient combination of event and time-triggered traffic with temporal isolation, maintaining the desired properties of both types of the traffic.

Other than the domain discussed above, there exists a set of approaches to the same problem. In one of the approaches, the distributed real time control application development for task allocation in heterogeneous systems is also reported in (Prayati et al., 2004), where authors address real time requirements in presence of interoperability of networked devices. The approach uses functional block allocation (FBALL) algorithm (International Electrotechnical Commission, 2007) to guarantee real time requirements. However, this work does not consider network situation during re-configurability of the process itself or its field devices during real time. Motivated by advent of pervasing communication and computing, the authors in (Recht & D'Andrea, 2004) consider spatially interconnected distributed systems with arbitrary discrete symmetry groups to discuss distributed controller design applicable to a much larger class of structure topologies. In another approach, the application domains and tasks faced by multi-robot systems of increasing complexity have been investigated. The authors (Farinelli et al., 2004) describe various forms of cooperation and coordination realized in the domain of multi-robot systems.

Although the literature was surveyed at length and considerable approaches were found related to network based control but the ones addressed in previous section are the most notable among many others. The issues addressed in literature cover network considerations (i.e., protocol change or delay characteristics), interoperability of devices; customization and reconfiguration of the control system. The approaches that address the network environment analyzed timing characteristics or protocols in order to reduce communication delays at various stages of information processing.

The process under our consideration is customary i.e., it comprises devices and nodes that are practically available today and mostly used. The environment can be visualized as shown in Figure 1. In this Figure, the controller is the main device for which distributed control environment is to be investigated. The complete system involves interconnection of control devices, some distributed field and intelligent devices connected to Profibus, and few intelligent control devices operating under legacy environment. The control devices at lower level of the system may involve standard PLCs for motion control of a typical drive, for example. The intelligent devices are normally needed for a specific application where some of the decisions and local information gathering is done at that level and for onward transmission to the main controller for final network control decision. The devices are assumed to be reprogrammable and reconfigurable to suit changing needs of the process control. The interface shown in the Figure 1 is for legacy intelligence devices which are now part of the main distributed control system. The network considered is typical with Profibus compatible network devices with wired or wireless connectivity to a remote device or a process controller. The PC shown is for management purpose, especially for SCADA system. All the field devices considered are IEC 61804 compliant, and PLCs are IEC 61131 compliant. Based on our discussion above, the requirements for DCS development for this process control include consideration of reconfigurability and intelligence in the form of framework within a Profibus compatible network.

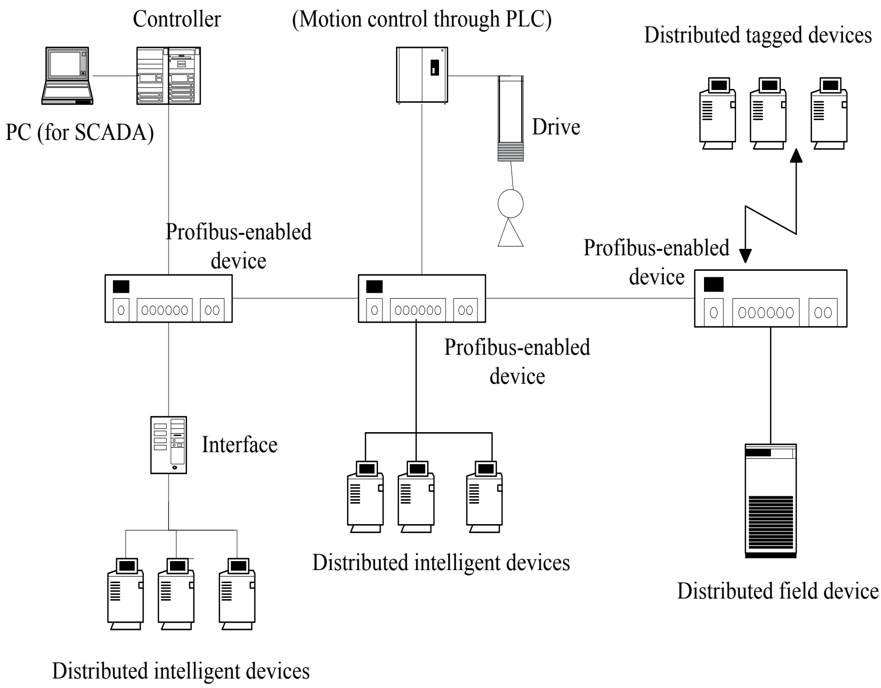


Fig. 1. Typical Process Control Network

For comparative purposes, we need to create a model that sets a baseline for performance. As the configuration of the network is distributed over a wide area, hence we consider a multiple input multiple output (MIMO) baseline that requires performance, which matches to that of the centralized MIMO (without network induced delays). A discrete time, causal and linear time-invariant systems with n states and m outputs can be described as (Goodwin et al., 2001):

$$X(k+1) = A\,X(k) + B\,[U(k) + D(k)];\; Y(k) = \mathrm{C\,X\,(k)} \tag{1}$$
$$with\; U(z) = K\,(z)\,E\,(z) = K\,(z)\;\; [R(z) - \;(C\,X\,(z) + \;N\,(z))]$$

where X, D, K, E, R, N are states of actual MIMO system in centralized control, disturbance, closed loop controller, error, reference, and sensor noise respectively. It should be noted that the MIMO system is assumed to be well designed meaning that its performance is exponentially stable (Goodwin et al., 2001), and that any MIMO control design technique can be used to develop an appropriate controller K. The parameters A, B, C are minimal realizations of the actual plant (Goodwin et al., 2001). Using equation (1), various states of the system can also be calculated:

$$X'(k+1) = A'X'(k) + \;B'\;U'(k);\; Y'(k)\; = C'X'(k) \tag{2}$$
$$with\; U'(z) = K\,(z)\,E'(z) = \;K\,(z)[R\,(z) - \;C'X'(z)]$$

where A′, B′, C′ represent minimal realizations of the model. For Figure 1 (as a distributed implementation of MIMO) to achieve same performance to that of the centralized MIMO, each error or reconfiguration request needs to be communicated over the network at any sample time, and that there are no communication delays. In order to categorize time delay $d(t)$ in the network is divided into three categories:

$$d(t) = d1(t)||d2(t)||d3(t) \quad (3)$$

Thus, the total time delay $d(t)$ is either of the time delays, depending on the prevailing situation in the environment. The d_1 corresponds to a normal situation (Green level), when only one field device communicates with the central process controller for a possible update or reconfiguration at any time. The d_2 corresponds to a situation (Yellow level), when more than one device (with number set as a threshold) communicate with central process controller for possible problem getting out of hand. The d_3 corresponds to a situation (Red level), when a set of devices (with number set as a threshold) tries to communicate with the central process controller. One obvious direction could be to minimize the delays described in equation (3) to optimize the performance to match a centralized MIMO system with no communication delays.

Since filed devices may include nodes or sensors that are non-stationary, protocol invariant and reconfigurable, the performance requirements stated by centralized MIMO would require approaches those are not based on legacy network control but should include recent advancements in sensor, information and communication technologies to match optimized performance. The combination of distributed control up to edges of the process containing sensors and tags, with recent information technology concepts like layered software and agents is an active area of research these days.

2.1 Relevant Standardization

The PLCs are the most widely used processing units and fieldbus is most widely used to interconnect the process controllers, sensors and actuators in a DCS system. The Profibus is the standard protocol for communication which is vendor independent so that the communication between devices from different manufacturers can be implemented without any special interface adjustment. To realize the plug and play type of operation, the DCS vendors ought to adopt certain standards both for PLCs and the communication. The International Electrotechnical Commission (IEC) and PLC manufacturing companies are actively involved in this development and establishment of standards (International Electrotechnical Commission, 2007). These standards define basic concepts for the design of modular, reusable, distributed components for the distributed control systems. Some of the most popular IEC standards are IEC 61131, IEC 61499 and IEC 61804. Because of the importance of these standards a brief introduction is presented in the next section.

IEC 61804: This standard addresses the need of common device description for representing the particular features of the field devices used in the DCS system. The standard is applicable to Function Blocks (FB) for process control and specifies the Electronic Device Description Language (EDDL) thus improving the device interoperability.

IEC 61131: This standard has different versions ranging from 61131-1 to 61131-8. The whole range covers general information about just the PLCs, the equipment requirement and tests, defines syntax and semantics of programming languages including the function block (FB) portion of standard for PLC languages, and sets the standards for the communication. To conclude IEC 61131 was first attempt to provide common terminology and reference model about hardware (HW) and software (SW) architectures, communication, and languages for a special class of control devices i.e., the PLCs.

IEC 61193-2: This standard IEC applies to the inspection of electronic components, packages, and modules for use in electronic and electric equipment, manufactured under suitable process control, which prevents the outflow of nonconforming products. The objective is to provide a method for stabilizing, monitoring, and improving processes.

IEC 61987-1: It defines a generic structure in which product features of industrial- process measurement and control equipment with analogue or digital output should be arranged, in order to facilitate the understanding of product descriptions when they are transferred from one party to another. It applies to the production of catalogues of process measuring equipment supplied by the manufacturer of the product and helps the user to formulate requirements.

IEC 61499: This standard defines a generic architecture and presents guidelines for the use of function blocks as the main building block for industrial process measurement and control system applications. It extends the FB language (defined in IEC 61131-3) to more adequately meet the requirements of the distributed control in a format that is independent of implementation. The architecture is presented in terms of implement-able reference models, textual syntax and graphical representations. The standard addresses the need for modular software that can be used for distributed industrial process control by defining the basic concepts for the design of modular, reusable, distributed components for industrial control systems. It also specifies the software tool requirements.

A variety of fieldbus technologies and digital fieldbus devices have been introduced within the process industries over the last fifteen years. There has been a gradual acceptance of the fact that a variety of communication technologies are needed to fully address the application requirements of a manufacturing facility. Process control systems that use fieldbus, function blocks, and device descriptive languages already exist in the market. It is clear that the majority of these systems will continue to use function blocks into the foreseeable future, as they are modular and reusable. The function blocks hide the changing details and thus provide a stable interface for the application developers.

3. Sensors and Tagging Technology for Automation

The RFID (Radio Frequency Identification) technology enables events that are in line with the vision of pervasive computing: using digital media to augment physical and social spaces rather than replacing them with disembodied virtual spaces. The diversification of consumer needs has made RFID and other sensor systems an important means within the manufacturing industry to achieve more efficient multi-product production. Within the RFID tag process, for example, control functions may be stored and updated as the product progresses through various stages of the operation. Specific uses of the data within the tag for process control include part routing, program operations, build orders, component needs and specific operations which may by unique to only a few products. While operating in a process control function, the RFID tag can also be used as a portable data base providing station to station data transfers, repair information, operation status, operation values, pass/fail data, host data transfers, part history and raw SPC data. One of the major advantages of using an RFID system in manufacturing is a major reduction in the size and complexity of the PC or PLC program reducing the cost associated with programming and also reduces the scan time of the controller.

The object's RFID tag may be integrated with a large object database to integrate object community into the space where social interaction occurs (Konomi et al., 2006). The Researchers at the University of Washington are curious to see what effects RFID technology could have on social networking (The RFID Ecosystem, University of Washington, 2008). To see what happens when the tags become ubiquitous, they installed two hundred antennae in and around a campus building and gave tags to twelve researchers. The result was such that their every move is recorded by computer. The system responds to nearby participants and uses algorithms to dynamically derive interconnected social clusters from a publication database - highlighting new opportunities and key design challenges. The Department of Computer Science and Engineering at the University of Washington has initiated a large scale RFID Ecosystem project to investigate user-centered RFID systems in connection with technology, business, and society and determine balance between privacy and utility (The RFID Ecosystem, University of Washington, 2008).

For vast field deployment, the authors (Bohn & Mattern, 2004) discuss distribution schemes where passive RFID tags are deployed in vast quantities and in random fashion. The authors suggest that certain deployment patterns help create novel RFID based services and applications. The density, structure of tag distributions, tag typing and clustering is also investigated by authors. For distributed business like process applications, the author (Palmer, 2007) discusses effective ways handling vast volume of RFID data for enterprise application management in real time. The steps applied together help enterprise manage volumes of raw RFID data in real time, provide reliability and accurate decisions to tackle the challenges, and exploit opportunities offered by RFID.

It is impossible to separate RFID technology from its issues and concerns surrounding its deployment, especially with respect to its potential for privacy infringement and identity theft. The RFID systems are designed to be asymmetric: tags are cheap and require low power while readers are expensive and power hungry. The various types and functionalities of these devices have been defined in an RFID class structure by the Auto-ID center, and later through Electronic Product Code (EPC) global (The EPC Global Standard, 2009), which has been subsequently refined and built-on. The number and use of standards within RFID and its associated industries is quite complex, involves a large number of bodies (ISO standard, 2009; ETSI standard, 2009; RFID Journal, 2009) and are in a continuous process of development. The various standards produced cover four key areas of RFID technology: its air interface (tag-reader communications) standards, data content and encoding, conformance and interoperability between different applications and RFID systems (RFID Journal, 2009). The various technological issues including its frequency range, standards, adoption and innovation has also been addressed at length in (Ward et al., 2006).

The RFID's have also been investigated in conjunction with software technology to optimize its usability. Integrating software agents into RFID architectures to accumulate information from tags and process them for specific use at different layers of the control plane has attracted many operational managers to investigate their deployment for real time use in industrial applications. The authors in (Naby & Giorgini, 2006) propose a possible integration between a Multi Agent framework and an RFID based application (back end). This is enabled by creating an agent for each person to develop its profile and correlating it with RFID tags movement inside a working environment. The accumulated information, after processing may be used to facilitate a concurrent mission. The RFID has also been explored for product identification in agent-based control applications to gain full control over all products and resources at shop floor level - aiming at installing a completely distributed agent-based production control system with full information control (Bratukhin & Treytl, 2006). The summary of literature survey for RFID deployment suggests investigating a multi-agent architecture as an application layer within existing RFID layered architecture that includes tag layer, RFID-reader layer, and back-end layer (Naby & Giorgini, 2006).

4. Agents for Intelligent Control

Recently there have been a number of advances in distributed artificial intelligence that provide the tools to move away from the traditional centralized, scan based programmable logic control (PLC) architecture towards a new architecture for the real-time distributed intelligent control. The Industrial control is typically implemented using large and often expensive hardware platforms that support monolithic computer control applications. The authors in (Bernnan et al., 2002) propose a general approach for dynamic and intelligent reconfiguration of real-time distributed control systems that utilizes the IEC 61499 function block model (International Electrotechnical Commission, 2007) to achieve shorter up-front commissioning time and significantly more responsive to changes, and is based on object oriented and agent based methods. Similarly, the authors in (Heck et al., 2003) discuss software technology for implementing reusable, distributed control components to make corresponding systems more flexible and dynamically re-configurable, so that they can be adapted in an easy way. Intelligent control and management of traffic and transportation systems in connected environments has been investigated in (Wang, 2005) that requires agent based technology to develop cheap, reliable and flexible control system. The authors in (Maturana et al., 2005) present distributive intelligent control architecture using agents by developing tools and methodologies for agent based solutions to a wide variety of industrial, financial and security problems, as classical control does not adapt dynamically well to the variability of the process. In another work, the authors in (Fregene et al., 2005) present a systems-and-control-oriented intelligent agent framework, as well as its composition into specific kinds of multi-agent systems to achieve the coordinated control of multiple multimode dynamical systems. The author in (Vyatkin, 2008) discusses distributed intelligent control using IEC 61499 (International Electrotechnical Commission, 2007) programming architecture for distributed automation systems in reconfigurable manufacturing systems. The testing environment consists of a network of identical software components for control of baggage handling system at the airport. The work in (Gingko Networks, 2008) provides simple decentralized ways to deal with a growing number of modern network requirements, by using distributed intelligent agents to deal with local situations in a more responsive way like sensing and observing events and changes occurring locally. In order to understand the potential use of agents in process industry, a survey has been presented in (Yang & Vyatkin, 2008) on design and validation of distributed control in process industry. The respective authors discuss distributed control benefits such as flexibility, reconfigurability and software reusability. The process control system is generally considered as a hybrid system as it usually contains both discrete and continuous dynamics. The authors highlight the importance of hybrid verification and simulation while handling validation of distributed systems.

In intelligent control design, scalability of a control system, active system reconfiguration, distributed intelligence and reduced communications are targeted benefits. In these situations, the issues like customization, decentralization and modularity at various levels of control are the main challenges in respective DCS design to facilitate distributed field operations in a process control. Intelligent network control together with active remote monitoring of a scalable distributed system has turned out to be an active area of research. The obvious choice seems to break the complex control process into two distinct processes (Memon, 2008): Local Process for simple task execution at field device level using social interaction of field process entities and generating operational parameters at the device level; Central Process for estimation of new characteristics of social entities only when entities need to be created or modified during operational stages. The independence at local process level minimizes communication between devices to avoid bottlenecks arising due to interoperability of devices, network protocol change or simple operational requirements at the device level; and provide a degree of re-configurability of field devices at the same time. Using local intelligence collected through interactions of local entities and combined with controller requirements at the central level helps solve combinatorial complexity present at any time during the operation. Thus, two areas are targeted: social interactions of entities with domain intelligence, and effective decision making set by the central process. The following is the discussion on both processes.

4.1 Local Process

Knowledge integration at central level requires balanced information from local entities. These entities tend to be distributed throughout the operational environment to support all operations. The job of these entities can be done effectively by distributed agents. The agents collaborate socially, learn and adjust their abilities within the constraints of the global process. Thus the agents are sophisticated software entities set by central process to execute trained intelligence at the local level. The functions within this process include: how agents collaborate and clusters are formed to accumulate intelligence and enable decision making; how access to another agent is facilitated; and the procedure for inter-cluster collaboration. To make sure that these agents form an intelligent system, the framework has to conform to four requirements (Maturana et al., 2005): that the system is decentralized; agents follow centricity; they use common language; and that the system is scalable.

4.1.1 Agent design & management

The agents deployed in existing infrastructures are designed to be manageable as other components with tools and procedures already in place at central process level. Each agent has software components adaptable to environmental changes. Each of these can be considered as a specialized function with some built-in capabilities, like sense→decide→act loop. The various categories of these specialized functions include:

- Producing knowledge in cooperation with other agents
- Evaluating the situation and decide to apply an appropriate action individually or through collaboration
- Acting onto the network element parameters, like fine tuning quality of service
- Forward useful information to the central process

Agents provide simple decentralized ways to deal with a growing number of modern process control network requirements. Agents can provide a set of management functions:

- Configuration functions for downloading of configuration parameters
- Control functions for monitoring the process environment and providing on-line activation and inspection of agents

- Data collection functions for enabling uploading of useful information collected by agents
- Deployment functions enable to add or remove agents dynamically

Agents can also request for additional capabilities once they discover that the task at hand cannot be fulfilled with existing ones. The programming of these agents is done at the central level where a set of heuristics is used for reasoning at the local level, and is stored as a function block diagram (like an internal script). The life cycle of the cluster is pre-set at the central process during agent design, and which can be re-negotiated upon an agent's request. The agents know about their equipment, and continuously monitor their state, as shown in Figure 2. The Figure 2 shows agents residing on tagged network elements, known as intelligent devices that can compute and communicate. Based on the intelligence, they can decide whether to participate in a mission or not, as shown in Figure 3. Thus the design of an agent is normally made open ended to add flexibility during operation.

4.1.2 Agent Collaboration

Each agent maintains its own view of the environment on the basis of the information obtained directly from its sensors and indirectly from the network through collaboration with its neighboring agents. As the local events are known and are properly documented in its view, the agent responds quickly and appropriately. Based on its view, agent may decide to automatically adapt its certain parameters of its own network element. The environmental view of the agent drives implicit cooperation amongst agents. This mode of cooperation is simple, robust and well situated for dynamically changing environments. The collaborating agents join (on their own will) and thus form a cluster in order to enable a decision making. Within their constraints, they accomplish their task. In addition to agents, there are other computing units that exist at the local level and which help to form a cluster. These are known as cluster directory (CD) and cluster facilitator (CF) respectively. The following steps describe the operational scene of agent collaboration in clusters:

- Agent N receives a request from central process for a task planning.
- It checks its internal scripts, if it can participate then it solves the local steps.
- For external steps, it contacts cluster directory to check for other agents if they also have external capability.
- In that case, CD provides contact details.
- Upon receipt of these details, agent N creates CF_N and passes on these details.
- CF_N understands coordination context.
- CF_N passes the request to specified agents, and thus cluster is formed.

For efficient collaboration, CD must remain updated for recording the information of its members, such as agent name, agent locator, service name, service type, and so on. Upon joining or leaving the cluster, an agent must register or cancel registration respectively through CF. Through a query, an agent can find out other members' services and locators. Through these steps, a trust is developed amongst agents.

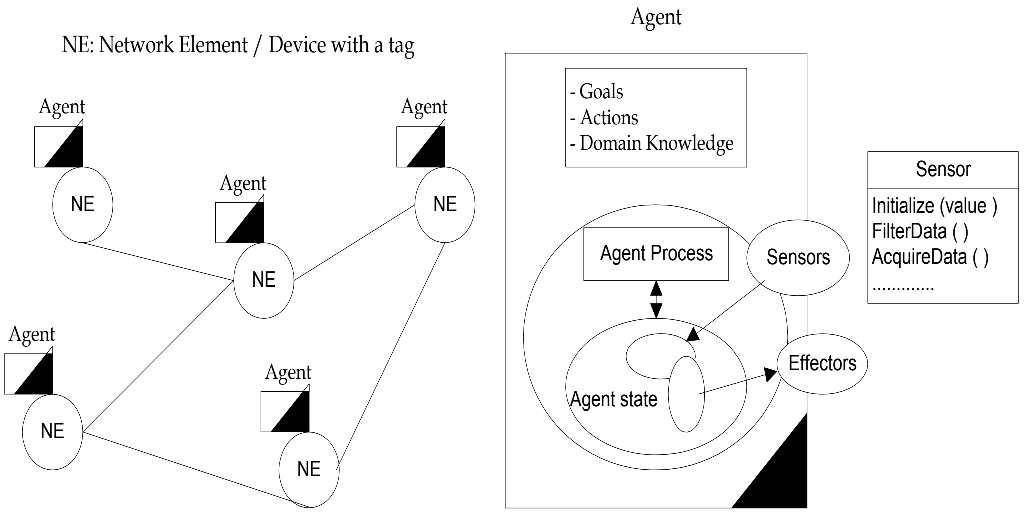
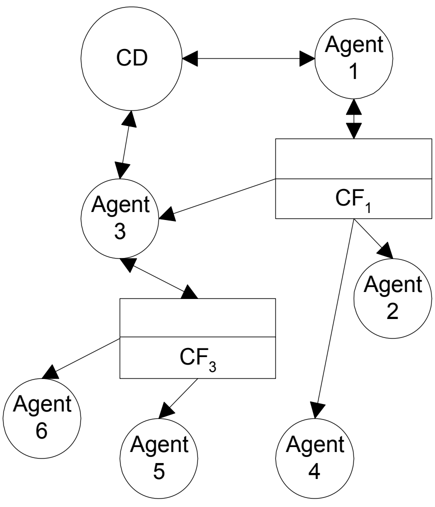


Fig. 2. Agents in a process network

Cluster 1: Agents (1,2,3,4); Cluster 2: Agents (3,5,6)



CD: Community Directory; CF: Community Facilitator

Fig. 3. Agent Collaboration

4.1.3 Agent Technology

Recently developed tools may be used to help design cluster facilitator (CF) and domain ontology, using for example DARPA Markup Language (DAML) (DAML website, 2000). The DAML extends XML (Extensible Markup Language) and RDF (Resource Description Framework) to include domain ontology. It provides rich set of constructs to create ontology and to markup information for attaining machine readability and understandability; it has capability of inference so that membership or service of cluster can be precisely defined. In case of XML and RDF, there seems to be no significant literature to manage ontology, and this is why, DAML is chosen to build cluster ontology. A number of agent standards have been in practice like Knowledge Query Manipulation Language (KQML), OMG's mobile agent system interoperability facility (MASIF) and the Foundation for Intelligent Physical Agents (FIPA). The Foundation for Intelligent Physical Agent (FIPA) Agent Management Specification (FIPA specification, 2002) is extended to develop the agent role called CF to manage cluster directory (CD) and cluster ontology. Using assistance from DAML-based ontology, the members of the cluster are able to form cluster and communicate with other agents. The interaction among domain ontology, CD and CF can be best understood using Figure 4. The Figure 4 shows how CF gets access to DAML files and facilitates the common goal of the cluster. There are tools available like Jena semantic web (HP Labs toolkit, 2003) that can be used to handle the cluster directory (CD) built using DAML, and to develop a Java class "Directory".

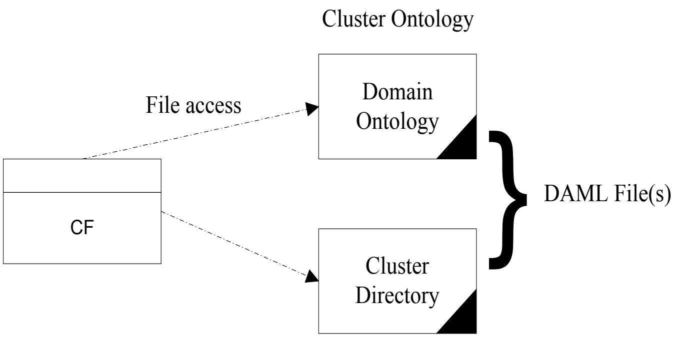


Fig. 4. Linking CF with DAML

From discussions above, the main functions of CD can be summarized, as:

- Add and Remove the information of an agent
- Get the list of agent names of all members
- Get the information of individual agent by name
- Get ontology used by members in the cluster
- Add external ontology if provided by an agent

Using local process mechanism and main functions of CD, the partial directory can be described as shown in Figure 5. It shows information of CF (lines 1-9) and members of cluster (lines 20-22), the cluster directory also records meta-data about cluster such as cluster name (line 12), cluster description (lines 13-15), ontology used in cluster (lines 16-18), and so on.

```
<cluster:CF rdf:ID="theCF">
<cluster:agentName>"CF"</cluster:agentName>
<cluster:agentDescription>
"DCS Cluster Facilitator"
</cluster:agentDescription>
<cluster:locator>
"http://dcs.ee.uaeu.ac.ae/DCS/agent/CF"
</cluster:locator>
</cluster:CF>

<cluster:Cluster rdf:ID="DCSCluster">
<cluster:clusterName>"DCS"</cluster:clusterName>
<cluster:clusterDescription>
"Distributed Control System"
</cluster:clusterDescription>
<cluster ontology>
"http://dcs.ee.uaeu.ac.ae/DCS/ontology/dcs.daml"
</cluster:ontology>

<cluster:hasCF rdf:Resource="#theCF"/>
<cluster:consistOf rdf:Resource="#agent1"/>
<cluster:consistOf rdf:Resource="#agent2"/>
</cluster:Cluster>
```

Fig. 5. DCS Cluster Directory

4.1.4 Example

An example can be illustrated to show how ontology may be updated (Fig. 6(b)) and that how interactions may develop in a local process. It should be noted here that basic cluster ontology (the knowledge of the local process) provided by CF remains the same but all members' domain knowledge (ontology) may not be the same. For example, user agent holds basic knowledge of the local process but does not understand the knowledge that a distributed field device holds. Through DAML-based ontology, members can communicate with each other to acquire requested service, as shown in Figure 6. It is clear from the Figure 6 that when distributed field device agent joins the cluster, it informs CF about corresponding ontology it provides (Figure 6(a)). Thus the CF maintains local process ontology plus the distributed field device ontology. When a user agent wants to perform a task, it asks CF about domain ontology and the agents that provide external capability. In response, CF informs the user agent if ontology is to be acquired (Figure 6(c)). Thus, the user agent can communicate with the distributed field device agent (Figure 6(d)).

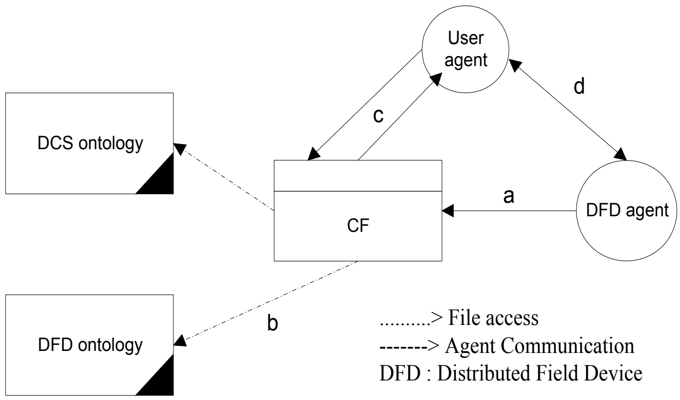


Fig. 6. Update in ontology provided by distributed field device agent

4.2 Central Process

This process handles core mechanism that glues organization's local processes to the central process. Some of the functions for example library of agents, their job description, definition of controller tasks, and domain ontology of each cluster can be defined offline before the implementation actually starts. The dynamic components are removing of agent deadlocks, security of agents, estimation of characteristics and relationships and decision making in cases of emergencies and when situation develops beyond the capabilities of agent clusters. It seems that all of these dynamic functions together may require computations, but the advantages gained are many: (i) reduced communications between central process controller and the device(s) (ii) provide simplicity to enable better interoperability (iii) intelligence gathering to build a degree of reconfigurability in a case estimated parameters exceed beyond a limit (iv) reduced human supervision. It can be also argued that complexity of this process is only a technology mismatch, and that if only small scale changes are to be decided at the central process like reconfiguration of device parameters, security of agents, then intelligence can further be distributed to the agents at the local level.

Based on presented work in section 3 and in 4.1, agents can be embedded in tagged devices within a layered architecture to support business operations and services in real time. In Figure 7, the model architecture of four tiers is drawn to implement objectives of the central process. At the bottom layer (Tier 1), active readers or Profibus/Profinet enabled devices collect data, often collected on a trigger similar to a motion sensor. These readers should be controlled by one and only one edge server to avoid problems related to network partitioning. In addition, this layer supports the notion that intelligence be introduced at the edges to reduce data traffic and improve reaction at the next layer. This layer also provides hardware abstraction for various Profibus/Profinet compatible hardware and network drivers for interoperability of devices. The edge sever (Tier 2) regularly poll the readers for any update from device agents, monitors tagged devices and distributed devices through readers, performs device management, and updates integration layer. This layer may also work with system through controls and open source frameworks that provide abstraction and design layer. The integration layer (Tier 3) provides design and engineering of various objects needed for central controller as well as for field processes and for simulation levels of reconfigurability. This layer is close to business application layer (Tier 4). The monitoring of agents behavior, its parameters and cluster characteristics are done at this layer to assess the degree of reconfigurability. This layer also takes care of parameters like handling device processes, applications, security of agents, resource allocation and scheduling of processes.

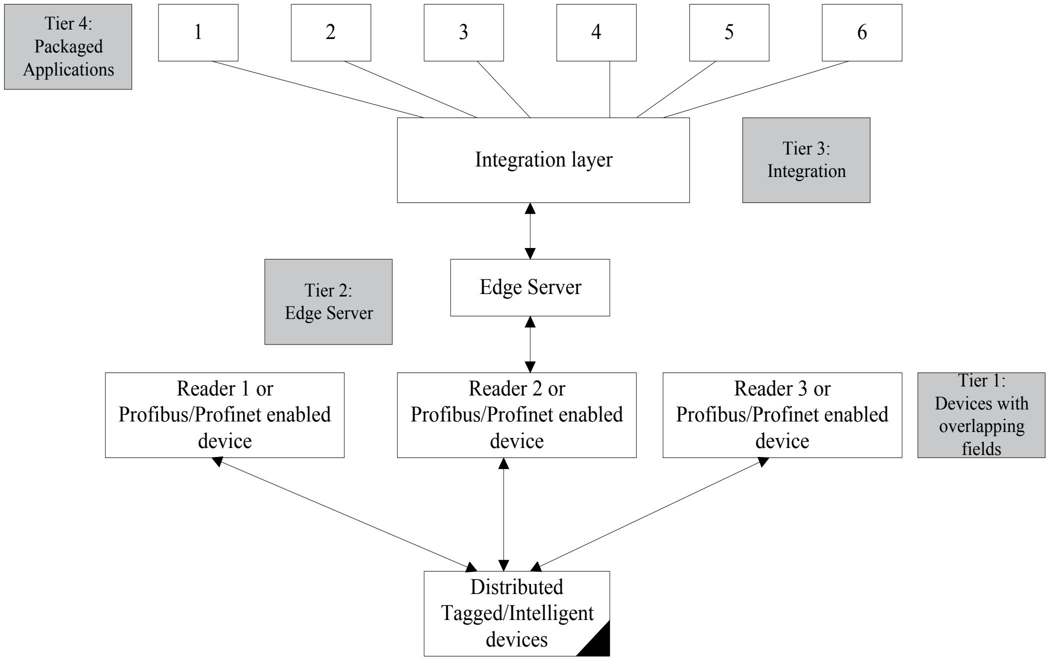


Fig. 7. 4-Tier Reference Architecture

The separation of edge server and integration layer improves scalability and reduces cost for operational management, as the edge is lighter and less expensive. The processing at the edge reduces data traffic to central point and improves reaction time. Similarly, the separation of integration from business applications helps in abstraction of process entities. The Tier 3 also enables it as self-healing and self-provisioning service architecture to increase availability and reduce support cost. Control messages flow into the system through business application portal to the integration layer, then on to the edge and eventually to the reader. Provisioning and configuration is done down this chain, while reader data is filtered and propagated up the chain.
The equation (3) may now be investigated again further, and evaluated using the model architecture, shown in Figure 7. The objective is to minimize the communication between field devices and the central process controller, and bring most of the local decision making intelligence at the local field level. Only when certain parameters need to be changed at local level device, then the values d_1, d_2, and d_3 need to be estimated. In order to evaluate these delay parameters, it is sufficient to estimate one communication between an RFID reader and the central process controller (i.e., d_1), as others just accumulate these delays over a number of communications. The communication between various nodes in Figure 7 may be abstracted using queuing models. In order to evaluate the performance modeling of this

architecture, queuing model can be used, since communication traffic may be considered as datagram that involve traversing multiple paths, which means M/M/1 queuing system can be used. Assume that the service rate between a reader and edge server is μ_{RE}, between an edge and integration layer is μ_{EI}, and that between integration and central process at packaged application is μ_{IC} respectively. Assume also that datagram arrival rate from an RFID reader is λ_{RE} and T_{IJ} is the average propagation delay between nodes *i* and *j*, where nodes *i* and *j* belong to one hop delay between any two nodes. Using these parameters, the RFID message delay from reader to central process controller may be written as accumulated delays across the entire path:

$$d_1 = d(R, \text{controller}) = \sum_{i,j}\left(\frac{\lambda ij}{uij(\mu ij - \lambda ij)} + \frac{1}{\mu ij} + Tij\right) \quad (4)$$

With Giga bits per second wireless transmission rates available today, T_{IJ} may be assumed negligible for one datagram traversing from one node to the other, along the entire path. The only terms left are the arrival rates and service rates along all node hops. Since 100 Mbps system (i.e., 0.014ms/message) is commonly available for RFID switches, network switches and servers (with exponential service waiting time), we may assume the corresponding values in equation (4) very easily. It may also be assumed that central process controller server message receiving rate at is 0.014ms/message, based on the same criterion. Thus, d_1 may be estimated once we insert λ_{IJ} in the equation (4). It turns out that for given typical value of λ from reader as (say) 0.2, the estimated delay from a reader to central process controller is less than 0.1msec, which is acceptable for a central process controller that is waiting to update a set of parameters for agents down the local process. For a set of RFID tags (say 50) generating communication signals to central process controller (i.e., Yellow level situation), the estimated delay is still few milliseconds. For a situation involving 1000 tags generating messages (i.e., Red level), the total estimated delay is still less than a second.

4.3 Performance Gains

As presented in section 4.2, the communication delay has largely been reduced at the cost of increased intelligence at the local level. In fact, if we look at equation (3) we see that d_1(t), d_2(t) and d_3(t) minimize to a level when problem of the node device exceeds the threshold level of the agent intelligence. Thus this approach sets practical performance limits. However, again this is just a technology mismatch. If agent design technology reaches its maturity i.e., if the collaborative intelligence within agents exceeds combinatorial complexity of device problems then there is no need of communication between devices and the controller. Thus the requirements of the central process reduce to that of the customized design of the agents only, and its performance matches to that of the centralized MIMO system. The mechanism set at the local process provides self-healing, reliability and scalability. If a reader or service goes down, additional units can take up the workload automatically. If bottlenecks develop, the RFID system software can dynamically provision new service agents to manage increased requirements. The scalability is assured by a design at the central process that grows horizontally and vertically – like a single-CPU, tag-and-ship pilot through N-way and multi-purpose device deployments, smoothing the growth path. At the central process, design and reconfigurability can help introduce features in agents to thwart external and intrusive agents, and thus help boost security of operational devices and processes during real time. This set of gains has not been addressed in either of the approaches described in (Konomi et al., 2006; Lian et al., 2002; Maturana et al., 2005; Prayati et al., 2004). The combination of agents and tagging technology uses programming and standardized components, which adds versatility to the process control. This type of process control is suited to a wide range of applications that need wide area sensing, and control points. The exploitation of agents is expected to rise over time as other enabling technologies grow in prominence.

5. Recent Standardization

There has been a large standardization effort conducted towards process control communications and systems, covering a range of industries. It is not possible to describe all of them here, but most recent, relevant to this work is presented below (International Electrotechnical Commission standard, 2006-2009; Hart Communication Foundation standard, 2009; Aim Global RFID Guideline, 2009):

IEC 60770-3 (2006): The standard specifies the methods for reviewing the functionality and the degree of intelligence in intelligent transmitters, for testing the operational behavior and dynamic performance of an intelligent transmitter as well as methodologies for determining the reliability and diagnostic features used to detect malfunctions; and determining the communication capabilities of the intelligent transmitters in a communication network.

IEC 69870-5-104 (2006): The standard defines telecontrol companion standard that enables interoperability among compatible telecontrol equipments. It applies to telecontrol equipment and systems with coded bit serial data transmission for monitoring and controlling geographically widespread processes.

IEC 61784-1-3 (2007): It defines a set of protocol specific communication profiles based primarily on the IEC 61158 series, to be used in the design of devices involved in communications in factory manufacturing and process control. It contains a minimal set of required services at the application layer and specification of options in intermediate layers defined through references.

IEC 62264-3 (2007): It defines activity models of manufacturing operations management that enable enterprise system to control system integration. The activities defined are consistent with the object models definitions given in IEC 62264-1. The modeled activities operate between business planning and logistics functions, defined as the Level 4 functions and the process control functions, defined as the Level 2 functions of IEC 62264-1. The scope of this standard is limited to: - a model of the activities associated with manufacturing operations management, Level 3 functions; - an identification of some of the data exchanged between Level 3 activities.

Hart 7.0 (2007): The Hart Communication Foundation (HCF) has released the HART 7 specification, enabling more capabilities for communication with intelligent field devices, and targeting wireless communication in industrial plant environment. The specification allows building on established and field-proven international standards including IEC 61158, IEC 61804-3, IEEE 802.15.4 radio and frequency hopping, spread spectrum and mesh networking technologies.

IEC 61298-1-4 (2008): The specification defines general methods and procedures for conducting tests, and reporting on the functional and performance characteristics of process measurement and control devices. The methods and procedures specified in this standard are applicable to any type of process measurement and control device. The tests are applicable to any such devices characterized by their own specific input and output variables, and by the specific relationship (transfer function) between the inputs and outputs, and include analogue and digital devices.

IEC 62424 (2008)E: It specifies how process control engineering requests are represented in a P&ID for automatic transferring data between P&ID and PCE tool and to avoid misinterpretation of graphical P&ID symbols for PCE. It also defines the exchange of process control engineering data between a process control engineering tool and a P&ID tool by means of a data transfer language (called CAEX). These provisions apply to the export/import applications of such tools.

IEC/PAS 62443-3 (2008): It establishes a framework for securing information and communication technology aspects of industrial process measurement and control systems including its networks and devices on those networks, during the operational phase of the plant's life cycle. It provides guidance on a plant's operational security requirements and is primarily intended for automation system owners/operators (responsible for ICS operation).

IEC 61850 (2009): This is a standard for the design of electrical substation automation. Multiple protocols exist for substation automation, which include many proprietary protocols with custom communication links. The objectives set for the standard are: a single protocol for complete substation, definition of basic services required to transfer data, promotion of high interoperability between systems from different vendors, a common method/format for storing complete data, and define complete testing required for the equipments which confirms to the standard.

IEC/PAS 62601 (2009): It specifies WIA-PA system architecture and communication protocol for process automation based on IEEE 802.15.4. WIA-PA network is used for industrial monitoring, measurement and control applications.

AIM Global RFID Guideline 396 (2008): This guideline describes RFID chips and transponders, verification and qualification of design and manufacture of chips. This guideline targets item level tagging where the RFID tag may be present in various formats including a label, incorporated into a patch, which then becomes permanently affixed to the inner or outer surface of a tire or incorporated during manufacture into the structure of the tire as an integral part of the tire.

6. Conclusions

The main idea behind two processes is decentralization The communication delay is reduced at the cost of increased intelligence at the local level. In fact, by looking at equation (1) it is clear that d_1(t), d_2(t) or d_3(t) minimize to a level when problem of the node device exceeds the threshold level of the agent intelligence. If collaborative intelligence exceeds combinatorial complexity then there is no need of communication between devices and the controller and requirements of the central process reduce to that of the design of agents only. Thus, the performance matches to that of the centralized MIMO system. The four-tier modular architecture at central level helps in implementation of distributed intelligence at field level and in designing of agents. The functionality more appropriate to the layer has been fit into respective tiers at central level. Additionally, design and reconfigurability can help introduce features in agents to thwart intrusive agents, during real time. It was also shown that the estimated delay due to communication from a tagged device to a central process controller is less than a second when one thousand tagged devices pass on their communication signal to central process controller at the same time. This set of gains has not been claimed in either of the approaches for distributed control system widely discussed in the literature.

7. References

Almeida, L., Pedreiras, P., & Fonseca, J. (2002). The FFT-CAN Protocol: Why and How, *IEEE Transactions on Industrial Electronics*, Vol. 49, No. 6, pp. 1189-1201, December, 2002

Alonso, J., Ribas, J., Coz, J., Calleja, A., & Corominas, E. (2000). Development of a Distributive Control Scheme for Fluorescent Lighting based on LonWorks Technology, *IEEE Transactions on Industrial Electronic*, Vol. 47, No. 6, pp. 1253-1262, December, 2000

Association for Automatic Identification and Mobility, AIM Global radio frequency identification (RFID) Guideline REG 396, www.aimglobal.org/, [last accessed 04/25/2009]

Bernnan, Fletcher, M., & Norrie, D. (2002). An Agent-Based Approach to Reconfiguration of Real-time Distributed Control Systems, *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 4, pp. 444-451, August, 2002

Bohn, J., & Mattern, F. (2004). Super-Distributed RFID Infrastructures, *Lecture Notes in Computer Science (LNCS)* No. 3295, pp. 1-12, Eindhoven, Netherlands, November 8-10, Springer-Verlag, 2004

Bratukhin, A., & Treytl, A. (2006). Applicability of RFID and Agent-Based Control for Product Identification in Distributed Production, *Proceedings of IEEE Conference on Emerging Technologies and Factory Automation*, Vol. 20, Issue 22, pp. 1198-1205, Prague, 2006

Cavinato, M, Manduchi, G., Luchetta, A., & Taliercio, C. (2006). General-Purpose Framework for Real Time Control in Nuclear Fusion Experiments, *IEEE Transactions on Nuclear Science*, Vol. 53, No. 3, pp. 1002-1008, June, 2006

DAML website, DARPA Agent Markup Language (2000) [Online], *http://www.daml.org/*; [last accessed 04/22/2006]

ETSI RFID standards by ETSI, http://www.etsi.org/WebSite/Standards/Standard.aspx, Available online [last accessed on 04/25/2009]

Farinelli, A., Iocchi, L., & Nardie, D. (2004). Multirobot Systems: A Classification Focused on Coordination, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 34, No. 5, pp. 2015-2028, October, 2004

FIPA website, Foundation for Intelligent Physical Agents (FIPA) Agent Management Specification (2002) [Online], *http://www.fipa.org/specs/fipa00023/*; [last accessed 09/21/2006]

Fregene, K., Kennedy, D., & Wang, D. (2005). Toward a Systems– and Control-Oriented Agent Framework, *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 35, No. 5, pp. 999-1012, October, 2005

Function Blocks (FB) for industrial-process measurement and control systems, IEC 61804, IEC 61131, and IEC 61499. http://www.iec.ch/searchpub/cur_fut.htm, [last accessed 02/26/2009]

Gingko Networks (2008). White Paper: Gingko Distributed Network Piloting System, *Gingo Networks*, September, 2008

Goodwin, G., Graebe, S., Salgado, M., (2001), Control System Design, New Jersey, Prentice Hall, 2001

Hart Communication Foundation website, Hart 7.0 Wireless Communication Protocol, http://www.hartcomm2.org/hart_protocol/wireless_hart/hart7_overview.html, [last accessed 04/27/2009]

Heck, B., Wills, L., & Vachtsevanos, G. (2003). Software Technology for Implementing Reusable, Distributed Control Systems, *IEEE Control Systems Magazine*, pp. 21-35, February, 2003

Hong, S. (2000). Experimental Performance Evaluation of Profibus-FMS, *IEEE Robotics and Automation Magazine*, pp. 64-72, December, 2000

HP Labs website, HP Labs (2003), [Online], Jena Semantic Web Toolkit Available: *http://www.hpl.hp.com/semweb/jena.htm/*; [last accessed September 2006]

International Electrotechnical Commission (IEC, 2007) Webstore, http://webstore.iec.ch/webstore/webstore.nsf/artnum/ [last accessed 04/28/2009]

Ioannides, M. (2004). Design and Implementation of PLC based Monitoring Control System for Induction Motor, *IEEE Transactions on Energy Conversion*, Vol. 19, No. 3, pp. 469-476, September, 2004

ISO RFID standards, http://www.iso.org/rfid, Available online [last accessed on 04/25/2009]

Kleines, H., Sarkadi, J., Suxdorf, F., & Zwoll, K. (2004). Measurement of Real Time Aspects of Simatic PLC Operation in the Context of Physics Experiments, *IEEE Transactions on Nuclear Science*, Vol. 51, No. 3, pp. 489-494, June, 2004

Konomi, S., Inoue, S., Kobayashai, T., Tsuchida, M., & Kitsuregawa, M. (2006). Supporting Colocated Interactions Using RFID and Social Network Displays, *IEEE Pervasive Computing Magazine*, Vol. 5, Issue 3, pp. 48-56, July-September, 2006

Lian, F., Moyne, J., & Tilbury, D. (2001). Performance Evaluation of Control Networks, *IEEE Control Systems Magazine*, pp. 66-83, February, 2001

Lian, F., Moyne, J., & Tilbury, D. (2002). Network Design Consideration for Distributed Control Systems, *IEEE Transactions on Control Systems Technology*, Vol. 10, No. 2, pp. 297-307, March 2002

Maturana, F., Staron, R., & Hall, K. (2005). Methodologies and Tools for Intelligent Agents in Distributed Control, *IEEE Intelligent Systems*, pp. 42-49, February, 2005

Memon, Q. (2008). A Framework for Distributed and Intelligent Process Control, *Proceedings of 5th International Conference on Informatics in Control, Automation and Robotics*, pp. 240-243, May 12-15, Madeira, Portugal, 2008

Naby, A. & Giorgini, S. (2006). Locating Agents in RFID Architectures, Technical Report No. DIT-06-095, University of Trento, Italy, 2006

O'Hearn, T., Cerff, J., & Miller, S. (2002). Integrating Process and Motor Control, *IEEE Industry Applications Magazine*, pp. 61-65, August, 2002

Palmer, M. (2007). Seven Principles of Effective RFID Data Management, *A Technical Primer*, *Real Time Division, Progress Software, Inc.*, http://www.progress.com/ [last accessed 03/21/2009]

Prayati, A., Koulamas, C., Koubias, S., & Papadopoulos, G. (2004). A Methodology for the Development of Distributed Real-time Control Applications with Focus on Task

Allocation in Heterogeneous Systems, *IEEE Transactions on Industrial Electronics*, Vol. 51, No. 6, pp. 1194-1206, December, 2004

Recht, B. & D'Andrea, R. (2004). Distributed Control of Systems over Discrete Groups, *IEEE Transactions on Automatic Control*, Vol. 49, No. 9, pp. 1446-1452, September, 2004

RFID Journal, A Summary of RFID Standards, http://www.rfidjournal.com/article/articleview/609/1/1/ [last accessed on 04/25/2009]

Stewart, G., Gorinevsky, D., & Dumont, G. (2003). Feedback Controller Design for a Spatially Distributed Systems: The Paper Machine Problem, *IEEE Transactions on Control Systems Technology*, Vol. 11, No. 5, pp. 612-628, September, 2003

The EPC global, http://www.epcglobalinc.org, Available online [last accessed on 04/25/2009]

The RFID Ecosystem, http://rfid.cs.washington.edu/, [last accessed on 04/24/2009]

Tovar, E., & Francisco, F. (1999). Real-time field bus communications using Profibus Networks, *IEEE Transactions on Industrial Electronics*, Vol. 46, No. 6, pp. 1241-1251, December, 1999

Vyatkin, V. (2008). Distributed IEC 61499 Intelligent Control of Reconfigurable Manufacturing Systems, *Technical Report*, University of Auckland, 2008

Wang, F. (2005). Agent-Based Control for Networked Traffic Management Systems, *IEEE Intelligent Systems*, pp. 92-96. October, 2005

Ward, M., Kranenburg, R., & Backhouse, G. (2006). RFID: Frequency, Standards, Adoption and Innovation, *Technical Report in JISC Technology and Standards Watch,* May, 2006

Willig, A. (2003). Polling based MAC protocols for improving real-time performance in a wireless Profibus, *IEEE Transactions on Industrial Electronics*, Vol. 50, No. 4, pp. 806-817, August, 2003

Yang, C., Vyatkin, V. (2008). Design and validation of distributed control with decentralized intelligence in process industries: A survey, *IEEE International Conference on Industrial Informatics*, pp. 1395 – 1400, Daejeon, July, 2008

Yang, Y. (2006). Attenuation Splice Control in the Manufacture of Fiber Optical Communication System, *IEEE Transactions on Control Technology*, Vol. 14, No. 1, pp. 170-175, January, 2006

12

Neural Generalized Predictive Control for Industrial Processes

Sadhana Chidrawar[1], Balasaheb Patre[2] and LaxmanWaghmare[3]
[1]*Assistant Professor, MGM's College of Engineering, Nanded (MS) 431 602,*
[2,3] *Professor, SGGS Institute of Engineering and Technology, Nanded (MS) 431 606 India*

1. Introduction

In the manufacturing industry, the requirement for high-speed, fast-response and high-precision performances is critical . Model predictive control (MPC) which, was developed in the late 1970's, refers to a class of computer control algorithms that utilizes an explicit process model to predict the future response of the plant (Qin & Badgwell, 2004). In the last two decades, MPC has been widely accepted for set point tracking and overcoming model mismatch in the refining, petrochemical, chemical, pulp and paper making and food processing industries (Rossiter, 2006). The model predictive control is also introduced to the positioning control of ultra-precision stage driven by a linear actuator (Hashimoto,Goko,et al.,2008). Some of the most popular MPC algorithms that found wide acceptance in industry are Dynamic Matrix Control (DMC), Model Algorithmic Control (MAC), Predictive Functional Control (PFC), Extended Prediction Self Adaptive Control (EPSAC), Extended Horizon Adaptive Control (EHAC) and Generalized Predictive Control (GPC) (Sorensen, Norgaard ,et al., 1999). In most of the controllers, the disturbances arising from manipulated variable are taken care off only after they have already influenced the process output. Thus, there is a necessity to develop the controller to predict and optimize process performance. In MPC the control algorithm that uses an optimizer to solve for the control trajectory over a future time horizon based on a dynamic model of the processes, has become a standard control technique in the process industries over the past few decades. In most applications of model predictive techniques, a linear model is used to predict the process behavior over the horizon of interest. But as most real processes show a nonlinear behavior, some work has to be done to extend predictive control techniques to incorporate nonlinearities of the plant. The most expensive part of the realization of a nonlinear predictive control scheme is the derivation of the mathematical model. In many cases it is even impossible to obtain a suitable physically founded process model due to the complexity of the underlying process or lack of knowledge of critical parameters of the model. The promising way to overcome these problems is to use neural network as a nonlinear models that can approximate the dynamic behavior of the process efficiently

Generalized Predictive Control (GPC) is an independently developed branch of class of digital control methods known as Model Predictive Control (MPC). (Clarke, Mohtadi, et al., 1987) and has become one of the most popular MPC methods both in industry and academia. It has been successfully implemented in many industrial applications, showing good performance and a certain degree of robustness. It can handle many different control problems for a wide range of plants with a reasonable number of design variables, which have to be specified by the user depending upon a prior knowledge of the plant and control objectives. GPC is known to control non-minimum phase plants, open loop unstable plants and plants with variable or unknown dead time. GPC is robust with respect to modeling errors and sensor noise. The ability of GPC for controlling nonlinear plants and to make accurate prediction can be enhanced if neural network is used to learn the dynamics of the plant. In this Chapter, we have discussed the neural network to form a control strategy known as Neural Generalized Predictive Control (NGPC) (Rao, Murthy, et al., 2006). The NGPC algorithm operates in two modes, i.e. prediction and control. It generates a sequence of future control signals within each sampling interval to optimize control effort of the controlled systems. In NGPC the control vector calculations are made at each sampling instants and are dependent on control and prediction horizon. A computational comparison between GPC and NGPC schemes is given in (Rao, Murthy, et al., 2007).The effect of smaller output horizon in neural generalized predictive control is dealt in (Pitche, Sayyer-Rodsari,et al.,2000). The nonlinear model predictive control using neural network is also developed in (Chen,Yuan,et al.,2002). Two model predictive control (MPC) approaches, an on-line and an off-line MPC approach, for constrained uncertain continuous-time systems with piecewise constant control input are presented (Raff & Sinz, 2008)

Numerous journal articles and meeting papers have appeared on the use of neural network models as the basis for MPC with finite prediction horizons. Most of the publications concentrate on the issues related to constructing neural network models. Very little attention is given to issues of stability or closed-loop performance, although these are still open and unresolved issues. A predictive control strategy based on improved back propagation neural network in order to compensate real time control in nonlinear system with time delays is proposed in (Sun,Chang,et al.,2002).For nonlinear processes, the predictive control would be unsatisfactory. Like neural networks, fuzzy logic also attracted considerable attentions to control nonlinear processes. There are many advantages to control nonlinear system since they has an approximation ability using nonlinear mappings. Generally, they do not use the parametric models such as the form of transfer functions or state space equations. Therefore, the result of modeling or controlling nonlinear systems is not the analytic consequence and we only know that the performance of those is satisfactory. Especially, if the controller requires the parametric form of the nonlinear system, there doesn’t exist any ways linking the controller and fuzzy modeling method. The fuzzy model based prediction is derived with output operating point and optimized control is calculated through the fuzzy prediction model using the optimization techniques in (Kim, Ansung, et al.,1998).

In this Chapter, a novel algorithm called Generalized Predictive Control (GPC) is shown to be particularly effective for the control of industrial processes. The capability of the algorithm is tested on variety of systems. An efficient implementation of GPC using a multi-layer feed-forward neural network as the plant’s nonlinear model is presented to extend the capability of GPC i.e. NGPC for controlling linear as well as nonlinear process very efficiently. A neural model of the plant is used in the conventional GPC stating it as a neural generalized predictive control (NGPC). As a relatively well-known example, we consider Duffing’s nonlinear equation for testing capability of both GPC and NGPC algorithms. The output of trained neural network is used as the predicted output of the plant. This predicted output is used in the cost function minimization algorithm. GPC criterion is minimized using two different schemes: a Quasi Newton algorithm and Levenberg Marquardt algorithm. GPC and NGPC are applied to the linear and nonlinear systems to test its capability. The performance comparison of these configurations has been given in terms of Integral square error (ISE) and Integral absolute error (IAE). For each system only few more steps in set point were required for GPC than NGPC to settle down the output, but more importantly there is no sign of instability. Performance of NGPC is also tested on a highly nonlinear process of continues stirred tank reactor (CSTR) and linear process dc motor.

The ideas appearing in greater or lesser degree in all the predictive control family are basically:

- Explicit use of a model to predict the process output at future time instants (horizon).
- Calculation of a control sequence minimizing an objective function.
- Receding strategy, so that at each instant the horizon is displaced towards the future, which involves the application of the first control signal of the sequence calculated at each step.

2. MPC Strategy

The methodology of all the controllers belonging to the MPC family is characterized by the following strategy, as represented in Fig.1:

1. The future outputs for a determined horizon N, called the prediction horizon, are predicted at each instant k using the process model. These predicted outputs $y(t+j/t)$ for $j = 1 \ldots N$ depend on the known values up to instant t (past inputs and outputs) and on the future control signals $u(t+j/t)$, $j = 0 \ldots N\text{-}1$, which are those to be sent to the system and to be calculated.
2. The set of future control signals is calculated by optimizing a determined criterion in order to keep the process as close as possible to the reference trajectory $w(t+j)$ (which can be the set point itself or a close). This criterion usually takes the form of a quadratic function of the errors between the predicted output signal and the predicted reference trajectory. The control effort is included in the objective function in most cases. An explicit solution can be obtained if the criterion is quadratic, the model is linear and there are no constraints; otherwise an iterative optimization method has to be used. Some assumptions about the structure of the future control law are made in some cases, such as that it will be constant from a given instant.
3. The control signal $u(t/t)$ is sent to the process whilst the next control signal calculated are rejected, because at the next sampling instant $y(t+1)$ is already known and step1 is repeated with this new value and all the sequences are brought up to date. Thus the $u(t+1|t)$ is calculated (which in principle will be different to the $u(t+1|t)$ because of the new information available) using receding horizon control.

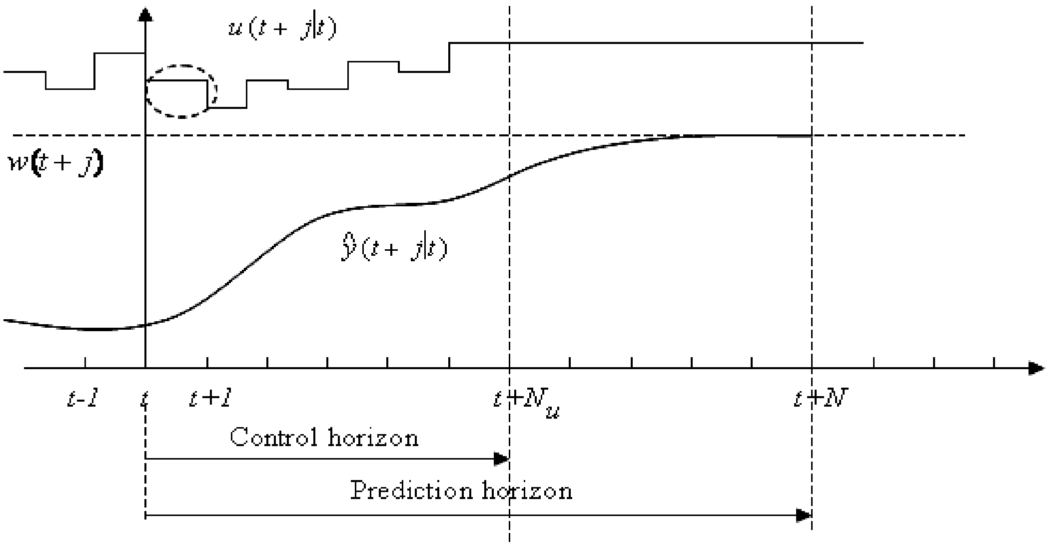


Fig. 1. MPC Strategy

3. Generalized Predictive Controller (GPC)

3.1 Introduction

The basic idea of GPC is to calculate a sequence of future control signals in such a way that it minimizes a multistage cost function defined over a prediction horizon. The index to be optimized is the expectation of a quadratic function measuring the distance between the predicted system output and some reference sequence over the horizon plus a quadratic function measuring the control effort. Generalized Predictive Control has many ideas in common with the other predictive controllers since it is based upon the same concepts but it also has some differences. As will be seen later, it provides an analytical solution (in the absence of constraints), it can deal with unstable and non-minimum phase plants and incorporates the concept of control horizon as well as the consideration of weighting of control increments in the cost function. The general set of choices available for GPC leads to a greater variety of control objective compared to other approaches, some of which can be considered as subsets or limiting cases of GPC. The GPC scheme is shown in Fig. 2. It consists of the plant to be controlled, a reference model that specifies the desired performance of the plant, a linear model of the plant, and the Cost Function Minimization (CFM) algorithm that determines the input needed to produce the plant's desired performance. The GPC algorithm consists of the CFM block. The GPC system starts with the input signal, $r(t)$, which is presented to the reference model. This model produces a tracking reference signal, $w(t)$, that is used as an input to the CFM block. The CFM algorithm produces an output which is used as an input to the plant. Between samples, the CFM algorithm uses this model to calculate the next control input, $u(t+1)$, from predictions of the response from the plant's model. Once the cost function is minimized, this input is passed to the plant.

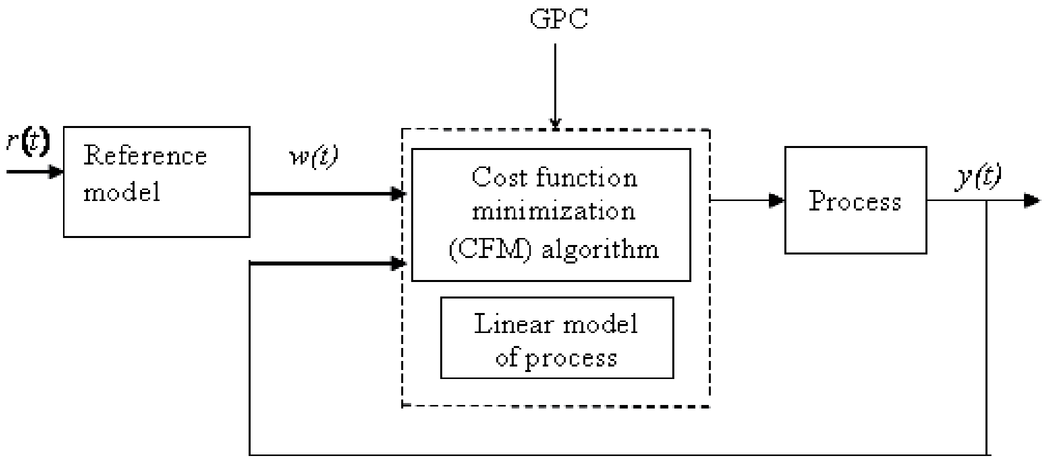


Fig. 2. Basic Structure of GPC

3.2 Formulation of Generalized Predictive Control

Most single-input single-output (SISO) plants, when considering operation around particular set-points and after linearization, can be described by the following:

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) + C(z^{-1})e(t) \tag{1}$$

where $u(t)$ and $y(t)$ are the control and output sequence of the plant and $e(t)$ is a zero mean white noise. A, B and C are the following polynomials in the backward shift operator z^{-1}:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots\ldots + a_{na} z^{-na}$$
$$B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots\ldots + b_{nb} z^{-nb}$$
$$C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} + \ldots\ldots + c_{nc} z^{-nc}$$

where, d is the dead time of the system. This model is known as a Controller Auto-Regressive Moving-Average (CARMA) model. It has been argued that for many industrial applications in which disturbances are non-stationary an integrated CARMA (CARIMA) model is more appropriate. A CARIMA model is given by,

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) + C(z^{-1})\frac{e(t)}{\Delta} \tag{2}$$

with $\Delta = 1 - z^{-1}$

For simplicity, C polynomial in (2) is chosen to be 1. Notice that if C^{-1} can be truncated it can be absorbed into A and B.

3.3 Cost Function

The GPC algorithm consists of applying a control sequence that minimizes a multistage cost function,

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta(j)[\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)[\Delta u(t+j-1)]^2 \tag{3}$$

where $\hat{y}(t+j\,|\,t)$ is an optimum *j*-step ahead prediction of the system output on data up to time *k*, N_1 and N_2 are the minimum and maximum costing horizons, N_u control horizon, $\delta(j)$ and $\lambda(j)$ are weighing sequences and $w(t+j)$ is the future reference trajectory, which can considered to be the constant.

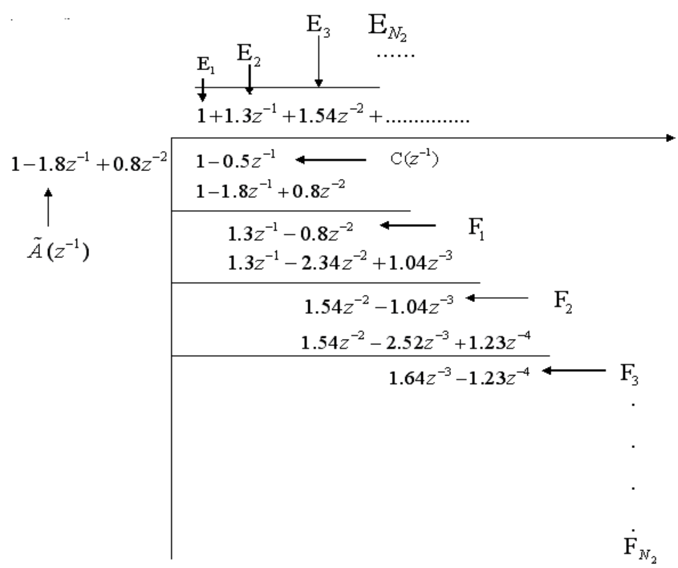
The objective of predictive control is to compute the future control sequence $u(t)$, $u(t+1), \ldots u(t+N_u)$ in such a way that the future plant output $y(t+j)$ is driven close to $w(t+j)$. This is accomplished by minimizing $J(N_1, N_2, N_u)$.

3.4 Cost Function Minimization Algorithm

In order to optimize the cost function the optimal prediction of $y(t+j)$ for $j \geq N_1$ and $j \leq N_2$ is required. To compute the predicted output, consider the following Diophantine equation,

$$1 = E_j(z^{-1})\tilde{A}(z^{-1}) + z^{-j}F_j(z^{-1}) \quad \text{with } \tilde{A}(z^{-1}) = \Delta A(z^{-1}) \tag{4}$$

The polynomials E_j and F_j are uniquely defined with degrees $j-1$ and na respectively. They can be obtained dividing 1 by $\tilde{A}(z^{-1})$ until the remainder can be factorized as $z^{-j}F_j(z^{-1})$. The quotient of the division is the polynomial $E_j(z^{-1})$. An example demonstrating calculation of E_j and *Fj* coefficients in Diophantine equation (4) is shown in Example 1 given below:



Example 1. Diophantine equation demonstration example

If (2) is multiplied by $\Delta E_j(z^{-1})z^j$ we get,

$$\tilde{A}(z^{-1})E_j(z^{-1})y(t+j) = E_j(z^{-1})B(z^{-1})\Delta u(t+j-d-1) + E_j(z^{-1})e(t+j) \tag{5}$$

Substituting (4), in (5) we get,

$$(1-z^{-j}F_j(z^{-1}))y(t+j) = E_j(z^{-1})B(z^{-1})\Delta u(t+j-d-1)) + F_j(z^{-1})y(t) + E_j(z^{-1})e(t+j)$$

which can be rewritten as:

$$y(t+j) = F_j(z^{-1})y(t) + E_j(z^{-1})B(z^{-1})\Delta u(t+j-d-1)) + F_j(z^{-1})y(t) + E_j(z^{-1})e(t+j) \tag{6}$$

degree of polynomial is $E_j(z^{-1}) = j-1$ the noise terms in equation are all in the future. The best prediction of $y(t+j)$ is given by,

$$\hat{y}(t+j \mid t) = G_j(z^{-1})\Delta u(t+j-d-1) + F_j(z^{-1})y(t) \tag{7}$$

where, $G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$.

It is very simple to show that, the polynomials E_{j+1} and F_j can be obtained recursively. Consider that polynomials E_j and F_j have been obtained by dividing 1 by $\tilde{A}(z^{-1})$ until the remainder of the division can be factorized as $z^{-j}F_j(z^{-1})$. These polynomials can be expressed as:

$$F_j(z^{-1}) = f_{j,0} + f_{j,1}z^{-1} + f_{j,2}z^{-2} + \dots\dots + f_{j,na}z^{-na} \tag{8}$$

$$E_j(z^{-1}) = e_{j,0} + e_{j,1}z^{-1} + e_{j,2}z^{-2} + \dots\dots + e_{j,j-1}z^{-(j-1)} \tag{9}$$

Suppose that the same procedure is used to obtain E_{j+1} and F_{j+1}, that is, dividing 1 by $\tilde{\mathbf{A}}(z^{-1})$ until the remainder of the division can be factorized as $z^{-(j-1)}F_{j+1}(z^{-1})$ with

$$F_{j+1}(z^{-1}) = f_{j+1,0} + f_{j+1,1}z^{-1} + \dots\dots + f_{j+1,na}z^{-na} \tag{10}$$

It is clear that only another step of the division is performed to obtain the polynomials E_{j+1} and F_{j+1}. The polynomial E_{j+1} will be given by:

$$E_{j+1}(z^{-1}) = E_j(z^{-1}) + e_{j+1,j}z^{-j} \tag{11}$$

with $e_{j+1,j} = f_{j,0}$. The coefficients of polynomial F_{j+1} can then be expressed as:

$$f_{j+1,i} = f_{j+1,i+1} - f_{j,0}\tilde{a}_{i+1} \qquad i = 0 \;....\; na-1 \tag{12}$$

The Polynomial G_{j+1} can be obtained recursively as follows:

$$G_{j+1} = E_{j+1}B = (E_j + f_{j,0}z^{-1})B \tag{13}$$

$$G_{j+1} = G_j + f_{j,0}z^{-1}B \tag{14}$$

That is, the first *j* coefficient of G_{j+1} will be identical to those of G_j and the remaining coefficients will be given by:

$$g_{j+1,j+i} = g_{j,j+i} + f_{j,0}b_i \qquad \text{for } i = 0 \;.....\; nb \tag{15}$$

To solve the GPC problem the set of control signals $u(t)$, $u(t+1)$, …, $u(t+N)$ has to be obtained in order to optimize expression. As the system considered has a dead time d sampling periods, the output of the system will be influenced by signal $u(t)$ after sampling period $d+1$. The values N_1, N_2 and N_u defining the horizon can be defined

by $N_2 = d + N$, $N_2 = d + N$ and $N_u = N$. Notice that there is no point in making $N_1 > d + 1$ as terms added to expression will only depend on the past control signals. On the other hand, if $N_1 > d + 1$, the first point in the reference sequence, being the ones guessed with most certainly, will not be taken into account.
Now consider the following set of *j* ahead optimal predictions given below:

$$\begin{aligned}
\hat{y}(t+d+1 \mid t) &= G_{d+1}\Delta u(t) + F_{d+1}y(t) \\
\hat{y}(t+d+2 \mid t) &= G_{d+2}\Delta u(t+1) + F_{d+2}y(t) \\
\hat{y}(t+d+N \mid t) &= G_{d+N}\Delta u(t+N-1) + F_{d+N}y(t)
\end{aligned} \tag{16}$$

which can be re-written as:

$$y = \mathbf{G}u + \mathbf{F}(z^{-1})y(t) + \mathbf{G}'(z^{-1})\Delta u(t-1) \tag{17}$$

where,

$$y = \begin{bmatrix} \hat{y}(t+d+1 \mid t) \\ \hat{\hat{y}}(t+d+2 \mid t) \\ \dots\dots \\ \hat{y}(t+d+N \mid t) \end{bmatrix};\ u = \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \dots\dots \\ \Delta u(t+N-1) \end{bmatrix};\ \mathbf{G} = \begin{bmatrix} g_0 & 0 & \dots\dots & 0 \\ g_1 & g_0 & \dots\dots & 0 \\ . & . & . & . \\ g_{N-1} & g_{N-2} & \dots\dots & g_0 \end{bmatrix}$$

$$\mathbf{G}'(z^{-1}) = \begin{bmatrix} (\mathbf{G}_{d+1}(z^{-1}) - g_0)z \\ (\mathbf{G}_{d+2}(z^{-1}) - g_0 - g_1 z^{-1})z^2 \\ \dots\dots \\ (\mathbf{G}_{d+N}(z^{-1}) - g_0 - g_1 z^{-1} \dots\dots g_{N-1}z^{-(N-1)})z^N \end{bmatrix};\ \mathbf{F}(z^{-1}) = \begin{bmatrix} F_{d+1}(z^{-1}) \\ F_{d+2}(z^{-1}) \\ \dots\dots \\ F_{d+N}(z^{-1}) \end{bmatrix}$$

Note that if the plant dead time is $d > 1$ the first d - 1 rows of **G** will be null, but if instead N_1 is assumed to be equal to d the leading element is non-zero. However, as d will not in general be known in the self-tuning case one key feature of the GPC approach is that a stable solution is possible even if the leading rows of **G** are zero.
Notice that the last two terms in (17) only depends on the past and can be grouped into f leading to:

$$\hat{y} = \mathbf{G}\mathrm{u} + \mathrm{f}$$

Notice that if all initial conditions are zero, the free response f is also zero. If a unit step is applied to the input at time *t;* that is

$$\Delta u(t) = 1, \Delta u(t+1) = 0, \ \dots, \ \Delta u(t+N-1) = 0$$

then expected output sequence $[\hat{y}(t+1), \hat{y}(t+2), \dots\dots \hat{y}(t+N)]^{\mathrm{T}}$ is equal to the first column of matrix $\mathbf{G}$. That is, the first column of matrix $\mathbf{G}$ can be calculated as the step response of the plant when a unit step is applied to the manipulated variable. The free response term can be calculated recursively by:

$$\mathbf{f}_{j+1} = z(1 - \tilde{A}(z^{-1}))\mathbf{f}_j + B(z^{-1})\Delta u(t - d + j) \tag{18}$$

with $\mathrm{f}_0 = y(t)$ and $\Delta u(t+j) = 0$ for $j \geq 0$.
The equation (3) can be written as:

$$J = (Gu + f - w)^T(Gu + f - w) + \lambda u^T u \tag{19}$$

where,

$$w = [w(t+d+1) \quad w(t+d+2) \quad ..... \quad w(t+d+N)]^T$$

It has been considered that the future reference trajectory keeps constant along the horizon or its evolution is unknown and therefore $w(t+i) = w(t)$.

The equation (19) can be written as:

$$J = \frac{1}{2} u^T H u + b^T u + f_0 \tag{20}$$

where,

$$\mathbf{H} = 2(\mathbf{G}^T \mathbf{G} + \mathrm{R\,I})$$
$$b^T = 2(\mathrm{f-w})^T \mathbf{G}$$
$$\mathrm{f}_0 = (\mathrm{f-w})^T (\mathrm{f-w})$$

The minimum of *J*, assuming there are no constraints on the control signals, can be found by making the gradient of J equal to zero, which leads to:

$$u = -H^{-1} b = (G^T G + \lambda I)^{-1} G^T (w - f) \tag{21}$$

The dimension of the matrix involved in equation (19) is *N* x *N*. Although in the non-adaptive case the inversion need be performed only once. In a self-tuning version the computational load of inverting at each sample would be excessive. Moreover, if the wrong value for dead-time is assumed, $G^T G$ is singular and hence a finite non-zero value of weighting λ would be required for a realizable control law, which is inconvenient because the accurate value for λ would not be known a priori. Notice that the control signal that is actually sent to the process is the first element of vector u, which is given by:

$$\Delta u = K(w - f) \tag{22}$$

where *K* is the first row of matrix $(G^T G + \lambda I)^{-1} G^T$. If there are no future predicted errors, that is, if $(w - f) = 0$, then there is no control move, since the objective will be fulfilled with the free evolution of the process. However, in the other case, there will be an increment in the control action proportional (with a factor *K*) to that future error. Notice that the action is taken with respect to *future* errors, not *past* errors, as is the case in conventional feedback controllers.

Also notice that when only the first element of u is applied, then at the next sampling instants, new data are acquired and a new set of control moves is calculated. Once again, only the first control move is implemented. These activities repeated at each sampling instant, and the strategy is referred to as a *receding horizon approach*. It may strange to calculate an N_u -step control policy and then only implement the first move. The important advantage of this receding horizon approach is that new information in the form of the most recent measurements *y*(*k*) is utilized immediately instead of being ignored for the next N_u sampling instants. Otherwise, the multi-step predictions and control moves would be based on old information and thus be adversely affected by unmeasured disturbances.

4. Introduction to Neural Generalized Predictive Control

The Generalized Predictive Control (GPC), introduced in above section, belongs to a class of digital control methods called Model-Based Predictive Control (MBPC). GPC is known to control a non-minimum phase plants, open-loop unstable plants and plants with variable or unknown dead time. GPC had been originally developed with linear plant predictor models which, leads to a formulation that can be solved analytically. But most of the real processes show nonlinear behavior. Some work has to be done to extend the predictive control techniques to incorporate nonlinear models. Developing adequate nonlinear empirical models is very difficult and there is no model form that is clearly suitable to represent general nonlinear processes. Part of the success of standard model based predictive techniques was due to the relative ease with which step and impulse responses or low order transfer functions could be obtained. A major mathematical obstacle to complete theory of nonlinear processes is the lack of superposition principal for nonlinear systems. The selection of the minimization algorithm affects the computational efficiency of the algorithm. Explicit solution for it can be obtained if the criterion is quadratic, the model is linear and there are no constraints; otherwise an iterative optimization method has to be used. In this work a Newton-Raphson method is used as the optimization algorithm. The main cost of the Newton-Raphson algorithm is in the calculation of the Hessian, but even with this overhead the low iteration numbers make Newton-Raphson a faster algorithm for real-time control (Soloway & Haley,1997).
The Neural Generalized Predictive Control (NGPC) scheme is shown in Fig. 3. It consists of four components, the plant to be controlled, a reference model that specifies the desired performance of the plant, a neural network that models the plant, and the Cost Function Minimization (CFM) algorithm that determines the input needed to produce the plant's desired performance. The NGPC algorithm consists of the CFM block and the neural net block.

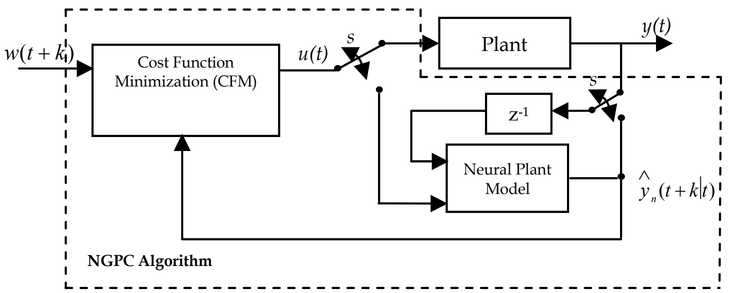


Fig. 3. Block Diagram of NGPC System

The NGPC system starts with the input signal, *w(t),* which is applied to the reference model. This model produces a tracking reference signal, *w(t+k),* that is used as an input to the CFM block. The CFM algorithm produces an output which is either used as an input to the plant or the plant's model. The double pole double throw switch, S**,** is set to the plant when the CFM algorithm has solved for the best input, *u(n),* that will minimize a specified cost function. Between samples, the switch is set to the plant's model where the CFM algorithm uses this model to calculate the next control input, *u(n+1)***,** from predictions of the response from the plant's model. Once the cost function is minimized, this input is passed to the plant. The computational performance of a GPC implementation is largely based on the minimization algorithm chosen for the CFM block. The selection of a minimization method can be based on several criteria such as: number of iterations to a solution, computational costs and accuracy of the solution. In general these approaches are iteration intensive thus making real-time control difficult. In this work Newton-Raphson as an optimization technique is used. Newton-Raphson is a quadratically converging. The improved convergence rate of Newton-Raphson is computationally costly, but is justified by the high convergence rate of Newton-Raphson. The quality of the plant's model affects the accuracy of a prediction. A reasonable model of the plant is required to implement GPC. With a linear plant there are tools and techniques available to make modeling easier, but when the plant is nonlinear this task is more difficult. Currently there are two techniques used to model nonlinear plants. One is to linearize the plant about a set of operating points. If the plant is highly nonlinear the set of operating points can be very large. The second technique involves developing a nonlinear model which depends on making assumptions about the dynamics of the nonlinear plant. If these assumptions are incorrect the accuracy of the model will be reduced. Models using neural networks have been shown to have the capability to capture nonlinear dynamics**.** For nonlinear plants, the ability of the GPC to make accurate predictions can be enhanced if a neural network is used to learn the dynamics of the plant instead of standard modeling techniques. Improved predictions affect rise time, over-shoot, and the energy content of the control signal.

5. Formulation of NGPC

5.1 Cost Function

As mentioned earlier, the NGPC algorithm is based on minimizing a cost function over a finite prediction horizon. The cost function of interest to this application is,

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta(j)\left[\hat{y}_n(t+j|t) - w(t+j)\right]^2 + \sum_{j=1}^{N_u} \lambda(j)\left[\Delta u(t+j-1)\right]^2 \quad (23)$$

N_1 = Minimum costing prediction horizon
N_2 = Maximum costing prediction horizon
N_u= Length of control horizon
$\hat{y}(t+k|t)$ = Predicted output from neural network
$u(t+k|t)$ = Manipulated input
$w(t+k)$ = Reference trajectory
δ and λ = Weighing factor

This cost function minimizes not only the mean squared error between the reference signal and the plant's model, but also the weighted squared rate of change of the control input with it's constraints. When this cost function is minimized, a control input that meets the constraints is generated that allows the plant to track the reference trajectory within some tolerance. There are four tuning parameters in the cost function, N_1, N_2, N_u, and λ**.** The predictions of the plant will run from N_1 to N_2 future time steps. The bound on the control horizon is N_u. The only constraint on the values of N_u and N_1 is that these bounds must be less than or equal to N_2. The second summation contains a weighting factor, λ that is introduced to control the balance between the first two summations. The weighting factor acts as a damper on the predicted *u(n+1)*.

5.2 Cost Function Minimization Algorithm

The objective of the CFM algorithm is to minimize *J* in equation (24) with respect to $[u(n+l), u(n+2), ..., u(n+N_u)]^T$, denoted as *U*. This is accomplished by setting the Jacobian of equation (23) to zero and solving for *U*. With Newton-Rhapson used as the CFM algorithm, *J* is minimized iteratively to determine the best *U*. An iterative process yields intermediate values for *J* denoted *J(k)*. For each iteration of *J(k)* an intermediate control input vector is also generated and is denoted as:

$$U(k) \stackrel{\Delta}{=} \begin{bmatrix} u(t+1) \\ u(t+2) \\ . \\ . \\ . \\ u(t+N_u) \end{bmatrix} \qquad k = 1, \; .....N_u \tag{24}$$

The Newton-Raphson method is one of the most widely used of all root-locating formula. If the initial guess at the root is x_i, a tangent can be extended from the point $[x_i, f(x_i)]$. The point where this tangent crosses the *x* axis usually represents an improved estimate of the root. So the first derivative at *x* on rearranging can be given as: $x_{(i+1)} = x_{(i)} - \dfrac{f(x_i)}{f'(x_i)}$

Using this Newton-Raphson update rule, $U(k+1)$ is given by,

$$U(k+1) = U(k) - \left(\frac{\partial^2 J}{\partial U^2}(k) \right)^{-1} \frac{\partial J}{\partial U}(k)' \quad \text{where} \quad f(x) = \frac{\partial J}{\partial U} \tag{25}$$

and the Jacobian is denoted as,

$$\frac{\partial J}{\partial U}(k) \equiv \begin{bmatrix} \dfrac{\partial J}{\partial u(t+1)} \\ . \\ . \\ . \\ \dfrac{\partial J}{\partial u(t+N_u)} \end{bmatrix} \tag{26}$$

Also the Hessian is given by,

$$\frac{\partial^2 J}{\partial U^2}(k) \equiv \begin{bmatrix} \frac{\partial^2 J}{\partial u(t+1)^2} & \cdot\ \cdot & \frac{\partial^2 J}{\partial u(t+1)\partial u(t+N_u)} \\ \cdot & \cdot\ \cdot & \cdot \\ \cdot & \cdot\ \cdot & \cdot \\ \frac{\partial^2 J}{\partial u(t+N_u)\partial u_k(t+1)} & \cdot\ \cdot & \frac{\partial^2 J}{\partial u(t+N_u)^2} \end{bmatrix} \tag{27}$$

Each element of the Jacobian is calculated by partially differentiating equation (23) with respect to vector *U*.

$$\frac{\partial J}{\partial u(t+h)} = 2\sum_{j=N_1}^{N_2} \delta(j)\left[\hat{y}_n(t+j) - w(t+j)\right]\frac{\partial \hat{y}_n(t+j)}{\partial u(t+h)} + 2\sum_{j=1}^{N_u} \lambda(j)\left[\Delta u(t+j)\right]\frac{\partial \Delta u(t+j)}{\partial u(t+h)} \tag{28}$$

where, $h = 1,\dots,N_u$.

Once again equation (28) is partially differentiated with respect to vector *U* to get each element of the Hessian.

$$\frac{\partial^2 J}{\partial u(t+m)\partial u(t+h)} = 2\sum_{j=N_1}^{N_2} \delta(j)\left\{\frac{\partial^2 \hat{y}(t+j)}{\partial u(t+m)\partial u(t+h)}\left[\hat{y}(t+j) - w(t+j)\right] - \frac{\partial \hat{y}(t+j)}{\partial u(t+m)}\frac{\partial \hat{y}(t+j)}{\partial u(t+h)}\right\}$$

$$+2\sum_{j=N_1}^{N_2} \lambda(j)\left\{\frac{\partial \Delta n(t+j)}{\partial u(t+m)}\frac{\partial \Delta n(t+j)}{\partial u(t+h)} + \frac{\partial^2 \Delta n(t+j)}{\partial u(t+m)\partial u(t+h)}\right\} \tag{29}$$

The m^{th}, h^{th} elements of the Hessian matrix in equation (27) are, $h = 1,\dots\dots,N_u$ and $m = 1,\dots\dots,N_u$.

The last computation needed to evaluate $U(k+1)$ is the calculation of the predicted output of the plant, $\hat{y}(t+j)$, and it's derivatives. The next sections define the equation of a multilayer feed forward neural network, and define the derivative equations of the neural network.

6. Neural Network for Prediction

In NGPC the model of the plant is a neural network. This neural model is constructed and trained using MATLAB Neural Network System Identification Toolbox commands (Norgaard, 2000). The output of trained neural network is used as the predicted output of the plant. This predicted output is used in the Cost Function Minimization Algorithm. If $y_n(t)$ is the neural network's output then it is nothing but plant's predicted output $\hat{y}_n(t+k|t)$. The initial training of the neural network is typically done off-line before control is attempted. The block configuration for training a neural network to model the plant is shown in Fig. 4. The network and the plant receive the same input, $u(t)$. The network has an additional input that either comes from the output of the plant, $y(t)$, or the neural network's, $y_n(t)$. The one that is selected depends on the plant and the application. This input assists the network with capturing the plant's dynamics and stabilization of unstable systems. To train the network, its weights are adjusted such that a set of inputs produces the desired set of outputs. An error is formed between the responses of the network, $y_n(t)$, and the plant, $y(t)$. This error is then used to update the weights of the network through gradient descent learning. In this work, a Levenberg-Marquardt method is used as gradient descent learning algorithm for updating the weights. This is standard method for minimization of mean-square error criteria, due to its rapid convergence properties and robustness. This process is repeated until the error is reduced to an acceptable level. Since a neural network is used to model the plant, the configuration of the network architecture should be considered. This implementation of NGPC adopts input/output models.

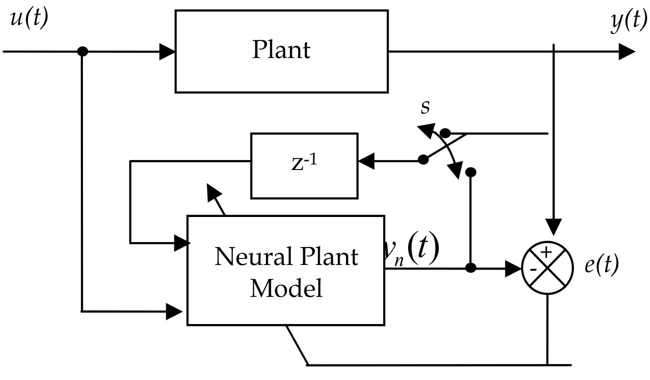


Fig. 4.Block Diagram of Off-line Neural Network Training

The diagram shown in Fig. 5, depicts a multi-layer feed-forward neural network with a time delayed structure. For this example, the inputs to this network consists of two external inputs, $u(t)$ and two outputs $y(t-1)$, with their corresponding delay nodes, $u(t)$, $u(t-1)$ and $y(t-1)$, $y(t-2)$. The network has one hidden layer containing five hidden nodes that uses bi-polar sigmoidal activation output function. There is a single output node which uses a linear output function, of one for scaling the output.

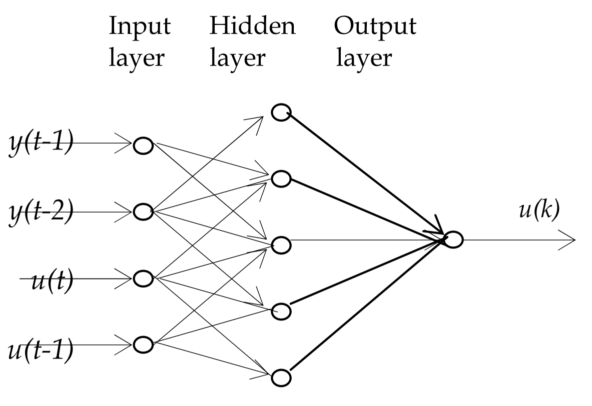


Fig. 5.Neural Network Architecture

The equation describing this network architecture is:

$$y_n(t) = \sum_{j=1}^{hid} w_j f_j\left(net(t)\right) \tag{30}$$

and

$$net_j(t) = \sum_{j=1}^{n_d} w_{j,i+1} u(t-i) + \sum_{j=1}^{d_d} w_{j,n_d+i+1} y(t-i) \tag{31}$$

where,

$y_n(t)$	is the output of the neural network
$f_j(.)$	is the output function for the j^{th} node of the hidden layer
$net_j(t)$	is the activation level of the j^{th} node's output function
hid	is the number of hidden nodes in the hidden layer
n_d	is the number of input nodes associated with *u(.)*
d_d	is the number of input nodes associated with *y(.)*
w_j	is the weight connecting the j^{th} hidden node to the output node
$w_{j,i}$	is the weight connecting the i^{th} hidden input node to the j^{th} hidden node
$y(t-i)$	is the delayed output of the plant used as input to the network
$u(t-i)$	is the input to the network and its delays

This neural network is trained in offline condition with plants input/output data.

Prediction Using Neural Network

The NGPC algorithm uses the output of the plant's model to predict the plant's dynamics to an arbitrary input from the current time, *t,* to some future time, *t+k.* This is accomplished by time shifting equations equation (30) and (31), by *k,* resulting in the following equations given by,

$$y_n(t+k) = \sum_{j=1}^{hid} w_j f_j\left(net_j(t+k)\right) \tag{32}$$

and

$$net_j(t+k) = \sum_{i=1}^{hid} w_{j,i+1} \begin{cases} u(n+k-i), k-N_u < i \\ u(n+N_u), k-N_u \geq i \end{cases} + \sum_{j=1}^{\min(k,d_d)} \left(w_{j,n_d+i+1} y_n(t+k-i)\right)$$
$$+ \sum_{i=k+1}^{d_d} \left(w_{j,n_d+i+1} y(t+k-i)\right) \tag{33}$$

The first summation in equation (33) breaks the input into two parts represented by the conditional. The condition where $k-N_u < i$ handles the previous future values of the u up to *u(t+Nu-1).*The condition where $k-N_u > i$ sets the input from $u(t+N_u)$ to $u(t+k)$ equal to $u(t+N_u)$. The second condition will only occur if $N_2 > N_u$. The next summation of equation (33) handles the recursive part of prediction. This feeds back the network output, y_n, for k or d_d times, which ever is smaller. The last summation of equation (33) handles the previous values of y. The following section derives the derivatives of equation (32) and (33) with respect to the input $u\left(t+h\right)$.

6.1 Neural Network Derivative Equations

To evaluate the Jacobian and the Hessian in equation (26) and (27) the network's first and second derivative with respect to the control input vector are needed.

Jacobian Element Calculation:

The elements of the Jacobian are obtained by differentiating $y_n(t+k)$ in equation (32) with respect to $u(t+h)$ resulting in

$$\frac{\partial yn\left(t+k\right)}{\partial u\left(t+h\right)} = \sum_{j=1}^{hid} w_j \frac{\partial f_j\left(net_j\left(t+k\right)\right)}{\partial u\left(t+h\right)} \tag{34}$$

Applying chain rule to $\partial f_j\left(net_j\left(t+k\right)\right)/\partial u\left(t+h\right)$ results in

$$\frac{\partial f_j\left(net_j\left(t+k\right)\right)}{\partial u\left(t+h\right)} = \frac{\partial f_j\left(net_j\left(t+k\right)\right)}{\partial net_j\left(t+k\right)} \frac{\partial net_j\left(t+k\right)}{\partial u\left(t+h\right)} \tag{35}$$

where $\partial f_j\left(net_j\left(t+k\right)\right)/\partial net_j\left(t+k\right)$ is the output function's derivative which will become zero as we are using a linear (constant value) output activation function and

$$\frac{\partial net_j\left(t+k\right)}{\partial u\left(t+h\right)} = \sum_{i=0}^{n_d} w_{j,i+1} \begin{cases} \delta\left(k-i,h\right), k-N_u < i \\ \delta\left(N_u,h\right), k-N_u \geq i \end{cases} + \sum_{i=0}^{\min(k,d_d)} w_{j,i+n_d+1} \frac{\partial yn\left(t+k-i\right)}{\partial u\left(t+h\right)} \delta_1\left(k-i-1\right) \tag{36}$$

Note that in the last summation of equation (36) the step function, δ**,** was introduced. This was added to point out that this summation evaluates to zero for $k\text{-}i<l$, thus the partial does not need to be calculated for this condition.

6.2 Hessian Element Calculation

The Hessian elements are obtained by once again differentiating equations (34) by $u(t+m)$, resulting in equation (37):

$$\frac{\partial^2 yn\left(t+k\right)}{\partial u(t+h)\partial u(t+m)} = \sum_{i=0}^{n_d} w_j \frac{\partial^2 f_j\left(net_j\left(t+k\right)\right)}{\partial u\left(t+h\right)\partial u\left(t+m\right)} \tag{37}$$

where,

$$\frac{\partial^2 f_j\left(net_j\left(t+k\right)\right)}{\partial u\left(t+h\right)\partial u\left(t+m\right)} = \frac{\partial f_j\left(net_j\left(t+k\right)\right)}{\partial net_j\left(t+k\right)} \frac{\partial^2 net_j\left(t+k\right)}{\partial u\left(t+h\right)\partial u\left(t+m\right)} + \frac{\partial^2 f_j\left(net_j\left(t+k\right)\right)}{\partial net_j\left(t+k\right)^2} \frac{\partial net_j\left(t+k\right)}{\partial u\left(t+h\right)} \frac{\partial net_j\left(t+k\right)}{\partial u\left(t+m\right)} \tag{38}$$

The equation (38) is the result of applying the chain rule twice.

7. Simulation Results

The objective of this study is to show how GPC and NGPC implementation can cope with linear as well as nonlinear systems. GPC is applied to the systems with changes in system order. The Neural based GPC is implemented using MATLAB Neural Network Based System Design Toolbox (Norgaard, 2000)

7.1 GPC and NGPC for Linear Systems

The above derived GPC and NGPC algorithm is applied to the different linear models with varying system order, to test its capability. This is done by carrying out simulation in MATLAB 7.0.1 (Mathworks Natic, 2007). Different systems with large dynamic differences are considered for simulation. GPC and NGPC are showing robust performance for these systems. In below figures, for every individual system the systems output with GPC and NGPC is plotted in single figure for comparison purpose. Also the control efforts taken by the both controllers are plotted in consequent figures for every individual figure.

In this simulation, neural network architecture considered is as follows. The inputs to this network consists of two external inputs, *u(t)* and two outputs *y(t-1),* with their corresponding delay nodes, *u(t), u(t-1)* and *y(t-1), y(t-2)*. The network has one hidden layer containing five hidden nodes that uses bi-polar sigmoidal activation output function. There is a single output node which uses a linear output function, of one for scaling the output.

For all the systems Prediction Horicon N_1 =1, N_2 =7 and Control Horizon (N_u) is 2. The weighing factor λ for control signal is kept to 0.3 and δ for reference trajectory is set to 0. The same controller setting is used for all the systems simulation. The following simulation results are obtained showing the plant output when GPC and NGPC are applied. Also the required control action for different systems is shown.

System I: The GPC and NGPC algorithms are applied to a second order system given below.

$$G(s) = \frac{1}{1 + 10s + 40s^2} \tag{39}$$

The Fig.6. Shows the plant output with GPC and NGPC for setpoint tracking. The Fig. 7 shows the control efforts taken by both controllers. The simulation results reveal that performance of NGPC is better than GPC.

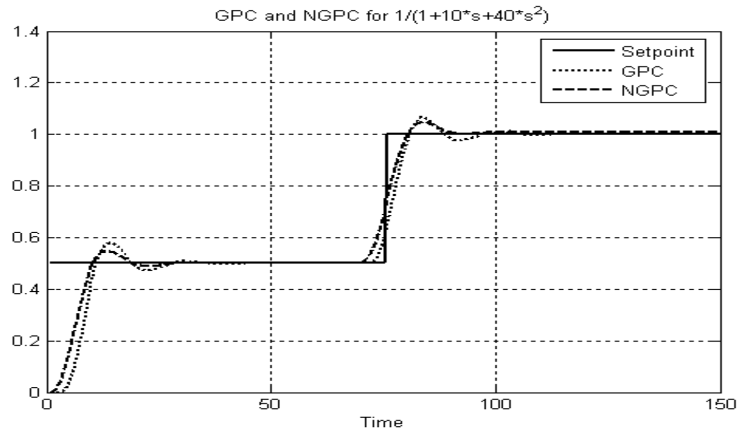


Fig. 6. System I Output using GPC and NGPC

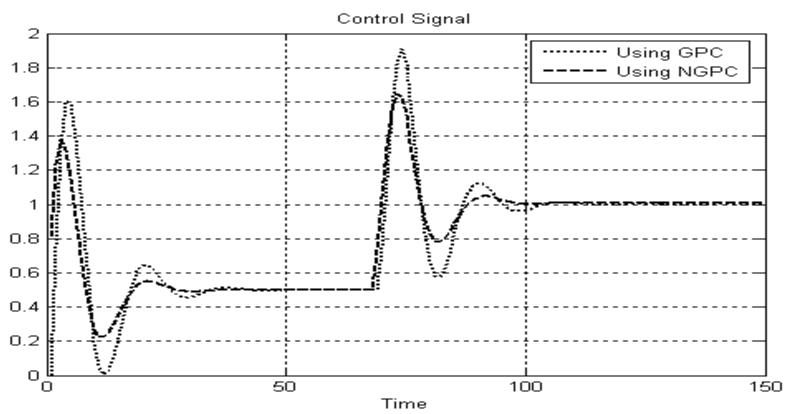


Fig. 7. Control Signal for System I

System II A simple first order system given below is to be controlled by GPC and NGPC.

$$G(s) = \frac{1}{1 + 10s} \tag{40}$$

Fig. 8 and Fig. 9 show the system output and control signal.

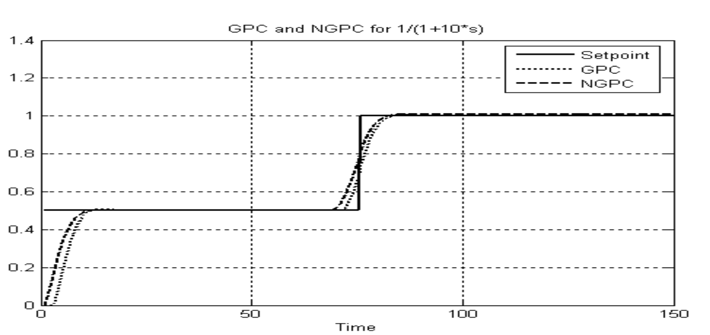


Fig. 8. System II Output using GPC and NGPC

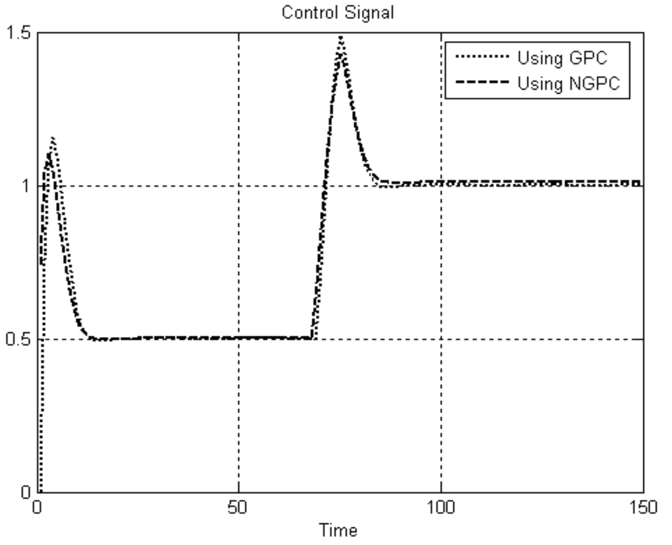


Fig. 9. Control Signal for System II

System III: A second order system given below is controlled using GPC and NGPC.

$$G(s) = \frac{1}{10s(1+2.5s)} \tag{41}$$

Fig. 10 and Fig. 11 show the predicted output and control signal.

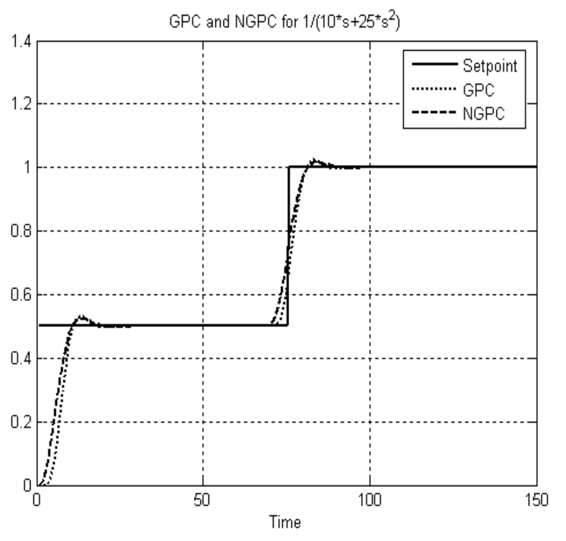


Fig. 10. System III Output using GPC and NGPC

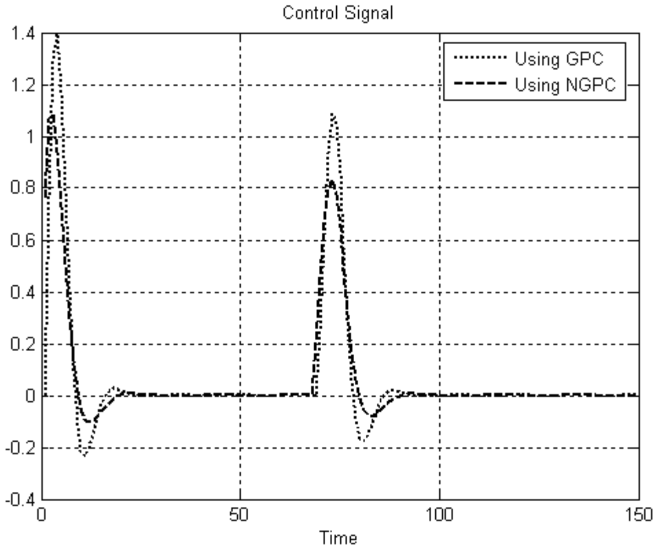


Fig. 11. Control Signal for System III

Before applying NGPC to the all above systems it is initially trained using Levenberg-Marquardt learning algorithm. Fig. 12 (a) shows input data applied to the neural network for offline training purpose. Fig. 12 (b) shows the corresponding neural network output.

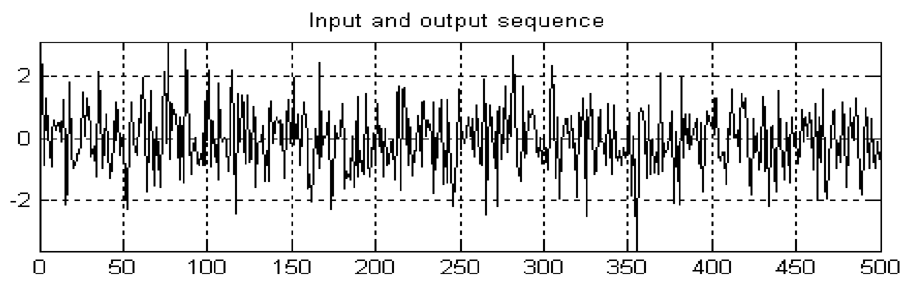


Fig. 12. (a). Input Data for Neural Network Training

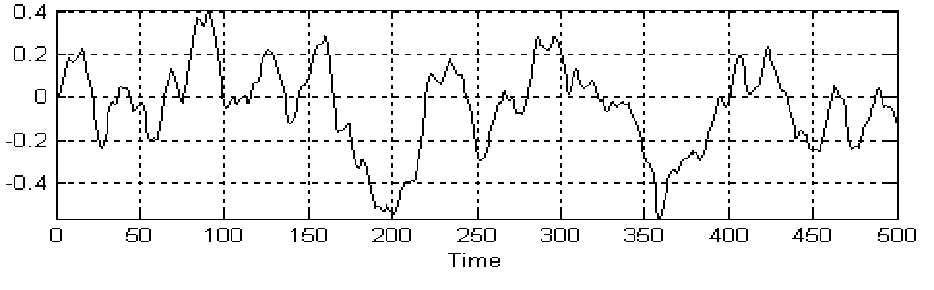


Fig. 12. (b). Neural Network Response for Random Input

To check whether this neural network is trained to replicate it as a perfect model or not, common input is applied to the trained neural network and plant. Fig. 13 (a) shows the trained neural networks output and predicted output for common input. Also the error between these two responses is shown in Fig. 13 (b).
The performance evaluation of both the controller is carried out using ISE and IAE criteria given by the following equations:

$$ISE = \int_0^t e^2 dt; \qquad IAE = \int_0^t |e| \, dt \tag{42}$$

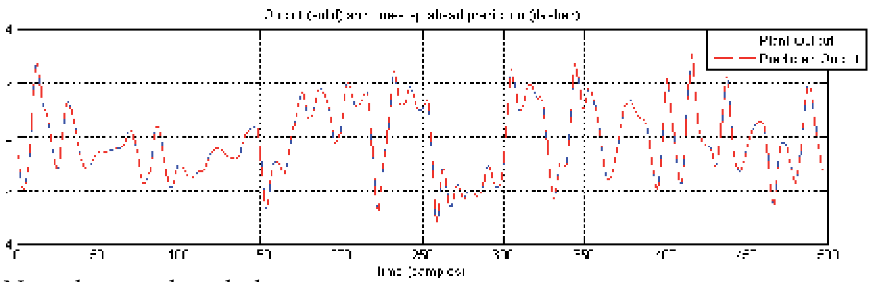


Fig. 13. (a) Neural network and plant output

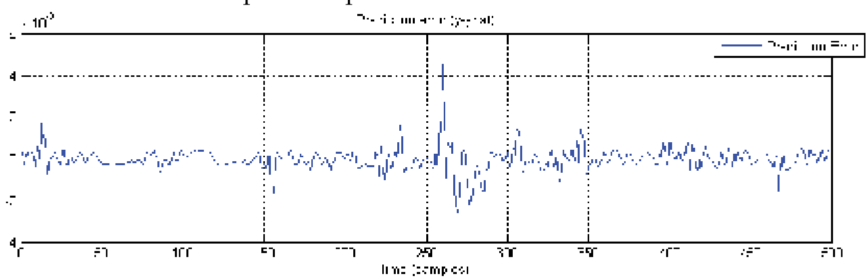


Fig. 13. (b) Error between neural network and plant output

The Table 1 gives ISE and IAE values for both GPC and NGPC implementation for all the linear systems given by equation (39) to equation (41). We can find that, for each system ISE and IAE for NGPC is smaller or equal to GPC. So using GPC with neural network i.e. NGPC control configuration for linear application, is also a better choice.

Systems	**Setpoint**	**GPC**		**NGPC**	
		ISE	IAE	ISE	IAE
System I	0.5	1.6055	4.4107	1.827	3.6351
	1	0.2567	1.4492	0.1186	1.4312
System II	0.5	1.1803	3.217	0.7896	2.6894
	1	0.1311	0.767	0.063	1.017
System III	0.5	1.4639	3.7625	1.1021	3.3424
	1	0.1759	0.9065	0.0957	0.7062

Table 1. ISE and IAE Performance Comparison of GPC and NGPC for Linear System

7.2 GPC and NGPC for Nonlinear System

In above Section GPC and NGPC are applied to the linear systems. Fig. 6 to Fig. 11. show the excellent behavior achieved in all cases by the GPC and NGPC algorithm. For each system only few more steps in setpoint were required for GPC than NGPC to settle down the output, but more importantly there is no sign of instability. In this Section, GPC and NGPC is applied to the nonlinear systems to test its capability. A well known Duffing's nonlinear equation is used for simulation. It is given by,

$$\ddot{y}(t) + \dot{y}(t) + y(t) + y^3(t) = u(t) \tag{43}$$

This differential equation is modeled in MATLAB 7.0.1 (Maths work Natic USA, 2007). Then using linearization technique ('linmod' function) available in MATLAB a linear model of the above system is obtained. This function returns a linear model in State-Space format which is then converted in transfer function. This is given by,

$$\frac{y(s)}{u(s)} = \frac{1}{s^2 + s + 1} \tag{44}$$

This linear model of the system is used in GPC algorithm for prediction. In both the controllers configuration, Prediction Horizon N_1 =1, N_2 =7 and Control Horizon (N_u) is 2 is set. The weighing factor λ for control signal is kept to 0.03 and δ for reference trajectory is set to 0. The sampling period for this simulation is kept at 0.1.

In this simulation, neural network architecture considered is as follows. The inputs to this network consists of two external inputs, *u(t)* and two outputs *y(t-1),* with their corresponding delay nodes, *u(t), u(t-1)* and *y(t-1), y(t-2)*. The network has one hidden layer containing five hidden nodes that uses bi-polar sigmoid activation output function. There is a single output node, which uses a linear output function, of one for scaling the output.

Fig. 14 shows the predicted and actual plant output for the system given in equation (43) when controlled using GPC and NGPC techniques. Fig.15. shows the control efforts taken by both the controller.

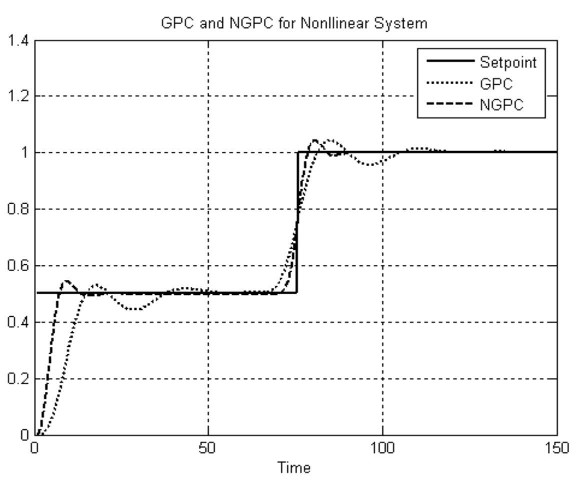


Fig. 14.Predicted Output and Actual Plant Output for Nonlinear System

The Fig.14, shows that, for set point changes the response of GPC is sluggish whereas for NGPC it is fast. The overshoot is also less and response also settles down earlier in NGPC as compared to GPC for nonlinear systems. This shows that performance of NGPC is better than GPC for nonlinear system. The control effort is also smooth in NGPC as shown in Fig. 15

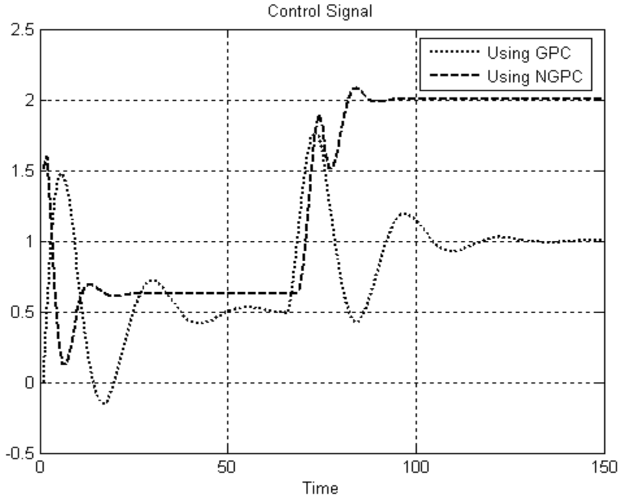


Fig. 15. Control Signal for Nonlinear System

Fig. 16 (a) shows input data applied to the neural network for offline training purpose. Fig. 16 (b) shows the corresponding neural network output.

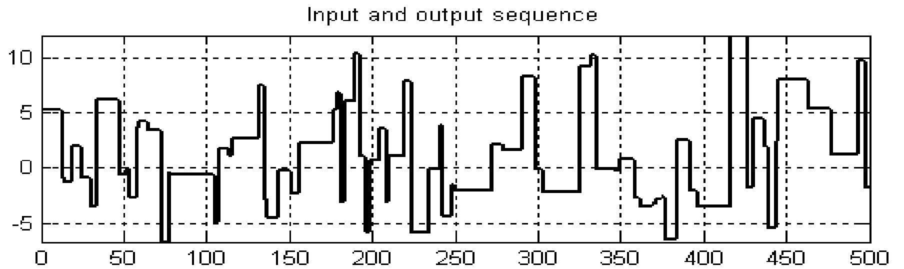


Fig. 16. (a) Input Data for Neural Network Training

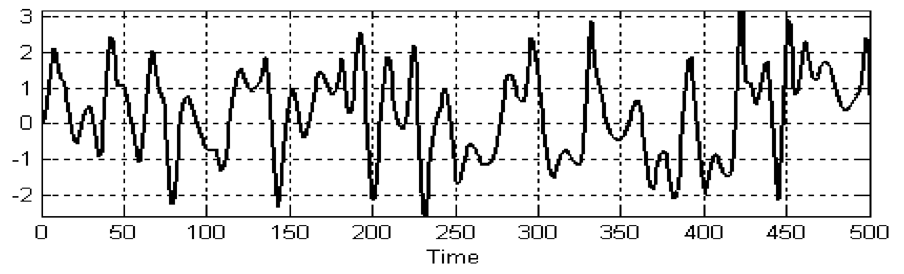


Fig. 16. (b) Neural Network Response for Random Input

The Table 2 gives ISE and IAE values for both GPC and NGPC implementation for the nonlinear system given by equation (43). Here a cubic nonlinearity is present. The NGPC control configuration for nonlinear application is better choice. Same results are also observed for set point equals to 1.

Setpoint	GPC		NGPC	
	ISE	IAE	ISE	IAE
0.5	1.8014	5.8806	0.8066	2.5482
1	0.1199	1.4294	0.0566	0.5628

Table 2. ISE and IAE Performance Comparison of GPC and NGPC for Nonlinear System

7.3 Industrial processes

To evaluate the applicability of the proposed controller, the performance of the controller has been studied on special industrial processes.

Example 1: NGPC for highly nonlinear process (Continues Stirred Tank Reactor)

Further to evaluate the performance of the Neural generalized predictive control (NGPC) we consider highly nonlinear process continuous stirred tank reactor (CSTR) (Nahas,Henson,et al.,1992) .Many aspects of nonlinearity can be found in this reactor, for instance, strong parametric sensitivity, multiple equilibrium points and nonlinear oscillations. The CSTR system, which can be found in many chemical industries, has evoked a lot of interest for the control community due to its challenging theoretical aspects as well as the crucial problem of controlling the production rate. A schematic of the CSTR system is shown in Fig.17. A single irreversible, exothermic reaction A→B is assumed to occur in the reactor.

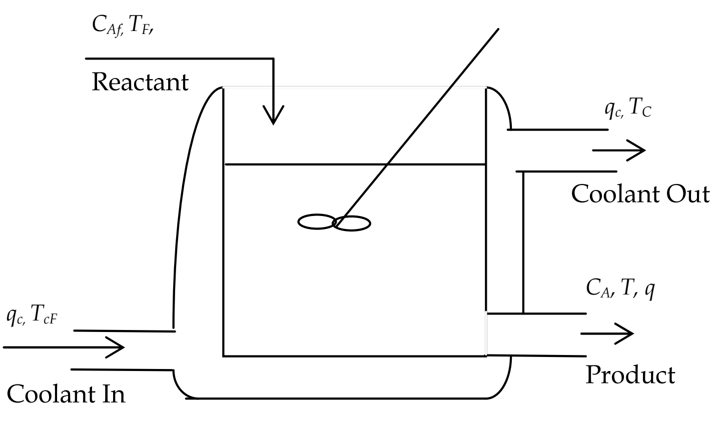


Fig. 17. Continuous Stirred Tank Reactor

The objective is to control the effluent concentration by manipulating coolant flow rate in the jacket. The process model consists of two nonlinear ordinary differential equations,

$$\frac{dC_A}{dt}=\frac{q}{V}(C_{Af}-C_A)-k_0C_Ae^{\left(-\frac{E}{RT}\right)}$$

$$\frac{dT}{dt}=\frac{q}{V}\left(T_f-T\right)+\frac{\left(-\Delta H\right)k_0C_A}{\rho C_p}e^{\left(-\frac{E}{RT}\right)}+\frac{\rho_c C_{pc}}{\rho C_p V}q_c(1-e^{\frac{-hA}{q_c\rho_c C_{pc}}})\left(T_{cf}-T\right) \quad (45)$$

where C_{Af} is feed concentration, C_A is the effluent concentration of component A, T_F, T and T_c are feed, product and coolant temperature respectively. q and q_c are feed and coolant flow rate. Here temperature T is controlled by manipulating coolant flow rate q_c. The nominal operating conditions are shown in Table 3.

$q = 100l\min^{-1}$	$E/R = 9.95\times10^3 K$
$C_{Af} = 1mol^{-1}$	$-\Delta H = 2\times10^5 calmol^{-1}$
$T_f = 350K$	$\rho,\rho_c = 1000gl^{-1}$
$T_{cf} = 350K$	$C_p, C_{pc} = 1cal\,g^{-1}K^{-1}$
$V = 100l$	$q_c = 103.41l\min^{-1}$
$hA = 7\times10^5 cal\min^{-1}K^{-1}$	$T = 440.2K$
$k_o = 7.2\times10^{10}\min^{-1}$	$C_A = 8.36\times10^{-2} mol^{-1}$

Table 3. Nominal CSTR operating conditions

The operating point in Table 3 corresponds to the lower steady state. For these conditions, there are three (two stable and one unstable) steady states. The objective is to control C_A by manipulating coolant flow rate q_c.
The corresponding model under certain assumptions is converted into transfer function form as,

$$\frac{y(s)}{u(s)}=\frac{0.42*e^{-0.75s}}{(13.4*s+1)} \quad (46)$$

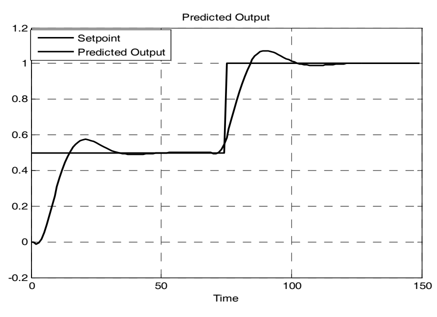


Fig. 18. System output using NGPC

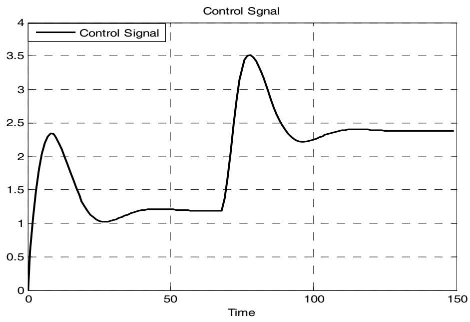


Fig. 19.Control signal for system

Fig. 18 shows the plant output for NGPC and Fig.19 shows the control efforts taken by controller. Performance evaluation of the controller is carried out using ISE and IAE criteria. Table 4 gives ISE and IAE values for NGPC implementation for nonlinear systems given by equation (46).

Systems	Setpoint	NGPC	
		ISE	IAE
System I	0.5	1.827	3.6351
	1	0.1186	1.4312

Table 4. ISE and IAE Performance Comparison of NGPC for CSTR

Example 2: NGPC for highly linear system **(dc motor)**
Here a DC motor is considered as a linear system from (Dorf & Bishop,1998). A simple model of a DC motor driving an inertial load shows the angular rate of the load, ω (*t*), as the output and applied voltage, V_{app}, as the input. The ultimate goal of this example is to control the angular rate by varying the applied voltage. Fig. 20 shows a simple model of the DC motor driving an inertial load *J*.

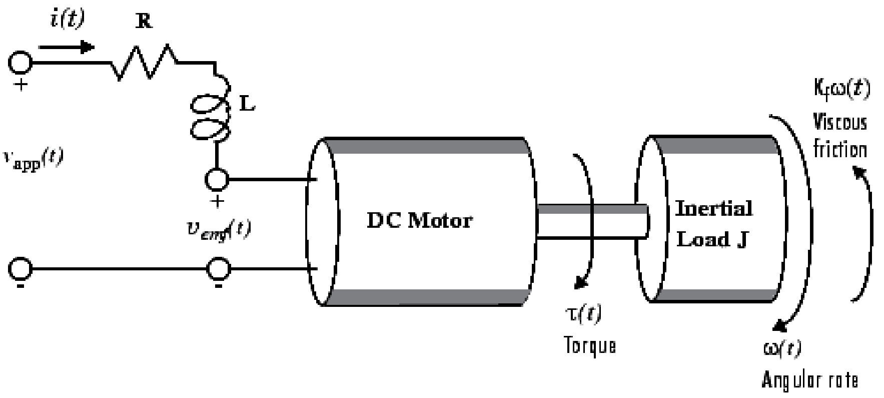


Fig. 20. DC motor driving inertial load

In this model, the dynamics of the motor itself are idealized; for instance, the magnetic field is assumed to be constant. The resistance of the circuit is denoted by R and the self-inductance of the armature by L. The important thing here is that with this simple model and basic laws of physics, it is possible to develop differential equations that describe the behavior of this electromechanical system. In this example, the relationships between electric potential and mechanical force are Faraday's law of induction and Ampere's law for the force on a conductor moving through a magnetic field.

A set of two differential equations describes the behavior of the motor. The first for the induced current, and the second for the angular rate,

$$\begin{aligned} \frac{di}{dt} &= -\frac{R}{L}\cdot i(t) - \frac{K_b}{L}\cdot \omega(t) + \frac{1}{L}\cdot V_{app} \\ \frac{d\omega}{dt} &= -\frac{K_F}{J}\omega(t) + \frac{K_m}{J}\cdot i(t) \end{aligned} \tag{47}$$

The objective is to control angular velocity ω by manipulating applied voltage, $V_{app.}$ The nominal operating conditions are shown in Table 5.

$K_b = 0.015$ (emf constant)	$K_m = 0.015$ (torque constant)
$K_f = 0.2Nms$	$J = 0.2Kgm^2 / \sec^2$
$R = 2\Omega$	$L = 0.5H$

Table 5. Nominal dc motor operating conditions

The corresponding model under certain assumptions is converted into transfer function form as,

$$\frac{y(s)}{u(s)} = \frac{1.5}{(60.31 * s + s \wedge 2)} \tag{48}$$

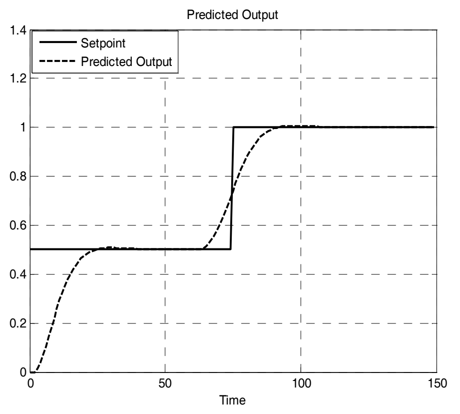


Fig. 21. System output using NGPC

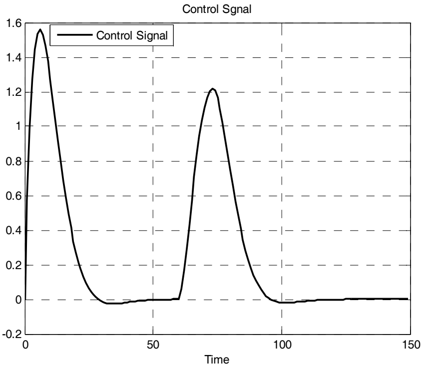


Fig. 22. Control signal for system

Fig. 21 shows the plant output for NGPC and Fig. 22 shows the control efforts taken by controller. Performance evaluation of the controller is carried out using ISE and IAE criteria. Table 6 gives ISE and IAE values for NGPC implementation for linear systems given by equation (48).

Systems	Setpoint	NGPC	
		ISE	IAE
System I	0.5	1.505	212.5
	1	1.249	202.7

Table 6. ISE and IAE Performance Comparison of NGPC for dc motor

8. Implementation of Quasi Newton Algorithm and Levenberg Marquardt Algorithm for Nonlinear System

To evaluate the performance of system two algorithms i.e. Newton Raphson and Levenberg Marquardt algorithm are implemented and their results are compared. The details about this implementation are given. The utility of each algorithm is outlined in the conclusion. In using Levenberg Marquardt algorithm, the number of iteration needed for convergence is significantly reduced from other techniques. The main cost of the Newton Raphson algorithm is in the calculation of Hessain, but with this overhead low iteration numbers make Levenberg Marquardt algorithm faster than other techniques and a viable algorithm for real time control. The simulation result of Newton Raphson and Levenberg Marquardt algorithm are compared. Levenberg Marquardt algorithm shows a convergence to a good solution. The performance comparison of these two algorithms also given in terms of ISE and IAE.

8.1 Simulation Results

Many physical plants exhibit nonlinear behavior. Linear models may approximate these relationships, but often a nonlinear model is desirable. This Section presents training a neural network to model a nonlinear plant and then using this model for NGPC. The

Duffing's equation is well-studied nonlinear system as given in equation (43). The Newton Raphson algorithm and Levenberg Marquardt algorithm has been implemented for the system in equation (43) and results are compared. Fig.23 shows Newton Raphson implementation and Fig. 24 shows implementation of LM algorithm. Fig. 25. Shows the control efforts taken by controller.

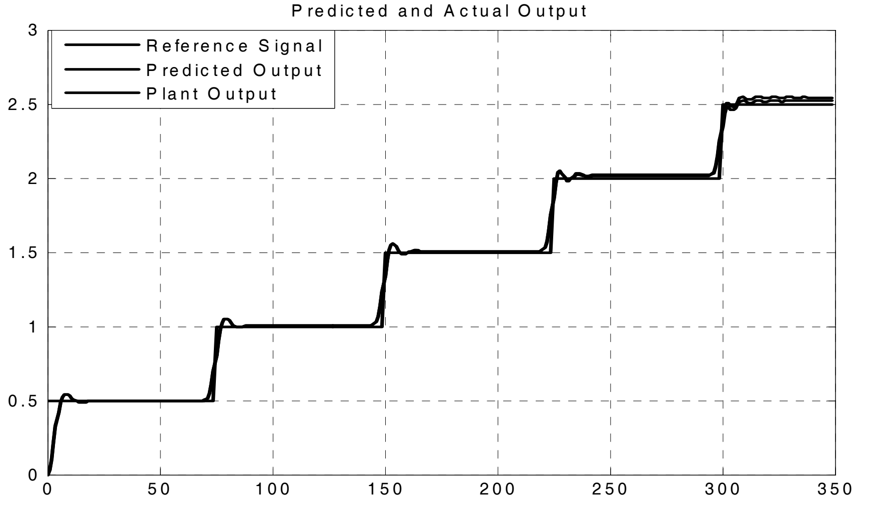


Fig. 23. Predicted Output and Actual Plant Output for Newton Raphson implementation

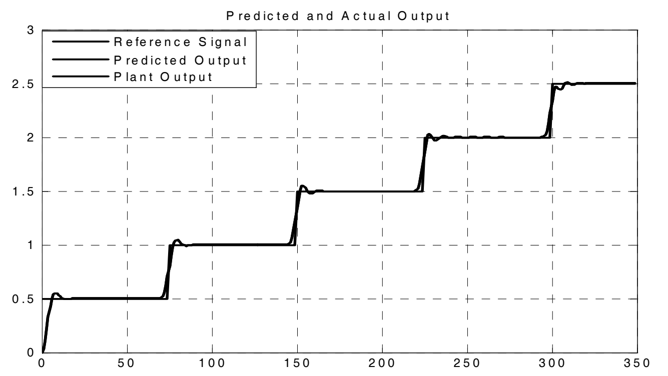


Fig. 24. Predicted Output and Actual Plant Output for Levenberg Marquardt implementation

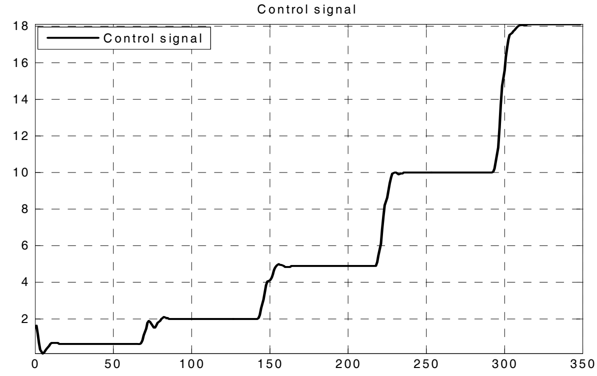


Fig. 25. Control signal for system

Set point	Newton Raphson		Levenberg Marquardt	
	ISE	IAE	ISE	IAE
0.5	0.8135	2.7950	0.8084	2.5537
1	0.1363	1.2961	0.1283	1.0315
1.5	0.1292	1.2635	0.1193	0.9931
2	0.1448	2.1251	0.1026	0.8791
2.5	0.0986	2.0367	0.0382	0.5326

Table 7. ISE and IAE Performance Comparison of Newton Raphson and Levenberg Marquardt algorithm

9. Conclusion

In this chapter we have combined predictive control and neural network to form a control strategy known as Neural Generalized Predictive Control (NGPC) The NGPC algorithm operates in two modes, i.e. prediction and control. It generates a sequence of future control signals within each sampling interval to optimize control effort of the controlled systems. The GPC and NGPC are applied to the variety of systems to test its capability. The output of trained neural network is used as the predicted output of the plant. This predicted output is used in the cost function minimization algorithm. GPC criterion is minimized using two different schemes: a Quasi Newton algorithm and Levenberg Marquardt algorithm. The performance comparison of these configurations has been given in terms of Integral square error (ISE) and Integral absolute error (IAE). The simulation result also reveals that, Levenberg Marquardt gives improved control performance over Newton Raphson optimization algorithm.The performance of NGPC is also tested on a highly nonlinear and linear industrial process such as continues stirred tank reactor (CSTR) and DC Motor. The simulation results show the efficacy of NGPC over GPC for controlling linear as well as nonlinear plants.

10. References

Chen, Z. ;Yuan, Z. & Yan, Z.(2002). "A survey on neural network based nonlinear predictive control", *Engineering of Control China,* Vol. 9, No. 4, pp. 7-11.

Clarke, D. W.; Mohtadi, C.& Tuffs, P. C.(1987). Generalized predictive control-part-I and Part-II the basic algorithms, *Automatica,* Vol. 23, pp. 137-163.

Dorf, R. C. and Bishop, R. H. (1998). Modern control systems, *Addison-Wesley, Menlo Park, CA, USA.*

Hashimoto, S. ; Goka, S.; Kondo, T. & Nakajima, K., (2008). Model predictive control of precision stages with nonlinear friction, *Advanced Intelligent Mechatronics*,.International Conference IEEE/ASME., Gunma Univ., Kiryu.

Kim J. H.; Ansung,D. & Gyonggi, D.(1998).Fuzzy model based predictive control, *IEEE Transactions on Fuzzy Systems,* pp. 405-409.

Mathworks USA,(2007) ,MATLAB 7.0.1 Software, Mathworks Natic, USA.

Nahas, E. P.; Henson, M. A. & Seborg, D. E. (1992). Nonlinear internal model control strategy for neural network models, *Computers Chemical Engineering*, Vol. 16, pp. 1039-1057.

Norgaard, M. (2000). Neural network based control system design toolkit Ver. 2.0" Technical Report, 00-E -892, *Department of Automation, Technical University of Denmark.*

Nørgaard,M.(2004),*Neural Network Based System Identification Toolbox*," Tech Report. 00-E-891, Department of Automation, Technical University of Denmark.

Piche, S.; Sayyar-Rodsari,B.; Johnson,D.; Gerules, M.; Technol, P. & Austin, T. X(2000). Nonlinear model predictive control using neural network , *IEEE Control Systems Magazine,* Vol. 20, No. 3, pp. 53-62.

Qin, S. J. and. Badgwell, T.(2003).A Survey of Industrial model predictive control technology", *Control Engineering Practice,* Vol. 11, No. 7, pp. 733-764.

Raff, T. ; Sinz, D. & Allgower, F. (2008). Model predictive control of uncertain continuous-time systems with piecewise constant control input: A convex approach, *American Control conference: Inst. for Syst. Theor. & Autom. Control,* Univ. of Stuttgart, Stuttgart;

Rao, D. N.;Murthy, . M. R. K,; Rao, . S. R. M. & Harshal D. N. ,(2006). Neural generalized predictive control for real time application, *International Journal of Information and Systems Sciences,* Vol. 3, No. 1, pp. 1-15.

Rao, D. N. ; Murthy, M. R. K. ; Rao, S. R. M. &. Harshal, D. N (2007). Computational comparisons of GPC and NGPC schemes", *Engineering Letters*, Vol. 14.

Rossiter, A.(*2006*).Recent Developments in predictive control, *UKACC Control,* Mini symposia, pp. 3-26.

Sorensen, P. H.; Norgaard, O. Ravn, & N. K. Poulsen,(1999). Implementation of neural network based nonlinear predictive control, *Neurocomputing,* Vol. 28, 1999, pp. 37-51.

Soloway, D. & Haley, P. J. (1997). Neural generalized predictive control- A Newton-Raphson implementation", *NASA Technical Report (110244).*

Sun, X.; Chang, R.; He, P. & Fan,Y.(2002). Predictive control based on neural network for nonlinear system with time delays, *IEEE Transactions on Neural Network,* pp. 319-322.

13

Forecasting, Diagnosis and Decision Making with Neural Networks and Self-Organizing Maps

Kazuhiro Kohara, Katsuyoshi Aoki and Mamoru Isomae
Chiba Institute of Technology
Japan

1. Introduction

Intelligent techniques such as back-propagation neural networks (BPNN) (Rumelhart et al., 1986), self-organizing maps (SOM) (Kohonen, 1995) , decision trees (Quinlan, 1993) and Bayesian networks (Jensen, 2001) have been extensively investigated, and various attempts have been made to apply them to identification, prediction and control (e.g., (Bishop, 1995); (Kil & Shin, 1996); (Pham & Liu, 1995)). This chapter describes three topics: (1) forecasting with BPNN and *selective learning*, (2) diagnosis with SOM and *ensemble learning*, and (3) decision making with SOM and Analytic Hierarchy Process (AHP) (Saaty, 1980).

The first section describes stock market prediction with BPNN and selective learning techniques for improving the ability of BPNN to predict large changes. *Selective-presentation* approach, in which the training data corresponding to large changes in the prediction-target time series are presented more often, *selective-learning-rate* approach, in which the learning rate for training data corresponding to small changes is reduced, and combining two approaches are described. The prediction of daily stock prices is used as a noisy real-world problem. The results of several experiments on stock-price prediction showed that the performances of these two approaches were similar and both better than the usual presentation approach, and combining them further improved the performance.

The second section shows an experimental study on medical diagnosis with SOM and ensemble learning. We apply SOM to medical diagnosis such as breast cancer, heart disease and hypothyroid diagnosis, comparing with decision trees and Bayesian networks. We used the UCI data sets as medical diagnosis problem. The accuracy for breast cancer and hypothyroid diagnosis was comparatively high and the accuracy for heart disease diagnosis was comparatively low. Therefore, we apply ensemble learning such as *bagging* and *boosting* of SOM to heart disease diagnosis. The accuracy with ensemble learning of SOM was much improved.

The third section describes purchase decision making with SOM and AHP. We proposed a purchase decision making method using SOM and AHP. First, we divide many products into several clusters using SOM. Secondly, we select some alternatives using the product maps. Finally, we make a final choice from the alternatives using AHP. As an example of real-world applications, we apply our method to a buying personal computer (PC) problem.

We considered one hundred and twenty kinds of notebook PCs. We evaluated our method through experiments conducted by 117 people and confirmed its effectiveness.

2. Forecasting with Neural Networks and Selective Learning

Prediction using back-propagation neural networks has been extensively investigated (e.g., (Weigend et al., 1990) ; (Vemuri & Rogers, 1994) ; (Mandic & Chambers, 2001)), and various attempts have been made to apply neural networks to financial market prediction (e.g., (Azoff, 1994) ; (Refenes & Azema-Barac, 1994) ; (White, 1988) ; (Baba & Kozaki, 1992) ; (Freisleben, 1992) ; (Tang et al., 1991) ; (Kohara, 2002)), electricity load forecasting (e.g., (Park et al., 1991)) and other areas. In the usual approach, all training data are equally presented to a neural network (i.e., presented in each cycle) and the learning rates are equal for all the training data independently of the size of the changes in the prediction-target time series. Also, network learning is usually stopped at the point of minimal mean squared error between the network's outputs and the desired outputs.

Generally, the ability to predict large changes is more important than the ability to predict small changes, as we mentioned in the previous paper (Kohara, 1995). When all training data are presented equally with an equal learning rate, the BPNN will learn the small and large changes equally well, so it cannot learn the large changes more effectively. We have investigated selective learning techniques for improving the ability of neural networks to predict large changes. We previously proposed the *selective-presentation* (Kohara, 1995) and *selective-learning-rate* (Kohara, 1996) approaches and applied them into stock market prediction. In the *selective-presentation* approach, the training data corresponding to large changes in the prediction-target time series are presented more often. In the *selective-learning-rate* approach, the learning rate for training data corresponding to small changes is reduced. The previous paper (Kohara, 1995) also investigated another stopping criterion for financial predictions. Network learning is stopped at the point having the maximum profit through experimental stock-trading.

We also previously proposed combining the *selective-presentation* and *selective-learning-rate* approaches (Kohara, 2008). By combining these two approaches, we can easily achieved fine-tuned and step-by-step selective learning of neural networks according to the degree of change. Daily stock prices were predicted as a noisy real-world problem.

2.1 Selective-Presentation and Selective-Learning-Rate Approaches

To allow neural networks to learn about large changes in prediction-target time series more effectively, we separate the training data into large-change data (L-data) and small-change data (S-data). L-data (S-data) have next-day changes that are larger (smaller) than a preset value. In the *selective-presentation* approach, the L-data are presented to neural networks more often than S-data. For example, all training data are presented every fifth learning cycle, while the L-data are presented every cycle. In the *selective-learning-rate* approach, all training data are presented in every cycle; however, the learning rate of the back-propagation training algorithm for S-data is reduced compared with that for L-data. These two approaches are outlined as follows.

Selective-Presentation **Approach**

1. Separate the training data into L-data and S-data.

2. Train back-propagation networks with more presentations of L-data than of S-data.

3. Stop network learning at the point satisfying a certain stopping criterion (e.g., stop at the point having the maximum profit).

***Selective-Learning-Rate* Approach**

1. Separate the training data into L-data and S-data.

2. Train back-propagation networks with a lower learning rate for the S-data than for the L-data.

3. Stop network learning at the point satisfying a certain stopping criterion (e.g., stop at the point having the maximum profit).

We combined these two approaches to achieve fine-tuned and step-by-step learning of neural networks according to the degree of change. The outline is as follows.

Combining *Selective-Presentation* and *Selective-Learning-Rate* Approaches

1. Separate the training data into L-data and S-data.

2. Separate L-data into two subsets: L1-data and L2-data, where changes in L2- data are larger than those in L1-data.

3. Separate S-data into two subsets: S1-data and S2-data, where changes in S2-data are larger than those in S1-data.

4. Train back-propagation networks with more presentations of L1- and L2-data than of S1- and S2-data, and with a lower learning rate for L1- and S1-data than for L2 and S2-data.

5. Stop network learning at the point satisfying a certain stopping criterion (e.g., stop at the point having the maximum profit).

2.2 Evaluation through Experimental Stock-Price Prediction

We considered the following types of knowledge for predicting Tokyo stock prices. These types of knowledge involve numerical economic indicators (Kohara, 1995).

1. If interest rates decrease, stock prices tend to increase, and vice versa.

2. If the dollar-to-yen exchange rate decreases, stock prices tend to decrease, and vice versa.

3. If the price of crude oil increases, stock prices tend to decrease, and vice versa.

We used the following five indicators as inputs to the neural network.

- TOPIX: the chief Tokyo stock exchange price index
- EXCHANGE: the dollar-to-yen exchange rate (yen/dollar)
- INTEREST: an interest rate (3-month CD, new issue, offered rates) (%)
- OIL: the price of crude oil (dollars/barrel)
- NY: New York Dow-Jones average of the closing prices of 30 industrial stocks (dollars)

TOPIX was the prediction target. EXCHANGE, INTEREST and OIL were chosen based on the knowledge of numerical economic indicators. The Dow-Jones average was used because Tokyo stock market prices are often influenced by New York exchange prices. We assume that tomorrow's change in TOPIX is determined by today's changes in the five indicators according to the knowledge. Therefore, the daily changes in these five indicators (e.g. Δ TOPIX(t) = TOPIX(t) - TOPIX(t-1)) were input into neural networks, and the next-day's change in TOPIX was presented to the neural network as the desired output (Fig. 1). The back-propagation algorithm was used to train the network. All the data of the daily changes were scaled to the interval [0.1, 0.9]. A 5-5-1 multi-layered neural network was used (five neurons in the input layer, five in the hidden layer, and one in the output layer).

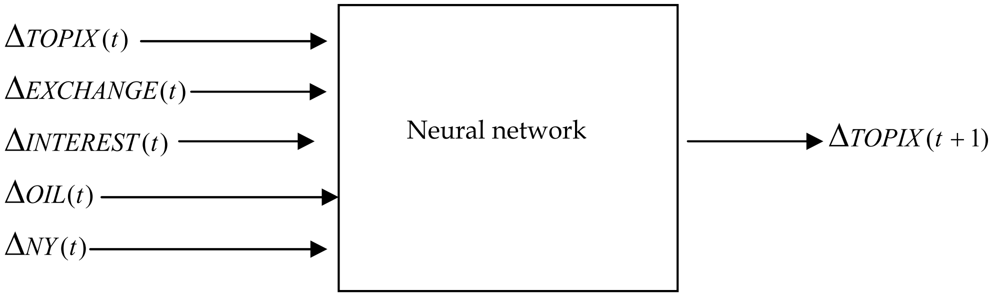


Fig. 1. Neural prediction model

2.3 Evaluation Experiments

We used data from a total of 409 days (from August 1, 1989 to March 31, 1991): 300 days for training, 30 days for validation (making decisions on stopping the network learning), and 79 days for making predictions. In Experiment 1, all training data were presented in each cycle with an equal learning rate (ε = 0.7). In Experiment 2, L-data were presented five times as often as S-data. Here, the large-change threshold was 14.78 points (about US$ 1.40), which was the median of absolute value of TOPIX daily changes in the training data. In Experiment 3, the learning rate for the S-data was reduced up to 20% (i.e., ε = 0.7 for the L-data and ε = 0.14 for the S-data). In each experiment, network learning was stopped at the point having the maximum profit (the learning was stopped at the point having the maximum profit for the validation data during 8000 learning cycles). The prediction error and profit were monitored after every hundred learning cycles.

When a large change in TOPIX was predicted, we tried to calculate "Profit" as follows: when the predicted direction was the same as the actual direction, the daily change in TOPIX was earned, and when it was different, the daily change in TOPIX was lost. This calculation of profit corresponds to the following experimental TOPIX trading system. A buy (sell) order is issued when the predicted next-day's up (down) in TOPIX is larger than a preset value which corresponds to a large change. When a buy (sell) order is issued, the system buys (sells) TOPIX shares at the current price and subsequently sells (buys) them back at the next-day price. Transaction costs on the trades were ignored in calculating the profit. The more accurately a large change is predicted, the larger the profit is.

In each experiment, the momentum parameter α was 0.7. All the weights and biases in the neural network were initialized randomly between -0.3 and 0.3. In each experiment the neural network was run four times for the same training data with different initial weights and the average was taken.

The experimental results are shown in Table 1. Multiple regression analysis (MR) was also used in the experiments. The "prediction error on large-change test data" is the mean absolute value of the prediction error for the test L-data. Applying our *selective-presentation* approach (Experiment 2) reduced the prediction error for test L-data and improved profits: the prediction-error on L-data was reduced by 7% (1- (21.3/22.9)) and the network's ability to make profits through experimental TOPIX-trading was improved by 30% (550/422) compared with the results obtained with the usual presentation approach (Experiment 1). The prediction error and profits in Experiment 3 (*selective-learning-rate* approach) were comparable to those in Experiment 2 (*selective-presentation* approach). Combining *selective-presentation* with *selective-learning-rate* approaches further reduced the prediction error for test L-data and improved profits: the prediction-error was reduced by 10% and the network's ability to make profits was improved by 38% (Kohara, 2008).

	MR	Experiment 1	Experiment 2	Experiment 3
Presentation method	equal	equal	*selective*	equal
Learning rate		equal	equal	*selective*
Prediction error for large-change data (relative value)	24.3 (1.06)	22.9 (1)	21.3 (0.93)	21.3 (0.93)
Profit on test data (relative value)	265 (0.62)	422 (1)	550 (1.30)	563 (1.33)

Table 1. Experimental results on daily stock price prediction

2.4 Summary of Section 2

We described selective learning techniques for forecasting. In the first approach, training data corresponding to large changes in the prediction-target time series are presented more often, in the second approach, the learning rate for training data corresponding to small changes is reduced, and in the third approach, these two techniques are combined. The results of several experiments on stock-price prediction showed that the performances of these two approaches were similar and both better than the usual presentation approach, and combining them further improved the performance. Next, we will apply these techniques today's stock market and other real-world forecasting problems. We also plan to develop a forecasting method that integrates statistical analysis with neural networks.

3. Diagnosis with Self-Organizing Maps and Ensemble Learning

We applied decision trees, Bayesian networks and SOM to medical diagnosis such as breast cancer, heart disease and hypothyroid diagnosis. The UCI data sets (Merz et al., 1997) were used as medical diagnosis problem. The accuracy of breast cancer and hypothyroid diagnosis was comparatively high and the accuracy of heart disease diagnosis was comparatively low. Therefore, we applied ensemble learning such as *bagging* (Breiman, 1994) and *boosting* (Schapire, 1990) of SOM to heart disease diagnosis. The accuracy with ensemble learning of SOM was much improved.

3.1 Medical Diagnosis with Decision Trees, Self-Organizing Maps, and Bayesian Networks

Viscovery SOMine 4.0 was used as SOM software and *See5 release 1.19* was used as decision tree software with default parameter values. We used *BayoNet 3.0.7* as Bayesian network software and constructed Bayesian networks semi-automatically. All problems were evaluated with 10-fold cross-validation. Experimental results on accuracy for medical diagnosis are shown in Table 2. Accuracy for breast cancer diagnosis (breast-w) was comparatively high and accuracy for heart disease diagnosis (heart-c) was comparatively low. In these problems, the accuracy with SOM was comparatively high and the accuracy with decision trees was comparatively low.

	Decision trees	SOM	Bayesian networks
Software	See5 release 1.19	Viscovery SOMine 4.0	BayoNet 3.0.7
Breast-w	95.3%	99.1%	96.0%
Heart-c	72.7%	89.4%	80.6%

Table 2. Experimental results on accuracy for medical diagnosis (1)

Therefore, we apply SOM and decision trees to hypothyroid data. All problems were evaluated with 10-fold cross-validation. Experimental results on accuracy are shown in Table 3. The accuracy for hypothyroid diagnosis was comparatively high. In this problem, the accuracy with decision trees was better than that with SOM.

	Decision trees	SOM
Software	See5 release 1.19	Viscovery SOMine 4.0
Hypothyroid	99.5%	96.8%

Table 3. Experimental results on accuracy for medical diagnosis (2)

3.2 Related Work

Tsoumakas et al. dealt with the combination of classification models that have been derived from running different (heterogeneous) learning algorithm on the same data set (Tsoumakas et al., 2004). They used WEKA implementations of the 10 base-level classification algorithms: decision tree (C4.5), support vector machine (SVM), naïve Bayes, k nearest neighbor, radial basis function and so on. Experimental results on accuracy for the same medical data as we used are shown in Table 4. Both worst and best results stated in the reference are shown.

Zhang & Su extended decision trees to represent a joint distribution and conditional independence, called conditional independence trees (CITrees) and reported that the CITree algorithm outperforms C4.5 and naïve Bayes significantly in classification accuracy (Zhang & Su, 2004). Their experimental results on accuracy for the same medical data as we used are also shown in Table 4.

Garcia-Pedrajas et al. presented a new approach to crossover operator in genetic evolution of neural networks and reported that their approach was compared to a classical crossover with an excellent performance (Garcia-Pedrajas et al., 2006). Their experimental results on accuracy for the same medical data as we used are also shown in Table 4.

Radivojac et al. investigated the problem of supervised feature selection within the filtering framework (Radivojac et al., 2004). In their approach, applicable to the two-class problems, the feature strength is inversely proportional to the p-value of the null hypothesis. Their experimental results for accuracy of heart disease data performed using naïve Bayes and SVM are shown in Table 4.

Robnik-Sikonja improved random forests (Breiman, 2001) and the experimental results on accuracy for breast cancer data are shown in Table 4 (Robnik-Sikonja, 2004). As mentioned above, the accuracy for breast cancer was high up to 99.5% (Garcia-Pedrajas et al., 2006) and the accuracy for hypothyroid was also high up to 99.7% (Tsoumakas et al., 2004). On the other hand, the accuracy for heart disease was comparatively low. The accuracy was 89.4% at most which was attained by SOM.

Reference	Tsoumakas et al., 2004	Zhang & Su, 2004	Garcia-Pedrajas et al., 2006
Classifier	Voting	DT, NB	NN, GA
Breast-w	97.0% to 98.0%	94.3% to 97.1%	93.8% to 99.5%
Heart-c	81.8% to 88.4%	78.1% to 84.4%	86.7% to 89.1%
Hypothyroid	97.2% to 99.7%	93.1% to 93.2%	94.4% to 94.9%

Reference	Radivojac et al., 2004	Robnik-Sikonja, 2004	This chapter
Classifier	NB, SVM	RF	SOM
Breast-w	Not available	96.6% to 96.7%	99.1%
Heart-c	83.5% to 83.9%	Not available	89.4%
Hypothyroid	Not available	Not available	96.8%

Voting: voting of heterogeneous classifiers including DT, NB, SVM etc.,
DT: decision trees, NB: naïve Bayes, SVM: support vector machine,
NN: neural networks, GA: genetic algorithm, RF: random forest.
Table 4. Experimental results on accuracy for medical diagnosis (3)

Quinlan investigated ensemble learning such as bagging and boosting of C4.5 (Quinlan, 1996). Experimental results on accuracy for the same medical data as we used are shown in Table 5. Bagging and boosting improved average of accuracy by about 1%. Therefore, we have decided to apply ensemble learning of SOM to heart disease data.

	C4.5	Bagged C4.5	Boosted C4.5
Breast-w	94.7%	95.8%	95.9%
Heart-c	77.0%	78.5%	78.6%
Hypothyroid	99.5%	99.6%	99.6%
Average	90.4%	91.3%	91.4%

Table 5. Experimental results on accuracy for medical diagnosis (4)

3.3 Ensemble Learning of Self-Organizing Maps

The steps of bagging and boosting for 10-fold cross-validation we used are as follows.

Bagging

Step 1: Divide all data into ten sets.
Step 2: Select one set for test data and use the remains for training set.
Step 3: Resample training subset from the training set mentioned in Step 2.
Step 4: Train SOM using the training subset resampled in Step 3.
Step 5: Repeat Step 3 to Step 4 nine times more and train the other nine SOM.
Step 6: Input test data to ten SOM and obtain results with mean summed outputs by ten SOM.
Step 7: Repeat Step 2 to Step 6 nine times more and obtain average accuracy for test data.

Boosting

Step 1: Divide all data into ten sets.
Step 2: Select one set for test data and use the remains for training set.
Step 3: Train SOM using training set, input test data to the SOM and obtain results with outputs by the SOM.

Step 4: Duplicate the misclassified data in Step 3 to the training set used in Step 3 and obtain modified training set.
Step 5: Repeat Step 3 to Step 4 nine times more to the other nine SOM.
Step 6: Input test data to ten SOM and obtain results with mean summed outputs by ten SOM.
Step 7: Repeat Step 2 to Step 6 nine times more and obtain average accuracy for test data.
We resampled 60% data of the training set in the Step 3 of bagging. When there were not misclassified data in the Step 5 of boosting, we didn't repeat Step 3 to Step 4 and the number of SOM in the Step 6 of boosting was not always ten.
Experimental results on accuracy for heart disease data with ensemble learning of SOM are shown in Table 6. Bagging and boosting improved accuracy by 1.7% and 5.0%, respectively. Especially, boosted SOM was very effective in heart disease data and its accuracy reached 94.4% which was better than that described in references mentioned above by about 5 to 10%.

	SOM	Bagged SOM	Boosted SOM
Heart-c	89.4%	91.1%	94.4%

Table 6. Experimental results on accuracy for medical diagnosis (5)

3.4 Summary of Section 3

We applied SOM to medical diagnosis such as breast cancer, heart disease and hypothyroid diagnosis, comparing with decision trees and Bayesian networks. We found that their accuracy of breast cancer and hypothyroid diagnosis was comparatively high and their accuracy of heart disease diagnosis was comparatively low. Then, we applied ensemble learning such as bagging and boosting of SOM to heart disease diagnosis. We found that their accuracy was much improved. Next, we will apply SOM and ensemble learning to the other medical diagnosis of the UCI data sets such as lung cancer and diabetes. We also plan to develop a diagnosis method that integrates decision trees and Bayesian networks with SOM.

4. Decision Making with Self-Organizing Maps and Analytic Hierarchy Process

According to Kotler (Kotler, 2002), marketing scholars have developed a *stages model* of the buying decision process. The consumer passes through five stages: problem recognition, information search, evaluation of alternatives, purchase decision, and postpurchase behavior. Five successive sets are involved in consumer decision making. The first set is the *total set* of brands available to the consumer. The individual consumer will come to know only a subset of these brands (*awareness set*). Some brands will meet initial buying criteria (*consideration set*). As the person gathers more information, only a few will remain as strong contenders (*choice set*). The brands in the choice set might all be acceptable. The person makes a final choice from this set.
Several intelligent decision support systems (DSS) have been proposed to aid in a variety of problems (see Table 7). We proposed a purchase decision support method using SOM and

AHP (Kohara & Isomae, 2006). First, we divide many products (*total set*) into several clusters using SOM. Secondly, we select some alternatives (*choice set*) using the product maps. Finally, we make a final choice from the alternatives using AHP. As an example of real-world applications, we apply our method to a buying personal computer (PC) problem. We considered one hundred and twenty kinds of notebook PCs. We evaluated our method through experiments conducted by 117 people and confirmed its effectiveness.

Application	Approach	Reference
Bankruptcy prediction	CBR and AHP	Park & Han, 2002
Weather prediction	Fuzzy reasoning	Riordan & Hansen, 2002
Economic prediction	Neural networks	Kohara, 2002
Electricity business planning	Knowledge-based system	Ha et al., 2003
Economic market model	Multi-agent system	Walle & Moldovan, 2003
Clinical decision support	AHP	Suka et al., 2003
Purchase decision support	SOM and AHP	Kohara & Isomae, 2006

Table 7. Examples of intelligent decision support systems

4.1 Self-Organizing Maps of Personal Computers

The SOM algorithm is based on unsupervised, competitive learning. It provides a topology preserving mapping from the high dimensional space to map units. Map units, or neurons, usually form a two-dimensional lattice and thus the mapping is a mapping from high dimensional space onto a plane. The property of topology preserving means that the mapping preserves the relative distance between the points. Points that are near each other in the input space are mapped to nearby map units in the SOM. The SOM can thus serve as a cluster analyzing tool of high-dimensional data. We considered one hundred and twenty kinds of notebook PCs which were sold in Japan on June 2004. We clustered these PCs using the following features: CPU speed (GHz), main memory capacity (MB), HDD storage capacity (GB), weight (kg), price (yen), battery life (hours), and so on.

We used the above features in two ways: continuous and classified data input. For classified data of CPU speed, we divide into three classes: under 1, 1 to 2, and over 2 GHz. For classified data of main memory capacity, we divide into two classes: 256 and 512 MB. For classified data of HDD storage capacity, we divide into three classes: under 40, 40 to 60, and over 60 GB. For classified data of weight, we divide into five classes: under 1, 1 to 2, 2 to 3, 3 to 4, and over 4 Kg. For classified data of price, we divide into six classes: under 100, 100 to 150, 150 to 200, 200 to 250, 250 to 300, and over 300 thousand yen. For classified data of battery life, we divide into six classes: under 1, 1 to 2, 2 to 3, 3 to 4, 4 to 5, and over 5 hours.

Viscovery SOMine 4.0 was used as SOM software. Fig. 2 and Fig. 3 show self-organizing map and an example of component map of PCs with classified data inputs, respectively. Fig. 4 and Fig. 5 show self-organizing map and an example of component map of PCs with continuous data inputs, respectively. There were five clusters in Fig. 2. When inspecting component maps, the feature of each cluster is clear. For example, when inspecting "under 1 GHz (1-GHz)" component map (see Fig. 3), we understand that one of the features of Cluster 5 is that CPU speed is under 1 GHz. In Fig. 3, originally red color (here, black) neurons correspond to under 1 GHz class and originally blue color (here, dark grey) neurons correspond to the other class.

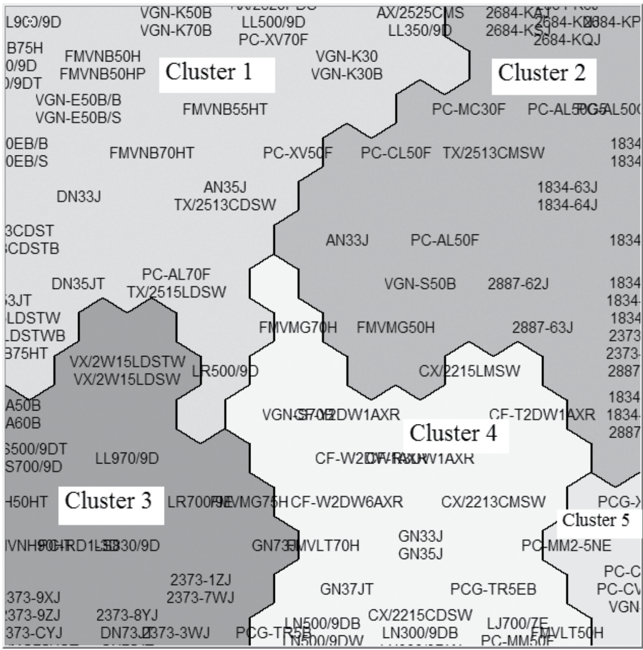


Fig. 2. Self-organizing map of PCs with classified data inputs

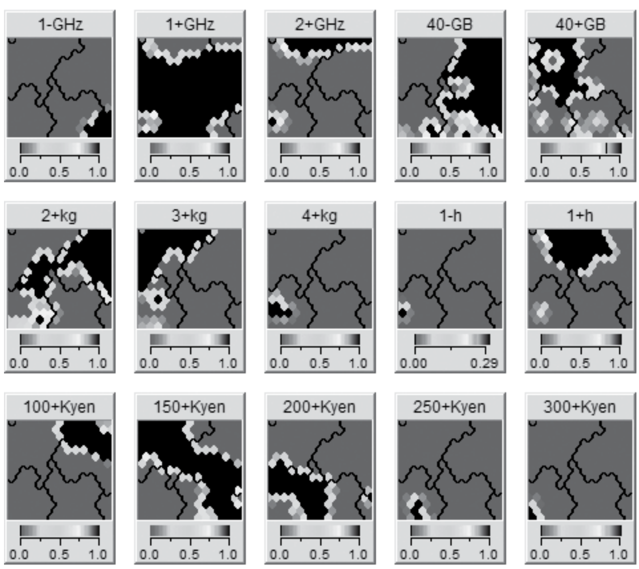


Fig. 3. Component maps of PCs with classified data inputs

There were four clusters in Fig. 4. In CPU (GHz) component map of Fig. 5, originally red (here, black) neurons correspond to 2.6 and more GHz and originally blue (here, dark grey) neurons correspond to 0.9 GHz CPU speed. Originally green and yellow (here, light grey) neurons correspond to intermediate values of CPU speed. When inspecting CPU component map of Fig. 5, the feature of each cluster is not clear. So, classified data input is better than continuous data input for clustering PCs. From now, we used classified data input only. We inspected every component map and understand that features of Cluster 1 to Cluster 5 are as in Table 8.

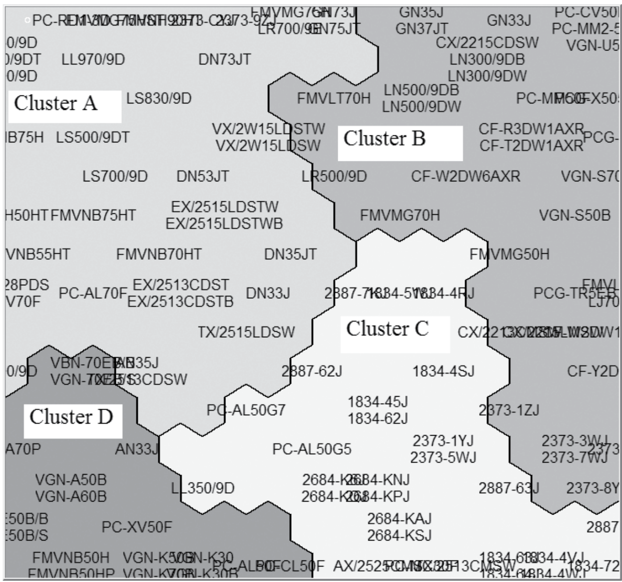


Fig. 4. Self-organizing map of PCs with continuous data inputs

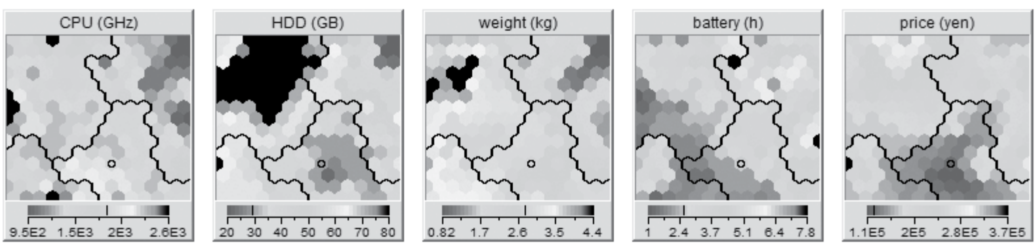


Fig. 5. Component maps of PCs with continuous data inputs

	Features	Main feature
Cluster 1	1 to 2 GHz (CPU), 40 to 60 GB (HDD), 3 to 4 Kg (weight), 150 to 200 thousand yen (price)	High performance
Cluster 2	under 40 GB (HDD), 256 MB (main memory), 100 to 150 thousand yen (price)	Low performance and low price
Cluster 3	over 60 GB (HDD), 512 MB (main memory), over 200 thousand yen (price)	Highest performance and high price
Cluster 4	1 to 2 Kg (weight), over 4 hours (battery life)	High mobility
Cluster 5	under 1 GHz (CPU), under 1 Kg (weight), 150 to 200 thousand yen (price)	Small size

Table 8. Features of clusters with classified data inputs

4.2 Decision Making with Analytic Hierarchy Process

AHP is a multi-criteria decision method that uses hierarchical structures to represent a problem and develop priorities for alternatives based on the user. Saaty (Saaty, 1980) has shown that weighting activities in multi-criteria decision-making can be effectively dealt with via hierarchical structuring and pairwise comparisons. Pairwise comparisons are based on forming judgment between two particular elements rather than attempting to prioritize an entire list of elements.

Fig. 6 shows the AHP model for a buying PC problem. Towards the goal on the first level (i.e. buying personal computer problem), four criteria in the second level and five alternatives on the third level were defined. Here, we used the following four criteria: low price, high mobility, high performance, and preference of design. Here, high mobility means small weight and long battery life. High performance means high CPU speed, large main memory capacity and large HDD storage capacity. We can select some alternatives using the PC maps in various ways. Here, we selected five alternatives using *neighborhood view* function of *Viscovery SOMine* software (this function displays all nodes that are topological similar to a reference node) and conducted AHP (see Tables 9 to 14). In this case, price is most important (see Table 9) and PC 2 is selected as final choice (see Table 14).

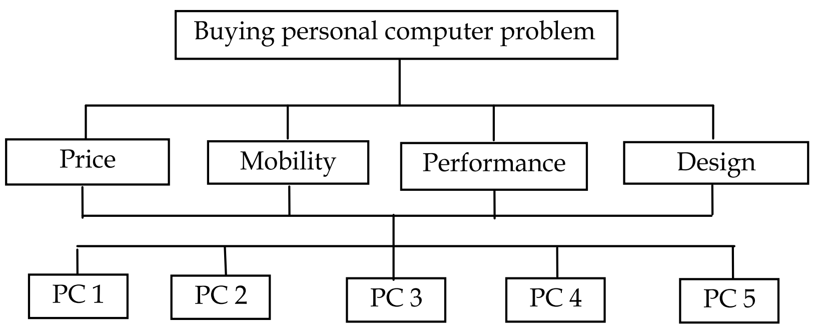


Fig. 6. AHP model for a buying PC problem

	Price	Mobility	Performance	Design	Weights
Price	1	7	3	5	0.565
Mobility	1/7	1	1/5	1/3	0.055
Performance	1/3	5	1	3	0.262
Design	1/5	3	1/3	1	0.118

Consistency index = 0.039

Table 9. Pairwise comparison matrix among four criteria

	PC 1	PC 2	PC 3	PC 4	PC 5	Weights
PC 1	1	1/5	1/4	1/4	1	0.063
PC 2	5	1	3	3	5	0.456
PC 3	4	1/3	1	1	4	0.209
PC 4	4	1/3	1	1	4	0.209
PC 5	1	1/5	1/4	1/4	1	0.063

Consistency index =0.031

Table 10. Pairwise comparison for price

	PC 1	PC 2	PC 3	PC 4	PC 5	Weights
PC 1	1	5	5	1	1/3	0.221
PC 2	1/5	1	1	1/5	1/5	0.056
PC 3	1/5	1	1	1/5	1/5	0.056
PC 4	1	5	5	1	1/3	0.221
PC 5	3	5	5	3	1	0.448

Consistency index =0.049

Table 11. Pairwise comparison for mobility

	PC 1	PC 2	PC 3	PC 4	PC 5	Weights
PC 1	1	2	2	1/2	1/2	0.186
PC 2	1/2	1	1	1/2	1/2	0.121
PC 3	1/2	1	1	1/2	1/2	0.121
PC 4	2	2	2	1	1/2	0.246
PC 5	2	2	2	2	1	0.326

Consistency index = 0.034

Table 12. Pairwise comparison for performance

	PC 1	PC 2	PC 3	PC 4	PC 5	Weights
PC 1	1	3	3	1/3	5	0.247
PC 2	1/3	1	1	1/5	3	0.104
PC 3	1/3	1	1	1/5	3	0.104
PC 4	3	5	5	1	6	0.497
PC 5	1/5	1/3	1/3	1/6	1	0.048

Consistency index =0.040

Table 13. Pairwise comparison for design

Alternatives	Results (rank)
PC 1	0.126 (5)
PC 2	0.305 (1)
PC 3	0.165 (3)
PC 4	0.253 (2)
PC 5	0.151 (4)

Table 14. Final results of AHP for a buying PC problem

4.3 Evaluation Experiments

We evaluated our decision support method by experimental purchase decision making with SOM and AHP (Kohara & Isomae, 2006). The number of people conducting evaluation experiments is 117. They were undergraduate students of the Department of Electrical, Electronics and Computer Engineering, Chiba Institute of Technology. The above buying PC problem was used. After the experiments, we asked subjects some questions. 89.7% subjects answered that self-organizing map of PCs is *extremely useful* or *useful* for selecting some alternatives. 85.4% subjects answered that AHP is *extremely useful* or *useful* for making a final choice from the alternatives and 91.5% subjects consent with the result.

4.4 Summary of Section 4

We described a purchase decision support method using SOM and AHP. First, we divide many products into several clusters using SOM. Secondly, we select some alternatives using the product maps. Finally, we make a final choice from the alternatives using AHP. As an example of real-world applications, we apply our method to a buying personal computer problem. We evaluated our method through experiments conducted by 117 people and confirmed its effectiveness. We will apply our method into another buying product problem.

5. Conclusion

This chapter has described forecasting with BPNN and *selective learning*, diagnosis with SOM and *ensemble learning*, and decision making with SOM and AHP. First, we described stock market prediction with BPNN and selective learning techniques for improving the ability of BPNN to predict large changes. Secondly, we showed an experimental study on medical diagnosis with SOM and ensemble learning. Thirdly, we described purchase decision making with SOM and AHP. These techniques can be applied to the other forecasting, diagnosis and decision making problems.

6. References

Azoff, E. (1994). Neural Network Time Series Forecasting of Financial Markets, John Wiley and Sons, West Sussex

Baba, N. & Kozaki, M. (1992). An intelligent forecasting system of stock price using neural networks, Proceedings of International Conference on Neural Networks, pp. I-371-I-377, Singapore

Bishop, C. (1995). Neural Networks for Pattern Recognition, Oxford University Press

Breiman L. (1994). Bagging predictors. Technical Report 421, Statistics Department, University of California, Berkley

Breiman, L. (2001). Random forests. Machine Learning, 45, (2001) 5-23

Freisleben, B. (1992). Stock market prediction with backpropagation networks. Lecture Notes in Computer Science, Vol. 604, 451-460, Springer-Verlag, Heidelberg

Garcia-Pedrajas, N.; Ortiz-Boyer, D. & Hervas-Martinez, C. (2006). An alternative approach for neural network evolution with a genetic algorithm: Crossover by combinatorial optimization. Neural Networks, Vol. 19, No. 4, (2006) 514-528

Ha, L.; Forgionne, G. & Wang, F. (2003). Facilitating Electronic Business Planning with Decision Making Support Systems, Lecture Notes in Artificial Intelligence, Vol. 2774, 45-51, Springer-Verlag, Heidelberg

Jensen, F. (2001). Bayesian Networks and Decision Graphs, Springer

Kil, D. & Shin, F. (1996). Pattern Recognition and Prediction with Applications to Signal Characterization, American Institute of Physics Press

Kohara, K. (1995). Selective Presentation Learning for Forecasting by Neural Networks, Proceedings of International Workshop on Applications of Neural Networks to Telecommunications, pp. 316-323, Stockholm

Kohara, K. (1996). Selectively Intensive Learning to Improve Large-Change Prediction by Neural Networks, Proceedings of International Conference on Engineering Applications of Neural Networks, pp. 463-466, London

Kohara, K. (2002). Neural networks for economic forecasting problems. In: Expert Systems - The Technology of Knowledge Management and Decision Making for the 21st Century-, Leondes, C. T. (Ed.) Academic Press. San Diego, CA

Kohara, K. & Isomae, M. (2006). Purchase Decision Support with Self-Organizing Maps and Analytic Hierarchy Process, Proceedings of International Conference on Knowledge Engineering and Decision Support, pp. 151-157, Lisbon

Kohara, K. (2008). Combining Selective-Presentation and Selective-Learning-Rate Approaches for Neural Network Forecasting of Stock Markets, Proceedings of International Workshop on Artificial Neural Networks and Intelligent Information Processing, pp. 3-9, Madeira

Kohonen, T. (1995). Self-Organizing Maps, Springer

Kotler, P. (2002). Marketing Management, eleventh edition, Prentice Hall

Mandic, D. & Chambers, J. (2001). Recurrent Neural Networks for Prediction, John Wiley & Sons

Merz, C.; Murphy, P. & Aha, D. (1997). UCI repository of machine learning databases. Dept of ICS, University of California, Irvine

Park, C-S. & Han, I. (2002). A CBR with feature weights derived by analytic hierarchy process for bankruptcy prediction. Expert Systems with Applications, Vol. 14, No. 3, (2002) 255-264

Park, D.; El-Sharkawi, M.; Marks II, R.; Atlas, L. & Damborg, M. (1991). Electric load forecasting using an artificial neural network. IEEE Transactions on Power Systems,Vol. 6, No. 2, (1991) 442-449

Pham, D. & Liu, X. (1995). Neural Networks for Identification, Prediction and Control, Springer

Quinlan, J. (1993). C4.5: Programs for Machine Learning, Morgan Kaufmann

Quinlan, J. (1996). Bagging, boosting and C4.5, Proceedings of the 13th National Conference on Artificial Intelligence and the 8th Innovative Applications of Artificial Intelligence Conference, pp. 725-730

Refenes, A. & Azema-Barac, M. (1994). Neural network applications in financial asset management. Neural Computing & Applications, Vol. 2, No. 1, (1994) 13-39

Riordan, D. & Hansen, B. K. (2002). A fuzzy case-based system for weather prediction. Engineering Intelligent Systems, Vol. 3, (2002) 139-145

Rumelhart, D.; Hinton, G. & Williams, R. (1986). Learning internal representations by error propagation. In: Parallel Distributed Processing 1, Rumelhart, D.; McClelland, J. & the PDP Research Group (Ed.), MIT Press, Cambridge, MA

Saaty, T. (1980). The Analytic Hierarchy Process, McGraw-Hill

Schapire, R. (1990). The strength of weak learnability. Machine Learning, 5, (1990) 197-227

Suka, M.; Ichimura, T. & Yoshida, K. (2003). Clinical Decision Support System Applied the Analytic Hierarchy Process. Lecture Notes in Artificial Intelligence, Vol.2774, 417-423, Springer-Verlag, Heidelberg

Tang, Z., Almeida, C. & Fishwick, P. (1991). Time series forecasting using neural networks vs. Box-Jenkins methodology. Simulation, Vol. 57, No. 5, (1991) 303-31

Tsoumakas, G.; Katakis, I. & Vlahavas, I. (2004). Effective Voting of Heterogeneous Classifiers. Lecture Notes in Artificial Intelligence, Vol. 3201, 465-476, Springer-Verlag, Heidelberg

Radivojac, P.; Obradovic, Z.; Dunker, A. & Vucetic, S. (2004). Feature Selection Filters Based on the Permutation Test. Lecture Notes in Artificial Intelligence, Vol. 3201, 334-346, Springer-Verlag, Heidelberg

Robnik-Sikonja, M. (2004). Improving Random Forests. Lecture Notes in Artificial Intelligence, Vol. 3201, 359-370, Springer-Verlag, Heidelberg

Vemuri, V. & Rogers, R. (Ed.) (1994). Artificial Neural Networks: Forecasting Time Series, IEEE Press, Los Alamitos, CA

Walle, B. V. & Moldovan, M. (2003). An Information Market for Multi-agent Decision Making: Observation from a Human Experiment. Lecture Notes in Artificial Intelligence, Vol. 2774, 66-72, Springer-Verlag, Heidelberg

Weigend, A., Huberman, B. & Rumelhart, D. (1990). Predicting the future: a connectionist approach. International Journal of Neural Systems, Vol. 1, No. 3, (1990) 193-209

White, H. (1988). Economic prediction using neural networks: the case of IBM daily stock return, Proceedings of International Conference on Neural Networks, pp. II-451-II-458, San Diego, CA

Zhang, H. & Su, J. (2004). Conditional Independence Trees. In: Boulicaut J-F, Esposito F, Giannotti F, Pedreschi D (eds). Lecture Notes in Artificial Intelligence, Vol. 3201, 513-524, Springer-Verlag, Heidelberg

14

Challenges of Middleware for the Internet of Things

Michal Nagy, Artem Katasonov, Oleksiy Khriyenko, Sergiy Nikitin, Michal Szydłowski and Vagan Terziyan
University of Jyväskylä
Finland

1. Introduction

Recent advances in networking, sensor and RFID technologies allow connecting various physical world objects to the IT infrastructure, which could, ultimately, enable realization of the Internet of Things and the Ubiquitous Computing visions. Also, this opens new horizons for industrial automation, i.e. automated monitoring, control, maintenance planning, etc, of industrial resources and processes. A much greater number of resources than that of today (machines, infrastructure elements, materials, products) can be connected to the IT systems, thus be monitored and potentially controlled. Such development will also necessarily create demand for a much wider integration with various external resources, such as data storages, information services, and algorithms, which can be found in other units of the same organization, in other organizations, or on the Internet.
The interconnectivity of computing and physical systems could, however, become "the nightmare of ubiquitous computing" (Kephart & Chess, 2003) in which human operators will be unable to manage the complexity of interactions in the system, neither even architects will be able to anticipate that complexity, and thus to design the system. The IBM vision of autonomic computing (Kephart & Chess, 2003) proclaims the need for computing systems capable of "running themselves" with minimal human management which is mainly limited to definition of some higher-level policies rather than direct administration. The computing systems will therefore be self-managed, which, according to the IBM vision, includes self-configuration, self-optimization, self-protection, and self-healing. The IBM vision emphasizes that the run-time self-manageability of a complex system requires its components to be to a certain degree autonomous themselves. Following this, we envision that the software agent technologies will play an important part in building such complex systems. Agent-based approach to software engineering is considered to be facilitating the design of complex systems (Jennings, 2001). In the multi-agent systems field, a significant amount of attention is paid to the task of building decentralized systems capable of supporting spontaneous configuration, tolerating partial failures, or arranging adaptive reorganization of the whole system (Mamei & Zambonelli, 2006).
A major problem is inherent heterogeneity in ubiquitous computing systems, with respect to the nature of components, standards, data formats, protocols, etc, which creates significant

obstacles for interoperability among the components. Semantic technologies are viewed today as a key technology to resolve the problems of interoperability and integration within heterogeneous world of ubiquitously interconnected objects and systems. Semantic technologies are claimed to be a qualitatively stronger approach to interoperability than contemporary standards-based approaches (Lassila, 2005). The Internet of Things should become in fact the Semantic Web of Things (Brock & Schuster, 2006). We subscribe to this view. Moreover, we apply semantic technologies not only to facilitate the discovery of heterogeneous components and data integration, but also for the behavioral control and coordination of those components (i.e. prescriptive specification of the expected behavior, declarative semantic programming).

It seems to be generally recognized that achieving the interoperability by imposing some rigid standards and making everyone comply could not be a case in ubiquitous environments. Therefore, the interoperability requires existence of some middleware to act as the glue joining heterogeneous components together. A consistent set of middleware, offering application programming interfaces, communications and other services to applications, will simplify the creation of applications and help to move from static programming approaches towards a configurable and dynamic composition capability (Buckley, 2006).

In this chapter, we describe our vision of such a middleware for the Internet of Things, which has also formed the basis for our research project UBIWARE. The project aims at a new generation middleware platform which will allow creation of self-managed complex systems, in particular industrial ones, consisting of distributed, heterogeneous, shared and reusable components of different nature, e.g. smart machines and devices, sensors, RFIDs, web-services, software applications, humans along with their interfaces, and others. Such middleware will enable various components to automatically discover each other and to configure a system with complex functionality based on the atomic functionalities of the components.

2. Semantic web technology for Internet of Things

2.1 Semantic Web technology for ubiquitous computing

According to (Buckley, 2006), the actual power of the Internet of Things arises from the fact that the devices are interconnected. Interoperability requires that clients of services know the features offered by service providers beforehand and semantic modeling should make it possible for service requestors to understand what the service providers have to offer. This is a key issue for moving towards an open-world approach, where new or modified devices and services may appear at any time, and towards device networks capable of dynamically adapting to context changes as may be imposed by application scenarios. This also has implications on requirements for middleware, as these are needed to interface between the devices that may be seen as services, and applications. Devices in the Internet of Things might need to be able to communicate with other devices anywhere in the world. This implies a need for a naming and addressing scheme, and means of search and discovery. The fact that devices may be related to an identity (through naming and addressing) raises in turn a number of privacy and security challenges. A consistent set of middleware, offering application programming interfaces, communications and other services to applications, will simplify the creation of services and applications. We need to move from static programming approaches towards a configurable and dynamic composition capability.

In (Lassila & Adler, 2003), the ubiquitous computing is presented as an emerging paradigm qualitatively different from current personal computing scenarios by involving dozens and hundreds of devices (sensors, external input and output devices, remotely controlled appliances, etc). A vision was presented for a class of devices, so called “Semantic Gadgets”, that will be able to combine functions of several portable devices users have today. Semantic Gadgets will be able to automatically configure themselves in new environments and to combine information and functionality from local and remote sources. Semantic Gadgets should be capable of semantic discovery and device coalition formation: the goal should be to accomplish discovery and configuration of new devices without “a human in the loop.” Authors pointed out that a critical to the success of this idea is the existence or emergence of certain infrastructures, such as the World Wide Web as a ubiquitous source of information and services and the Semantic Web as a more machine- and automation-friendly form of the Web.

Later, (Lassila 2005a) and (Lassila, 2005b) discussed possible application of Semantic Web technologies to mobile and ubiquitous computing arguing that ubiquitous computing represents the ultimate “interoperability nightmare”. This work is motivated by the need for better automation of user’s tasks by improving the interoperability between systems, applications, and information. Ultimately, one of the most important components of the realization of the Semantic Web is “serendipitous interoperability”, the ability of software systems to discover and utilize services they have not seen before, and that were not considered when and where the systems were designed. To realize this, qualitatively stronger means of representing service semantics are required, enabling fully automated discovery and invocation, and complete removal of unnecessary interaction with human users. Avoiding a priori commitments about how devices are to interact with one another will improve interoperability and will thus make dynamic ubiquitous computing scenarios without any choreographing more realistic. Semantic Web technologies are qualitatively stronger approach to interoperability than contemporary standards-based approaches.

2.2 Semantic Web technology for coordination

When it comes to developing complex, distributed software-based systems, the agent-based approach was advocated to be a well suited one (Jennings, 2001). From the implementation point of view, agents are a next step in the evolution of software engineering approaches and programming languages, the step following the trend towards increasing degrees of localization and encapsulation in the basic building blocks of the programming models (Jennings, 2000). After the structures, e.g., in C (localizing data), and objects, e.g., in C++ and Java (localizing, in addition, code, i.e. an entity’s behavior), agents follow by localizing their purpose, the thread of control and action selection. An agent is commonly defined as an encapsulated computer system situated in some environment and capable of flexible, autonomous action in that environment in order to meet its design objectives (Wooldridge, 1997).

The problem of “crossing the boundary” from the domain (problem) world to the machine (solution) world is widely recognized as a major issue in software and systems engineering. Therefore, when it comes to designing software, the most powerful abstractions are those that minimize the semantic distance between the units of analysis that are intuitively used to conceptualize the problem and the constructs present in the solution paradigm (Jennings, 2000). A possibility to have the same concept, i.e. agent, as the central one in both the problem analysis and the solution design and implementation can make it much easier to design a good solution and to handle the complexity. In contrast, e.g. the object-oriented approach has its conceptual basis determined by the underlying machine architecture, i.e. it is founded on implementation-level ontological primitives such as object, method, invocation, etc. Given that the early stages of software development are necessarily based on intentional concepts such as stakeholders, goals, plans, etc, there is an unavoidable gap that needs to be bridged. (Bresciani et al., 2004) even claimed that the agent-oriented programming paradigm is the only programming paradigm that can gracefully and seamlessly integrate the intentional models of early development phases with implementation and run-time phases. In a sense, agent-oriented approach postpones the transition from the domain concepts to the machine concepts until the stage of the design and implementation of individual agents (given that those are still to be implemented in an object-oriented programming language).

Although the flexibility of agent interactions has many advantages when it comes to engineering complex systems, the downside is that it leads to unpredictability in the run time system; as agents are autonomous, the patterns and the effects of their interactions are uncertain (Jennings, 2000). This raises a need for effective coordination, cooperation, and negotiation mechanism. (Those are in principle distinct, but the word "coordination" is often used as a general one encompassing all three; so for the sake of brevity we will use it like that too). (Jennings, 2000) discussed that it is common in specific systems and applications to circumvent these difficulties, i.e. to reduce the system's unpredictability, by using interaction protocols whose properties can be formally analyzed, by adopting rigid and preset organizational structures, and/or by limiting the nature and the scope of the agent interplay. However, Jennings asserted that these restrictions also limit the power of the agent-based approach; thus, in order to realize its full potential some longer term solutions are required.

The available literature sketches two major directions of search for such a longer term solution:

- D1: Social level characterization of agent-based systems. E.g. (Jennings, 2000) stressed the need for a better understanding of the impact of sociality and organizational context on an individual's behavior and of the symbiotic link between the behavior of the individual agents and that of the overall system.
- D2: Ontological approaches to coordination. E.g. (Tamma et al., 2005) asserted a need for common vocabulary for coordination, with a precise semantics, to enable agents to communicate their intentions with respect to future activities and resource utilization and get them to reason about coordination at run time. Also (Jennings et al., 1998) put as an issue to resolve the question about how to enable individual agents to represent and reason about the actions, plans, and knowledge of other agents to coordinate with them.

In our work, we attempt to provide a solution advancing into both D1 and D2 and somewhat integrating both. Some basic thinking, leading our work, follows.

3. The Vision and Approach

We believe that the ultimate goal is the vision of the Global Understanding Environment (GUN) (Terziyan, 2003; Terziyan, 2005; Kaykova et al., 2005a). We made the first step in the SmartResource project (2004-2006). Figure 1 depicts our research roadmap.

Global Understanding Environment (GUN) aims at making heterogeneous resources (physical, digital, and humans) web-accessible, proactive and cooperative. Three fundamentals of such platform are Interoperability, Automation and Integration. Interoperability in GUN requires utilization of Semantic Web standards, RDF-based metadata and ontologies and semantic adapters for the resources. Automation in GUN requires proactivity of resources based on applying the agent technologies. Integration in GUN requires ontology-based business process modeling and integration and multi-agent technologies for coordination of business processes over resources.

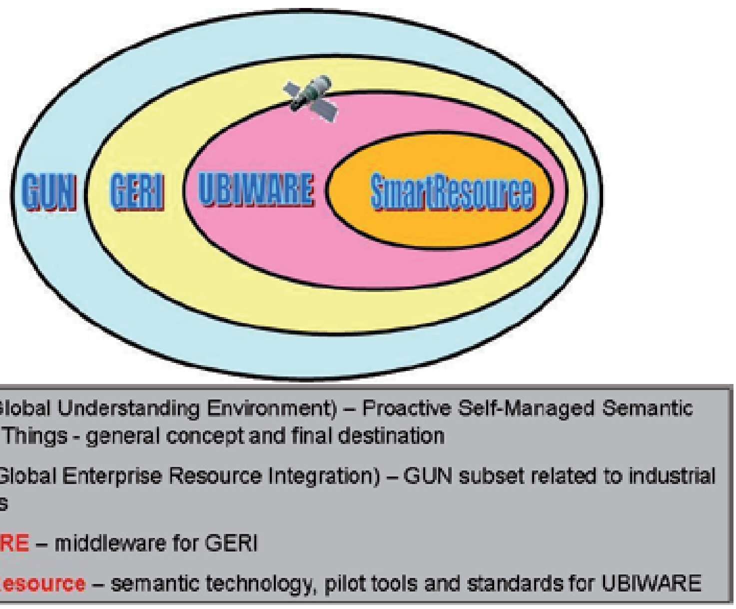


Fig. 1. The research roadmap towards GUN.

When applying Semantic Web in the domain of ubiquitous computing, it should be obvious that Semantic Web has to be able to describe resources not only as passive functional or non-functional entities, but also to describe their behavior (proactivity, communication, and coordination). In this sense, the word "global" in GUN has a double meaning. First, it implies that resources are able to communicate and cooperate globally, i.e. across the whole organization and beyond. Second, it implies a "global understanding". This means that a resource A can understand all of (1) the properties and the state of a resource B, (2) the potential and actual behaviors of B, and (3) the business processes in which A and B, and maybe other resources, are jointly involved.

Main layers of GUN can be seen in Figure 2. Various resources can be linked to the Semantic Web-based environment via adapters (or interfaces), which include (if necessary) sensors with digital output, data structuring (e.g. XML) and semantic adapter components (XML to

Semantic Web). Software agents are to be assigned to each resource and are assumed to be able to monitor data coming from the adapter about the state of the resource, make decisions on the behalf of the resource, and to discover, request and utilize external help if needed. Agent technologies within GUN allow mobility of service components between various platforms, decentralized service discovery, FIPA communication protocols utilization, and multi-agent integration/composition of services.

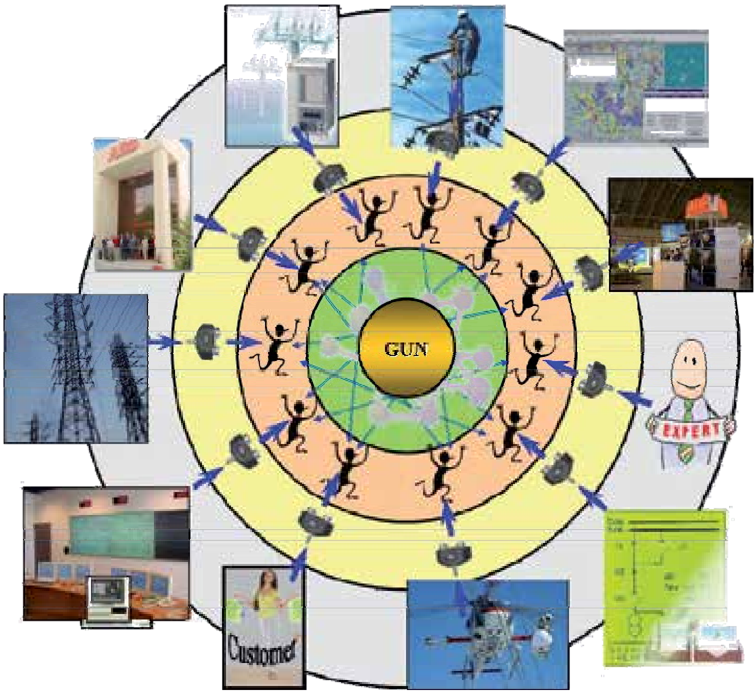


Fig. 2. Main layers of GUN.

When applying the GUN vision, each traditional system component becomes an agent-driven "smart resource", i.e. proactive and self-managing. This can also be recursive. For example, an interface of a system component can become a smart resource itself, i.e. it can have its own responsible agent, semantically adapted sensors and actuators, history, commitments with other resources, and self-monitoring, self-diagnostics and self-maintenance activities. This could guarantee high level of dynamism and flexibility of the interface. Such approach definitely has certain advantages when compared to other software technologies, which are integral parts of it, e.g. OOSE, SOA, Component-based SE, Agent-based SE, and Semantic SE. This approach is also applicable to various conceptual domain models. For example, domain ontology can be considered a smart resource, what would allow having multiple ontologies in the designed system and would enable their interoperability, on-the-fly mapping and maintenance, due to communication between corresponding agents.

In one sense, our intention is to apply the concepts of automatic discovery, selection, composition, orchestration, integration, invocation, execution monitoring, coordination, communication, negotiation, context awareness, etc (which were, so far, mostly related only to the Semantic Web-Services domain) to a more general "Semantic Web of Things" domain. Also we want to expand this list by adding automatic self-management including (self-*) organization, diagnostics, forecasting, control, configuration, adaptation, tuning, maintenance, and learning.

According to a more global view to the Ubiquitous Computing technology:

- UBIWARE will classify and register various ubiquitous devices and link them with web resources, services, software and humans as business processes' components;
- UBIWARE will consider sensors, sensor networks, embedded systems, alarm detectors, actuators, communication infrastructure, etc. as "smart objects" and will provide similar care to them as to other resources.

Utilization of the Semantic Web technology should allow:

- Reusable configuration patterns for ubiquitous resource adapters;
- Reusable semantic history blogs for all ubiquitous components;
- Reusable semantic behavior patterns for agents and processes descriptions;
- Reusable coordination, design, integration, composition and configuration patterns;
- Reusable decision-making patterns;
- Reusable interface patterns;
- Reusable security and privacy policies.

Utilization of the Distributed AI technology should allow:

- Proactivity and autonomic behavior;
- Communication, coordination, negotiation, contracting;
- Self-configuration and self-management;
- Learning based on live blog histories;
- Distributed data mining and knowledge discovery;
- Dynamic integration;
- Automated diagnostics and prediction;
- Model exchange and sharing.

4. The core of the middleware

4.1 Challenges

The main objectives of the Ubiware core (UbiCore) are the following. It has to give every resource a possibility to be smart (by connecting a software agent to it), in a sense that it would be able to proactively sense, monitor and control its own state, communicate with other components, compose and utilize own and external experiences and functionality for self-diagnostics and self-maintenance. It has to enable the resources to automatically discover each other and to configure a system with complex functionality based on the atomic functionalities of the resources. It has to ensure a predictable and systematic operation of the components and the system as a whole by enforcing that the smart resources act as prescribed by their organizational roles and by maintaining the "global" ontological understanding among the resources. The latter means that a resource A can understand all of (1) the properties and the state of a resource B, (2) the potential and actual behaviors of B, and (3) the business processes in which A and B, and maybe other resources, are jointly involved.

Several Agent Programming Languages (APLs) have been developed by researchers working in internal architectures and approaches to implementation of software agents. Examples of such languages are AGENT-0 (Shoham, 1993), AgentSpeak(L) (Rao, 1996), 3APL (Dastani et al., 2004) and ALPHA (Collier et al., 2005). All of those are declarative rule-based languages and are based on the first-order logic of n-ary predicates. All of them are also inspired by the Beliefs-Desires-Intentions architecture (Rao & Georgeff, 1991). For example, an agent program in ALPHA consists of declarations of the beliefs and goals of that agent and declaration of a set of rules, including belief rules (generating new beliefs based on existing ones), reactive rules (invoking some actions immediately) and commitment rules (adopting a commitment to invoke an action). Sensors (perceiving environment and generating new beliefs) and actuators (implementing the actions to be invoked) are then pieces of external code, namely in Java.

Based on the review of the languages mentioned above, we list the following important features of them, which should also be realized in UBIWARE's language for specification of behavior models:

- Ability to specify beliefs (something that the agent believes to be true) and goals (something that the agent does not believe to be true but wants to eventually become true).
- Ability to describe behavior rules, i.e. actions taken when a certain condition is met (can be either presence or absence of certain beliefs or presence of certain goals). Ability to have as the result of firing rule all of following:
  - adding/removing beliefs
  - engaging sensors and actuators
  - creating commitments (actions to be executed later)
- Ability to describe plans, i.e. predefined sequences of actions.
- Ability to describe commitments that are:
  - executed when certain condition is met
  - dropped when certain condition is met (or is not met anymore)

Agent-oriented approach postpones the transition from the domain concepts to the machine concepts until the stage of the design and implementation of individual agents. The advantage of using an APL is that the transition is postponed even further, until the implementation of particular sensors and actuators. This advantage seems to be, however, the only one that is considered. We did not encounter in literature approaches that would extend the role of APL code beyond the development stage. APL code is assumed to be written by the developer of an agent and either compiled into an executable program or interpreted in run-time but remaining an agent's intrinsic and static property. APL code is not assumed to ever come from outside of the agent in run-time, neither shared with other agents in any way.

Such export and sharing of APL code would, however, probably make sense. Methodologies for design of agent-based systems like OMNI (Vazquez-Salceda, 2005) describe an organizational role with a set of rules, and an APL is a rule-based language. So, using an APL for specifying a role sounds as a natural way to proceed. The difference is that APL code corresponding to a role should naturally be a property of and controlled by the organization, and accessed by the agents' enacting the role potentially even in the run-time. Run-time access would also enable the organization to update the role code if needed.

The second natural idea is that the agents may access a role's APL code not only in order to enact that role, but also in order to coordinate with the agents playing that role. As one option, an agent can send to another agent a part of its APL code to communicate its intentions with respect to future activities (so there is no need for a separate content language). As another option, if a role's code is made public inside the organization, the agents may access it in order to understand how to interact with, or what to expect from, an agent playing that role.

However, when thinking about using the existing APLs in such a manner, there are at least two issues:

- The code in an APL is, roughly speaking, a text. However in complex systems, a description of a role may need to include a huge number of rules and also a great number of beliefs representing the knowledge needed for playing the role. Also, in a case of access of the code by agents that are not going to enact this role, it is likely that they may wish to receive only a relevant part of it, not the whole thing. Therefore, a more efficient, e.g. a database-centric, solution is probably required.
- When APL code is provided by an organization to an agent, or shared between agents, mutual understanding of the meaning of the code is obviously required. While using first-order logic as the basis for an APL assures understanding of the semantics of the rules, the meaning of predicates used in those rules still needs to be consistently understood by all the parties involved. On the other hand, we are unaware of tools allowing unambiguous description of the precise semantics of n-ary predicates.

As a solution to these two issues, we see creating an APL based on the W3C's Resource Description Framework (RDF). RDF uses binary predicates only, i.e. triples (n-ary predicates can be represented nevertheless, of course, using several approaches). For RDF, tools are available for efficient database storage and querying, and also for explicit description of semantics, e.g. using OWL. Our proposition for such an RDF-based APL is the Semantic Agent Programming Language (S-APL).

4.2 Proposed solution

The answer to the requirements mentioned above is our Semantic Agent Programming Language (S-APL). S-APL is built on top of these axioms:

- Everything is a belief. All other mental attitudes such as desires, goals, commitments, behavioral rules are just complex beliefs.
- Every belief is either a semantic statement (subject-predicate-object triple) or a linked set of such statements.
- Every belief has the context that restricts the scope of validity of that belief. Beliefs have any meaning only inside their respective contexts.
- Statements can be made about contexts, i.e. contexts may appear as subjects or/and objects of triples. Such statements give meaning to contexts. This also leads to a hierarchy of contexts (not necessarily a tree structure though).
- There is the general context G, which is the root of the hierarchy. G is the context for global beliefs (as opposed to local ones). Nevertheless, every local belief, through the hierarchical chain of its contexts, is linked to G.

The notation that is selected for use in S-APL is a subset of Notation3 (Berners-Lee, 2006). Notation3 was proposed by Berners-Lee as an alternative to the dominant notation for RDF which is RDF/XML. Notation 3 is a language which is more compact and probably better readable than RDF/XML, and is also extended to allow greater expressiveness.

UbiCore uses a basic 3-layer agent structure that is common to the APL approach (see e.g. Collier et al., 2005). The architecture can be seen in Figure 3. There is a behavior engine implemented in Java, a declarative middle-layer, and a set of so-called Reusable Atomic Behaviors (RABs). A RAB is a Java component and its main purpose is to act as sensor or actuator. However, we do not restrict RABs to be only sensors or actuators, i.e. components concerned with the agent's environment. A RAB can also be a reasoner (data-processor) if some of the logic needed is impossible or is not efficient to realize with S-APL, or if one wants to enable an agent to do some other kind of reasoning beyond the rule-based one.

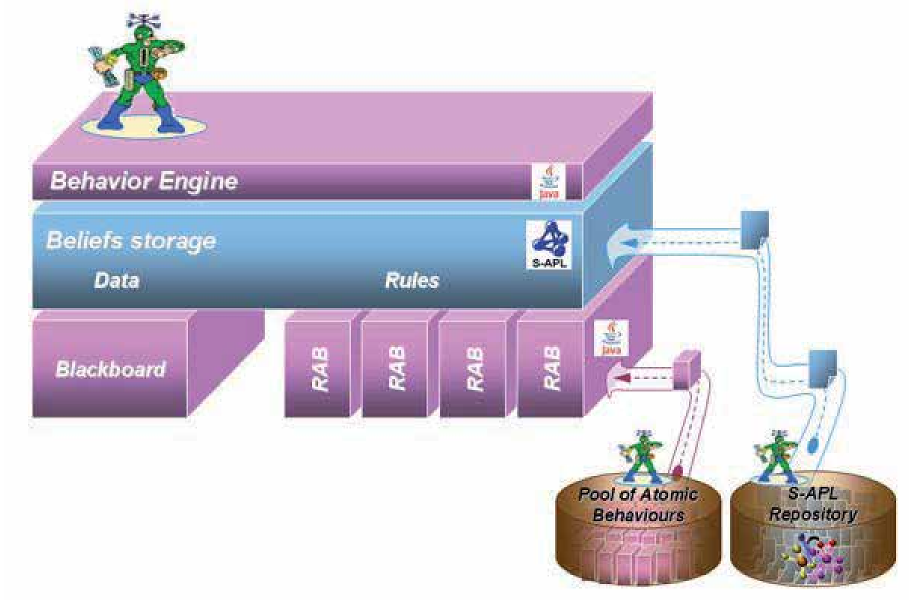


Fig. 3. The agent architecture.

The middle layer of the architecture acts as beliefs storage. What differentiates S-APL from traditional APLs is that S-APL is RDF-based. This provides the advantages of the semantic data model and reasoning. Another advantage of S-APL is that the data and the program code are treated in a similar way. Data and code use the same storage instead of two separate ones. This also means that: a rule upon its execution can add or remove another rule, the existence or absence of a rule can be used as a premise of another rule, and so on.
In traditional APL approaches this cannot be performed, because the rules are separated from the data. S-APL is very symmetric with respect to this – anything that can be done to a simple statement can also be done to any belief structure of any complexity. An S-APL agent can obtain data and rules also by querying S-APL repositories. For example a repository can contain a list of ready-made scripts that correspond to organizational roles. In our implementation, such querying is performed as inter-agent action with FIPA ACL messaging but does not involve any query or content languages beyond S-APL itself. As can be seen from Figure 3, agents also can load RABs remotely. This is done as an exception mechanism triggered when a rule prescribes engaging a RAB while the agent does not have it available. Thus, organizations are able to provide not only the rules to follow but also the tools needed for that.

An Ubiware-based application consists of a set of S-APL scripts (behavior models and data) and a set of Reusable Atomic Behaviors. In order to increase the reusability of the platform, we provide some ready-made scripts and behaviors. Therefore, our platform as such can be seen as consisting of the following three elements: (1) the behavior engine, (2) a set of "standard" SAPL models, (3) a set of "standard" RABs. The set of currently provided standard S-APL models includes for example communication models. The set of currently provided standard RABs includes RABs for such operations like downloading a web document, sending an email, loading and transforming into an S-APL presentation a text table document, loading an XML document, etc (see details in Katasonov, 2008).
Technically, our implementation is built on the top of the Java Agent Development Framework (JADE) (Bellifemine at al., 2007), which is a Java implementation of IEEE FIPA specifications. JADE provides communication infrastructure, agent lifecycle management, agent directory-based discovery and other standard services.

5. Intelligent resource visualization

5.1 Motivation

One of the reasons for intelligent resource visualization springs from the necessity of resource search and browsing processes enhancement. In (Marcos et al., 2005), several features of Classical Model of information search are criticized. First of all, this model does not adequately distinguish between the needs of a user and what a user must specify to get it. Very often, users may not know how to specify a good search query, even in natural language terms. Analyzing what is retrieved from the first attempt is usually not used to select useful results, but to find out what is there to be searched over. A second important criticism of the Classical Model is that any knowledge generated during the process of formulation a query is not used later in the sequence of search process steps, to influence the filtering step and presenting step of the search results, or to select the results. Finally, Classical Model provides an essentially context-free process. There is no proper way in which knowledge of the task context and situation, and user profile can have an influence on the information search process.
Thus, when we come to the vision of GUN (where all the resources of the virtual and the real world are connected and interoperate with each other) and the amount of information becomes huge, we have to elaborate new visualization techniques that simplify the search and browsing processes through reducing amount of queries via context-dependent resource visualization. Following this approach, we have a need to visualize the resource properties (in different from "directed arc between objects" representation way), the various relations between resources and the inter-resource communication process. And even more, we have a need to make this visualization more context-dependent, to be able to represent information in a handy way and adequate to a certain case (context), to reach the plasticity of UIs (Thevenin & Coutaz, 1999). Thus, the main focus in GUI development will be concentrated on the resource visualization aspects and we have a challenging task of semantically enhanced context-dependent multidimensional resource visualization.

5.2 4i: A new form of resource visualization

This part will study dynamic context-aware A2H (Agent-to-Human) interaction in UBIWARE, and will elaborate on a technology which we refer to as 4i (FOR EYE)

technology. 4i enables creation of a smart human interface through flexible collaboration of an Intelligent GUI Shell, various visualization modules, which we refer to as MetaProvider-services, and the resources of interest.
In UBIWARE, humans are important resources, which can play several distinct roles: a resource under care, a service provider, a user, and an administrator. Obviously, the humans need some graphical interfaces to interact with the rest of the system. The same person can play several roles, switch between them depending on the context, and, in result, require different interfaces at different times. In addition, a UBIWARE-based system presents a large integration environment with potentially huge amounts of heterogeneous data. Therefore, there is a need for tools facilitating information access and manipulation by humans. A semantic context-aware multimodal visualization approach would provide an opportunity for creating smart visual interfaces able of presenting relevant information in a more suitable and more personalized form.

Fig. 4. Human-Resource Adapter based on 4i (FOR EYE) technology.

From the GUN point of view, a human interface is a special case of a resource adapter (Human-Resource AdaPter - HRAP). We believe, however, that it is unreasonable to embed all the data acquisition, filtering and visualization logic into such an adapter. Instead, external services and application should be effectively utilized. Therefore, the intelligence of a smart interface (HRAP) will be a result of collaboration of multiple agents: the human's agent, the agents representing resources of interest (those to be monitored or/and controlled), and the agents of various visualization services – MetaProviders via an agent of HRAP (see Figure 4). This approach makes human interfaces different from other resource adapters and indicates a need for devoted research.
Based on 4i technology, an infrastructure will be embedded into UBIWARE enabling effective realization of the following system functions:

- visualization of data provided by a service in response to a request;

- search, retrieving and visualization of data required by a human expert,
- providing access to contextual information, and visualization of it;
- visualization of resource registration, configuration, and security policy establishment processes;
- resource discovery via MetaProviders (because they act as thematic portals).

5.3 4i infrastructure

As was mentioned in the previous section, one of the requirements for visual interfaces is an ability to represent information regarding to chosen contextual property of a resource, an interface should allow a user to simply choose a context for data representation, and should even support cases of multiple contextual property selection for complex views and filtering purposes. Such requirements can be met by MetaProviders - sui generis portals of resources with specific visualization view. It is named MetaProvider in a sense that it provides an access and presents other resources, which in turn are providers of own information (data). All GUN resources have certain own location (physical and digital). But it does not mean that they should have just one way of getting access to them. MetaProvider is an interface-mediator that gives a possibility to mark/select a resource (object) on its interface and provide the link to original resource location. In other words, it allows resource registration for further access to its data. At the same time, any resource can be registered on variety of different MetaProviders in different views. The main feature of MetaProviders is that each party which takes care of some GUN-Resource registers the resource itself. It causes fast filling of information accessible through MetaProvider. And each user/resource in one moment has an access to related information of amount of others. But such interoperability brings a new requirement for the MetaProviders and users. They should share common ontology to be interoperable on semantic level. Additionally to semantic interoperability, GUN-Resources are proactive/goal-driven resources and supplied with a Resource Agent for resource-to-resource (R2R)/ agent-to-agent (A2A) communication.

4i (FOR EYE) is an ensemble of GUN Resource Platform Intelligent GUI Shell (smart middleware for context dependent use and combination of a variety of different MetaProviders depending on user needs) and MetaProviders, visualization modules/platforms that provide context-dependent filtered representation of resource data and integration on two levels (information/data integration of the resources to be visualized and integration of resource representation views with a handy resource browsing) (see Figure 5). Context-awareness and intelligence of such interface brings a new feature that gives a possibility to the user to get not just the raw data, but required integrated information based on a specified context. GUI Shell allows the user dynamic switching between MetaProviders for more suitable information representation depending on a context or resource nature. MetaProvider plays these four main roles:

- Context-aware resource visualization module that presents information regarding to specified context in more suitable and personalized for user form;
- Interface for integrated information visualization with intelligent context-aware filtering mechanism to present only relevant information, avoiding a glut of unnecessary information;
- Visual Resource Platform that allows resource registration, search, access and modification of needed information/data in a space of registered resources;
- Mediator that facilitates resource to resource (R2R) communication.

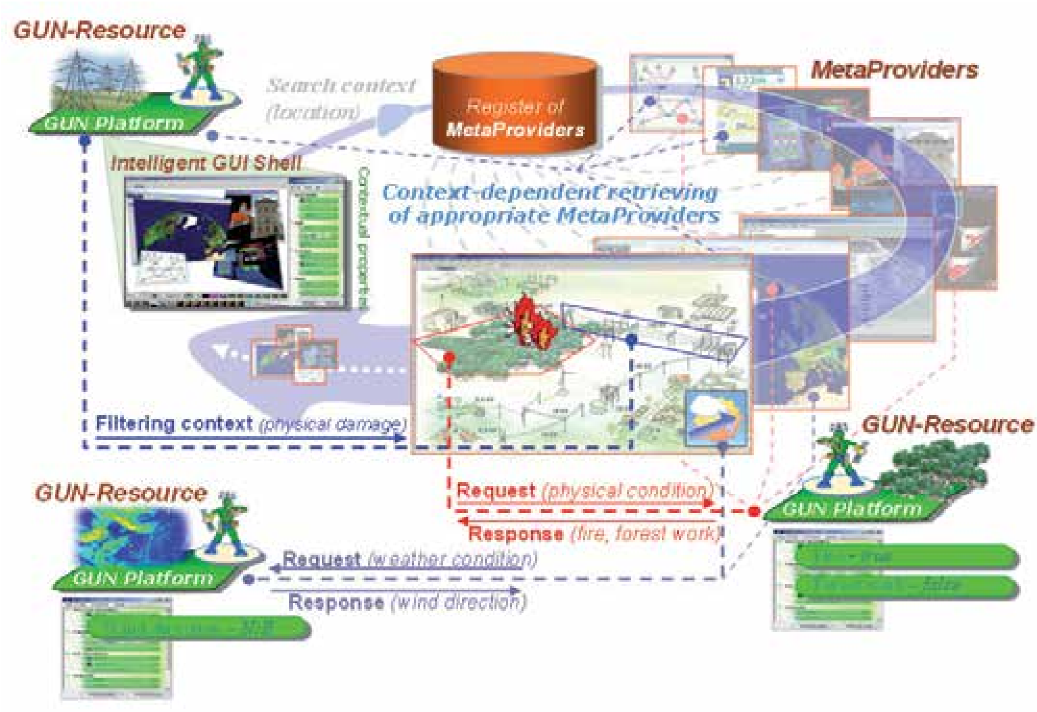


Fig. 5. Intelligent Interface for Integrated Information (4i technology).

5.4 Conclusion

4i (FOR EYE) technology is an integral part of UBIWARE. On one side, the technology is a base for Human-Resource adaptation and will be elaborated according to the principles and visions of UBIWARE. The intelligence of this smart interface is the result of collaboration of multiple agents: the human's agent, the agents representing resources of interest (those to be monitored or/and controlled or requesting a human), and the agents of various visualization services – MetaProviders via an agent of Human-Resource Adapter. From the other side, accordingly to the 4i, specific interfaces (MetaProviders) will be developed to provide functionality for a human (Platform administrator) to configure functionality of a UBIWARE-based system.

6. Resource histories

6.1 Motivation

In UBIWARE, every resource is represented by a software agent. Among major responsibilities of such an agent is monitoring the condition of the resource and the resource's interactions with other components of the system and humans. The beliefs storage of the agent will, therefore, naturally include the history of the resource, in a sense "blogged" by the agent. Obviously, the value of such a resource history is not limited to that particular resource. A resource may benefit from the information collected with respect to other resources of the same (or similar) type, e.g. in a situation which it faces for the first time while others may have faced that situation before. Also, mining the data collected and integrated from many resources may result in discovery of some knowledge important at the level of the whole ubiquitous computing system. A scalable solution requires mechanisms for inter-agent information sharing and data mining on integrated information which would allow keeping the resource histories distributed without need to copy those histories to a central repository.

The work in this area is aimed at answering the following research question:

- How to semantically markup the history of a resource in a system, in order to make it reusable for other resources and at the system-level?
- What mechanisms are needed for effective and efficient sharing of information between the agents representing different resources?

6.2 General approach

The general approach, which was selected in this task, is to avoid designing some special protocols and languages for inter-agent information sharing, but rather to reuse as much as possible the tools developed as part of the UbiCore mentioned in the Section 4.

Obviously, there is a certain similarity in the following. On one hand, an agent always needs to query its own beliefs base in order to evaluate the left sides of its behavior rules in order to identify rules that are to be executed. On the other hand, when an agent asks another agent for some information, it, in a sense, queries the belief base of that other agent.

Our approach is therefore to design the external querying process so it would be almost the same as if the agent itself would query its belief base to check the conditions for executing a rule. This also means that we plan to use the Semantic Agent Programming Language (S-APL) not only as the means for prescribing the agents' behaviors, but also as the inter-agents communication content language (to be used instead of FIPA-SL or other languages of this type). The advantages of this should be obvious as the symmetry and expressive power in the UBIWARE platform will be maximized. The agents will be able to query each other not only for some facts (present or historical) about the external world (the domain) but also, for example:

- Query if the other agent knows a plan for achieving a certain goal,
- Query if the other agent knows a rule that should be applied in a particular situation.

6.3 Ontonuts

In this section we introduce Ontonuts concept to facilitate the presentation of modular scripts and plans in the UBIWARE platform. The idea is somewhat similar to the notion of Enterprise Java Beans, whereas it is semantic by nature and has its own architectural peculiarities. Instances of Ontonut concept in general represent a capability with known input and expected output. Then we adapt Ontonuts to the problem of distributed querying. There are two main viewpoints towards distributed querying in the UBIWARE: adapter-based and service-based. The former tackles the adaptation of data sources that are external to the platform (databases, files, web services, etc.), when the latter deals with the agent-to-agent querying. Nevertheless, both target the same goal: to make distributed querying transparent to an agent (see Figure 6).

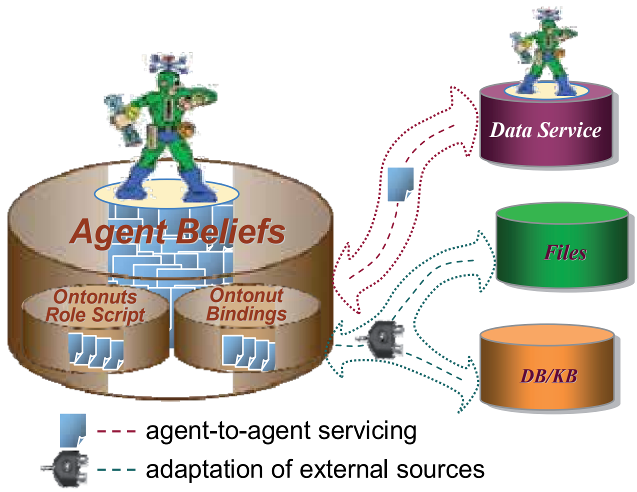


Fig. 6. Distributed querying in UBIWARE.

The agent-to-agent querying follows servicing paradigm and particularly data servicing discussed in (Quilitz & Leser, 2008). The adaptation of external sources (RDF-based adaptation is discussed in e.g. Langegger et al, 2007) resolves the problem of connectivity and interaction with those resources that are external to the platform, i.e. communicating with them in their native language. However, from the business logic developer's point of view, any remote resource should be transparent, in order to keep business logic script as modular and clear as possible. Ontonuts become wrapper, adapter and connector in one place.

For example, there can be a database that contains data in a native format about events of some machine. Let us consider the situation, when an agent needs to instantly analyze event series and, hence, perform queries regularly. An agent can have a flat solution, where all the calls to the database, as well as transformations are written within the same S-APL code that performs the analysis. Or, it may declare an Ontonut and specify a pattern that defines the query types allowed. Then the code that performs the analysis will be purely separated.

The Ontonuts technology is implemented as a combination of a Semantic Agent Programming Language (S-APL) script and Reusable Atomic Behaviors (RABs), and hence, can be dynamically added, removed or configured. Being run on the platform, Ontonuts can be understood as an engine or agent role. From the programming point of view Ontonuts allow componentization of S-APL code by introducing a semantic annotation to it. Such annotated pieces of code are called capabilities (analog of function in procedural programming). The capabilities further can be dynamically combined into plans and put to the execution by the Ontonuts engine.

The Ontonut capabilities are S-APL descriptions with explicitly defined preconditions and effects. The semantic annotation of Ontonut (by precondition and effect) allows us to automatically plan (compose) agent's activities to achieve a specified goal. The script part has an S-APL code that produces the effect based on the precondition. The whole data model of the UBIWARE platform is triple-based, therefore goals, preconditions and effects are defined as triple sets in S-APL. Suppose we have an initial data set `{A1 A2 A3}`, a goal `G1` defined as `{C1 C2 C3}` and we have two ontonuts `O1` and `O2`, defined as:

```
O1 :type :Ontonut
O1 :precondition {A1 A2 A3}
O1 :effect {B1 B2 B3}
O1 :script {{A1 A2 A3}=>... =>{B1 B2 B3}}

O2 :type :Ontonut
O2 :precondition {B1 B2 B3}
O2 :effect {C1 C2 C3}
O2 :script {{B1 B2 B3}=>...=>{C1 C2 C3}}
```

The appearance of the goal G1 will activate Ontonuts engine that will match G1 against available effects and apply planning, which will result in an execution plan: O1=>O2=>G1.

6.4 Conclusion

The goal of this work is to structure the S-APL code of the UBIWARE agent in order to simplify programming of the agent and allow automated goal-driven planning. Although, the work applies to a narrow domain, it involves generic problems of agent planning and semantic programming. The main topic is the automated planning of distributed queries and inevitably interferes with distributed RDF queries and so-called virtual graphs, when the graph being queried does not have content beneath; instead, the RDF queries are used to generate SQL queries to the corresponding database.

The approach proposed here, does not try to map models, but uses patterns to define the desired result and applies queries or any other code to fill the patterns requested by the agent.

7. Middleware for Peer-to-Peer Discovery

7.1 Motivation

With the growing popularity of multi-agent systems, the number of deployed multi-agent platforms will continue to increase. We see the necessity of existence of mechanisms which allow creating connections between these platforms. We believe that UBIWARE should be extended with functionality which would provide agents issuing search request not only with local results, but also with global results from the entire network of interconnected platforms. Our goal is to find the way to extend the UBIWARE functionality so that it could handle inter-platform communication and discovery in peer-to-peer (P2P) manner. Such extension will allow us to create a connection between separate UBIWARE systems where agents residing on different platforms could exchange information and cooperate.

The goal is to design mechanisms which will extend the scale of semantic resource discovery in UBIWARE with peer-to-peer discovery. Such mechanisms have to enable an agent:

- To discover agents playing a certain organizational role,
- To discover an agent (or agents) possessing certain needed information,
- To discover resources (through its agents) of certain type or possessing certain properties (e.g. a device in state X, or a web-resource providing some information searched for).

In all cases, existence of a central Directory Facilitator is not assumed, so the discovery is to be performed in peer-to-peer manner. This has at least two goals:

- Improving the survivability of UBIWARE-based systems. P2P is a complementary mechanism which can be utilized in an exception situation where the central Directory Facilitator became for some reason unavailable.
- Discovery across UBIWARE-based systems. There could be several UBIWARE-based agent platforms (applications) started up independently, each having own Directory Facilitator. While JADE (underlying agent framework of UBIWARE) supports communication among agents residing on different platforms, it does not provide for the cross-platform discovery.

7.2 Inter-platform and agent discovery in multi-platform environment

Let us analyze the situation, where several independent UBIWARE-based agent platforms exist in the network, each having a set of agents running and registered with DF.

Within each platform, inter-agent discovery and communication is not a problem, since the platform's components take care of these acts. The question arises, would it be possible (and how) to allow the agents from different platforms to search remote DFs and interact with agents located on foreign platforms. Such solution deployed in UBIWARE platform would enhance its functionality – if agent cannot locate a service or obtain needed information within its home platform, it can query the DFs on known remote platforms in search of needed data. Agents would benefit from this possibility in two situations:

The service or information required by an agent cannot be located within the native platform,

An agent possessing such service/information no longer exists within the platform or cannot be contacted.

We see the similarities between peer-to-peer file sharing networks and multi agent systems. In P2P networks the nodes exchange messages in order to locate their neighbors and resources they share. In multi agent systems the agents attempt to discover other existing agents and services they provide in order to utilize them, thus enhancing the overall functionality of the platform. In this section we analyze the existing approaches used to build file sharing P2P networks that could be utilized to address the problem – the centralized approach of general Directory Facilitator and solutions known from pure peer-to-peer file sharing networks with connection to JADE's mechanism of DF federations.

The centralized approach for inter-platform agent discovery borrows from the hybrid model known from file sharing peer-to peer networks implemented in Napster platform. In hybrid model the central server stores a list of available resources each node is sharing. Moving this idea to multi-platform environment, one of the existing platforms is designated to be a central server. Every agent wishing to publish its services contacts the central server and registers with the general Directory Facilitator. To discover an agent playing certain role, agents query the central server, which will return the list of addresses (AIDs) of agents providing required service. Figure 7 shows the idea of multi-platform environment with central Directory facilitator.

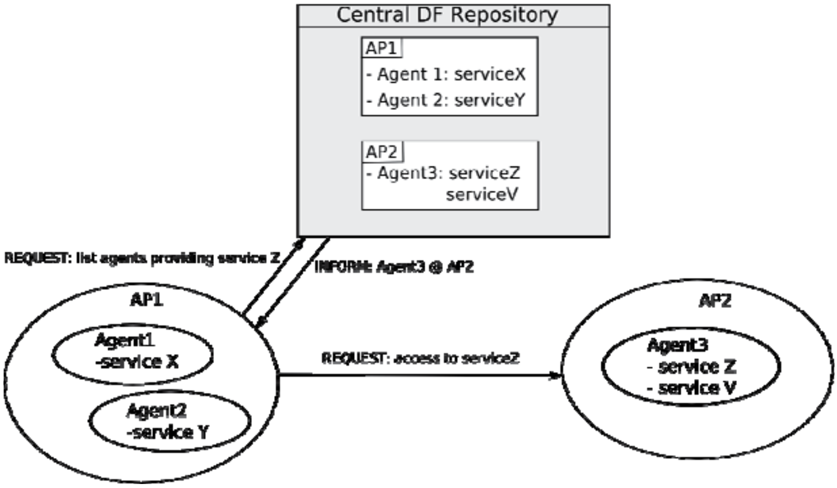


Fig. 7. Multi-platform environment with centralized DF.

The second way to achieve inter-platform agent discovery is through Federated Directory Facilitators. According to FIPA specifications, Directory Facilitators should be able to register with each other, creating federation within a single (also distributed) platform. When DF agent registers with another one, it becomes its child, and will from this point receive all search queries from its parent DF, also forwarding it to each of its child. The propagation of the query within federation is controlled by a parameter MaxDepth in search constraints, which specifies how many times the query should be forwarded. The parameter MaxResults limits the number of agents' descriptions that will be returned to agent issuing a query. Figure 8 shows an example of Agent Platform with multiple federated DFs each having set of agents registered with it.

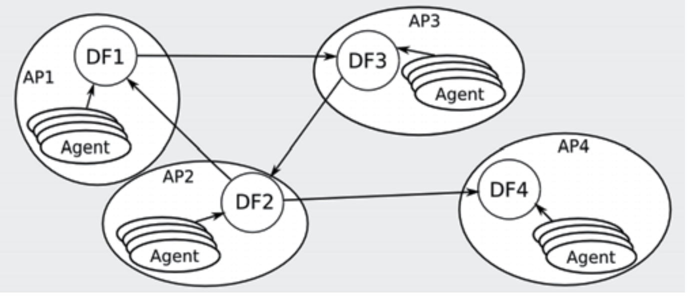


Fig. 8. Federation of Directory Facilitators.

The third way is to create a dynamic peer-to-peer topology. We borrow the idea of Gnutella system (Klingberg & Manfredi, 2002). Gnutella is a decentralized peer-to-peer system, which means it operates without presence of centralized server. Each participant acts as both server and a client; therefore the term "servent" was introduced. All servents share resources allowing others to locate and download them. By using messages passing system all servents execute two types of operation. First, by exchanging messages servents increase or maintain their degree of connectivity. Secondly, the search queries are exchanged in order to locate required resources that may be shared by some servents connected to the network. The protocol requires that within the network exists at least one always-on node, which provides a new participant with addresses of the servents already operating. Each servent upon startup obtains a pool of addresses and connects to them. In order to discover other participants it starts the PING / PONG process, presented in Figure 9.

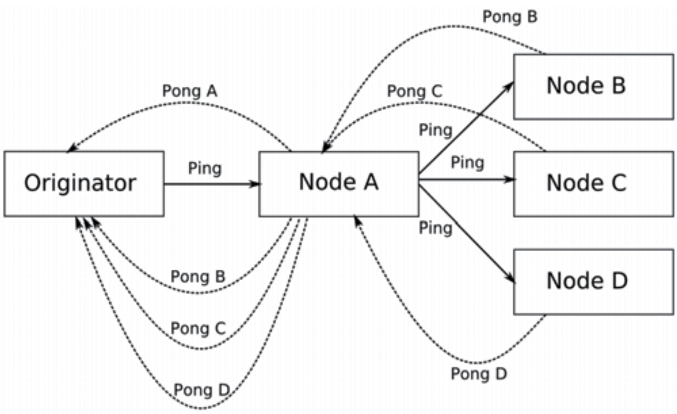


Fig. 9. Propagation of PING and PONG messages in Gnutella discovery process.

7.3 Conclusions and future work

We have presented three different approaches of building distributed peer-to-peer infrastructure in multiplatform environments. By the means of inter-platform discovery we give agents the opportunity to communicate, share services and resources beyond the boundaries of their home platforms.
Future work will include incorporating one of the described methods into the UBIWARE prototype. We also plan to conduct further research upon improving the efficiency of created network of agent platforms.

8. Conclusion and future work

In this chapter we present several challenges for achieving the vision of the Internet of Things and Ubiquitous Computing. Today's development in the field of networking, sensor and RFID technologies allows connecting various physical world objects to the IT infrastructure. However the complexity of such a system may become overwhelming and unmanageable. Therefore there is a need for computing systems capable of "running themselves" with minimal human management which is mainly limited to definition of some higher-level policies rather than direct administration. We believe that this complexity can be solved by incorporating the principles of multi-agent systems because of its ability to facilitate the design of complex systems.
Another challenge that has to be faced is the problem of heterogeneity of resources. Semantic technologies are viewed today as a key technology to resolve the problems of interoperability and integration within heterogeneous world of ubiquitously interconnected objects and systems. Semantic technologies are claimed to be a qualitatively stronger approach to interoperability than contemporary standards-based approaches. For this

reason we believe that Semantic Web technologies will play an important role in the vision of Internet of Things.

We do not believe that imposing some rigid standards is the right way to achieve the interoperability. Instead of that we suggest using middleware that will act as glue joining heterogeneous components together.

Based on these beliefs we describe our vision of such a middleware for the Internet of Things, which has also formed the basis for our research project Ubiware. Ubiware is one of the steps needed to achieve a bigger vision that we refer to as Global Understanding Environment (GUN). Global Understanding Environment (GUN) aims at making heterogeneous resources (physical, digital, and humans) web-accessible, proactive and cooperative. Three fundamentals of such platform are Interoperability, Automation and Integration.

The most important part of the middleware is the core. In the Ubiware project we refer to it as UbiCore. The goal of UbiCore is to give every resource a possibility to be smart (by connecting a software agent to it), in a sense that it would be able to proactively sense, monitor and control its own state, communicate with other components, compose and utilize own and external experiences and functionality for self-diagnostics and self-maintenance.

In order to be able to describe our intentions we needed a language. There are several existing agent programming languages (APLs) like AGENT-0, AgentSpeak(L), 3APL or ALPHA. All of those are declarative rule-based languages and are based on the first-order logic of n-ary predicates. All of them are also inspired by the Beliefs-Desires-Intentions architecture. However none of them considers the possibility of sharing the APL code with other agents or leaving the agent in the run-time.

Export and sharing of APL code would, however, make sense because of two main reasons. Firstly, this approach can be used for specifying the organizational roles since organizational roles are specified with a set of rules and APL is a rule-based language. Secondly, the agents may access a role's APL code not only in order to enact that role, but also in order to coordinate with the agents playing that role. In this way an agent can communicate its intentions with respect to future activities.

When thinking about using the existing APLs in way that mentioned above, there are at least two issues present. Firstly, the code in an APL is, roughly speaking, a text. However in complex systems, a description of a role may need to include a huge number of rules and also a great number of beliefs representing the knowledge needed for playing the role. Therefore, a more efficient, e.g. a database-centric, solution is probably required. Secondly, when APL code is provided by an organization to an agent, or shared between agents, mutual understanding of the meaning of the code is obviously required.

As a solution to these two issues, we see creating an APL based on the W3C's Resource Description Framework (RDF). RDF uses binary predicates only, i.e. triples. Our proposition for such an RDF-based APL is the Semantic Agent Programming Language (S-APL). We decided to use Notation3 as the base of this language because it is compact and better readeable than RDF/XML.

We use a basic 3-layer agent structure that is common for the APL approach. There is a behavior engine implemented in Java, a declarative middle-layer, and a set of sensors and actuators which are again Java components. The latter we refer to as Reusable Atomic Behaviors (RABs). In general a RAB can be any component concerned with the agent's environment, i.e. reasoner. The middle layer is the beliefs storage. What differentiates S-APL from traditional APLs is that S-APL is RDF-based. This provides the advantages of the semantic data model and reasoning.

The architecture of our platform implies that a particular application utilizing it will consist of a set of S-APL documents (data and behavior models) and a set of atomic behaviors needed for this particular application. There is a set of standard RABs and a set of standard S-APL scripts. They create the base of the Ubiware core. On top of them, the user can specify his/her own S-APL scripts and/or RABs.

We believe that the vision of Internet of Things also needs a new approach in the field of resource visualization. The classical model of information search has several disadvantages. Firstly, it is difficult for the user to transform the idea of the search into the proper search string. Many times, the first search is used just to find out what is there to be searched. Secondly, the classical model introduces a context-free process.

In order to overcome these two disadvantages of the classical model, we introduce For Eye (4i) concept. 4i is studying a dynamic context-aware A2H (Agent-to-Human) interaction in Ubiware. 4i enables the creation of a smart human interface through flexible collaboration of an Intelligent GUI Shell, various visualization modules, which we refer to as MetaProvider-services, and the resources of interest.

MetaProviders are visualization modules that provide context-dependent filtered representation of resource data and integration on two levels - data integration of the resources to be visualized and integration of resource representation views with a handy resource browsing. GUI Shell is used for binding MetaProviders together.

The fact that all resources are represented by an agent responsible for this resource implies that such an agent has knowledge of the state of this resource. The information about this state may be beneficial for other agents. Other agents can use this information in a situation which they face for the first time while others may have faced that situation before. Also, mining the data collected and integrated from many resources may result in discovery of some knowledge important at the level of the whole ubiquitous computing system.

We believe that the creation of a central repository is not the right approach. Instead of that we propose the idea of distributed resource histories based on a transparent mechanism of inter-agent information sharing and data mining. In order to achieve this goal we introduce the concept of Ontonut.

The Ontonuts technology is implemented as a combination of a Semantic Agent Programming Language (S-APL) script and Reusable Atomic Behaviors (RABs), and hence, can be dynamically added, removed or configured. Each Ontonut represents a capability of accessing some information. An Ontonut is annotated by precondition, effect and script property. Precondition defines a state required for executing the functionality of desired ontonut. Effect defines the resulting data that can be obtained by executing this Ontonut. The script property defines the way how to obtain the data. A part of the Ontonuts technology is also a planner that automatically composes a querying plan from available ontonuts and a desired goal specified by the agent.

In the future several Ubiware-based platforms may exist. Our goal is to design mechanisms which will extend the scale of semantic resource discovery in Ubiware with peer-to-peer discovery. We analyzed three approaches: Centralized Directory Facilitator, Federated Directory Facilitators and creation of a dynamic peer-to-peer topology. We believe that this type of discovery should not be based on a central Directory Facilitator. This will improve the survivability of the system.

In the future we would like to concentrate on the core extension. Currently we are working on an extension for agent observable environment. This opens new possibilities for coordination and self-configuration. In the area of peer-to-peer inter-platform discovery we plan to conduct further research on improving the efficiency of created network of agent platforms. Another topic that we are researching is the area of self-configuration and automated application composition.

9. References

Bellifemine, F. L., Caire G., Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*, Wiley, ISBN 978-0470057476

Berners-Lee, T., (2006). *Notation 3*, online (May 2009): http://www.w3.org/DesignIssues/Notation3.html

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., and Mylopoulos, J. (2004) Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems* 8(3): 203-236

Brock, D.L., Schuster, E. W., Allen, S.J., and Kar, Pinaki (2005) An Introduction to Semantic Modeling for Logistical Systems, *Journal of Business Logistics*, Vol.26, No.2, pp. 97-117 (available in: http://mitdatacenter.org/BrockSchusterAllenKar.pdf).

Buckley, J. (2006) From RFID to the Internet of Things: Pervasive Networked Systems, *Final Report on the Conference organized by DG Information Society and Media*, Networks and Communication Technologies Directorate, CCAB, Brussels (online :http: //www.rfidconsultation.eu/docs/ficheiros/WS_1_Final_report_27_Mar.pdf).

Collier, R., Ross, R., O'Hare, G. (2005). Realising reusable agent behaviours with ALPHA. *In: Eymann, T., Klugl, F., Lamersdorf,W., Klusch, M., Huhns,M.N. (eds.) MATES 2005*. LNCS (LNAI), vol. 3550, pp. 210-215. Springer, Heidelberg

Dastani, M., van Riemsdijk, B., Dignum, F., Meyer, J.J. (2004). A programming language for cognitive agents: Goal directed 3APL. *In: Dastani, M., Dix, J., El Fallah-Seghrouchni, A. (eds.) PROMAS 2003*. LNCS (LNAI), vol. 3067, pp. 111-130. Springer, Heidelberg

Jennings, N.R., Sycara K. P., and Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems 1(1)*: 7-38.

Jennings, N.R. (2000) On agent-based software engineering. *Artificial Intelligence 117(2)*: 277-296.

Jennings, N.R. (2001) An agent-based approach for building complex software systems. *Communications of the ACM 44(4)*: 35-41.

Katasonov, A. (2008). *UBIWARE Platform and Semantic Agent Programming Language (S-APL). Developer's guide*, Online: http://users.jyu.fi/~akataso/SAPLguide.pdf.

Kaykova O., Khriyenko O., Kovtun D., Naumenko A., Terziyan V., and Zharko A. (2005a) General Adaption Framework: Enabling Interoperability for Industrial Web Resources, *In: International Journal on Semantic Web and Information Systems, Idea Group*, Vol. 1, No. 3, pp.31-63.

Kephart J. O. and Chess D. M. (2003). The vision of autonomic computing, *IEEE Computer*, Vol. 36, No. 1, pp. 41-50

Klingberg, T., Manfredi, R. (2002) *Gnutella 0.6*, online: http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html

Langegger, A., Blochl, M., Woss, W., (2007). Sharing Data on the Grid using Ontologies and distributed SPARQL Queries, *Proceedings of 18th International Conference on Database and Expert Systems Applications*, pp.450-454, Regensburg, Germany, 3-7 Sept. 2007

Lassila, O. (2005a) Applying Semantic Web in Mobile and Ubiquitous Computing: Will Policy-Awareness Help?, in Lalana Kagal, Tim Finin, and James Hendler (eds.): *Proceedings of the Semantic Web Policy Workshop, 4th International Semantic Web Conference*, Galway, Ireland, pp. 6-11.

Lassila, O. (2005b) Using the Semantic Web in Mobile and Ubiquitous Computing, in: Max Bramer and Vagan Terziyan (eds.): *Proceedings of the 1st IFIP WG12.5 Working Conference on Industrial Applications of Semantic Web*, Springer IFIP, pp. 19-25.

Lassila, O., and Adler, M. (2003) Semantic Gadgets: Ubiquitous Computing Meets the Semantic Web, In: D. Fensel et al. (eds.), Spinning the Semantic Web, MIT Press, pp. 363-376.

Mamei, M, Zambonelli F. (2006). *Field-Based Coordination for Pervasive Multiagent Systems*, Soringer, ISBN 9783540279686, Berlin

Quilitz, B., Leser, U. (2008) Querying Distributed RDF Data Sources with SPARQL, The Semantic Web: Research and Applications, *5th European Semantic Web Conference, ESWC 2008*, Tenerife, Canary Islands, Spain, June 1-5, 2008, pp.524-538.

Rao, A.S. and Georgeff, M.P.(1991) Modeling rational agents within a BDI architecture. *Proc. 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pp. 473-484.

Rao, A.S. (1996) AgentSpeak(L): BDI agents speak out in a logical computable language. *Proc. 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World,* LNCS vol.1038, pp. 42-55.

Tamma, V.A.M., Aart, C., Moyaux, T., Paurobally, S., Lithgow-Smith, B., and Wooldridge, M. (2005) An ontological framework for dynamic coordination. *Proc. 4th International Semantic Web Conference'05*, LNCS vol. 3729, pp. 638-652.

Terziyan V. (2003) Semantic Web Services for Smart Devices in a "Global Understanding Environment", *In: R. Meersman and Z. Tari (eds.), On the Move to Meaningful Internet Systems 2003: OTM 2003 Workshops*, Lecture Notes in Computer Science, Vol. 2889, Springer-Verlag, pp.279-291.

Terziyan V. (2005) Semantic Web Services for Smart Devices Based on Mobile Agents, *In: International Journal of Intelligent Information Technologies*, Vol. 1, No. 2, Idea Group, pp. 43-55.

Thevenin, D. and Coutaz, J., (1999). Plasticity of User Interfaces: Framework and Research Agenda. *In Proceedings of Interact'99*, vol. 1, Edinburgh: IFIP, IOS Press, 1999, pp. 110-117.

Vázquez-Salceda, J., Dignum, V., and Dignum, F. (2005) Organizing multiagent systems. *Autonomous Agents and Multi-Agent Systems 11(3)*: 307-360.

Wooldridge, M. (1997) Agent-based software engineering. *IEE Proceedings of Software Engineering 144(1)*: 26-37.

15

Artificial Intelligence Methods in Fault Tolerant Control

Luis E. Garza Castañón and Adriana Vargas Martínez
Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM)
Monterrey, México

1. Introduction to Fault Tolerant Control

An increasing demand on products quality, system reliability, and plant availability has allowed that engineers and scientists give more attention to the design of methods and systems that can handle certain types of faults. In addition, the global crisis creates more competition between industries and plant shutdowns are not an option because they cause production losses and consequently lack of presence in the markets; primary services such as power grids, water supplies, transportation systems, and communication and commodities production cannot be interrupted without putting at risk human health and social stability.

On the other hand, modern systems and challenging operating conditions increase the possibility of system failures which can cause loss of human lives and equipments; also, some dangerous environments in places such as nuclear or chemical plants, set restrictive limits to human work. In all these environments the use of automation and intelligent systems is fundamental to minimize the impact of faults.

The most important benefit of the Fault Tolerant Control (FTC) approach is that the plant continues operating in spite of a fault, no matter if the process has certain degradation in its performance. This strategy prevents that a fault develops into a more serious failure. In summary, the main advantages of implementing an FTC system are (Blanke et al., 1997):

- Plant availability and system reliability in spite of the presence of a fault.
- Prevention to develop a single fault in to a system failure.
- The use of information redundancy to detect faults instead of adding more hardware.
- The use of reconfiguration in the system components to accommodate a fault.
- FTC admits degraded performance due to a fault but maintain the system availability.
- Is cheap because most of the time no new hardware will be needed.

Some areas where FTC is being used more often are: aerospace systems, flight control, automotive engine systems and industrial processes. All of these systems have a complex structure and require a close supervision; FTC utilizes plant redundancy to create an intelligent system that can supervise the behavior of the plant components making these kinds of systems more reliable.

Since a few years ago, emerging FTC techniques have been proposing new controller designs capable to tolerate system malfunctions and maintain stability and desirable performance properties. In order to achieve its objectives, two main tasks have to be considered on an active FTC system: fault detection and diagnosis and controller reconfiguration. The main purpose of fault detection and diagnosis is to detect, isolate and identify the fault, determining which faults affect the availability and safety of the plant. The controller reconfiguration task accommodates the fault and re-calculates the controller parameters in order to reduce the fault effects.

Although several schemes of FTCS have been proposed, most of them are closely related to a general architecture. (Blanke et al., 1997) introduce an approach for the design of an FTC system, shown in figure 1, which included three operational levels: single sensor validation, fault detection and isolation using analytical redundancy, and an autonomous supervision and reconfiguration system. The single sensor validation level involves the control loop with actuators, sensors, the controller and the signal conditioning and filtering. The second level (FDI) is composed of detectors and effectors that will perform the remedial actions. And finally, the supervision level deals with state-event logic in order to describe the logical state of controlled objects.

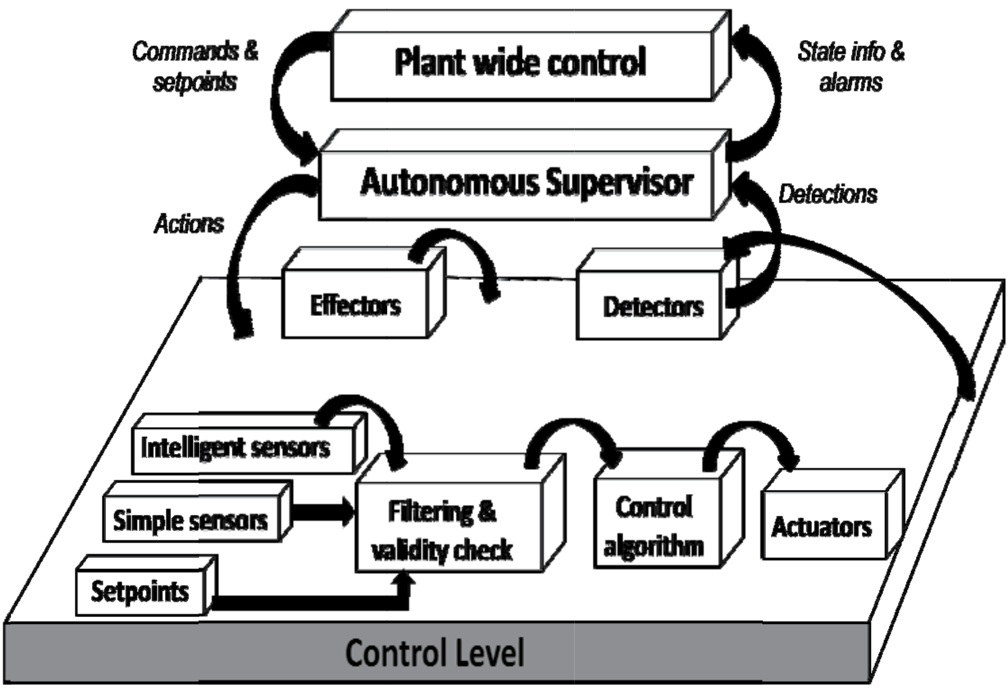


Fig. 1. Architecture for Fault Tolerant Autonomous Control Systems proposed by (Blanke, 1997).

A slightly different architecture is presented in (Karsai et al, 2003). They introduce a scheme of Fault-Adaptive Control Technology (FACT), centered on model-based approaches for fault detection, fault isolation and estimation, and controller selection and reconfiguration for hybrid systems (see figure 2). Hybrid models derived from hybrid bond graphs are used to model the continuous and discrete system dynamics. The supervisory controller, modeled as a generalized finite state automaton, generates the discrete events that cause reconfigurations in the continuous energy-based bond graph models of the plant. Fault detection involves a comparison between expected behaviors of the system, generated from the hybrid models, with actual system behavior.

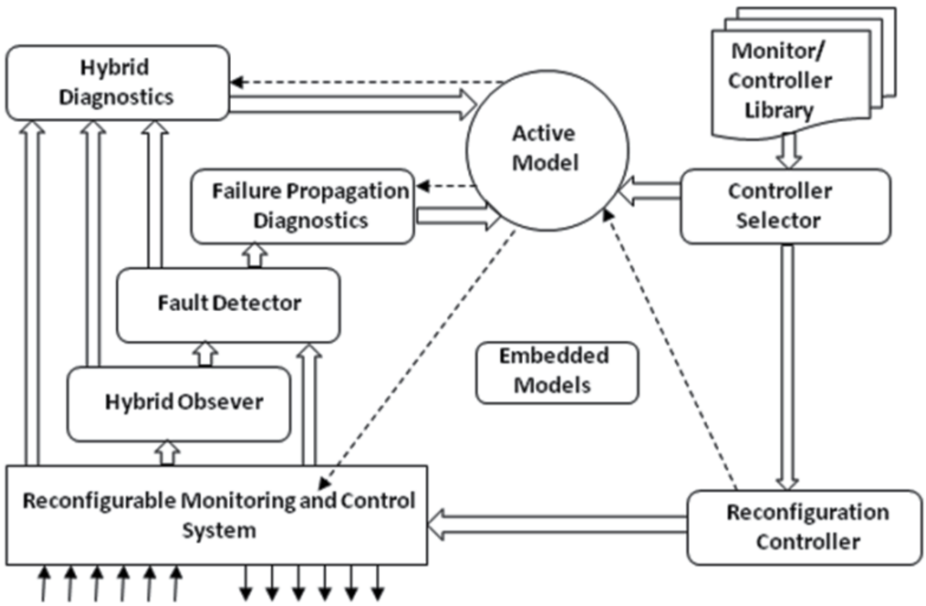


Fig. 2. Architecture for Fault-Adaptive Tolerant Control Technology (FACT) proposed by (Karsai et al, 2003).

2. Classification of the Fault Tolerant Control Methods

Some authors have proposed different classifications for the FTC methods (Blanke et al., 2003; Eterno et al., 1985; Farrel et al., 1993; Lunze & Richter, 2006; Patton, 1997; Stengel, 1991). The classification shown in figure 3 includes all the methods explained by these authors. We can also find a recent and very complete survey of FTC methods and applications in (Zhang & Jiang, 2008).

Regarding the design methods, fault tolerant control can be classified into two main approaches: active or passive. In Active Fault Tolerant Control (AFTC), if a fault occurs, the control system will be reconfigured using some properties of the original system in order to maintain an acceptable performance, stability and robustness. In some cases degraded system operations have to be accepted (Blanke et al., 2001; Patton, 1997; Mahmoud et al., 2003). In Passive Fault Tolerant Control (PFTC) the system has a specific fixed controller to counteract the effect and to be robust against certain faults (Eterno et al., 1985).

To implement the AFTC approach two tasks are needed: fault detection and isolation and controller reconfiguration or accommodation. FDI means early detection, diagnosis, isolation, identification, classification and explanation of single and multiple faults; and can be accomplished by using the following three methodologies (Venkatasubramanian et al., 2003a, 2003b, 2003c):

Quantitative Model-Based: requires knowledge of the process model and dynamics in a mathematical structural form. Also, the process parameters, which are unknown, are calculated applying parameter estimation methods to measured inputs and outputs signals of the process. This approach uses analytical redundancy that can be obtained by implementing Kalman filters, observers and parity space.

Qualitative Model-Based: Are based on the essential comprehension of the process physics and chemical properties. The model understanding is represented with quality functions placed in different parts of the process. This methodology can be divided in abstraction hierarchies and causal models. Abstraction hierarchies are based on decomposition and the model can establish inferences of the overall system behavior from the subsystems law behavior. This can be done using functional or structural approaches. Causal models take the causal system structure to represent the process relationships and are classified in diagraphs, fault trees and qualitative physics.

Process History-Based: uses a considerable amount of the process historical data and transform this data into a priori knowledge in order to understand the system dynamics. This data transformation is done using qualitative or quantitative methods. The quantitative methods are divided in expert systems (solves problems using expertise domain) and trend modeling (represents only significant events to understand the process). Quantitative methods can be statistical (use PCA, DPCA, PLA, CA) and non statistical (neural networks) to recognize and classify the problem.

After the detection and isolation of the fault, a controller reconfiguration or accommodation is needed. In controller accommodation, when a fault appears, the variables that are measured and manipulated by the controller continue unaffected, but the dynamic structure and parameters of the controller change (Blanke et al., 2003). The fault will be accommodated only if the control objective with a control law that involves the parameters and structure of the faulty system has a solution (Blanke et al., 2001). In order to achieve fault accommodation, two approaches can be used: adaptive control and switched control. Adaptive control means to modify the controller control law to handle the situation where the system's parameters are changing over time. It does not need a priori information about the parameters limits. The goal is to minimize the error between the actual behavior of the system and the desirable behavior. In the other hand, switched control is determined by a bank of controllers designed for specifics purposes (normal operation or fault) that switch from one to another in order to control a specific situation (Lunze & Richter, 2006).

Meanwhile, controller reconfiguration is related with changing the structure of the controller, the manipulated and the measured variables when a fault occurs (Steffen, 2005). This is achieved by using the following techniques:

Controller Redesign. The controller changes when a fault occurs in order to continue achieving its objective (Blanke et al., 2003). This can be done by using several approaches: pseudo inverse methods (modified pseudo inverse method, admissible pseudo inverse method), model following (adaptive model following, perfect model following, eigen structure assignment) and optimization (linear quadratic design, model predictive control) (Caglayan *et al.*, 1988; Gao & Antsaklis, 1991; Jiang, 1994; Lunze & Richter, 2006; Staroswiecki, 2005).

Fault Hiding Methods. The controller continues unchanged when a fault is placed, because a reconfiguration system hides the fault from the controller. This method can be realized using virtual actuators or virtual sensors. (Lunze & Richter, 2006; Steffen, 2005).

Projection Based Methods. A controller is designed a priori for every specific fault situation and replaces the nominal controller if that specific fault occurs. This can be done by a bank of controllers and a bank of observers (Mahmoud et al., 2003).

Learning Control. This methodology uses artificial intelligence like neural networks, fuzzy logic, genetic algorithms, expert systems and hybrid systems which can learn to detect, identify and accommodate the fault (Polycarpou & Vemuri, 1995; Stengel, 1991; Karsai et al, 2003).

Physical Redundancy. This is an expensive approach because it uses hardware redundancy (multiple sensor or actuators) and decision logic to correct a fault because it switches the faulty component to a new one. An example of this is the voting scheme method (Isermann et al., 2002; Mahmoud et al., 2003).

On the other hand, passive FTC is based on robust control. In this technique, an established controller with constant parameters is designed to correct a specific fault to guarantee stability and performance (Lunze & Richter, 2006). There is no need for online fault information. The control objectives of robust control are: stability, tracking, disturbance rejection, sensor noise rejection, rejection of actuator saturation and robustness (Skogestad & Postlethwaite, 2005). Robust control involves the following methodologies:

H_∞ controller. This type of controller deals with the minimization of the H-infinity-norm in order to optimize the worst case of performance specifications. In Fault Tolerant Control can be used as an index to represent the attenuation of the disturbances performances in a closed loop system (Yang & Ye, 2006) or can be used for the design of robust and stable dynamical compensators (Jaimoukha et al., 2006; Liang & Duan, 2004).

Linear Matrix Inequalities (LMIs). In this case, convex optimization problems are solved with precise matrices constraints. In Fault Tolerant Control is implemented to achieve robustness against actuator and sensor faults. (Zhang et al., 2007).

Simultaneous Stabilization. In this approach multiple plants must achieve stability using the same controller in the presence of faults. (Blondel, 1994).

Youla-Jabr-Bongiorno-Kucera (YJBK) parameterization. This methodology is implemented in Fault Tolerant Control to parameterize stabilizing controllers in order to guarantee system stability. YJBK in summary is a representation of the feedback controllers that stabilize a given system (Neimann & Stoustrup, 2005).

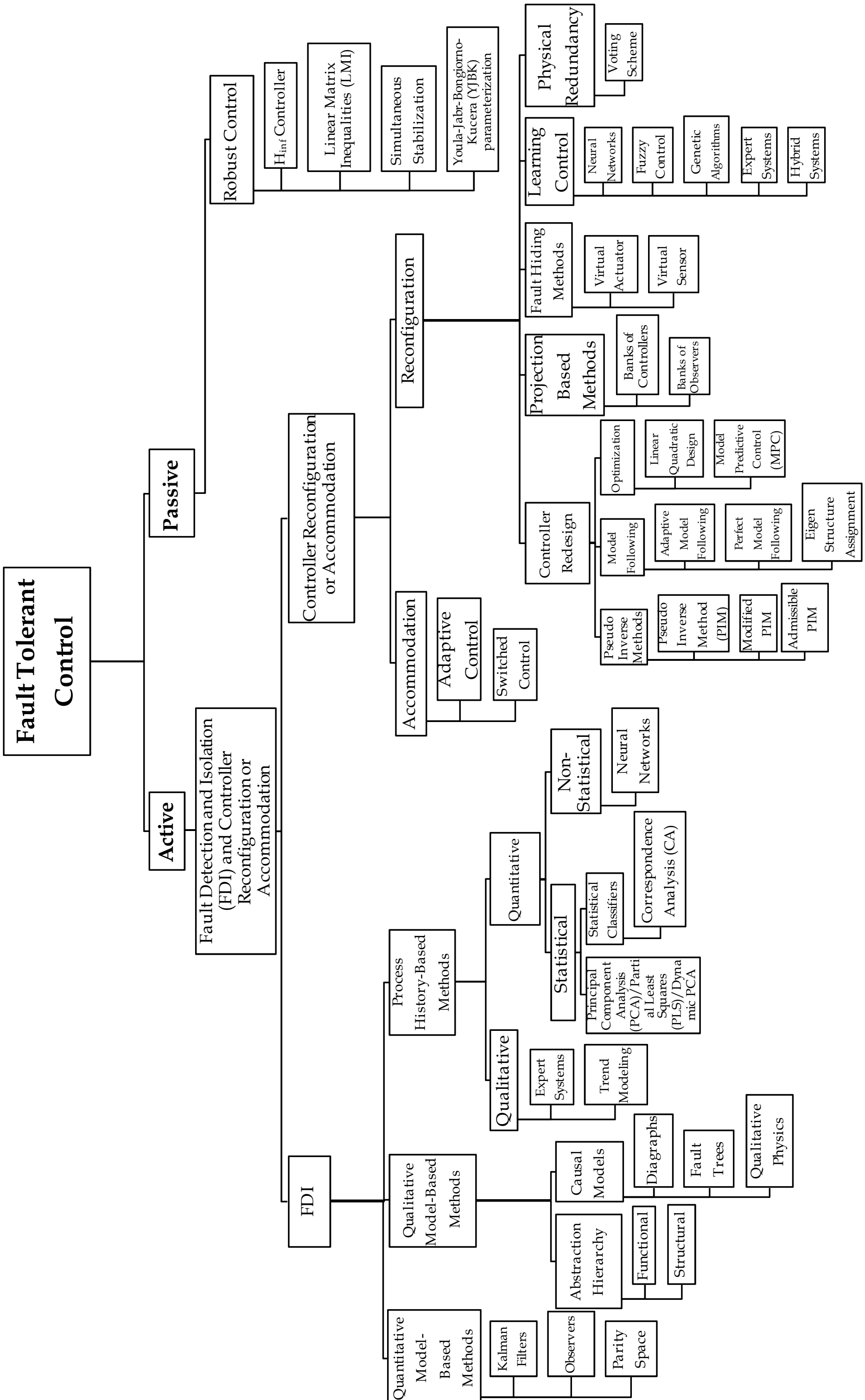


Fig. 3. FTC classification approaches

3. Artificial Intelligence Methods

The use of AI in fault tolerant control has been suggested in the past (Bastani & Chen, 1988). Methods such as Neural Networks (NNs), Fuzzy Logic and Neuro-Fuzzy Systems, offer an advantage over traditional methods (state observers, statistical analysis, parameter estimation, parity relations, residual generation, etc) because can reproduce the behavior of non linear dynamical systems with models extracted from data. This is a very important issue in FTC applications on automated processes, where information is easily available, or processes where accurate mathematical models are hard to obtain. In the other hand, AI optimization tools such as Genetic Algorithms (GAs) provide a powerful tool for multiobjective optimization problems frequently found on FTC.

3.1 Neural Networks

Artificial Neural Networks (ANNs) are mathematical models that try to mimic the biological nervous system. An artificial neuron have multiple input signals x_1, x_2, …,x_n entering the neuron using connection links with specific weights w_1, w_2, …, w_n or $\sum_{i=1}^{n} w_n x_i$ named the net input, and also have a firing threshold b, an activation function f and an output of the neuron that is represented by $y = f\left(\sum_{i=1}^{n} w_i x_i - b\right)$. The firing threshold b or bias can be represented as another weight by placing an extra input node x_0 that takes a value of 1 and has a w_0=-b. (Nguyen et al., 2002). This can be represented in the figure 4.

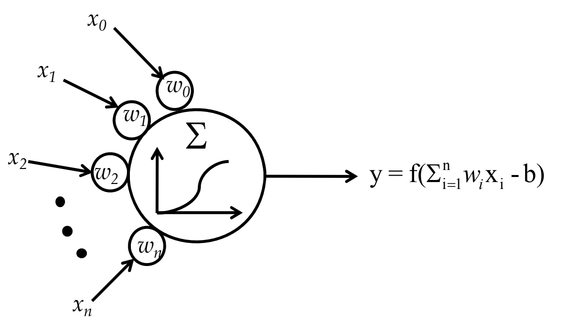


Fig. 4. Artificial Neuron.

A neural network with more than one input layer of neurons, a middle layer called the hidden layer and an output layer is named a multi-layer neural network.

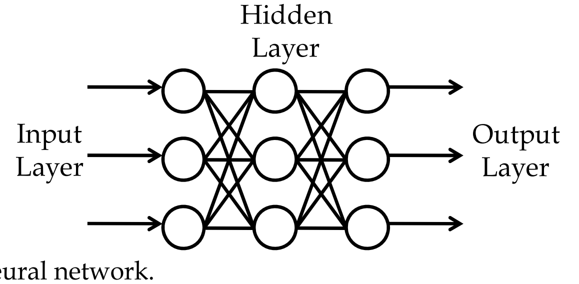


Fig. 5. Multi-layer neural network.

A neural network can have a feedback or a feed forward structure. In the feedback structure the information can move back and forward. In the feedforward structure, the information moves only forward from the input nodes through the outputs nodes with no cycles in the network (Ruan, 1997).
The neural networks need to be trained from examples, in a process called supervised learning. Once a successfully training is done, the neural network is ready if and only if the networks reproduce the desired outputs from the given inputs. The most common methodology for this kind of learning is the backpropagation algorithm, where the weights of the neural network are determined by using iteration until the output of the network is the same as the desired output (Rumelhart et al., 1986). In addition, unsupervised learning uses a mechanism for changing values of the weights according to the input values, this mechanism is named self-organization. An example of this algorithm is the Hebbian learning algorithm (Ruan, 1997).

3.1.1 Neural Networks in Fault Tolerant Control

Artificial neural networks have been applied in fault tolerant control because they are helpful to identify, detect and accommodate system faults. The application of ANNs to FTC can be divided in three groups. The first group includes neural networks used as fault detectors by estimating changes in process models dynamics *(Polycarpou* & Helmicki*, 1995; Patton et al., 1999; Polycarpou, 2001; Gomaa, 2004)*. The second group includes neural networks used as controllers (*Wang & Wang, 1999; Pashilkar et al., 2006*), and the third group integrates neural networks which performs both functions: fault detection, and control (*Perhinschi et al., 2007; Yen & DeLima 2005*).
(Polycarpou & Helmicki*, 1995)* proposed a construction of automated fault detection and accommodation architecture that uses on-line approximators and adaptive-learning schemes. The online approximator is a neural network model that monitors changes in the system dynamics due to a failure.
(Patton et al., 1999) use a scheme of neural network to detect and isolate a fault in two steps: residual generation and decision making. In the first step a residual vector characterizes the fault and then the second step process the residual vector information in order to locate the fault and the time of occurrence. Once the residual is trained, qualitative knowledge of the plant can be added. This combination of qualitative and quantitative approached is helpful to decrease the number of false alarms in the fault decision making step.
(Polycarpou, 2001) proposed a methodology for fault accommodation of a multivariable nonlinear dynamical system using a learning approach that monitors and approximates any abnormal behavior using neural networks and adaptive nonlinear estimation. When a fault occurs the neural network is used to estimate the nonlinear fault function supplying a framework for fault identification and accommodation. The neural network at the beginning of the monitoring stage is capable of learning the modeling errors in order to improve the system robustness.
(Gomaa, 2004) recommended a fault tolerant control approach based on multi-ANN system faulty models. The nominal plant is nonlinear and is vulnerable to faults. A feedforward neural network is trained as the nominal model; two PID controllers are used, one for the nominal plant and the other for the neural network imitating the nominal plant (reference model). Both PIDs controllers were tuned using genetic algorithms. If there exist a difference between the nominal plant (y_p) and the reference model (y_{rm}) a nonzero residual is

generated. Then, depending on the magnitude of the residual an ANN faulty model and its respective compensation path are selected to repair the fault and improve the system operating conditions. This can be observed in figure 6.

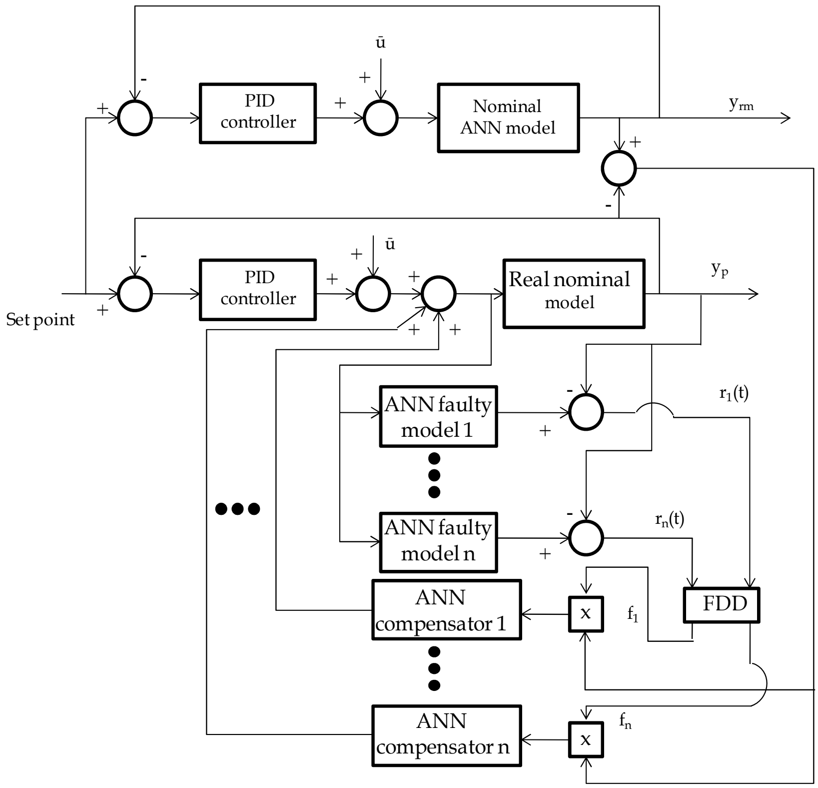


Fig. 6. Multi-ANN faulty models FTC scheme (Gomaa, 2004).

(Wang & Wang, 1999) proposed a Neural Network-FTC where the nominal system is controlled by a Pseudolinear Neural Network (PNN) based one-step-ahead controller that uses a modified gradient approach. When the system is with no fault, a PNN model connected in parallel to the plant model can be trained and used for the design of the control algorithm. The PNN model is helpful for the detection of a fault. In addition, when a fault is present, a residual signal is generated and an extra neural network based fault compensation loop is imported in order to provide the closed loop stability. This last neural network is a two layers perceptron network and its weights are updated using the modified gradient approach. This FTC system is shown in figure 7.

(Pashilkar et al. 2006) proposed a neural controller that improves the fault tolerant potential of a fighter aircraft during landing. The faults are caused by severe winds or stuck control surfaces and can be divided in single faults (aileron or elevator stuck) or double fault (aileron and elevator stuck). This neural network controller employs a feedback error learning method with a dynamic radial basis function neural network. The neural network uses on-line training and not a-priori training. This kind of controller helped to improve the capability of handling large faults and also helps to achieve the desired requirements.

(Perhinschi et al., 2007) presented a methodology for detection, identification and accommodation of sensor and actuator failures inside fault tolerant control laws. The fault detection and identification uses neural estimators. The accommodating control laws design for the actuator fault is done using nonlinear dynamic inversion with neural network augmentation. Whereas the accommodation of sensor fault is accomplished by changing the failed sensor output for neural estimates calculated in the detection and identification process. This approach can handle sensor and actuator faults successfully. It uses membership functions to describe the mathematical model of process.
(Yen & DeLima, 2005) presented a neural network trained on-line with a global dual heuristic programming architecture. This approach has also a supervision structure made from decision logic. This supervision level is very efficient to identify the controller faults in early stages and can supply new values to improve the convergence utilizing dynamic model bank information.

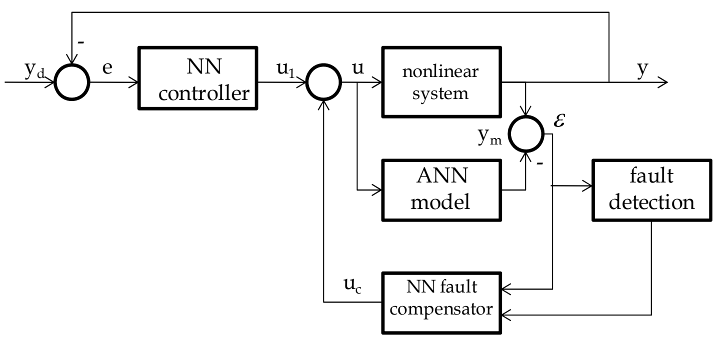


Fig. 7. Neural Network FTC scheme proposed by (Wang & Wang, 1999).

3.2 Fuzzy Logic

Fuzzy logic is a research field of fuzzy set theory; this methodology uses heuristic and mathematical tools to approximate reasoning in order to achieve knowledge processing when lack of information exists. In addition, this approach tries to mimic the control logic rules used by persons.
In Fuzzy systems, a mathematical model of the system is not needed because the desired input and output relationships are defined by "if" and "then" rules. Also, due to the lack of system information it is not possible to precise the exact value of a variable "x" but it is possible to know the membership function. The membership function is created to describe range of values that the variable can have, this is known as input fuzzification (Passino & Yurkovich, 1997; Ruan, 1997; Nguyen, 2002). This membership function can have different shapes that are defined according to a specific problem; the shapes forms are: triangular, Gaussian trapezoidal, sigmoidal S and sigmoidal Z functions.
Once the membership functions of variables are established, the inference rules "if (antecedent)-then (consequent)" are set. Then an inference method is used to quantify each premise, this is an application of a fuzzy operator in the "if" part of the rule and can be the math function "or" – "and", once this is done, the inference method fire rules, which is known as the implication method. Two inference methods are commonly used: Mamdani method, where the "if" and "then" parts are fuzzy expressions, and the Takagi-Sugeno

method where the "if" part of the rule is a fuzzy expression, but the "then" part of the rule is a deterministic expression (Passino & Yurkovich, 1997; Nguyen, 2002). After applying the chosen inference method, the output fuzzy set is form. This can be shown in the next rules:

If *(0.4 or 0.2)* then *input (0.4)*

If *max (0.4, 0.2)* then *input (0.4)*

Finally***:* If *(0.4)* then *input (0.4)***

The above means that if the antecedent part of the rule is partially true which means that have a value minor than one, then the output fuzzy set will be truncated by the implication method, as showed in the next figure:

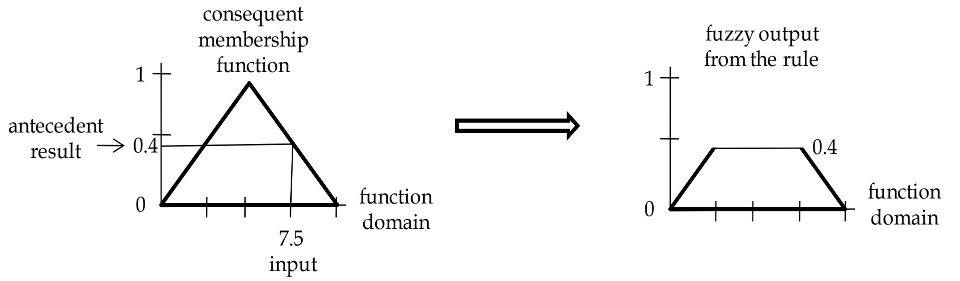


Fig. 8. Implication method representation.

The last step in the inference method is the aggregation of all the fuzzy outputs from the rules, as can be seen in the next figure:

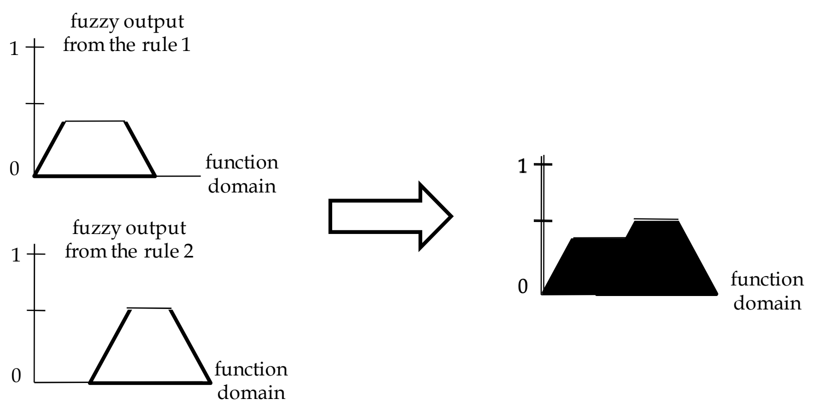


Fig. 9. Fuzzy outputs aggregation representation.

Finally, the fuzzy controller implements defuzzification, where the input of the defuzzification process is a fuzzy set and the output is just a number. This can be done using several methods as: center of area method also known as the center or gravity or centroid method, the height-center of area, the max criterion, the first of maxima and the middle of maxima method. The deffuzification result will be the output of the fuzzy system (Passino & Yurkovich, 1997; Nguyen, 2002). This is showed in the following figure:

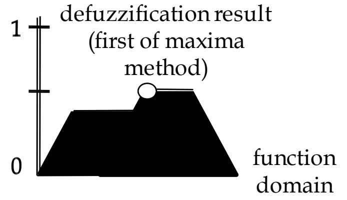


Fig. 10. Defuzzification representation.

In summary, the main advantages of Fuzzy Logic are: robustness, easy controller reconfiguration, works well with multi input multi output systems and adequate to model complex nonlinear systems.

3.2.1 Fuzzy Logic in Fault Tolerant Control

Fuzzy models are capable of handling nonlinear dynamic systems and the nonlinearities of the faults. In the following we describe several works reported with fault detectors and/or controllers based on fuzzy logic.

(Kwong et al., 1995) presented an expert supervision approach of Fuzzy learning system for fault tolerant control in aircrafts. This approach, shown in figure 11, uses a fuzzy model reference learning controller (FMRLC) to reconfigure the aircraft nominal controller in order to accommodate actuator failures without needing specific information about the failures. This approach also demonstrated that the FMRLC can be improved using fault detection and identification information to acquire an adaptive system that can reconfigure its performance level when a failure occurs.

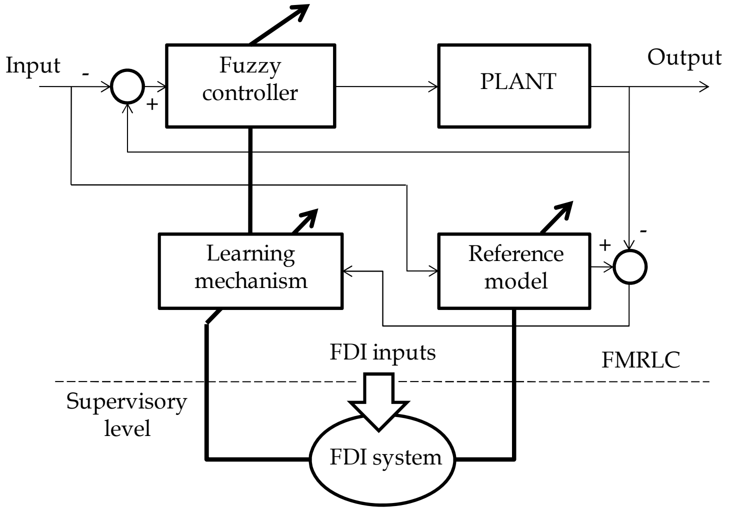


Fig. 11. Supervisory FMRLC scheme proposed by (Kwong et al., 1995).

(Ballé et al., 1998) presented a model-based adaptive control and reconfiguration approach based on fault detection and diagnosis implemented to a heat exchanger. The fault detection and adaptive tasks are based on Takagi-Sugeno (T-S) fuzzy model of the process and the

fault diagnosis task uses a self-organizing fuzzy structure. In this approach an on-line adaptation scheme is used for reconfiguration when the measurements are erroneous because of the sensor faults. If all the input signal and the disturbances can be measured the model is used for model-based predictive control. When a sensor fault occurs, the signals are not longer used for prediction. Instead of these measurements, a reduced model of the system process is required to adapt these erroneous signals as external disturbances. The T-S fuzzy model is also used for residuals generation in order to detect and isolate sensor faults. Then the decision making is sustained by identifying particular patterns of the residual that are link to different faults. This is done by using a fuzzy classification tree that can learn sample data patterns. The next figure shows the explained approach:

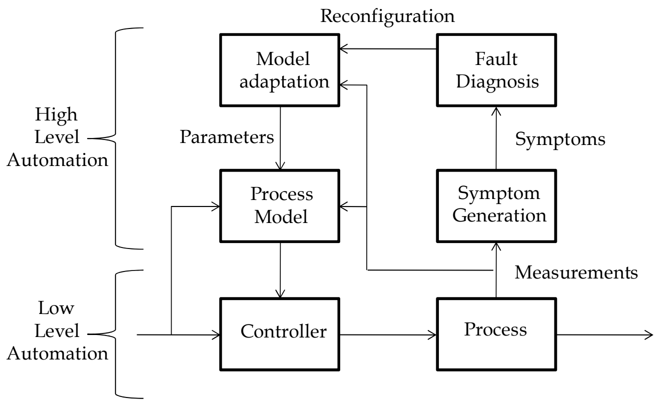


Fig. 12. Model-based control, diagnosis and reconfiguration scheme proposed by (Ballé al., 1998).

(Patton et al., 1999) proposed the use of Takagi-Sugeno fuzzy observers for nonlinear dynamic systems. In this methodology the plant is represented by a T-S fuzzy model and a fuzzy observer estimate the system state vector. The fuzzy model is locally observable in order to design the fuzzy observer. A linear time invariant observer is created with every real of the fuzzy model. Then, the whole complete observer's dynamic is a weighted sum of the individual observers. A residual is generated from the observers and is compared against the fuzzy model output. If the residual is nonzero a fault is present in the system.
(Holmes & Ray, 2001) proposed a damage mitigating control system in which the main objective is to accomplish optimal performance of the system with a high level of reliability, availability, component durability and maintainability. This scheme was implemented in a two-tier structure. A lineal robust sample data controller was proposed for the lower tier to track a reference trajectory vector. On the other hand, for the upper tier, a fuzzy logic based damage controller was implemented. This controller adjusts the system dynamic performance with the structural durability of critical components. This scheme demonstrated that a substantial gain in structural durability of critical components can be reached without significant performance degradation.
(Diao & Passino, 2001) implemented an intelligent Fault Tolerant Control scheme using a hierarchical learning structure for the design of the plant. This learning structure was a Takagi-Sugeno Fuzzy system. Then, the fault tolerant control system is based on stable

adaptive fuzzy/neural control that has online learning properties which are used to acquire the unknown dynamics developed as consequence of system faults.
(Diao & Passino, 2002) presented a fault tolerant control scheme using adaptive estimation and control methods based on the learning properties of fuzzy systems and neural networks. An online approximation-based stable adaptive neural/fuzzy control was implemented in an input-output feedback time-varying nonlinear system. Also, the adaptive controller enhanced the fault tolerance capability by using the estimated information from the fault diagnosis unit, chosen by a multiple model interface with a supervisory expert approach.
(Oudghiri et al., 2008) proposed a Fault Tolerant scheme for lateral vehicle dynamics implementing static output feedback control with sliding mode observers in order to enhance the vehicle stability and handling in the presence of sensor fault. This approach uses three blocks, one for the fault and detection, a second for the static output feedback controller and a third for a switcher. The lateral vehicle dynamics are represented in a nonlinear two degrees of freedom vehicle model implemented in a Takagi-Sugeno fuzzy model. The fault detection and isolation block uses a bank of observers' constructed using sliding mode in order to calculate the system state vector. Then, a set of diagnostic signal residuals are produced, associating the measured and the estimated outputs. After doing this, the sensor in which the fault happened is isolated.

3.3 Genetic Algorithms

Genetic Algorithms (GAs) are searching and optimizing algorithms motivated by natural selection evolution and natural genetics (Goldberg, 1989). The simplest GA follows the next steps: Generate a random initial population of chromosomes, calculate the fitness of every chromosome in the population, apply selection, crossover and mutation and replace the actual population with the new population until the required solution is achieved. The main advantages of GAs are: powerful computational effect, robustness, fault tolerance, fast convergence to a global optimal, capability of searching in complex landscape where the fitness function is discontinuous, can be combined with traditional optimization techniques (Tabu search) and have the ability to solve problem without needing human experts (Goldberg, 1989; Mitchell, 1996; Ruan, 1997).

3.3.1 Genetic Algorithms in Fault Tolerant Control

Recently genetic algorithms have been applied in fault tolerant control as a strategy to optimize and supervise the controlled system in order to accommodate system failures. Some applications of this technique are the following:
(Schroder et al., 1998) proposed a fault tolerant control technique for an active magnetic bearing. In this approach a nonlinear model of a turbo machine rotor from the rolls-royce lifted up by an active magnetic bearing was presented. This model is capable of modeling difference configuration of magnetic bearings. A multi-objective genetic algorithm was used to generate and adequate PID controller for the active magnetic bearing with different bearing configuration. Also the fault accommodation was done using a centralized fault compensation scheme.
(Sugawara et al., 2003) showed a fault tolerant control approach using multi-layer neural networks with a genetic algorithm. The proposed of this approach was to develop a self-

recovery technique implemented for large scale neural networks programmed in a single ship to accommodate faults without the needing of a host computer. This FTC scheme uses hardware redundancy and weight retraining using a genetic algorithm in order to reconfigure the neural network to accommodate the fault. The objective of the genetic algorithm is to reduce the error between the actual output and the desired output.

4. The Proposed Methodology, Experiments and Results

We propose a new FTC schema, where a Model Reference Adaptive Control (MRAC) is used in combination with a neural network controller, in order to achieve a better performance when faults are present in the system. We use an experimental model of a heat exchanger where abrupt and gradual faults (also called *soft faults*) are induced in sensors and actuators. To compare our schema, we also have made experiments with a simple MRAC and MRAC-PID structures.

4.1 MRAC Controller

The Model Reference Adaptive Controller, shown in figure 13, implements a closed loop controller that involves the parameters that should be optimized, in order to modify the system response to achieve the desired final value. The adaptation mechanism adjusts the controller parameters to match the process output with the reference model output.

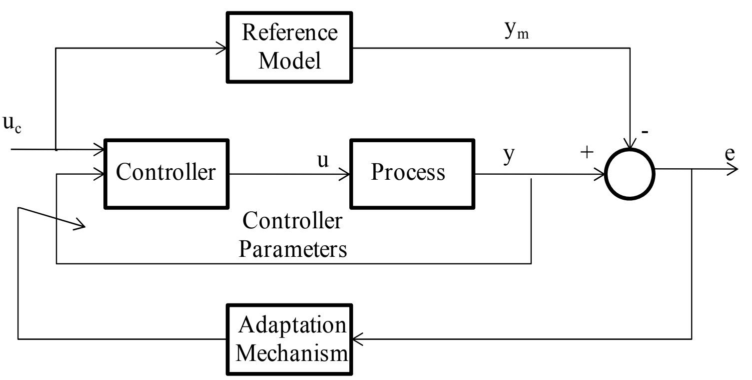


Fig. 13. MRAC scheme (Nagrath, 2006).

The controller error is calculated as follows:

$$e = y_{process} - y_{reference} \tag{1}$$

To reduce the error, a cost function was used, in the form of:

$$J(\theta) = \frac{1}{2} e^2(\theta) \tag{2}$$

The function above can be minimized if the parameters $\boldsymbol{\theta}$ change in the negative direction of the gradient $\boldsymbol{J}$, this is called the gradient descent method and is represented by:

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \tag{3}$$

where γ helps to adjust the speed of learning. The above equation is known as the MIT rule and determines how the parameter θ will be updated in order to reduce the error.

The implemented MRAC scheme in our process, shown in figure 14, has two adaptation parameters: adaptive feedfoward gain (θ_1) and adaptive feedback gain (θ_2). These parameters will be updated to follow the reference model.

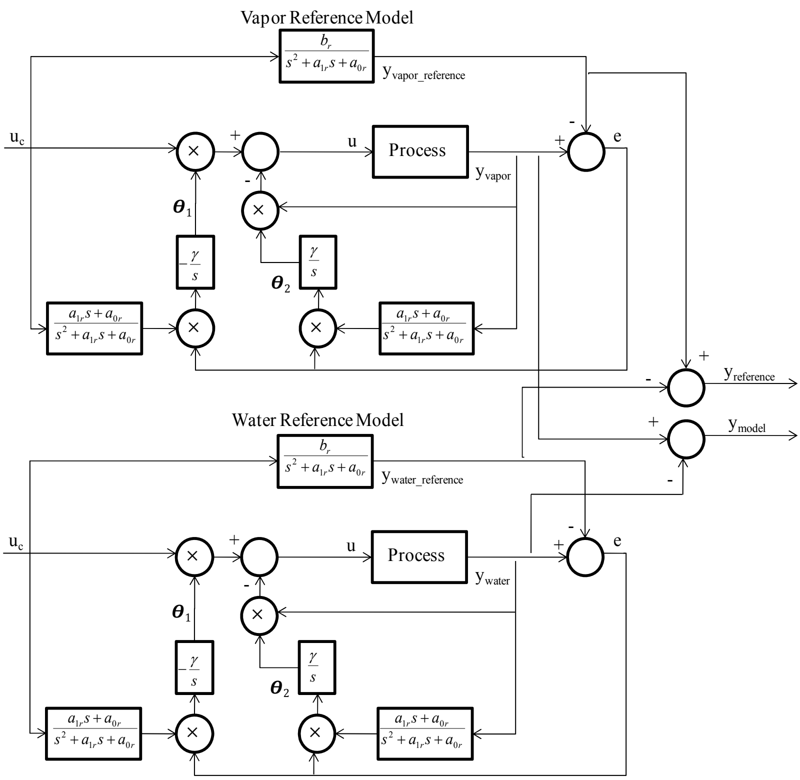


Fig. 14. Fault tolerant MRAC scheme.

4.2 PID Controller

To overcome the limitations of the simple MRAC structure, a classical PID controller, in the form of equation 4, was introduced in the feedforward part of simple MRAC scheme. The resulting structure is shown in figure 15. The desired response is introduced in terms of rise time, stabilization time and overshoot.

$$G_{PID\_controller} = K_p + {K_i}/{s} + K_d s \tag{4}$$

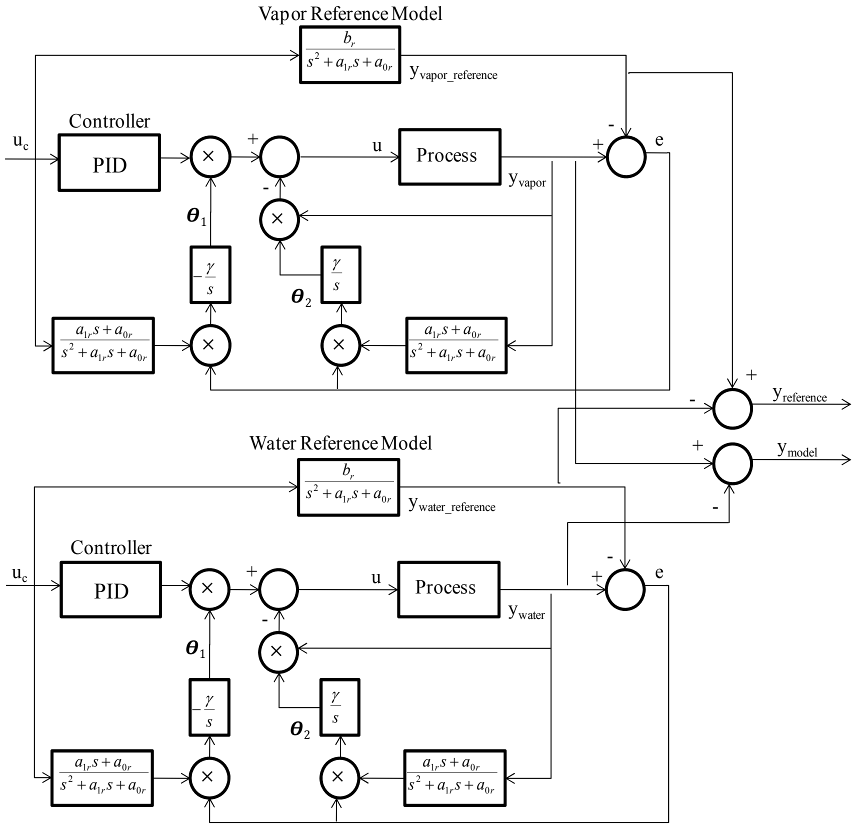


Fig. 15. Fault tolerant MRAC-PID Controller structure.

4.3 Neural Network Controller

In this scheme, we substitute the PID controller by a neural network (shown in figure 16). The neural network was trained with the original process inputs as well as the desired outputs.

4.4 Experiments

Two different faults were simulated: abrupt faults and gradual faults. In the abrupt faults case, the whole magnitude of the fault is developed in one moment of time and was simulated with a step function. On the other hand, gradual faults are developed during a period of time and are implemented with a ramp function.
Both types of faults, abrupt and gradual, can be implemented in sensors (feedback), in which the properties of the process are not affected, but the sensor readings are mistaken. Also, can be implemented in actuators (process entry) in which the process properties are not affected either, but the process behavior can change or can be interrupted.
We use an industrial heat exchanger to test our approach (shown in figure 17). The process has two inputs: water and steam flows controlled by pneumatic valves, and one output, the water temperature measured by a termocouple. Variations in water and steam flows are determined by flow transmitters. To develop the continuous model of this process, an

identification experiment was performed, where a Pseudo Random Binary Sequence (PRBS) was applied to water and steam valves, and variations in water temperature were recorded.

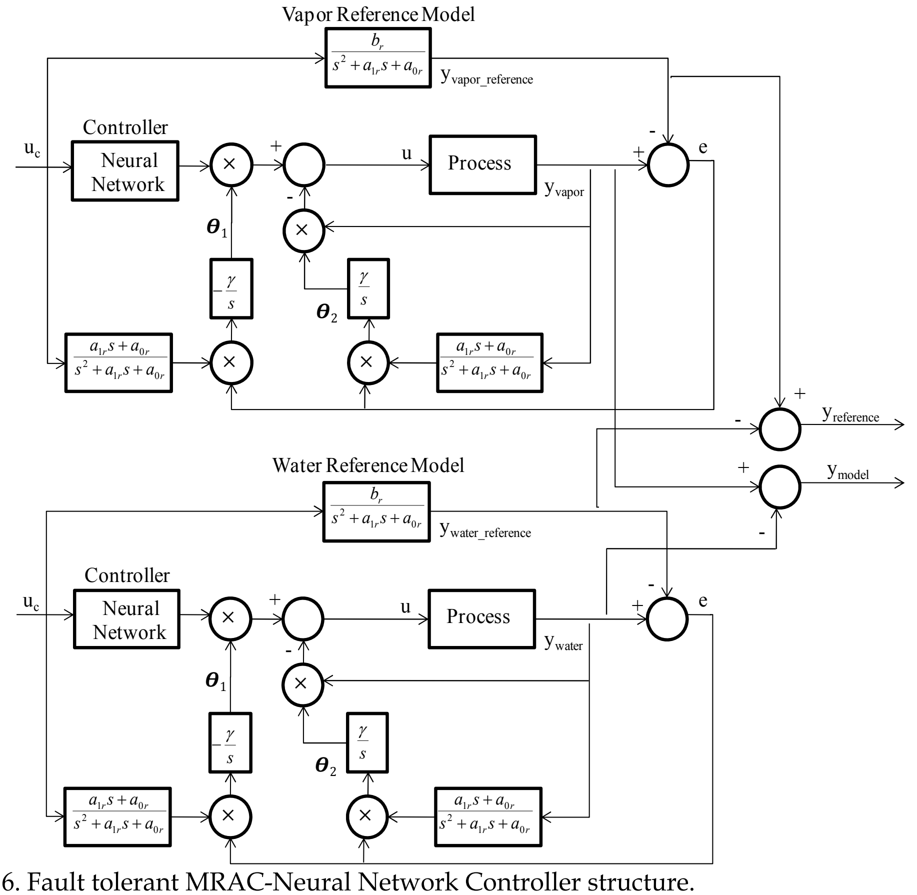


Fig. 16. Fault tolerant MRAC-Neural Network Controller structure.

With the data obtained in the PRBS test, the identification was achieved using software developed in Matlab®.

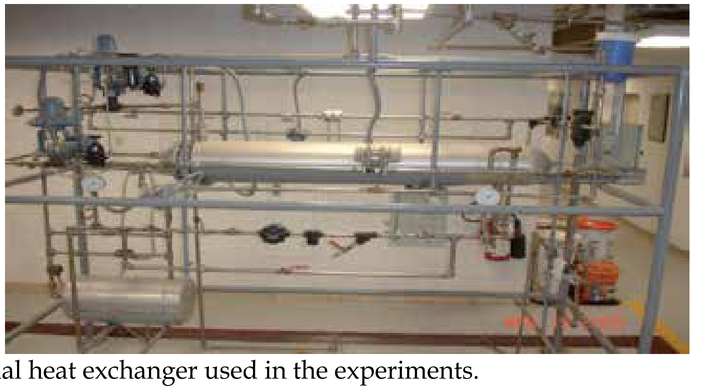


Fig. 17. Industrial heat exchanger used in the experiments.

The following model was obtained:

$$G_p = G_{vapor} - G_{water}$$

$$G_p = \frac{0.00002}{s^2 + 0.004299s + 0.00002} - \frac{0.000013}{s^2 + 0.007815s + 0.00008} \tag{5}$$

This model was used to implement the 3 different FTC Schemes mentioned above.

4.5 Results

A total of six different experiments were developed in Simulink®. Table 1 explains the results of each simulated experiment and shows the numerical performance of every method by using the Mean Square Error (MSE) obtained during the application of the fault.

Method & Fault Type	*Results when a the Fault was applied In Sensor*	*Results when an Fault was applied In Actuator*
MRAC-Abrupt Fault	- If the fault magnitude is < 0.44, the system is robust against the fault. - If the fault magnitude is between [0.44, 1.52] the system accommodates the fault. -If the fault magnitude is > 1.52, the system becomes unstable. - **MSE= 0.501236189**	- If the fault magnitude is 1 the system response varies around +/- 3%. This means that the system is degraded but still works. This degradation becomes smaller over time, because the system continues accommodating the fault. - **MSE=0.10521016**
MRAC-Gradual Fault	- If the fault has saturation < +/- 0.44, the system is robust against the fault. - If the fault has a saturation between +/- [0.44, 1.52] the system accommodate the fault. -If the fault has saturation > 1.52, the system becomes unstable. **MSE=0.50777113**	- If the fault saturation is +/- 1 the system response varies around +/- 4%. This means that the system is degraded but still works. This degradation becomes smaller over time, because the system continues accommodating the fault. - **MSE=0.09163081**
MRAC-PID-Abrupt Fault	- If the fault magnitude is < 1.75, the system is robust against the fault. - If the fault magnitude is between [1.75, 1.97] the system accommodates the fault. -If the fault magnitude is > 1.97, the system becomes unstable. - **MSE=0.00036942**	- If the fault magnitude is 1 the system response varies around +/- 3.5%. This means that the system is degraded but still works. This degradation becomes smaller over time, because the system continues accommodating the fault. - **MSE=0.13283874**
MRAC-PID-Gradual Fault	- If the fault has saturation < +/- 1.75, the system is robust against the fault. - If the fault has a saturation between +/- [1.75, 1.97] the system accommodates the fault. -If the fault has saturation > 1.97, the system becomes unstable. - **MSE=0.0003694**	- If the fault saturation is +/- 1 the system response varies around +/- 4%. This means that the system is degraded but still works. This degradation becomes smaller over time, because the system continues accommodating the fault. - **MSE=0.13508005**

MRAC-Neural Network-Abrupt Fault	- The system is robust against sensor faults **- MSE=0.00030043**	- If the fault magnitude is 1 the system response varies around +/- 3%. This means that the system is degraded but still works. This degradation becomes smaller over time, because the system continues accommodating the fault. - **MSE=0.13154736**
MRAC-Neural Network-Gradual Fault	- The system is robust against sensor faults. **- MSE=0.00030043**	- If the fault saturation is +/- 1 the system response varies around +3% and - 4%. This means that the system is degraded but still works. This degradation becomes smaller over time, because the system continues accommodating the fault. - **MSE=0.13149647**

Table 1. Results of experiments with abrupt and gradual faults simulated in the 3 different fault tolerant MRAC schemes.

The following graphs represent a comparison between the different simulated experiments.
Figure 18 represents system behavior when abrupt faults are simulated. The three graphs on the left column are sensor faults and the graphs from the right column are actuator faults. The sensor faults have a magnitude of 1.8 and the actuator faults a magnitude of 1. It is observed that the MRAC-Neural Network represents the best scheme because is insensitive to abrupt sensor faults and has a good performance when abrupt actuator faults are developed.
Figure 19 graphs represent system behavior when gradual faults are present on the system. The fault magnitude of the sensor fault is of 1.8 and the magnitude of the actuator fault is of 1. It can be seen also that the MRAC-Neural Networks Controller scheme is the better option because is robust to sensor faults and has a less degraded performance in actuator faults. In conclusion, the proposed MRAC-Neural Network scheme gives the best fault tolerant control scheme developed in this work.

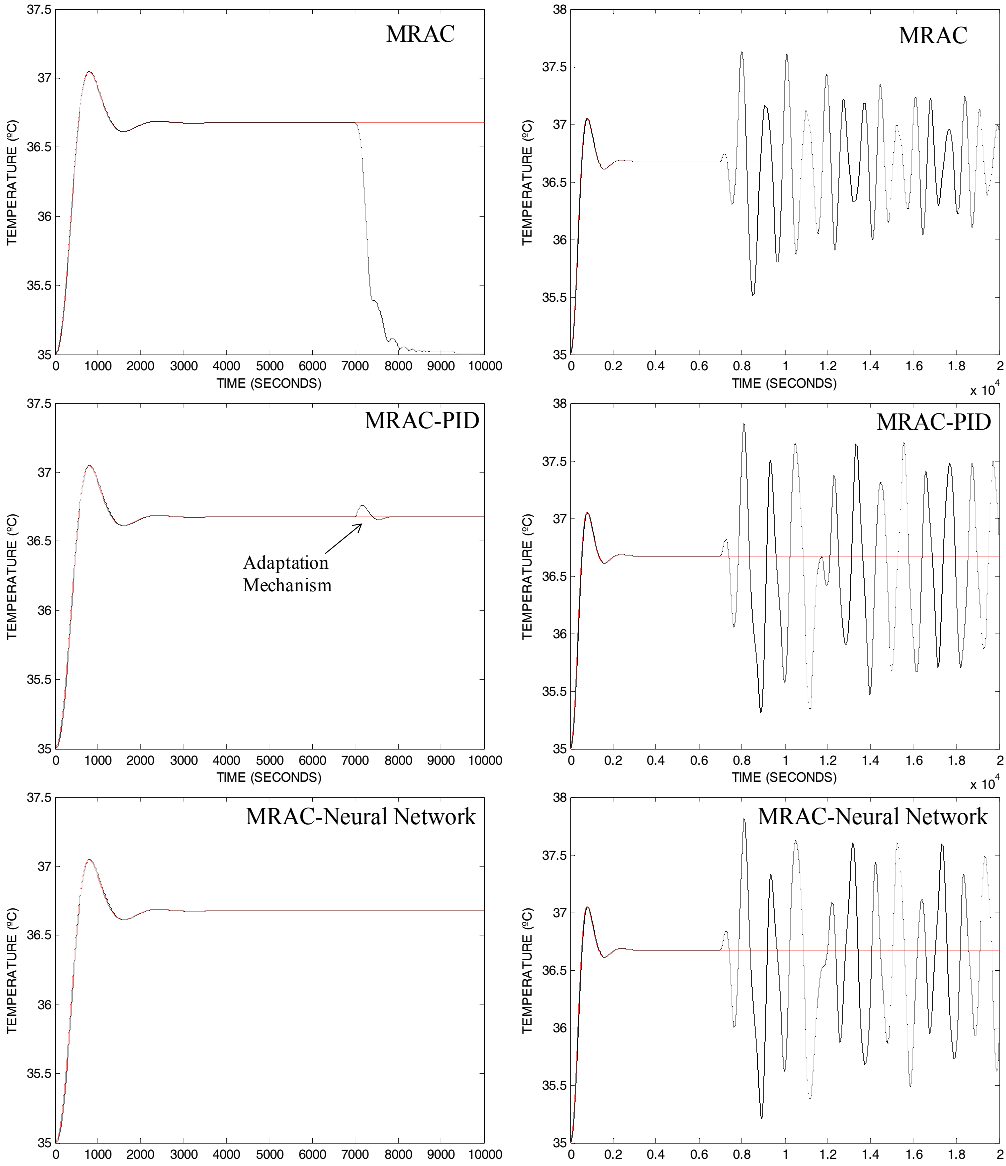


Fig. 18. Abrupt-Sensor Faults (left column) and Abrupt-Actuator Faults (Right column) of the three different proposed schemes, the fault started at time 7000 secs.

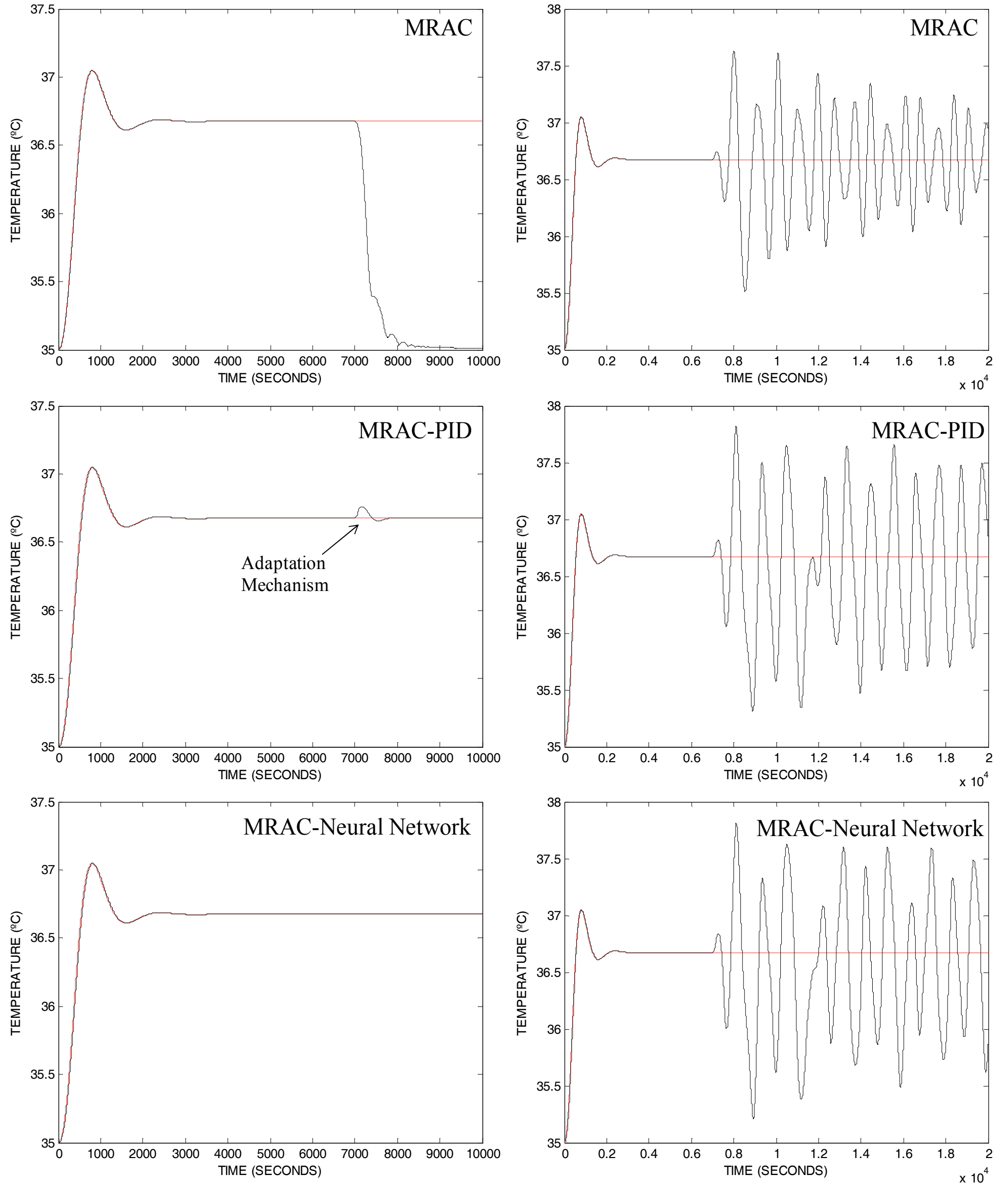


Fig. 19. Gradual-Sensor Faults (left column) and Gradual-Actuator Faults (Right column) of the three different proposed schemes, the fault started at time 7000 secs.

5. References

Ballé, P.; Fischera, M.; Fussel, D.; Nells, O. & Isermann, R. (1998). Integrated control, diagnosis and reconfiguration of a heat exchanger. *IEEE Control Systems Magazine*, Vol. 18, No. 3, (June 1998) 52–63, ISSN: 0272-1708.

Bastani, F., & Chen, I. (1988). The role of artificial intelligence in fault-tolerant process-control systems. *Proceedings of the 1st international conference on Industrial and engineering applications of artificial intelligence and expert systems*, pp. 1049-1058, ISBN:0-89791-271-3, June 1988, ACM, Tullahoma, Tennessee, United States.

Blanke, M.; Izadi-Zamanabadi, R.; Bogh, R. & Lunau, Z. P. (1997). Fault tolerant control systems—A holistic view. *Control Engineering Practice*, Vol. 5, No. 5, (May 1997) 693–702, ISSN: S0967-0661(97)00051-8.

Blanke, M., Staroswiecki, M., & Wu, N. E. (2001). Concepts and methods in fault-tolerant control. *In Proceedings of the 2001 American Control Conference*, pp. 2606–2620, Arlington, Virginia, ISBN: 0-7803-6495-3, June 2001, IEEE, United States.

Blanke, M.; Kinnaert, M.; Lunze, J. & Staroswiecki, M. (2003). *Diagnosis and Fault-Tolerant Control*. Springer-Verlag, ISBN: 3540010564 , Berlin, Germany.

Blondel, V. (1994). *Simultaneous Stabilization of Linear Systems*. Springer Verlag, ISBN: 3540198628, Heidelberg, Germany.

Caglayan, A.; Allen, S. & Wehmuller, K. (1988). Evaluation of a second generation reconfiguration strategy for aircraft flight control systems subjected to actuator failure/surface damage. *Proceedings of the 1988 National Aerospace and Electronics Conference*, pp. 520–529, May 1988, IEEE, Dayton , Ohio, United States.

Diao, Y. & Passino, K. (2001). Stable fault-tolerant adaptive fuzzy/neural control for turbine engine. *IEEE Transactions on Control Systems Technology*, Vol. 9, No. 3, (May 2001) 494–509, ISSN: 1063-6536.

Diao,Y. & Passino, K. (2002). Intelligent fault-tolerant control using adaptive and learning methods. *Control Engineering Practice*, Vol. 10, N. 8, (August 2002) 801–817, ISSN: 0967-0661.

Eterno, J.; Looze, D; Weiss, J. & Willsky, A. (1985). Design Issues for Fault-Tolerant Restructurable Aircraft Control, *Proceedings of 24th Conference on Decision and Control*, pp. 900-905, December 1985, IEEE, Fort Lauderdale, Florida, United States.

Farrell, J.; Berger, T. & Appleby, B. (1993). Using learning techniques to accommodate unanticipated faults. *IEEE Control Systems Magazine,* Vol. 13, No. 3, (June 1993) 40–49, ISSN: 0272-1708.

Gao, Z. & Antsaklis, P. (1991). Stability of the pseudo-inverse method for reconfigurable control systems. *International Journal of Control,* Vol. 53, No. 3, (March 1991) 717–729.

Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, ISBN: 0201157675, Reading, Massachusetts, United States.

Gomaa, M. (2004). Fault tolerant control scheme based on multi-ann faulty models. Electrical, Electronic and Computer Engineering. *ICEEC International Conference*, Vol. , No. , (September 2004) 329 – 332, ISBN: 0-7803-8575-6.

Gurney, K. (1997). *An Introduction to Neural Networks*, CRC Press Company, ISBN: 1857285034, London, United Kingdom.

Holmes, M. & Ray, A. (2001). Fuzzy damage-mitigating control of a fossil power plant. *IEEE Transactions on Control Systems Technology*, Vol. 9, No. 1, (January 2001) 140– 147, ISSN: 1558-0865.

Isermann, R.; Schwarz, R. & Stölzl, S. (2002). Fault-tolerant drive-by-wire systems. *IEEE Control Systems Magazine*, Vol. 22, No. 5, (October 2002) 64-81, ISSN: 0272-1708.

Jaimoukha, I.; Li, Z. & Papakos, V. (2006). A matrix factorization solution to the H-/H infinity fault detection problem. *Automatica*, Vol. 42, No. 11, 1907 – 1912, ISSN: 000-1098.

Jiang, J. (1994). Design of reconfigurable control systems using eigenstructure assignments. *International Journal of Control,* Vol. 59, No. 2, 395–410, ISNN 00-7179.

Karsai, G.; Biswas, G.;Narasimhan, S.; Szemethy, T.; Peceli, G.; Simon, G. & Kovacshazy, T. (2002). Towards Fault-Adaptive Control of Complex Dynamic Systems, In: *Software- Enabled Control*, Tariq Samad and Gary Balas, Wiley-IEEE press, 347-368, ISBN: 9780471234364, United States.

Kwong,W.; Passino, K.; Laukonen, E. & Yurkovich, S. (1995). Expert supervision of fuzzy learning systems for fault tolerant aircraft control. *Proceedings of the IEEE*, Vol. 83, No. 3, (March 1995) 466–483, ISSN: 0018-9219.

Liang, B. & Duan, G. (2004). Robust H-infinity fault-tolerant control for uncertain descriptor systems by dynamical compensators. *Journal of Control Theory and Applications*, Vol. 2, No. 3, (August 2004) 288-292, ISSN: 1672-6340.

Lunze, J. & J. H. Richter. (2006). *Control reconfiguration: Survey of methods and open problems.* , ATP, Bochum, Germany.

Mahmoud, M.; Jiang, J. & Zhang, Y. (2003). Active fault tolerant control systems: Stochastic analysis and synthesis, Springer, ISBN: 2540003185, Berlin, Germany.

Mitchell, M. (1996). *An introduction to genetic algorithms*, MIT Press, ISBN: 0262631857, Cambridge, Massachusetts, United States.

Nagrath, .J (2006). *Control Systems Engineering*, Anshan Ltd, ISBN: 1848290039, Indian Institute of Technology, Delhi, India.

Neimann, H. & Stoustrup, J. (2005), Passive fault tolerant control of a double inverted pendulum - a case study. *Control Engineering Practice*, Vol. 13, No 8, 1047-1059, ISNN: 0967-0661.

Nguyen, H.; Nadipuren, P.; Walker, C. & Walker, E. (2002). *A First Course in Fuzzy and Neural Control*, CRC Press Company, ISBN: 158488241, United States.

Oudghiri, M.; Chadli, M. & El Hajjaji, A. (2008). Sensors Active Fault Tolerant Control For Vehicle Via Bank of Robust H∞ Observers. *17th International Federation of Automatic Control (IFAC) World Congress*, July 2008, IFAC, Seoul, Korea.

Passino, K. and Yurkovich, S. (1997). *Fuzzy Control*, Addison-Wesley Longman, ISBN: 020118074, United States.

Pashilkar,A.; Sundararajan, N.; Saratchandran, P. (2006). A Fault-tolerant Neural Aided Controller for Aircraft Auto-landing. *Aerospace Science and Technology*, Vol. 10, pp. 49-61.

Patton, R. J. (1997). Fault-tolerant control: The 1997 situation. *Proceedings of the 3rd IFAC symposium on fault detection, supervision and safety for technical processes,* pp. 1033–1055, Hull, United Kingdom.

Patton, R.; Lopez-Toribio, C. & Uppal, F. (1999). Artificial intelligence approaches to fault diagnosis. *IEEE Condition Monitoring: Machinery, External Structures and Health, I*, pp. 5/1 – 518, April 1999, IEEE, Birmingham, United Kingdom.

Perhinschi, M.; Napolitano, M.; Campa, G., Fravolini, M.; & Seanor, B. (2007). Integration of Sensor and Actuator Failure Detection, Identification, and Accommodation

Schemes within Fault Tolerant Control Laws. *Control and Intelligent Systems*, Vol. 35, No. 4, 309-318, ISSN: 1480-1752.

Polycarpou, M. & Helmicki, A. (1995). Automated fault detection and accommodation: A learning systems approach. *IEEE Transactions on Systems*, Vol. 25, No. 11, (November 1995) 1447–1458.

Polycarpou, M. & Vemuri, A. (1995). Learning methodology for failure detection and accommodation. *IEEE Control Systems Magazine*, Vol. 15, No. 3, (June 1995) 16–24, ISSN: 0272-1708.

Polycarpou, M. (2001). Fault accommodation of a class of multivariable nonlinear dynamical systems using a learning approach. *IEEE Transactions on Automatic Control*, Vol. 46, No.5, (May 2001) 736–742, ISSN: 0018-9286.

Rumerhart, D.; McClelland, J.; & the PDP Research Group. (1986). *Parallel distributed processing: explorations in the microstructure of cognition*, MIT Press, ISBN: 0262631105, Cambridge, Massachusetts, United States.

Ruan, D. (1997). *Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms*, Kluwer Academic Publishers, ISBN: 0792399994, United States.

Schroder, P.; Chipperfield, A.; Fleming, P. & Grum, N. (1998). Fault tolerant control of active magnetic bearings. IEEE International Symposium on Industrial Electronics, pp. 573-578, ISBN: 0-7803-4756-0, July 1998, IEEE, Pretoria, South Africa.

Skogestad, S., & Postlethwaite I**.** (2005). *Multivariable Feedback Control-Analysis and Design*, John Wiley & Sons, ISBN: 9780470011676, United States.

Staroswiecki, M. (2005). Fault tolerant control: The pseudo-inverse method revisited. *Proceedings 16th IFAC World Congress*, pp. Th-E05-TO/2, IFAC, Prague, Czech Republic.

Steffen, T. (2005). *Control reconfiguration of dynamic systems: Linear approaches and structural tests*, Springer, ISBN: 3540257306, Berlin, Germany.

Stengel, R. (1991). Intelligent Failure-Tolerant Control. *IEEE Control Systems Magazine*, Vol. 11, No. 4, (June 1991) 14-23, ISSN: 0272-1708.

Sugawara, E.; Fukushi, M. & Horiguchi, S. (2003). Fault Tolerant Multi-layer Neural Networks with GA Training. *The 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI systems*, pp. 328-335, ISBN: 0-7695-2042-1, IEEE, November 2003 Boston, Massachusetts, United States.

Venkatasubramanian, V.; Rengaswamy, R.; Yin, K. & Kavuri, S. (2003a). A review of process fault detection and diagnosis. Part I. Quantitative modelbased methods. *Computers and Chemical Engineering*, Vol. 27, No. 3, 293–311, ISSN-0098-1354.

Venkatasubramanian, V.; Rengaswamy, R. & Kavuri, S. (2003b). A review of process fault detection and diagnosis. Part II. Qualitative models and search strategies. *Computers and Chemical Engineering*, Vol. 27, No. 3, 313–326, ISSN: 0098-1354.

Venkatasubramanian, V.; Rengaswamy, R.; Kavuri, S. & Yin, K. (2003c). A review of process fault detection and diagnosis. Part III. Process history based methods. *Computers and Chemical Engineering*, Vol. 27, No. 3, 327–346, ISSN: 0098-1354.

Wang, H. & Wang, Y. (1999). Neural-network-based fault-tolerant control of unknown nonlinear systems. *IEE Proceedings – Control Theory and Applications*, Vol. 46, No. 5, (September 1999) 389–398, ISSN; 1350-2379.

Yang, G. & Ye, D. (2006). Adaptive fault-tolerant Hinf control via state feedback for linear systems against actuator faults, *Conference on Decision and Control*, pp. 3530-3535, December 2006, San Diego, California, United States.

Yen, G. & DeLima, P. (2005). An Integrated Fault Tolerant Control Framework Using Adaptive Critic Design. *International Joint Conference on Neural Networks*, Vol. 5, pp. 2983-2988, ISBN: 0-7803-9048-2.

Zhang, D.; Wang Z. & Hu, S. (2007). Robust satisfactory fault-tolerant control of uncertain linear discrete-time systems: an LMI approach. *International Journal of Systems Science*, Vol. 38, No. 2, (February 2007) 151-165, ISSN: 0020-7721.

Zhang, Y., & Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Elsevier Annual Reviews in Control*, Vol. 32, (March 2008) 229-252.

16

A Real Time Expert System For Decision Making in Rotary Railcar Dumpers

Osevaldo Farias, Sofiane Labidi, João Fonseca Neto, José Moura and Samy Albuquerque
Federal University of Maranhão and VALE
Brazil

1. Introduction

In a great deal of industrial production mechanisms approaches able to turn automatic a wide range of processes have being used. Such applications demand high control pattern, tolerance to faults, decision taking and many other important factor that make large scale systems reliable (Su et al., 2005), (Su et al., 2000) and.

In particular, Artificial Intelligence (AI) presents a wide applicability of those approaches implementing their concepts under the form of Expert Systems (Fonseca Neto et al., 2003). Applications with this architecture extend knowledge-based systems and allow the machine to be structured into a model apt to act and behave in the most similar way a human specialist uses its reasoning when facing a decision taken problem (Feigenbaum, 1992).

The VALE production system comprehends several mining complexes, among which is notorious the Ponta da Madeira Dock Terminal (PMDT). In this complex macro level processes of Unloading, Storing and Minerals Shipping are performed, supervised by a very reliable Operational Control Center (OCC).

This article discusses the development of an on-line expert system applied to decision taken when facing faults occurred in the VV311-K01 used to unload minerals at the VALE's PMDT. This project attends the handling of a large quantity of available operative data created at production time, and cares of the organization, interpretation and understanding of these data.

Besides automation technologies, in order to attend our proposal, we apply some information technologies such as: the JESS, the JAVA language and also XML (eXtensible Markup Language) aiming the real time running of the Expert System.

This article is organized as follows: Section 2 describes the Expert System proposal; in Section 3 are described the particularities and the operation of the rotary railcar dumper system, the real time hardware and the monitoring performed by the supervisor system. Faults occurrence is also described starting from the behaviour of the VV311-K01 rotary railcar dumper. In Section 4 are detailed the Expert System Development steps using techniques of Knowledge Engineering within the context of CommonKADS methodology. In addition, in this Section are also presented resources of the JESS environment used as inference motor for the system's decision module, the system's application and implementation global architecture and the final remarks.

2. Expert System Proposal

The system's proposal is to reach the decision process considering as input the faults detected by the VV311-K01 rotary railcar dumper system components, aiming at furnishing enhancement and speed to the decisions to be taken when facing faults in the minerals unloading system.
The faults identification actually is obtained through Microsoft electronic spreadsheets and Access database analysis. This means a lot of operative data and potential information that have not integration with VALE's Plant Information Management System (PIMS). The decision process in order to achieve the possible solutions for a fault in VV311-K01 positioner car, the engineers and technician team need to deal with several relevant devices tracing it fault mode, effects and it related causes. Stated another way this is made according to follow model.

$$H_{fault} : x_{devices} \rightarrow y_{relevant}$$

Being x the set of VV311-K01 devices or subsystems. The Expert System propose consider the plant devices mapping dealing and inferring the functional relationship (i.e. fault-device) between the set of plant devices and faults mode. By example:

$$x = \{x_i \mid devices\}$$

Being x_i , **shaft**, **engine**, **sensors**, coupling, shock absorbers and furthermore VV311-K01 car positioner devices. Associated to this propose, these sets are inputs to begin the system modelling and discovery in which conditions the decision making procedure is sustained. In addition, the Expert System is built by using the AI symbolic reasoning paradigms (Luger and Stablefield, 2008) to be modelled for the industrial sector.
Notice that the Expert System considers the VV311-K01 significant characteristics based upon the knowledge of experts and the domain agents (i.e. engineers, operation analysts and operators), during positioner car operation **in order to improve the unloading system's productivity** along the execution of the involved tasks at the VALE industrial complex.

3. The Rotary Railcar Dumper System

The minerals unloading mechanism initiates at the rotary railcar dumper with the arrival of the locomotive pulling behind it 102 to 104 rail-wagons that will be positioned in the dumper, and from there on the goal of each rotary comes to be the unloading of 2 rail-wagons per iteration. That iteration is the time the positioner car needs to fix the rail-wagons in the dumping cycle.
To attain the rotation a positioner car fixes the rail-wagons in the rotary and this, consequently, unloads the material by performing a 160° rotation – it can eve be programmed to rotate up to 180° - in the carrier-belts (Fonseca Neto et al., 2003). Remember that while the rail-wagons material is been unloaded and at the same pass as the positioner car is already returning to fix the next rail-wagons, the railroad-cargo is kept immobilized by means of latches, until the rail-wagons that are in the rotation are freed.

3.1 Real Time Hardware

The physical components of the devices that command the dumper are typically compounded by peripherals such as inductive and photoelectric sensors, charge cells, presostates and thermostats, limit electromechanical switches and cam switches.
Really, dumper's peripherals play an important role in the behaviour of the following functions: displacement stop or interruption, position signalling, pressure and temperature monitoring, beside other aspects characterized in this context.
Thus, rail-wagons dumper's hardware are potentially something like an intermediate layer (i.e. a middleware) important for the communication between the Expert System and the VV311-K01 hydraulic and mechanical components at the operation time.

3.2 Supervision Control System

Supervision is conducted by means of the programmable logic controllers (PLCs) which receive all the information from the dumper hardware through input cards, commanding also the Motors Control Centre (MCC) through output cards. In the dumper, the programmable logic controllers command actuators and action drives (converters).
The programming, developed in LADDER, is structured in such a way that the first mesh are destined to the faults; to the command mesh and finally to the output mesh. The program is developed in subroutines by moves, with one subroutine for each component (e.g. positioner car, rotation, latches and etc.) present in the dumper. The command mesh was developed such that they depended only on the supervisory command to be closed.
The Operational Process is supervised by the Supervisory Control and Data Acquisition (SCADA), a system composed by two servers that run the InTouch software from Wonderware and by four clients that collect data for the SCADA system through the Dynamic Data Exchange (DDE) from Microsoft.

3.3 Faults occurrence

The faults that occur in the production process and in the system's stopping for a long period of time due to equipments overloading, sensors defaults and problems with other component sets of the rotary railcar dumper, have currently caused much financial damage to the VALE industrial pole, based on the monthly unloading average of the VV311-K01, which is around 16120 rail-wagon cycles (i.e. 155 trains, each with 208 rail-wagons).
Among the faults in the dumper, most of them occur at the positioner car once, according to the statistical VALE reports, this component can be responsible for the reduction in the monthly average in 1095 cycles of rail-wagons.
From this information, the VV311-K01 positioner car was selected as one of the critical points to be analyzed in already mentioned production sector.

4. The Expert System Development

Before initiating the Expert System developing stages, it is necessary to select some important characteristics that will be used to build the system, such as the JESS and the CommonKADS methodology.

4.1 JESS

The JESS is a tool for constructing the Expert System developed by Friedman Hill at Sandia National Laboratories. The JESS is totally developed in JAVA, and is characterized as an API for creating the expert Systems based on production rules. Its architecture involves cognition components defined like: Inference Engine, Agenda and Execution Engine. All these structures catch assertions or domain facts and also create new assertions.
The inference JESS engine is constituted by the Pattern-Matching mechanism (i.e. patterns joining) that decides which rules will be activated. The Agenda programs the order in which the activated rules will be fired, and the Execution Engine is in charge of the triggering shot (Friedman-Hill, 2006). Besides that, such rules can contain function callings that care of code statements in JAVA.
In JESS the facts have attributes or fields called slots, which must be grouped in templates in order to keep common feature assertions, and have some of their properties grouped in classes like Object-Oriented.
The reasoning formalism used by the JESS presents rules composed by if...then patterns, represented by the LHS (Left-Hand Side) and RHS (Right-Hand Side), respectively.The inference process is given by the Rete algorithm (Forgy, 1982) that combines the facts according to the rules and selects the one that will be shot to execute their corresponding actions.
Having JESS as decision core, the Expert System will operate by matching the facts, which are the right statements on the attributes contained in the VV311-K01 knowledge base, with the rules that translate the domain of the agent's explicit knowledge of the VALE unloading system's.

4.2 CommonKADS

The historical scope of the CommonKADS methodology was confirmed by the results of several projects of the ESPRIT program for building knowledge based systems. Even though it was conceived at the Amsterdam University, initially under the name KADS (Knowledge Acquisition Design System), it referred to a method for knowledge acquisition; later some contributions papers and European Science Societies developed various knowledge systems through it. As a consequence of the good results obtained with the KADS technique, they decided to expand it towards a set of techniques or methods applied to all development phases of systems based upon knowledge, creating the CommonKADS methodology, becoming acknowledged by several companies as a full pattern for knowledge engineering (Labidi,1997).
Products arisen from Expert Systems development that use this methodology are the result of the performed phases modelling activities, and characterize the input artifacts for the successive refinements undergone in the next steps of the CommonKADS life cycling.
Having in hands the particularities that will be used in the Expert System building, the steps of the system with actions such as Acquisition and Knowledge representation are organized– also including the analysis phase – Rules representation – ruling the Design phase – and the System's Settling– satisfying the settling phase.

4.3 Acquisition and Knowledge representation

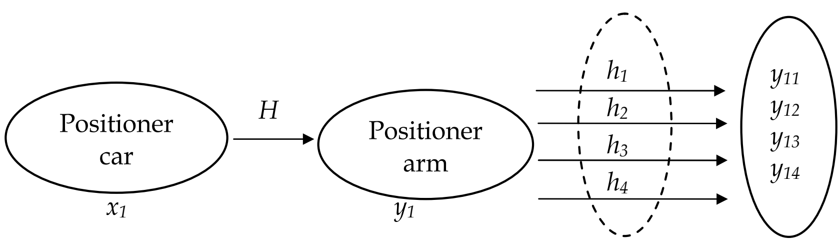
Knowledge acquisition is the most important step when developing Expert Systems, and aims at the detailed attainment of the knowledge used by the expert to relate problems. All the knowledge elicitation was done by means of interviews with the expert through information kept in the operational reports, spredsheets and off-line database. The method used to the knowledge representation was built based upon production rules. These rules map the knowledge of the VV311-K01 operation expert onto computing artefacts take into consideration the set of relevants faults (i.e. y) instance and its generator sources, modeling the conditions in which the faults deduction can points out the diagnosis or support the expert's decision making. Highlighting the relevant fauls, they establish a vector in which the positoiner car devices are relevant faults attributes according to following set.

$$y = \left[y_1 \, y_2 \, y_3 \right]$$

In this set, y_1 is the kind of generator source, y_2 is the priority and the y_3 is the historic, reminding that only generator source is treated in this chapter. In order to undestand the relevant faults model instance, was considered the car positioner in agreement with the following set.

$$y(x) = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Being x the VV311-K01 positioner car and y, the set of faults belongs to it, y_1 is the engines situation, y_2 positioner arm situation, y_3 latches situation and y_4 coupling situation. These mathematical elucidation are early requirements to understand the amount of situations that enginers and technicians team have to deal during the productivy system. The following model represents the possibilites during a decision making scenario.



In fact, all this situations and conditions will be handled by JESS inference engine. The JESS handles the knowledge representation as production systems and rules them like condition-action pairs. A rule condition is treated as a pattern that decides when it can be applied, while the action will define the associated problem solution step (Friedman-Hill, 2003). In this way, there were defined the sort of problems presented by the positioner car, along the mineral unloading process, for the elaboration of production rules.

There were observed the main concepts related with the dumper's positioner car along activities in the operational productive system, aiming at getting knowledge elements description by elaborating the organizational model that complements the CommonKADS (Breuker et al., 1994) Analysis phase.

SLOT	OPPORTUNITIES	PROBLEMS
Situation	Engine	Vibration
		Broken Rollers
		locked
		Lack of voltage
	Positioner arm	Short-circuit
		Broken fixing screws
		Broken Counter-bolts
	Latches	Disruption
		Infiltration
		Internal Fugue
		Insufficient outflow
	Coupling	High oil level
		Low oil level

Table 1. Organization Model.

The domain facts and experiences deal with the equipment situation and the potential causes that promote the main system stopping or reduce its productivity. Therefore, in Table 1 is presented the organizational model, correlating problems and opportunities that can be solved or enhanced by the Expert System from which extracted the identified slots for building the VV311-K01 templates. The slot called 'Situation' is one of the units that comprise the templates for representing the knowledge in the JESS inference engine (Friedman-Hill, 2006).

SLOT	INFERENCE LEVEL	TASK LEVEL
Cause	Vibration	Motor basement snap
		Resonance
		Bend axle
	Short-circuit	Terminal out of order
		Low isolation
		Falling's wire material
	Insufficient outflow	Worn Pump
		Obstructed Tabulation
		Safety Valve with insufficiently fixed
	Obstruction	Dirt

Table 2. Knowledge Model.

It was observed that the causes that lead the dumper to reach certain circumstances are pointers for guiding what must be done as to specify derivations that constitute a method for the VV311-K01 positioner car problem resolution, and the strategies to attain this solution. The efforts spent in this stage are described through the knowledge model of the CommonKADS methodology, as shown in Table 2.

According with (Labidi, 1997), the inference and task levels are layers that describe the expert Knowledge; thus, the model in Table 2 constitutes a set of knowledge instances on the VV311-K01 positioner car component. Starting from the model in Table 2, in order to better characterize the system's knowledge mechanism in agreement with the CommonKADS methodology, the activities organized in the inference model presented in Figures 1, were decomposed.

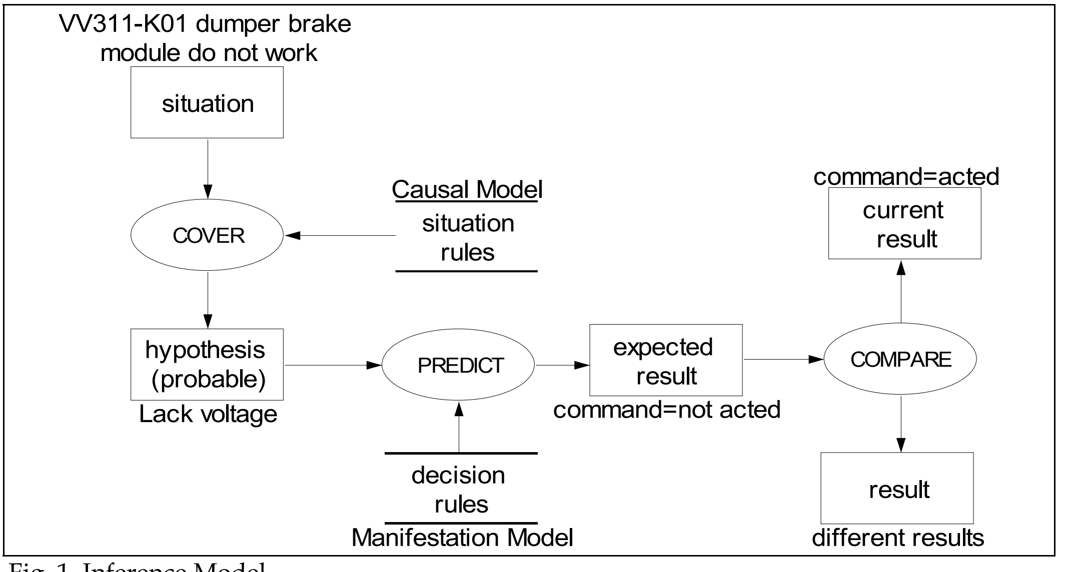


Fig. 1. Inference Model.

This model aims at elaborating a declarative specification of input and output properties, and the inference actions used in the Expert System reasoning. The Inference Model in Figure 1 describes a deduction example done for the VV311-K01 positioner car component, and can be explained as follows: the knowledge's roles are functional names that capture the elements participant in the reasoning process as diagnostic the positioner car state, and can present variable actual status (i.e. temporal stopping, overheating, etc). Inference actions assume as inputs static roles, represented by the manifestation and causal models.

Within the causal model, the rules relate the positioner car fault modes taking into account their attribute's values, while in the manifestation model are reunited the production rules that express their responsibilities through the attributes' values, which satisfy some given conditions (Labidi, 1997).

The COVER, PREDICT and COMPARE inference concepts, represent reasoning axioms that will be mapped by the JESS inference engine used in the Expert System. Basically, is done a transition from abstract concepts as input artefacts to synthesized concrete concepts in a set of assertions as output artefacts (Friedman-Hill, 2003).

Once we have in hands the analysis of the main elements that model the general goal of the knowledge specification stage, according to the CommonKADS and taking into account the granularity of the acquired information for the VV311-K01, significant levels of detail were obtained for representing the knowledge, under the form of production rules.

4.4 Rules Representation

After establishing the knowledge sources, the effort used in this step are linked to the structuring of rules of the type if...then. Firstly, the selection of this type of representation was justified in function of the shell JESS inference engine promotes the building of rules in production systems, and also as a consequence of the attained results in a system built by (Fonseca Neto et al., 2003), which used production rules to detect faults in the mineral belts conveyors at the VALE PMDT in off-line mode.

The example below shows the 'dumper' template that will treat the attributes 'situation' and 'cause' at the Expert System knowledge base.

```
 (deftemplate dumper
(slot situation (default NF))
(slot cause (default NF))
)
```

The 'dumper' template gathers the characteristics considered in the interviews for the possible situations (new status) that the VV311-K01 can generate and the pre-attributed values – NF: Not Found – in events within which there is no chance of pattern unification (i.e. pattern-matching).

In the 'decision' template are grouped the possible causes as particularities that can correspond to 'dumper' template situations (current status) expressed by means of the sensors. The next template was also defined with slots and predefined default valued.

```
(deftemplate decision
   (slot dec-sit (default NF))
   (slot dec-cause (default NF))
   (slot dec-decision(default NF))
 )
```

After the templates specifications, the rules were built taking as basement their slots definition. These slots in the JESS rules structure are part of the so called LHS patterns (i.e. premises) and will be unified (pattern-matching) by the inference engine through the Rete algorithm. The templates and the rules were built with the 7.0p1 version of shell JESS, licensed for academic use.

4.5 System Implementation

For the system's encoding it was built an executable little version of the Expert System as a means for finding functionalities omitted during the interviews looking for reminiscent knowledge. The tool selection was also done by means of the linguistic resources that the JESS makes available in terms of inter-operationability and portability that can be added to the VALE operational productive system.

In order to facilitate the systems operation handling, the Expert System decision module performs a scanning on the faults detected by the sensors present in the VV311-K01 instrumentation. Through the checking of these faults addresses, which are generated by the programmable logical controllers and mapped into faults tagnames stored in the relational database, the Expert System rules that active a given condition are activated in the working memory of the JESS inference engine. Tagnames (e.g. AFF_CEP_F01@VV311K01) are part of historical registry from PIMS. In the Expert System they plays input data for the rules that deduce the situation of the VV311-K01 components pass through, that is why they form the rules LHS pattern in the JESS language syntax, as can be seeing in the excerpt bellow:

```
 (defrule rule198
```

```
(test
(eq TRUE(actedTag "AFF_CEP_F01@VV311K01")))
=>
(store RESULT198 "Loose-Wire-Connection")
(assert (decision(decCausa Loose-Wire-Connection)
(decPrint "Loose-Wire-Connection"))))
```

The above rule presents in its LHS pattern, a actedTag function, which returns true for 1 and false for 0, according to the result read in the XML file generated by stored procedure into the PIMS (Plant Information Management System) oracle database server (see Figure 2), location at which are stores the faults tagnames. The XML document is shown as following structure:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <tags>
 - <tagname id="ASC_B11@VV311K01">
       <value>0</value>
   </tagname>
 - <tagname id="AIN_ALI_001@VV311K01">
       <value>1</value>
   </tagname>
 - <tagname id="AFL_BEP_001@VV311K01">
       <value>0</value>
   </tagname>
 - <tagname id="ATA_BRT_M01@VV311K01">
       <value>0</value>
  </tagname>
 ...
  </tags>
```

The function 'actedTag' assumes as input the description of a tagname to check whether its value acted or not using set and get methods upon JAVA call functions. The 'actedTag' function can be seen in JESS language statement as following:

```
 (deffunction actedTag (?tagName)
    (bind ?sb (new BeansTag))
    (definstance beans ?sb)
    (call ?sb setTgName ?tagName)
    (if (eq (call ?sb getTgValue)1)
         then return TRUE else
         (if (eq (call ?sb getTgValue)0)then return FALSE))
)
```

Starting from an acted tagname, the rule's LHS pattern is evaluated by the conditional function test (Friedman-Hill, 2006) – which is part of the own JESS language – responsible for determining that this pattern will only be unified if the result of the actedTag function returns true. Taking as example the following parcel of the system's decision rule, is possible to observe that the cause 'Loose-Wire-Connection' deduces that the ultrasound on VV311-K01 must be made. The RHS pattern of the rule's parcel stores the result through the command store, and next inserts in the 'decision' template, the action to be done.

```
 (defrule rule204
(decision (decCause Loose-Wire-Connection))
 =>
(store RESULT204 "Do ultrasound on VV311-K01")
```

```
(assert (decision
         (decDecision  Do-ultrasound-on-VV311-K01)
         (decPrint  "Do ultrasound on VV311-K01")
)))
```

The architecture components collaboration is achieved by integration of JESS inference engine and PIMS database server, which is supplied by historical database connected to VV311-01 throughout PLC device. The key point is the usage of the XML pattern aiming convert the SQL queries from PIMS database to knowledge base.

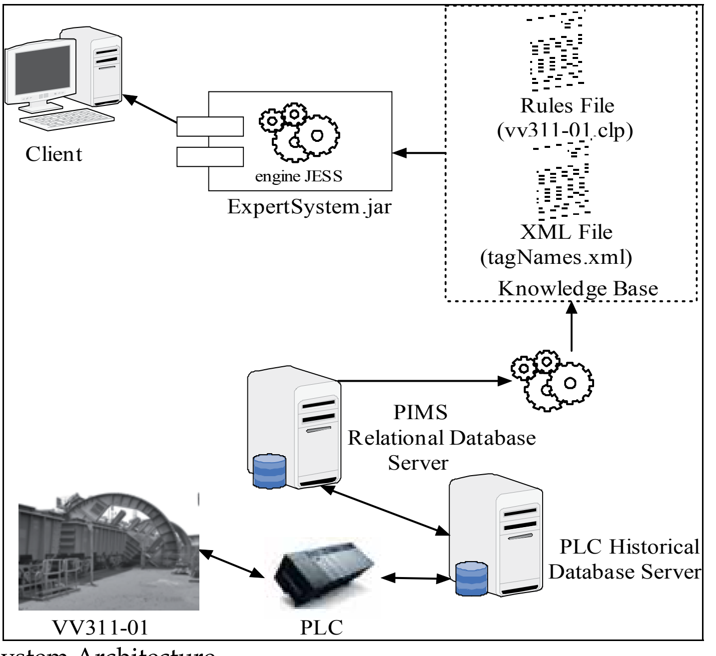


Fig. 2. Expert System Architecture.

Aiming to highlight the global and external view of the system, in Figure 2 was described the Expert System architecture and the nodes distribution that form its structure in order to summarize the Expert System solution and its knowledge base composition.

At the end of the performed deductions, in the shell JESS working memory is shown a window with the recommendations that were stored by the command store (Friedman-Hill, 2006), based in the causes that led to the inference process pointed out by the system, along the responses given by the user. Figure 3 shows the decision making delivered by the system.

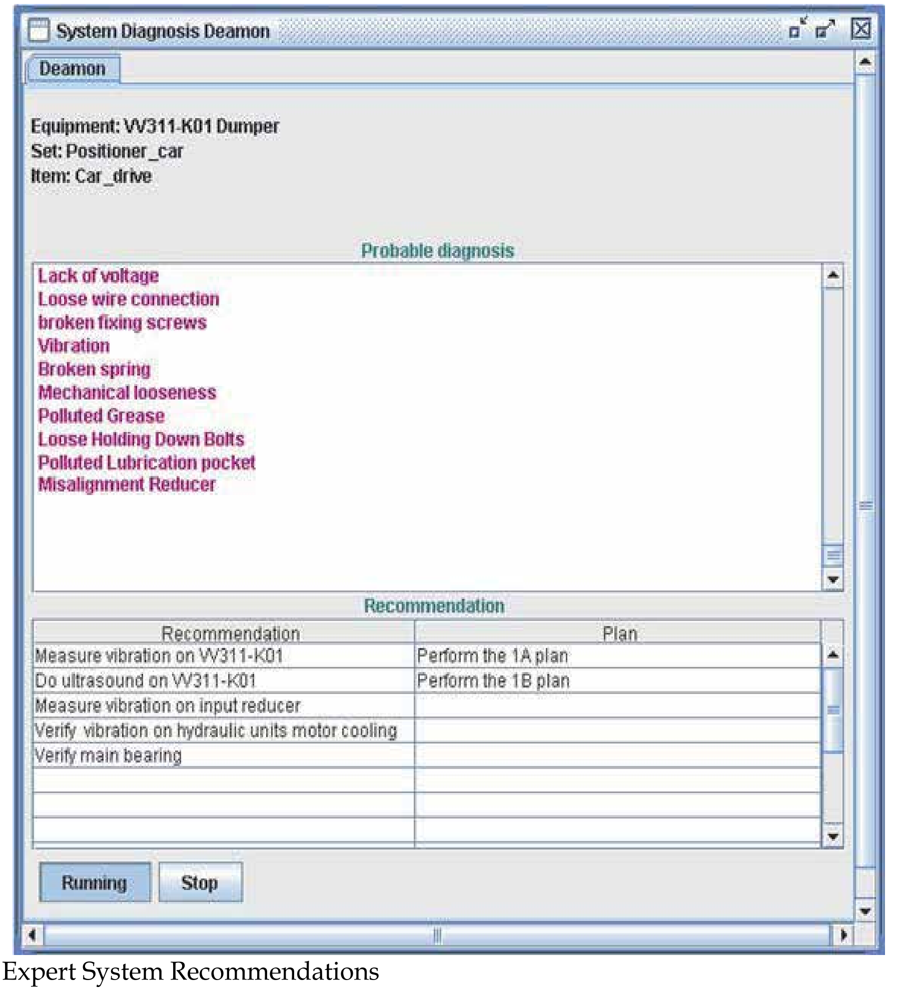


Fig. 3. Expert System Recommendations

Foreach plan recommended, the Expert System detail the needful activities in order to its running. Besides, the amount of people that will perform the tasks related to the faults. By example, in Figure 3 the necessary information to measure vibration onVV311-K01 is listed by class of maintain plans.

Starting from recommendation, the user (technician or engineer) can access its details by double clicking over plans line, according to Figure 4. Through this screen is possible to perform the activities related with decisions and maintain plans lunched by Expert System. Notice that these plans comprise maintain methods developed by VALE's especialists and include all background acumulated by themself too. In fact, this feature can promote the activity decomposition by steps, the labor envolved with each recommendation.

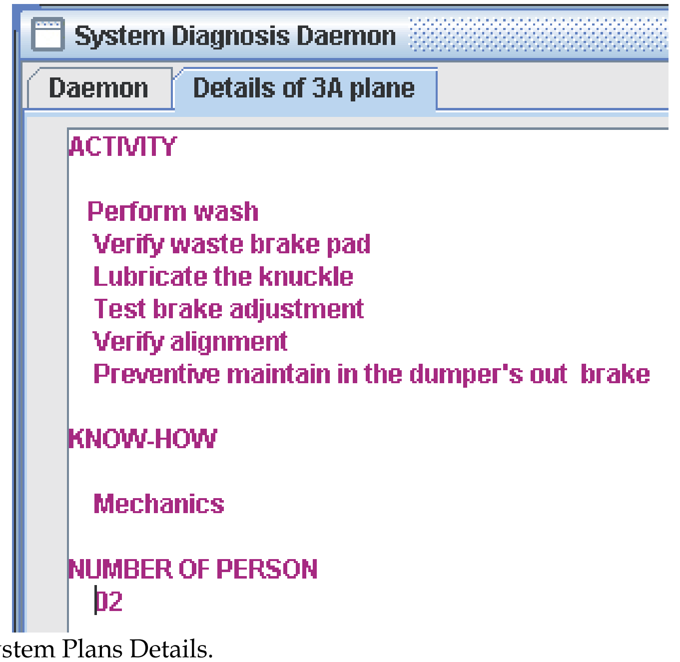


Fig. 4. Expert System Plans Details.

The recommendations viewed in Figure 5, represent the rules that were activated in the working memory of the JESS engine were triggered (shot) only because they got the unification of patterns present in its structure, deduced by the Rete algorithm. The control structure used for the Expert System rules chaining was based on backward chaining (Friedman-Hill, 2003).

5. Experiments and Results

The developed expert system approach is focused in the information and automation architecture and infrastructure. For the first results obtained through its putting into operation, the system was set on the corporative mineral unloading mesh of the VALE Company at the OCC installments.

The system's performance while processing operative data in the JESS inference engine was considered satisfactory once the search frequency of such data elements was tested at 5 s. interval for the deduction of 250 rules, although its standard configuration requires 5 minutes of response time, in order to update in the working memory of the expert system the events occurred at the PIMS server.

The speed and proper timing obtained in terms of the updating processes of the expert system rules base was due to the use of the XML technology as to feed the systems' knowledge base. In addition, the expert system is nor directly connected to the PIMS server, thus avoiding the expenditure of computational resources due to the flux of information between the application and the VALE's data manager, once all the programming necessary for generating the file XML was implemented in the own database.

The comparison of the results between the system and the human expert was characterized by the need of fast and reliable decision taking, fact that allowed operators and maintenance teams to minimize the time spent from the fault's detection to its correction. These efforts are shown in Box 1.

Set	Fault	Is it correctly detected by technician?	Is it correctly detected by Expert System?
Arm command	Damaged contact block	Yes	Yes
Car command	Lack of Measuring	No	Yes
Drive command	Pulley out of place	Yes	No
Drive command	Defective switching	Yes	Yes
Drive command	Vibration	No	Yes
Drive command	Heat of rollers	Yes	Yes
Arm command	Misalignment	No	No
Drive command	Burn	No	Yes

Table 3. Comparison between Expert System and Technician.

As a critical point observed during the results comparison are highlighted atypical events such as a slight misalignment and pulley out of place which was noticed by the system, but did not work in the next cycle. These experiments are very important because the results can contribute to modify the knowledge base inference logic in order to improve these decisions which human expert still have some advantage over the system.

6. Conclusion

The system developed in this work presented the conception and automation of the strategic knowledge required by VALE's mineral unloading system activities. Considering that minerals exploitation is a task that requires a vast specialized knowledge, starting from the domain's understanding, knowledge representation based on production rules using the JESS inference engine was got.

The building of the Expert System in JESS, turned available the use of existing and well succeeded methods for the developing of systems based on knowledge, like the CommonKADS. The direct handling of JAVA technology objects and XML allowed better results when implementing the detailed system's architecture in real time, assuring that the operations to be executed will shown at the very instant they are occurring, as well as its easiness interoperability with any database.

Finally, this system furnished enhancement and relative readiness to the knowledge processing, as a guide for the decision making of the VALE's rail-wagon unloading system experts.

7. References

Breuker, J. Velde, Van de. CommonKADS Library for Expertise Modeling: Reusable problem solving components. IOS Press, Amsterdam, 1994.

Schreiber, Guus. Akkermans, Hans. Anjewierden, Anjo. Hoog, Robert de. Knowledge engineering and management: The CommonKADS methodology. 1st ed. The MIT Press. 2000.

Feigenbaum, A.E. Expert Systems: principles and practice. In: Encyclopedia of computer science and engineering, 1992.

Forgy, C.L. Rete: A fast algorithm for the many pattern/many object pattern match problem, Artificial Intelligence 1982.

Friedman-Hill, E. J. Jess in Action: Rule-based systems in java. USA: Manning Press, 2003.

17

Modular and Hybrid Expert System for Plant Asset Management

Mario Thron and Nico Suchold
ifak e.V. Magdeburg
Germany

1. Introduction

Industrial companies have to improve their production process as an important precondition for their survival on the market in a situation of a strong national and international concurrency. This includes a protection of the availability for use of the production equipment. This equipment includes simple devices like sensors and actuators but also complex machines and even more complex production lines and plants. The term "improvement" relates to the maximization of the production performance and a minimization of unintentional production down times.

This chapter contains an approach for the support of the plant asset management, which is an important part of the business management of a company. The plant asset management is focussed on the production equipment of an industrial company, while the general asset management involves all tangible and intangible assets of a company including buildings and patents. Furthermore plant asset management is mostly oriented to the operation phase of the production equipment. In the following we will take the term asset as synonym for production equipment.

Plant asset management has the goals to improve the reliability and efficiency of the assets and to reduce the effort to keep them in a state ready for production. Thus equipment maintenance is an important aspect of the plant asset management.

Important strategies of maintenance are:

- **Preventive maintenance**: The fitness to work of the assets is preserved by maintenance actions before any failure occurs. The action planning is based on production time and/or load cycles.
- **Predictive maintenance**: Maintenance actions are executed before any damages. The abrasion state of the assets is continuously monitored and logged here. The remaining useful life is predicted. This information builds the base for the planning of maintenance actions. This strategy is also called condition based maintenance. It leads to longer useful life time of assets compared to the preventive maintenance strategy.
- **Reactive maintenance**: Maintenance actions are executed after emerging failures of the assets. Repair actions or exchange of assets are necessary actions to take in most cases. This strategy has been used widely in former times when technical preconditions like computer aided planning systems and measurement systems where not available.

The reliability of production processes may be significantly increased by application of the preventive or predictive maintenance strategies in comparison with the application of the reactive maintenance strategy. However technical systems may and will fail. Thus it is not possible to completely pass on the reactive maintenance strategy.
The application of predictive and preventive maintenance strategies and economical constraints lead to a situation where the maintenance manpower is drastically reduced. The complexity of the technical systems grows in parallel for example by introduction of advanced automation systems and technical items dedicated to save energy and material during the production process. Thus in many cases a small group of maintenance employees is confronted with high complex systems in rarely but very critical situations, where production loss is caused by equipment failures.
Computerized assistance systems may support the maintenance employees at the completion of their tasks. This chapter describes an approach of such a system for the diagnosis of failure causes in complex production systems. The application of discrete theories is not sufficient for an optimal support of diagnosis tasks. Thus the approach contains the combination of various theories of the expert system technology and of probability theory. The reliability of detection of the right failure causes and the effort to formalize the necessary diagnosis knowledge are considered here as optimality criteria.

2. System Requirements

Expert systems for the diagnosis of technical systems are known since the 1960's and early 1970's. They have been successfully used for special systems. But they have not been applied for big, heterogeneous and complex systems. An analysis of end user requirements is necessary in order to find solutions for a broader appliance of such systems.
Figure 1 provides a summary of the use cases, which have been identified within the research project WISA (see acknowledgements in section 5). It is noteworthy that the diagnosis process is only one among a variety of use cases. A successful introduction of expert system technology depends a lot on the comfort for the creation of the knowledge base. Thus the consideration of the needs of other person groups is necessary beside the main focus on the maintenance personnel. Important user roles are:

- **Diagnostics user**. This is the user of the expert system. Mostly this will be a member of the maintenance staff. She is responsible for the availability of the assets.
- **Asset manager**. The asset manager is responsible for the efficient production. She controls the processes on business level. In most cases she is supervisor of the diagnostic user.
- **Asset expert**. The asset expert has special knowledge about the disturbance behaviour of the assets. This knowledge contains relations between failure causes and their symptoms. In many cases the expert is the constructor of the asset, but it is also likely that she is the asset user with long term experiences.

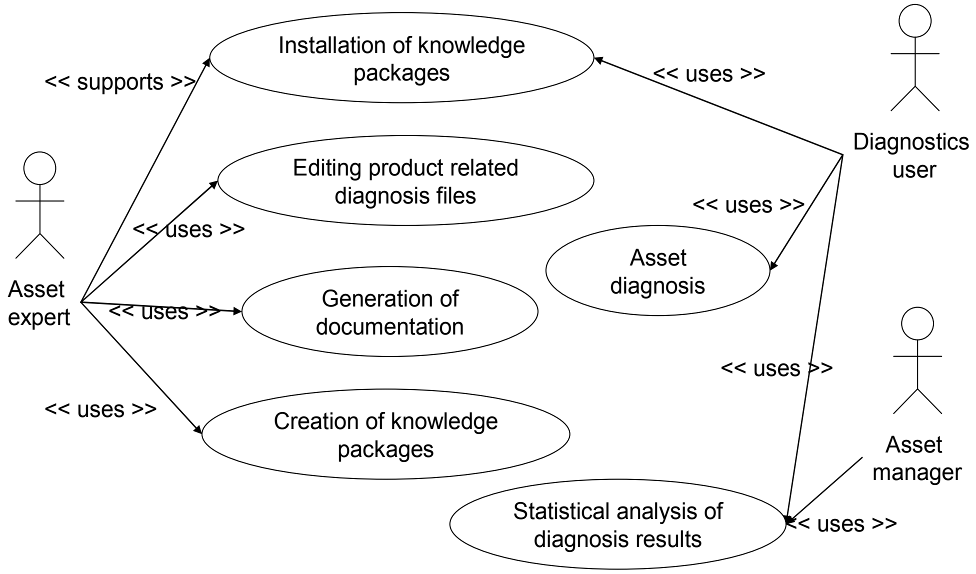


Fig. 1. Summary on use cases for a diagnosis expert system framework

The following list gives some details about the requirements resulting from the use case analysis:

- **Asset diagnosis**: This is the main use case. A failure event occurs and becomes visible for the diagnostic user who searches for the failure cause. The information about the failure may be distributed via industrial networks or may be a perception of the operation staff or the diagnosis user itself. Requirements are:
  - The input of the diagnosis user into diagnosis dialogs should be minimized.
  - If there are multiple failure causes possible then the list of causes should be sorted according to their probability.
  - The diagnosis system should ask back for additional symptoms, which make sense.
- **Editing product related diagnosis files**: The asset expert has to formalize his diagnosis knowledge. In most cases the asset expert is not an IT expert. Thus the input of the knowledge should be done not directly by using a programming language. Instead a forms based editor will be used. Requirements regarding such an editor are:
  - The graphical user interface of the editor should use maintenance terminology.
  - The editor should recognize and inform about syntactic and semantic errors within the knowledge base as far as possible.
- **Creation of knowledge packages**: The diagnosis knowledge of a complex asset includes the knowledge about the asset elements and the causal relations between them. A system is needed, which gathers all the knowledge, which contain the necessary information in order to create a knowledge package for the considered asset. Requirements regarding such a packaging system are:
  - The packaging system should gather all necessary knowledge in a file container.
  - The file container should be compressed for hard drive space efficiency.

- **Generation of documentation**: The equipment manufacturers should act as asset experts. Thus they investigate more time into the development of the products. The generation of a part of the necessary product documentation is a pay-back possibility. Requirements on a generator system are:
  - The documentation generator system should be included into the editor framework.
  - The generator system should only be parameterized by the asset main context. Necessary descriptions of asset elements will be included automatically.
  - The documentation result should be linked HTML pages, which may be transformed into other text processing formats.
- **Installation of knowledge packages**: The diagnosis expert system to be used by the diagnostics user will include the knowledge of a variety of products, which aggregate other components. Thus there is the need for installation of asset descriptions. Some requirements are:
  - The installation system should inform about type and version conflicts regarding descriptions of assets.
  - The installation system should contain features to gather knowledge packages via the Internet and from local resources.
- **Statistical analysis of diagnosis results**: If the failure behaviour of assets is tracked over a long time period, then there is the possibility to identify the assets, which are responsible for the highest production loss. Failure frequency and amount of production loss has to be tracked for this purpose. Requirements for this analysis are:
  - The statistical analysis system should identify the assets, which are responsible for the highest production loss.
  - The result presentation may be done in text form on-screen.

These system requirements should be satisfied by an intelligent language construction for the formalized asset descriptions and by a related software tool chain.

3. Design of the Knowledge Base

3.1 Special Criteria for the Knowledge Base

The introduction and the system requirements state, that the knowledge base will be constructed by multiple person groups. Thus there must be a way of collaboration between them. In the WISA project the following approach has been used:

1. A language has been designed for the formal description of diagnosis knowledge. It has been named **Hybrid Logic Description Language (HLD)** for purposes described below.
2. Tools have been designed for creation, transportation and installation of HLD files.
3. An HLD interpreter has been designed for the diagnostic process.

A variety of approaches are known to work fine for special diagnostic purposes. The problem is to select the right one(s) to meet industrial needs and strength. Thus a selection had to be made by matching against a catalogue of criteria for such a language. Some of these criteria are:

- **Processing of discrete causal relations should be possible.** That means that a specific symptomatic is uniquely related to a distinct failure cause. Such relations are often described in tabular form in conventional machinery and plant documentation.

- **Processing of vague diagnosis information should be possible.** Vague information here means such information, which don't may be acquired by measurement or only with inadequately hard effort. This kind of information is estimated by the diagnostics user mostly based on his/her visual, acoustical and tactile perceptions.
- **Processing of uncertain causal relations should be possible.** In opposite to the discrete causal relations it is often necessary to formalize relations between symptomatic and a variety of causes. These relations are weighted by different probabilities. Thus the language should provide constructs to formalize these kinds of relations.
- **The information processing concepts should be kept simple.** The concepts should be easy to understand for asset experts.
- **There should be the possibility to validate language instances.** The process of editing HLD files should be supported by checking them for syntactic and semantic errors.
- **There should be the possibility to re-use asset type based diagnosis knowledge.** The asset experts know the scope of the assets they describe, which includes all aggregates of the considered asset. The diagnosis knowledge about the aggregates should simply be referenced by the type and version of the aggregate in the description of the asset.
- **There should be efficient algorithms for the information processing concepts.** The knowledge base for a complex technical system will itself become very complex. The algorithm implementations should provide answers on the failure causes in an appropriate time (some seconds).

The following section describes the design of the HLD language, which has been developed to satisfy this list of criteria.

3.2 Hybrid Logic Description Language (HLD)

3.2.1 Propositional Logic and Its Representation as HLD Rules

The first means of expression is the propositional logic. Later on we want to introduce certainty factors. In order to motivate the introduction of these concepts let us have a look on the following example dialog between a machinery operator and an expert for the machinery:

Operator: "Production line 3 is currently out of order"
Expert: "How do the last products look like?"
Operator: "The surfaces are scratched."
Expert: "Did the forming machine made noise?"
Operator: "No."
Expert: "Check if there are crumbs in the raw material."
Operator: "Yes, there are crumbs."
Expert: "Then we have got low quality of the component X within the raw material.

This example shows a systematically diagnosis controlled by the expert. This expert has to be replaced by the expert system. Thus there is the need to formalize the shown diagnosis knowledge.

The kind of knowledge as mentioned in the example may be expressed by a rule base like the following:

(1) products_scratched IF line_3_not_working.
(2) tool_loose IF products_scratched AND forming_machine_noisy.
(3) crumbs_in_material IF products_scratched.

(4) component_X_bad_quality IF crumbs_in_material

These rules constitute only a small section of a diagnosis knowledge base for a real world application. The causes of symptoms are situated on the left side of the IF statement, while the symptoms itself are positioned on the right side. This is in opposite direction of the causal direction. The results of the application of the reasoning algorithms are the conclusions of the rules on the left-hand side of the IF statement and the result should be by definition the cause of symptoms.

The syntax of the propositional logic has been defined in various books like (Kreuzer and Kühling, 2006), (Russel and Norvig, 2003) or (Poole et al., 1998). Propositional formulae deal only with the truth values {TRUE, FALSE} and a small set of operations is defined including negation, conjunction, disjunction, implication and bi-conditional relations. The possibility to nest formulae enables arbitrary large formulae.

The HLD restricts the propositional logic to Horn-logic, which is not a big limitation. A Horn formula is a propositional formula in conjunctive normal form (a conjunction of disjunctions) in which each disjunction contains in maximum one positive literal. The set of elements of these disjunctions is also called a Horn clause. A set of Horn clauses build a logic program. If a Horn clause contains exactly one positive and at least one negative literal, then it is called a rule. The positive literal is the conclusion (or head) of the rule, while the negative literals constitute the condition part (or the body) of the rule. If a rule is part of a HLD file, then we call it a HLD rule.

The form of horn clauses is chosen for the HLD, since there exist an efficient reasoning algorithm for this kind of logic - namely the SLD resolution. This resolution algorithm may be combined with the breadth-first search (BFS) or with the depth-first search (DFS) strategy.

- **Breadth-first search**: The algorithm proves for each rule whether the conclusion is a consequence of the values of the conditions. Each condition value is either looked up in a variable value mapping table or it will be determined by consideration of rules, which have the same literal as conclusion. If there exist such rules, but a direct evaluation of their conclusion is not possible, then a reference to this rule is stored, but the algorithm proceeds with the next condition of the original rule. If there is no condition left in the original rule, then references are restored and the same algorithm as for the original rule is applied to the referenced rules. This approach needs a huge amount of memory.
- **Depth-first search**: This algorithm proves for each rule whether the conclusion is a consequence of the values of the conditions. Each condition value is looked up in a variable value mapping table or it will be determined by consideration of rules, which have the same literal as conclusion. If there exist such rules, but a direct evaluation of the conclusion is not possible then the first of these rules is evaluated directly. Therefore this algorithm does not need the references and saves a lot of memory compared to BFS.

It may be shown that the SLD resolution with BFS strategy is complete for Horn logic while the combination with DFS is incomplete. However, DFS is much more memory efficient than BFS and in practise it leads often very quickly to the result values. Thus both resolution algorithms have been prototypically implemented for evaluation of HLD files. The syntax of the HLD does not depend on the selection of search algorithms.

The propositional variables of HLD rules have special meanings for the diagnosis purposes. Following has been defined:

- **Symptoms** are propositional variables, which appear only as conditions within HLD rules.
- **Indirect failure cause**s are propositional variables, which appear as conclusion in some HLD rules and in other HLD rules condition part.
- **Direct failure causes** are propositional variables, which appear only as conclusions of HLD rules.

Thus simple propositional logic is modelled in the HLD by direct and indirect failure causes as conclusion of rules and by symptoms and indirect failure causes as conditions of rules.

3.2.2 HLD Rules with Empirical Uncertainty Factors

The application of HLD rules is not always applicable or at least not very comfortable because of the following reasons:

- A huge amount of rules and symptoms have to be defined in order to find failure causes in complex technical systems. This is accompanied by very large condition parts of the rules. The establishment of the knowledge base becomes too expensive.
- A diagnosis expert system, which has a high complex knowledge base, has to ask the users for a lot of symptoms in order to find a failure cause. Guided diagnosis becomes too time-consuming.
- Complex knowledge bases lead to long-term reasoning.

All these effects should be avoided according to the defined requirements. The mapping of simple cause-effect relations with simple HLD rules continues to be applicable. But complex circumstances need other kinds of expressivity.

A simple extension of HLD rules is the introduction of certainty factors (CF). Therein the conclusion of a rule is weighted with a certainty factor. Such systems are described for example in (Bratko, 2000), (Norvig, 1992) and (Janson, 1989). In these resources the value range for the certainty factors is the interval [-1, +1]. For a better comparability of the CFs with probabilities the interval [0, 1] has been chosen for certainty factors of HLD rules.

All propositions, which are evaluated by application of an algorithm on a HLD knowledge base, are weighted by a CF, since the conclusion parts of the rules are weighted by certainty factors. Certainty factors of propositions have the following semantic within HLD files:

CF = 0.0 The proposition is false.
CF = 0.5 It is unknown if the proposition is true or false.
CF = 1.0 The proposition is true.

CF values between 0.5 and 1.0 have the meaning that the related propositions are more likely true than false, while CF values between 0.5 and 0.0 mean, that the related propositions are more likely false than true.

Two algorithms for the evaluation of HLD rules with certainty factors have been tested. These are the simple evaluation algorithm according to (Janson, 1989) and the EMYCIN algorithm as shown in (Norvig, 1992). The simple algorithm is based on the following instructions:

1. The CF of the condition part of a rule is the minimum CF of all the conditions.
2. The CF of the conclusion of a rule is the CF of the condition part of this rule multiplied with the CF value for this rule.
3. If the knowledge base contains multiple rules with the same conclusion, then the CF of this conclusion is the maximum of the related CF values.

The algorithms for certainty factors are proved to provide incorrect results in some situations. On the other hand for MYCIN it has been shown that such systems may provide better results than human experts. In addition the rule CFs may be empirically determined and thus the creation of a knowledge base is very easy. For these reasons the concept of certainty factors has been included into the HLD language.

3.2.3 Fuzzy Logic as Part of the HLD

Rule sets as described in the previous sections use mappings of diagnosis relevant physical values to discrete values as propositions. Thus rules for each discrete value interval have to be provided. This leads to a big effort for the creation of the knowledge base. In this section we introduce Fuzzy Logic as one opportunity to improve the preciseness of the reasoning and to reduce the necessity for fine grained discretization levels of physical values. An example of a HLD fuzzy logic rule is the following:

motor_defect WITH 0.9 IF motor_windings_hot AND load_low.

The way of diagnosis is different from that of the propositional logic. The diagnosis user inputs values of continuous value spaces (in the example for motor winding temperature and mechanical load), instead of providing discrete symptoms and binary answering of questions. The result is again a value out of a continuous value space (in the example an estimation of the degree of abrasion of the motor). Special diagnosis relevant output variables have been defined for the HLD language.

The use of Fuzzy Logic for diagnosis purposes works in following steps:

1. Definition of the knowledge base: (A) Fuzzy variables have to be defined and (B) a Fuzzy rule set has to be integrated into the knowledge base.
2. Evaluation of the knowledge base: (C) The user inputs variable values and (D) the implementation of a Fuzzy Logic interpreter provides results by fuzzyfication of input variables, applying of inferences and by defuzzyfication of output variables.

Fuzzy variables may be defined by mapping of triangles, trapezoids or more round function shapes to terms of natural language. Input variables within the HLD fuzzy logic may be defined by piecewise linear membership functions, while output variables are defined by singletons (see figure 2).

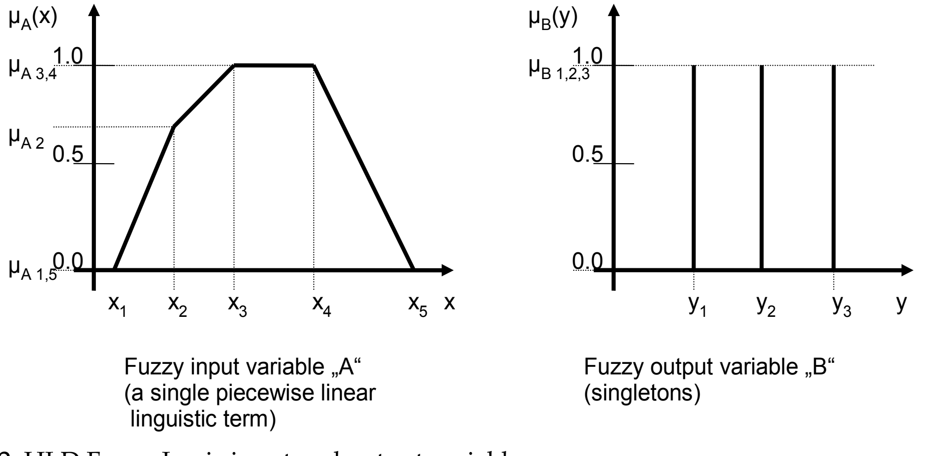


Fig. 2. HLD Fuzzy Logic input and output variables

This definition is in line with the standard (IEC61131-7, 1997). This is the standard for programming languages for programmable logic controllers (PLC). PLCs are the most used automation systems for machinery and plant control. Thus if the maintenance employees know something about Fuzzy Logic then it is very likely, that they know the terminology and semantics of this standard.

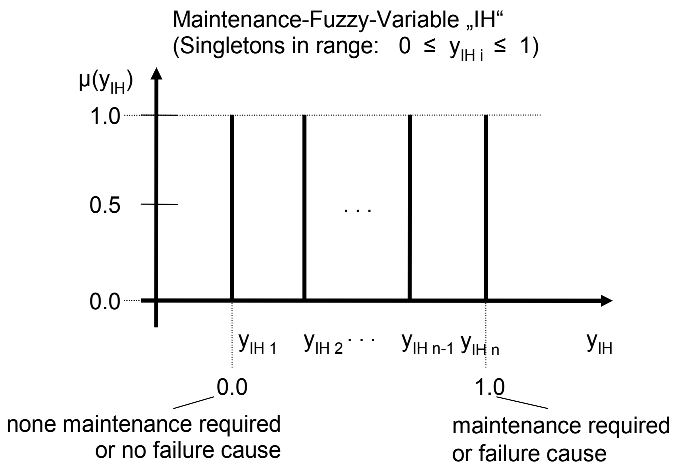


Fig. 3. HLD maintenance fuzzy variables

Beside the common semantics of Fuzzy output variables there are special definitions for maintenance variables in the HLD specification. This is illustrated in figure 3. The values of such variables y_{IH} are defined only within a range of [0, 1.0]. Only within this value range singletons may be defined. Similar to the definitions for certainty factors following conventions have been defined for these maintenance variables:

$y_{IH} = 0.0$ Maintenance is not necessary or this is not a failure cause.
$y_{IH} = 0.5$ It is not decidable if this is a failure cause.
$y_{IH} = 1.0$ Maintenance is necessary since this is a failure cause.

As mentioned above the processing of the maintenance knowledge base is done in three steps:

1. **Fuzzyfication**: Memberships are computed for each linguistic term of the input variables if there are numerical values available for the physical input variables.
2. **Inference**: The inference is done very similar to the approach used for rule sets with certainty factors:
    a. The membership of the condition part of a fuzzy rule is the minimum of all the memberships of the condition variables.
    b. The membership of the conclusion of a fuzzy rule is the membership of the condition part of this rule multiplied with the weighting factor for this rule.
    c. If the knowledge base contains multiple fuzzy rules with the same conclusion, then the membership of this conclusion is the maximum of the membership values of the conclusion variables.

3. **Defuzzyfication**: Within the basic level of conformance of the standard (IEC61131-7, 1997) the method "Center of Gravity for Singletons" (COGS) has been defined as defuzzyfication method. This has been taken over for the HLD specification. The result value of the fuzzy logic output variable is computed by evaluation of following formula:

$$y = \frac{\sum_{i=1}^{p} (\mu^*_{Bi} y_i)}{\sum_{i=1}^{p} \mu^*_{Bi}}$$

This formula uses the terminology as presented in figure 2. The μ^*_{Bi} are the membership values computed in the inference process for the *p* singletons at the values y_i. The result value *y* is the value of the output variable. Thus it is not a membership but a value of the value range defined for this output variable. Especially for the maintenance output variables the value range is [0, 1].

The approach of using singletons fits the need of fast computations as specified in the requirements analysis, since only multiplication and addition operations are used.

3.2.4 Bayesian Networks

Bayesian Networks have been introduced into the HLD, since the handling of uncertainty with certainty factors is not as mathematically correct as the probability theory does.
The example introduced in the propositional logic section could be extended by probabilities as follows

component_X_bad_quality (p=0.9) IF crumbs_in_material.
component_X_bad_quality (p=0.5) IF product_color_grey.

This example has the meaning that if there are crumbs in the raw material then the probability are very high (90%) that the material component X has not a good quality. In other words there are not many other reasons for crumbs than a bad material X. But there is another phenomenon in that approach: the variables crumbs_in_material and product_color_grey are not independent from each other. If there are crumbs in the material, then it is likely that the component X has a bad quality, but then there is also a good chance that the product looks a little bit grey.
Bayesian Networks are graphical representations (directed acyclic graphs) of such rules as shown in the example. (Ertel, 2008) gives a good introduction to Bayesian Networks based on (Jensen, 2001). One of the earlier literature references is (Pearl, 1988). There are following principles of reasoning in Bayesian Networks:

- **Naive computations of Bayesian Networks**. This algorithm computes the probabilities for every node of the network. The computation is simple but very inefficient. (Bratko, 2000) presents an implementation of this algorithm for illustration of the principles.
- **Clustering algorithms for Bayesian Networks**. This approach uses special properties of Bayesian Networks (d-Separation) for dividing the network into smaller pieces (clusters). Each of the clusters may be separately computed. For each cluster it is decided if it is influenced by evident variables. The computation of probabilities is done only for these clusters. The approach is much more efficient than the naive approach.

- **Approximation of Bayesian Networks**. Algorithms of this concept estimate the probability of variables. Such algorithms may be used even in cases where clustering algorithms need too much time.

The naive algorithm has been implemented for the evaluation of the usability of Bayesian Networks for the HLD. Further evaluation has been done by using the SMILE reasoning engine for graphical probabilistic models contributed by the Decision Systems Laboratory of the University Pittsburgh (http://dsl.sis.pitt.edu).

3.2.5 Summary and the HLD Language Schema

XML has been chosen as basic format of the HLD. Thus an XML schema according to W3C standards has been developed, which contains language constructs for the methodologies described in the previous sections. The structure of this schema is shown in figure 4.

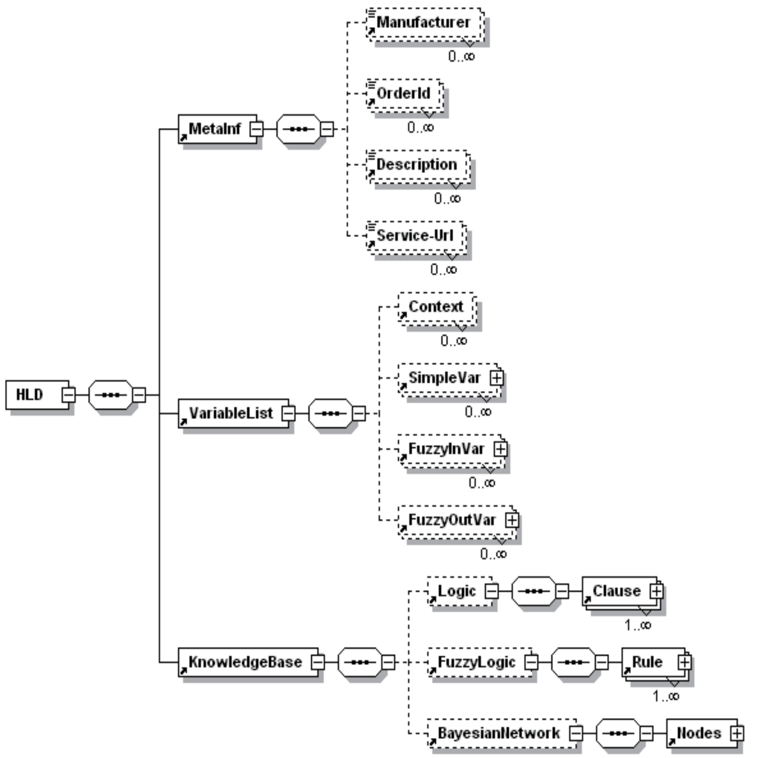


Fig. 4. HLD schema overview.

The HLD schema contains following top level information:

- Meta Information. The element MetaInf contains various common information about the asset described by the HLD file. This includes for example the manufacturer name, an ordering number, a short description and a service URL for getting further information from the manufacturer.

- Variable Declarations. The element VariableList contains lists of variables. Propositional variables (with and without certainty factors) are separated from Fuzzy Logic input and output variables due to their different representation models.
- Knowledge Base. This element contains the following sub elements:
  - Logic: This element contains rules with and without the use of certainty factors.
  - Fuzzy Logic: This element contains fuzzy logic rules and it references the Fuzzy Logic input and output variables.
  - Bayesian Network: This element contains the definition of a Bayesian Network for discrete variables. It contains conditional probability tables and references to the declarations of propositional variables.

The other attributes and elements define the semantics as specified in the sections above. The full HLD scheme may be downloaded at "http://i2service.ifak.eu/wisa/".

4. Framework for the Handling of the Knowledge Base

The central application of the HLD framework is the diagnosis system. It is implemented as a web application. This provides the possibilities to:

- maintain the knowledge base on one place,
- enable the access to the diagnosis system from any place,
- reduce the necessity of installation of special software (a Web browser is expected to be installed on any modern operating system by default).

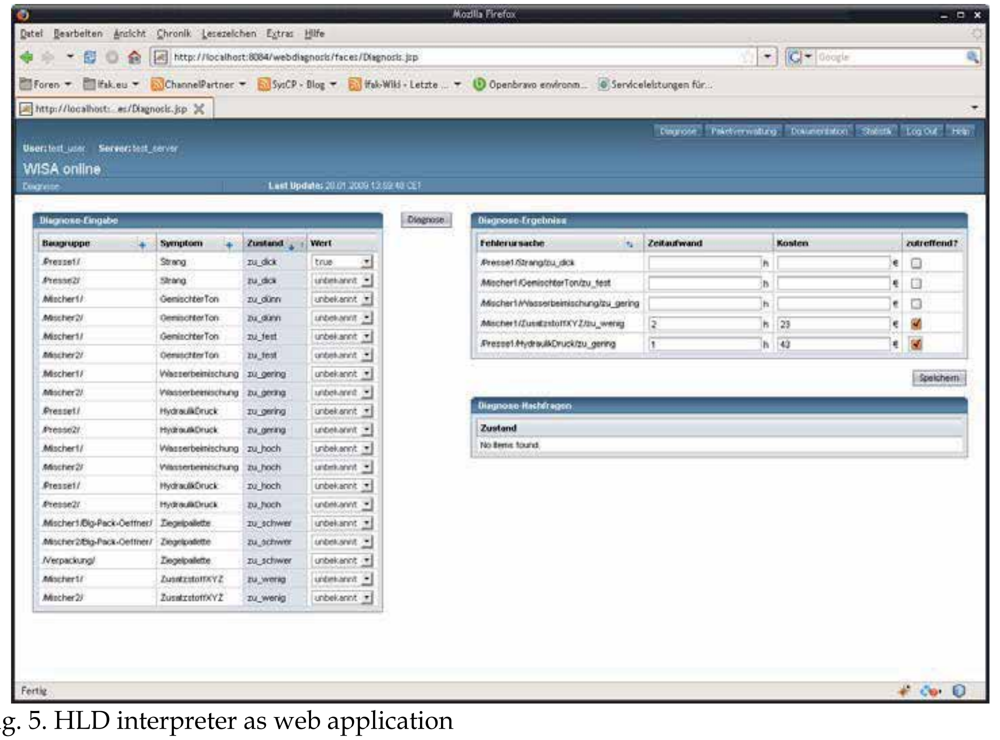


Fig. 5. HLD interpreter as web application

Figure 5 gives an overview of this application. On the left side the expert system provides a list of possible symptoms. The diagnosis user marks, which symptoms he has percepted. The diagnosis results are listed on the right side, sorted by their probability or their membership to a maintenance fuzzy variable.
The expert has another application for the creation of the knowledge for a specific asset type. This is an editor for HLD files. A screenshot of a prototype application is shown in figure 6. On the left side there is a tree representing the asset hierarchy. Elements of this tree are set into relations by definition of rules. This is done by entering some input into the forms of the right side. The screenshot shows the forms for input of logic rules. The entry fields are labelled by using the maintenance terminology. Thus a transformation of the terminology of artificial intelligence terminology to the application domain is done by this user frontend for the asset experts. The HLD editor uses for example the term "failure cause" ('Schadensursache') instead of the term "conclusion" or "clause head".

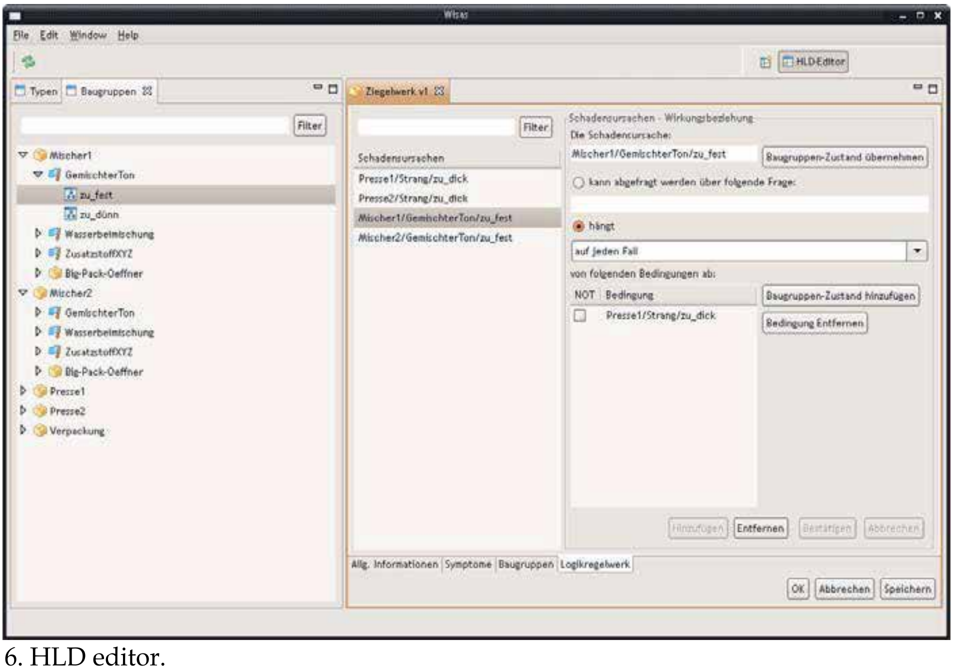


Fig. 6. HLD editor.

Assets like machine and plants are recursively nested when considering the aggregation relation. This is illustrated in fig. 7. If we consider a plant as asset, then the machines are the asset elements. If we further consider the machines as assets, then the tools, HMI elements and the control system are the asset elements. The HLD language introduces elements with the name "Context" in order to reference aggregated asset elements (see also fig. 4).

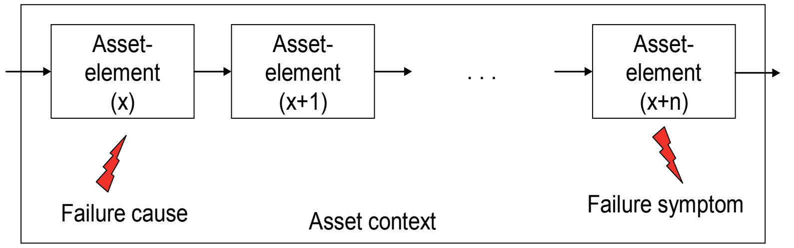


Fig. 7. Failure cause and symptom relation within an asset

In many cases failures occur in one asset element and cause symptoms in another asset element. These relations may be described in HLD files dedicated to the upper asset context, which contains the related asset elements directly or indirectly.
All HLD descriptions of the assets and their recursively nested aggregates build the knowledge base of the diagnosis expert system. They are positioned side by side in a HLD file repository. Each HLD description is dedicated to an asset type and its version, which are represented by structure elements of the repository. Thus the repository is not free form. It is obviously from fig. 7, that an asset description must be the assembly of the asset context description and the descriptions of all asset elements. Thus a HLD description is a package of HLD files with the same structure like a HLD repository. The tool set contains a packaging system, which assembles all necessary HLD descriptions from the repository of the asset expert and compresses them. Furthermore the tool set contains a package installation system, which decompresses the packages and installs them in a HLD repository, while paying attention to asset type and version information. In addition a documentation generation system has been set up, which generates HTML files out of a repository by a given asset context.

5. Conclusions and Future Research Work

An expert system has been introduced with a hybrid knowledge base in that sense that it uses multiple paradigms of the artificial intelligence research work. There was a gap between the success of the theoretical work and the acceptance in the industry. One key problem is the necessary effort for the creation of the knowledge base, which is overcome by the concept of a collaborative construction of the knowledge base by contributions of manufacturers of the production equipment.
Further research work will be spent to structure and parameter learning algorithms for the Bayesian Networks. The results have to be integrated into the HLD editor. Furthermore an on-line data acquisition will be integrated into the diagnosis system, which is especially necessary for an effective application of the Fuzzy Logic reasoning.
Most of the work has been done as part of the research project WISA. This project work has been funded by the German Ministry of Economy and Employment. It is registered under reg.-no. IW06215. The authors gratefully thank for this support by the German government.

6. References

Bratko, Ivan (2000). PROLOG - Programming for Artificial Intelligence, 3.Ed., Addison-Wesley

Bronstein, Semendjajew, Musiol, Mühlig (1997). Taschenbuch der Mathematik, 3. Ed., Verlag Harri Deutsch

Ertel, W. (2008). Grundkurs künstliche Intelligenz, 1. Ed., Vieweg Verlag

IEC61131-7 (1997). IEC 61131 - Programmable Logic Controllers, Part 7 - Fuzzy Control Programming, Committee Draft 1.0, International Electrotechnical Commission (IEC)

Janson, Alexander (1989). Expertensysteme und Turbo-Prolog, 1. Ed., Franzis Verlag GmbH München

Jensen, Finn V. (2001). Bayesian networks and decision graphs, Springer Verlag

Kreuzer, M.; Kühling, S. (2006). Logik für Informatiker, 1. Ed, Pearson Education Deutschland GmbH

Norvig, Peter (1992). Paradigms of Artificial Intelligence Programming - Case Studies in Lisp, 1. Ed., Morgan Kaufman Publishers, Inc.

Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann Publishers, Inc.

Poole, D.; Mackworth, A.; Goebel, R. (1998). Computational Intelligence - A Logical Approach, 1. Ed., Oxford University Press, Inc.

Russell, S. and Norvig, P. (2003). Artificial Intelligence - A Modern Approach , 2. Ed., Pearson Education, Inc.

18

Image Retrieval System in Heterogeneous Database

Khalifa Djemal, Hichem Maaref and Rostom Kachouri
University of Evry Val d'Essonne, IBISC Laboratory
France

1. Introduction

This chapter content try to give readers theoretical and practical methods developed to describes and recognize images in large database through different applications. Indeed, many computer vision system applications, such as computer-human interaction, controlling processes: autonomous vehicles, and industrial robots, have emerged a new need for searching and browsing visual information. Furthermore, due to the fast development of internet technologies, multimedia archives are growing rapidly, especially digital image libraries which represent increasingly an important volume of information. So, it is judicious to develop powerful browsing computer systems to handle, index, classify and recognize images in these large databases. Different steps can be composes an image retrieval system where the most important are, features extraction and classification.
Feature extraction and description of image content: Each feature is able to describe some image characteristics related to shape, color or texture, but it cannot cover the entire visual characteristics. Therefore, using multiple and different features to describe an image is approved. In this chapter, the extraction of several features and description for heterogeneous image database is presented and discussed. Test and choice of adequate classifiers to manage correctly clustering and image recognition: The choice of the used classifier in CBIR system (Content Based Image Retrieval) is very important. In this chapter we present the Support vector machines (SVMs) as a supervised classification method comprising mainly two stages: training and generalization. From these two important points, we try how we can decide that the extracted features are relevant in large heterogeneous database and the response to query image is acceptable. In these conditions we must find compromise between precision of image recognition and computation time which can be allowed to the CBIR System. In this aim, an heterogeneous image retrieval system effectiveness is studied through a comparison between statistical and hierarchical feature type models. The results are presented and discussed in relation with the used images database, the selected features and classification techniques. The different sections of this chapter recall and present the importance and the influence of the features relevance and classification in image recognition and retrieval system. Indeed, different features extraction methods (section 3) and classification approaches (section 4) are presented and

discussed. We illustrate the principle and obtained results of the optimization methods in sections 5.

2. Related works

The image recognition system, consists of extracting from a Database all the similar images to query images chosen by the user, figure 1, has attracted much research interest in recent years. Principal difficulties consist on the capacity to extract from the image the visual characteristics, the robustness to the geometrical deformations and the quantification of the similarity concept between images. Indexation and recognition are given from a classification methods accomplished on images features. To understand the influence of the images database on the description method and the appropriate classification tool, it is more convenient to subdivide the image databases into two categories.

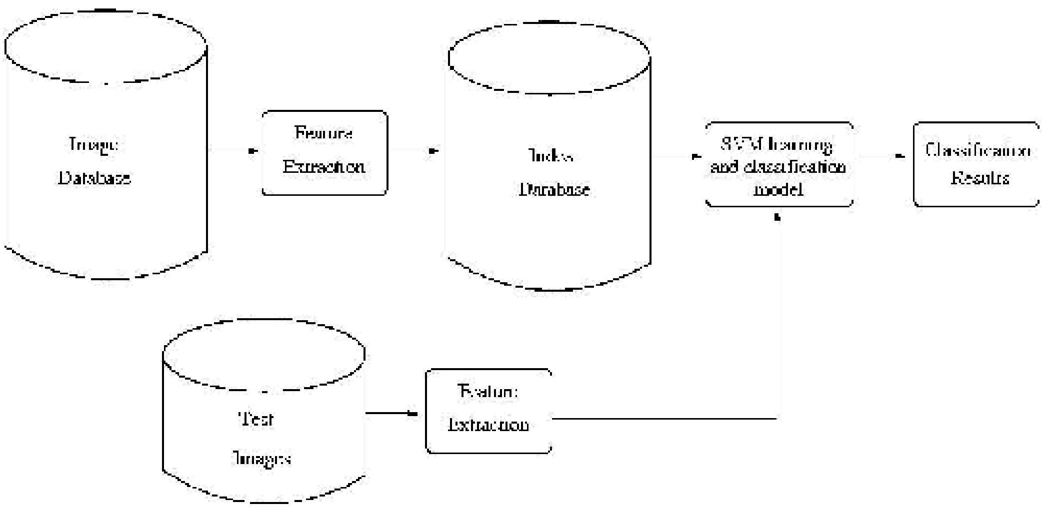


Fig. 1. Content Based Image Recognition system.

The first category consists of image databases usually heterogeneous. In this context, the objective of the images recognition system is to assist the user to intelligently search in images database a particular subject adapting to the subjective needs of each user. The second category concerns specific image databases. In this context, the images are most often a uniform semantic content. The concerned applications are generally professional. To index this image databases, the user must integrate more information defined by the expert to develop a specific algorithm, the objective is to optimize system efficiency and its ability to respond as well as the expert. This categorization is to be taken into account when developing any content image retrieval system. Indeed, to obtain satisfactory results, the choice or development of new description methods must be appropriate for the type of the considered images database. This is due simply to the great difficulty to obtain a universal description algorithm.

The content-based image retrieval that we have just described it in figure 1, have attracted the attention of several specialists in different domains and caused for a few years an important effort of research. Indeed, there has been growing interest in indexing large biomedical images by content due to the advances in digital imaging and the increase development of computer vision. Image recognition in biomedical databases, for example, are critical assets for medical diagnosis. To facilitate automatic indexing and recognition of image databases, several methods has been developed and proposed. As described above, two important steps composed an image retrieval system, features extraction and classification. We present in this section some important related work to these stages. We find several choices of features from the low level to high level: shape, geometry, symbolic features, .etc. The basic goal in content-based image retrieval and recognition is to bridge the gap from the low-level image properties. Consequently, we can directly access to the objects, and users generally want to find in image databases. For example, color histograms (Stricker & Swain, 1994), (Swain & Ballard, 1991), are commonly used in image description and have proven useful, however, this global characterization lacks information about how the color is distributed spatially. Several researchers have attempted to overcome this limitation by incorporating spatial information in the features extraction stage. Stricker and Dimai (Stricker & Dimai, 1997), store the average color and the color covariance matrix within each of five fuzzy image regions.

(Huang et al., 1997) store a color correlogram that encodes the spatial correlation of color-bin pairs. (Smith & Chang, 1996), store the location of each color that is present in a sufficient amount in regions computed using histogram back-projection. (Lipson et al., 1997) retrieve images based on spatial and photometric relationships within and across simple image regions. Little or no segmentation is done; the regions are derived from low-resolution images. In (Jacobs et al., 1995), authors use multiresolution wavelet decompositions to perform queries based on iconic matching. Some of these systems encode information about the spatial distribution of color features, and some perform simple automatic or manually-assisted segmentation. However, none provides the level of automatic segmentation and user control necessary to support object queries in a very large image databases. (Carson et al., 2002), see image retrieval ultimately as an object recognition problem and they proceed in three steps. Firstly, pixels into regions which generally correspond to objects or parts of objects were grouped. These regions are described in ways that are meaningful to the user. The proposed system allows access to these region descriptions, either automatically or with user intervention, to retrieve desired images. In this approach the features do not encode all the important information in images and the image retrieval is obtained without classification what can poses a problem of recognition in large image databaes. (Antani et al., 2002), presents a comprehensive survey on the use of pattern recognition methods which enable image and video retrieval by content where the classification methods are considered. Research efforts have led to the development of methods that provide access to image and video data. These methods have their roots in pattern recognition. The methods are used to determine the similarity in the visual information content extracted from different low level features.

A multi-level semantic modeling method is proposed in (Lin et al., 2006), which integrates Support Vector Machines (SVM) into hybrid Bayesian networks (HBN). SVM discretizes the continuous variables of medical image features by classifying them into finite states as middle-level semantics. Based on the HBN, the semantic model for medical image semantic retrieval can be designed at multi-level semantics. To validate their method, a model is built to achieve automatic image annotation at the content level from a small set of astrocytona MRI (magnetic resonance imaging) samples. Indeed, Multi-level annotation is a promising solution to enable image retrieval at different semantic levels. Experiment results show that this approach is very effective to enable multi-level interpretation of astrocytona MRI scan. This study provides a novel way to bridge the gap between the high-level semantics and the low-level image features.

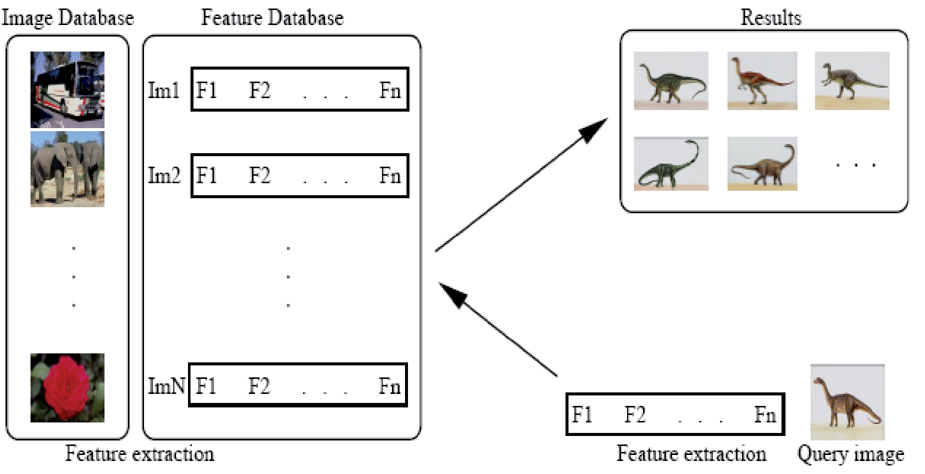


Fig. 2. Features extraction and image recognition: form a query image, a relevant features extraction allows to obtain all similar images in database.

Many indexation and recognition systems were developed based on image content description and classification in order to perform image recognition in large databases (figure 2). These systems use low level features such as the colors and orientations histograms, Fourier and wavelet transforms. In spite of these acceptable results, the classification based on a similarity distance computing is not enough robust to manage great dimensions of the extracted features vectors. To resolve this problem, other proposed systems calculate for each pixel of each image a characteristics vector. This vector contains the components associated with the color, the texture and position descriptors, which gives a better description of the image. But the performances of the image description remain to be improved. Moreover, several works were based on the wavelet tools. The authors in (Serrano et al., 2004), have enhanced their image representation by the use of the texture features extracted by wavelet transform. A new extraction technique of rotation invariants is proposed in (Sastry et al. 2004), this method offers satisfactory results, taking into account the rotation features. For more precision and representativeness of images, a new transform called Trace transform is proposed in (Kadyrov & Petrou, 2001). This transform offers at the same time a good description of image and is invariant to rotation, translation and scaling. After the features extraction, classification is made afterwards by the means of a classifier, such as KNN classifier. But this classifier is slow considering his incapacity to manage great dimensions of feature vectors. A more effective method based on a Bayesian approach (Sclaroff & Pentland, 1995), which consists in concatenation of feature blocks gave better results. The artificial training techniques were also used in the field of the image classification (artificial neural networks (ANN), genetic algorithms,. . . etc). Their broad use is due to the facilities which they offer to the level computing time and their performances in term of classification. Indeed, we find many works based on the ANN. The obtained results show a great capacity thanks to the speed offered by the ANN and the simplicity of their implementation (Takashi & Masafumi, 2000), (Egmont-Petersen et al. 2002). In (Djouak et al., 2007), a combination of features vectors is proposed. These vectors are obtained by a visual search based on the colors histograms, the geometrical and texture features. The rotation invariants and texture features are given using wavelet transform. In addition, the Trace transform is introduced in order to obtain more invariance. A good compromise between effectiveness and computation simplicity are obtained using RBF classification technique. While every content heterogeneous image recognition system, as mentioned above, has two stages: Extraction of discriminate features and classification (figure 1), feature extraction is the most important one, because satisfactory image classification rely basically on a better image description.

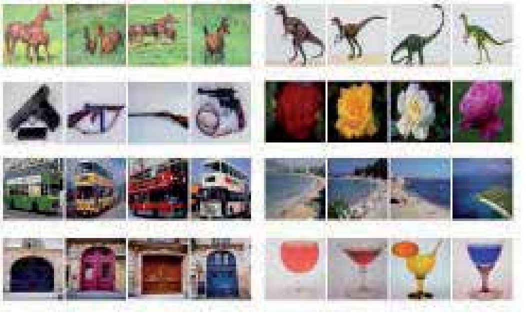


Fig. 3. Images database sample.

In fact, classification algorithm choice is generally based on index content data-sets. Moreover in case of heterogeneous image databases, relevant feature selection is recommended. In section 3, we discuss feature extraction and selection. We evaluated our modular statistical optimization and hierarchical type features model, presented in section 5 on a very well-known heterogeneous images database, chosen from the Wang image database available on this Web site: http://wang.ist.psu.edu/docs/home.shtml/
This image collection contains 1000 images of various types having large difference in colors, shapes, and textures. Some samples are shown in figure 3.
In this chapter, we present in section 3 and 4 the two important steps of an image recognition and retrieval system. In section 5, two CBIR systems are proposed, the first one is based on modular statistical methods and the second on hierarchical best features type selection.

3. Features extraction

In this section, several feature extraction and evaluation for an heterogeneous image database recognition are detailed. Indeed, the main objective of feature extraction is to find, for each image, a representation (signature) that is, in one hand compact to be quickly accessible and easily comparable, and in the other hand enough comprehensive to well characterize the image. Most used features, mainly, reflect the low level characteristics in image, such as color, texture, and/or shape (Bimbo 2001). Color features are the first used in

CBIR systems, and they still be the most used due to their extraction simplicity, rich description and their recognition efficiency. Average color (Faloutsos et al. 1994) and color histograms (Hafner et al. 1995) are very useful, they are position and scale variation insensitive. Correlogram feature was proposed by (Huang et al. 1997) as improvement of color histogram. It is a matrix that describes the spatial correlation between colors in an image according to some distance and a certain degree of orientation. Then, the auto-correlogram, a sub-feature of the correlogram one was defined in (Huang et al. 1999), it captures only the spatial auto-correlation between the same color intensities in the image. (Bimbo 2001), have provided an extensive study of different color indexing methods. Also, a set of color features was tested for inclusion in the standard MPEG-7 (Manjunath et al., 2001). Texture is increasingly used in image indexing, because it mitigates certain problems arising from color indexing, in particular when the color distributions are very close. The existing texture descriptors can be classified into two categories: The first one is deterministic and refers to a spatial repetition of a basic pattern in different directions. This structural approach corresponds to macroscopic textures. First order statistics (Press 1987) is one of the most simple feature computations of this kind. It is extracted from the normalized image histogram. Co-occurrences matrix proposed by (Haralick & Dinstein, 1973) is also used to analyze texture in the spacial field. This matrix allows to compute the same gray level pixel numbers in the image, which are separated by certain distance and positioned according to certain direction. From this matrix, thirteen texture features are computed (Haralick & Dinstein 1973). The second category of texture is probabilistic, it seeks to characterize the chaotic aspect which does not include localizable pattern or main repetition frequency. This approach is called microscopic, or multi-resolution. Examples of texture features derived from the wavelet transformation are presented in (Serrano et al. 2004). As color feature, a set of texture features was tested for inclusion too in the standard MPEG-7 (Manjunath et al. 2001). In addition of color and texture feature, which describes the general aspect of an image, shape features are able to characterize the different objects in the image.

Generally, this kind of feature indicates the object outline, then segmentation as image preprocessing is often necessary, moreover reliable segmentation is especially critical for shape feature characterization. Image segmentation subdivides image into regions of interest allowing its understanding or further processing. Ones region segmentation is realized, two shape feature kinds could be extracted (Chen et al. 2004), (Chen et al. 2006). The first is based on image region geometry. The second is based on pixel grey level statistics in different image regions. Among the most popular features of this kind, we cite invariant moments of HU (Hu 1962). These moments are invariant to different geometric transformation (translation, rotation and scale variation). In our case, because of the complex heterogeneous image contents, a rich description covering the different visual characteristics is needed (Manjunath et al. 2001). For this, we employ in this chapter a set of features belonging the three visual kinds: We use the four average color feature vectors (Faloutsos et al. 1994) computed in different color spaces, histograms, and correlogram (Huang et al. 1997), (Huang et al. 1999), as color features. First order statistics (Press 1987), cooccurence matrix (Haralick & Dinstein 1973) and Daubeshies wavelet transformation are used as texture features. The gradient norm (Delingette & Montagnat 2001) and the invariant moments of Hu (Hu 1962) are used as shape features. Figures 4, 5 and 6 show some extracted feature samples.

3.1 Used and improved features

We present in this section the different used and improved features. Three types of features are discussed, color, textures and shape features. The improvement is given from gradient norm projection and Hu moments considered and obtained from gradient norm image after the suppression of the local maxima.

3.1.1 Color features

Fist we compute the average color in the RGB, HSV, HMMD and YCbCr spaces.

$$R_{avg} = \frac{1}{N}\sum_{p=1}^{N} R(p), \quad G_{avg} = \frac{1}{N}\sum_{p=1}^{N} G(p), \quad B_{avg} = \frac{1}{N}\sum_{p=1}^{N} B(p)$$

Where $X = (R_{avg}, G_{avg}, B_{avg})^T$ and N the image pixel number. The second color features used are the histograms:

$$histR(x) = \frac{1}{N}\sum_{p=1}^{N} R(x_p = x) \quad histG(x) = \frac{1}{N}\sum_{p=1}^{N} G(x_p = x)$$

$$histB(x) = \frac{1}{N}\sum_{p=1}^{N} B(x_p = x)$$

With $x \in [0,1,2,.......,255]$.

We use the color autocorrelogram, defined in (Huang et al. 1999), captures the spatial correlation between identical colors only, it is a subset of the correlogram (Huang et al. 1997) which expresses how the spatial correlation of pairs of colors changes with distance.

3.1.2 Textures features

As texture features we use the First Order Statistics, the Daubeshies Wavelett transform and the Spatial Gray Level dependence matrix.

The first order statistics are considered by the mean, the standard deviation and the coefficient of variation.

$$Mean = \frac{1}{255}\sum_{x=0}^{255} x.hist(x), \quad S\tan dardDeviation = \frac{1}{255}\sum_{x=0}^{255} x - Mean^2 .hist(x)$$

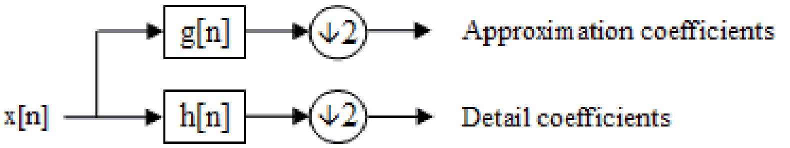
$$CoefficientOfVariation = \frac{S\tan dardDeviation}{Mean}$$

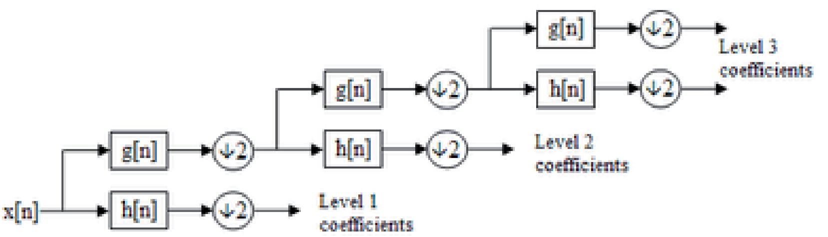
The Daubeshies Wavelett transform in the fourier space is given by the following equations:

$$\hat{\varphi}(\omega) = \Pi_{k=1}^{\infty} \frac{h\left(2^{-k}\omega\right)}{\sqrt{2}}$$

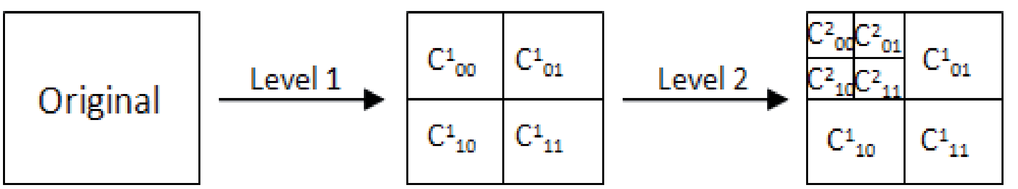
$$\hat{\psi}(\omega)=\frac{1}{\sqrt{2}}g\left(\frac{\omega}{2}\right)\hat{\varphi}\left(\frac{\omega}{2}\right) \qquad h(\omega)=\sum_{k\in Z}h_k e^{-ik\omega} \qquad g(\omega)=\sum_{k\in Z}g_k e^{-ik\omega}$$

And the decomposition is given as follow:





The Wavelet coefficients are $c^l_{ij}(x,y)$, where l is the decomposition level.



The lowest frequency coefficients $c^2_{00}(x,y)$ are not inherently useful for texture analysis. Therefore, a direction-independent measure of the high-frequency signal information is obtained by filtering the raw coefficients $c^2_{00}(x,y)$ with the Laplacian. The texture features are obtained by computing the subband energy of all wavelet coefficients (including the Laplacian filtered $c^2_{00}(x,y)$ coefficients) :

$$e^l_{ij}=\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left|C^l_{ij}(mn)\right|^2$$

where M and N are the dimensions of coefficients $c^l_{ij}(x,y)$, (Wu, 2003).

For the Spatial Gray Level dependence matrix (SGLD Matrix), the co-occurrence matrix (Julezs, 1975), called also the spatial gray level dependence (SGLD) matrices, counts how often pairs of gray level of pixels, separated by a certain distance and lying along certain direction, occur in an image. From these SGLD matrices, thirteen features related to the image texture, could be calculated, (Haralick & Destein, 1973).

3.1.3 Shape features

The invariant moments of Hu and the Gradient norm projection are used as shape features. The Hu invariants (Hu, 1962) are obtained from quotients or powers of moments. A moment is a sum over all pixels of the image model by weighted polynomials related to the positions of pixels. Let *I(x,y)* the grayscale of a pixel in the image *I*, we define the moment of order *(p+q)(p,q>0)* of an image *I* by:

$$m_{p,q} = \int_{R^2} x^p y^q I(x,y)dxdy$$

Let (x_0,y_0) the centroid of the function *I* and the centered image $I_T(x,y)=I(x+x_0,y+y_0)$ is invariant by translation. The central moment of order *(p+q)* of the function *I* is written as follows:

$$v_{p,q} = \int_{R^2} x^p y^q I(x+x_0, y+y_0)dxdy$$

The central moments are invariant by translation. We introduce the normalized moments as follows:

$$\mu_{p,q} = \frac{v_{p,q}}{v_{0,0}^{(1+(p+q)/2)}}$$

These moments are invariant to translation and scale changes. Hu moments are calculated from normalized moments and are invariant by translation, rotation and change of scale:

$$\phi_1 = \mu_{2,0} + \mu_{0,2}$$
$$\phi_2 = (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2$$
$$\phi_3 = (\mu_{3,0} - 3\mu_{1,2}^2 + (3\mu_{2,1} - \mu_{0,3})^2$$
$$\phi_4 = (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{2,1} + \mu_{0,3})^2$$
$$\phi_5 = (\mu_{3,0} - 3\mu_{1,2})(\mu_{3,0} + \mu_{1,2})[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{2,1} + \mu_{0,3})^2]$$
$$+ (3\mu_{2,1} - \mu_{0,3})(\mu_{2,1} + \mu_{0,3})[3(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2]$$
$$\phi_6 = (\mu_{2,0} - \mu_{0,2})[(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2] + 4\mu_{1,1}(\mu_{3,0} + \mu_{1,2})(\mu_{2,1} + \mu_{0,3})$$
$$\phi_7 = (3\mu_{2,1} - \mu_{0,3})(\mu_{3,0} + \mu_{1,2})[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{2,1} + \mu_{0,3})^2]$$
$$- (\mu_{3,0} - 3\mu_{1,2})(\mu_{2,1} + \mu_{0,3})[3(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2]$$

The Gradient Norm projection feature is obtained after some processing steps. First, we apply Sobel masks to obtain the directional gradients according to x and y:

$$G_x(i,j) = h_x(i,j) \otimes I(i,j) \qquad G_y(i,j) = h_y(i,j) \otimes I(i,j)$$

where *I(i, j)* is the image gray level information and $h_x(i, j)$, $h_y(i, j)$ are Sobel masks:

$$h_x(i,j) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad h_y(i,j) = \begin{bmatrix} -1 & 0 & -1 \\ -2 & 0 & -2 \\ -1 & 0 & -1 \end{bmatrix}$$

then, gradient norm is computed as follow:

$$G(i,j) = \sqrt{G_x(i,j)^2 + G_y(i,j)^2}$$

As the coefficient number in the gradient norm is the same as the pixel number in the image, we compute the gradient norm projection according to x and y, in order to reduce this feature size:

$$P_{Xi} = \frac{1}{max\, G_{i,j}} \sum_j G(i,j) \qquad P_{Yi} = \frac{1}{max\, G_{i,j}} \sum_j G(i,j)$$

3.2 Examples of description results and discussion

From the presented variety feature vector extraction, an improvement of certain vectors (Kachouri et al., 2008b) is ensured, in order to reduce their size while reserving their relevance description and even improve it.

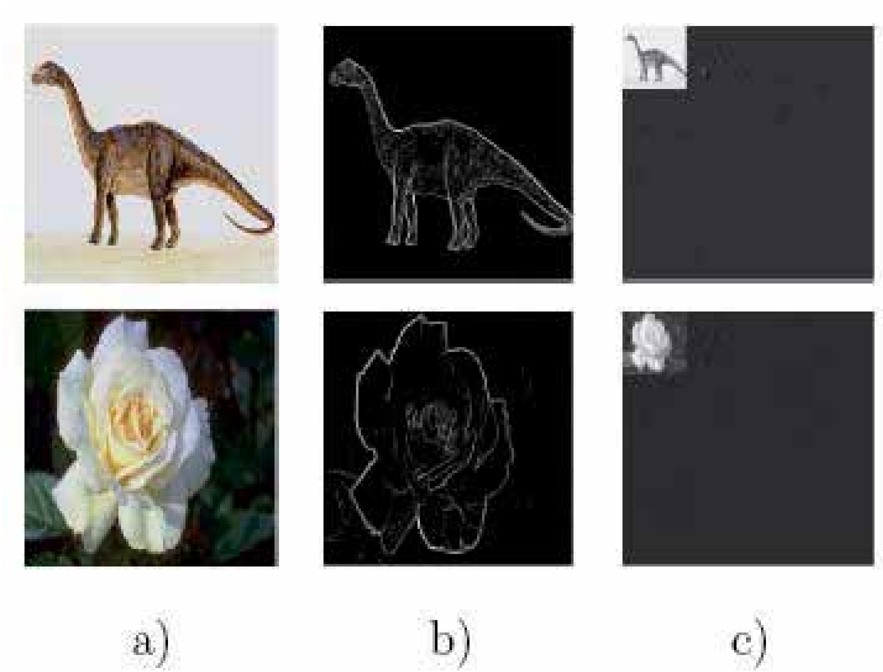


Fig. 4. Extracted feature samples: a) Dinosaur and Flower images , b) Dinosaur and Flower image respectively gradient norm and c) Dinosaur and Flower image respectively two level decomposition Daubechies wavelet coefficients.

For this, we employ the auto-correlogram, computed from correlogram feature (Huang et al. 1999), we retain only the eight most commonly used features since thirteen extracted from image co-occurrence matrix. Also a condensation to compute single value resulting from different level decomposition detail coefficients of the Daubeshies wavelet transformation and the Laplacian approximation filtering, is made, figure 4.

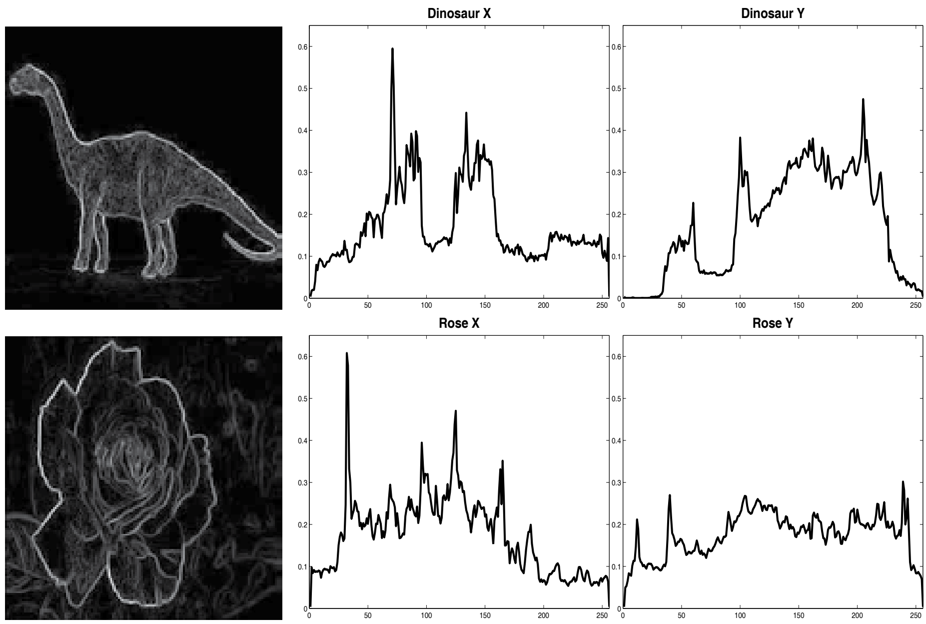


Fig. 5. Extracted features samples: Gradient Norm projection (/x and /y), obtained from Dinosaur and Rose images.

Moreover we use instead of the initial gradient norm matrix, just a gradient norm projection according to x and y axes, which allow to reduce the dimension of this feature while reserving its same description quality. Figure 5 shows discrimination capability of this new feature between images from different classes.

Finally, we retain, from seven extracted, only the first four invariant Hu moments. In addition, these moments are computed from the gradient norm instead of the initial pixel values. This helps to consider the contour more than other information in the image which enhances furthermore the shape description of this feature (figure 6).

This information features will help to build models of objects for the recognition and indexing in a database. To properly accomplish this task, we need a classification system able to categorize all objects and images in a database from their description. In the rest of this chapter, we note the different average color vectors as RGB, HSV, HMMD and YCrCb, histograms as Hist, auto-correlogram as Auto-Corr, first order statics as FOS, co-occurence matrix as SGLD, Daubeshies wavelet transformation as Daub, gradient norm as G-norm and the invariant moments of Hu as Hu.

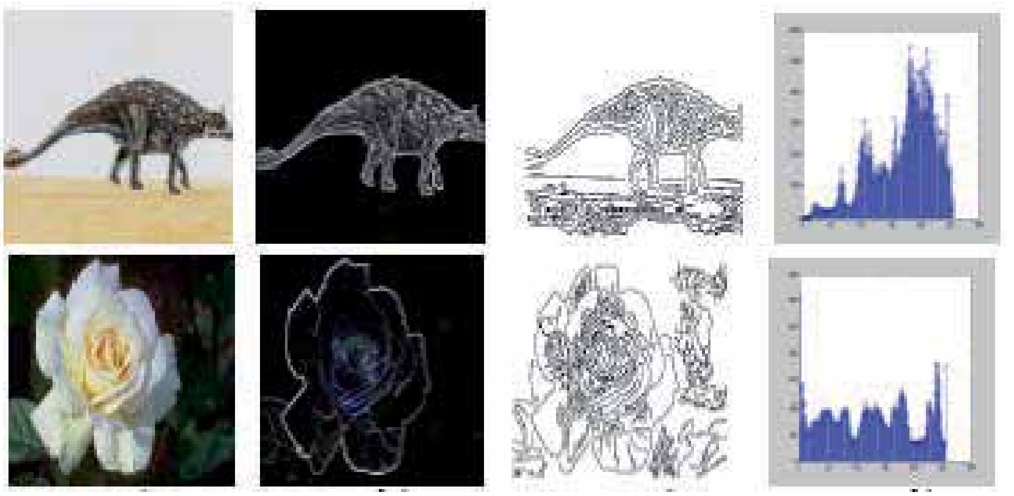


Fig. 6. Contour extraction and Histograms examples: From real images, we compute first the gradient norm, the contours extraction, the histograms and the Hu moments.

4. Classification

We call automatic classification, the algorithmic categorization of the objects and images. This one is to determine in which class or category you can "store" each object, based on similarities criteria. Many classification or learning methods have been already developed, and used in a wide spectrum of applications: diagnosis, bioinformatics, fraud detection, speech recognition, etc.

Once training image feature set is built, a classifier is used to ensure image recognition. There are two classification kinds in the literature. The first is unsupervised classification where the classifier deals directly feature dataset without any prior knowledge (Kachouri et al. 2008c). Classification here is based, generally, on clustering approach. So, CBIR systems allow image access according to their visual characteristics by means of similarity measures. The smaller the similarity distance is, the closer the two images are. The most used distance in literature is the Euclidian one which is really a particular case of the Minkowski metric (Kachouri et al. 2008c), various similarity measures are discussed in (Ramesh et al. 1995). The second type is supervised classification (Kachouri et al. 2008a), where the classifier assumes that there is already image classification, a training stage is required, it is the case of a library or a search engine. Since the Nineties, Support vector machines (SVMs) did not cease arousing the interest of several researcher communities of various expertise fields. Nowadays, SVM performance, in several applications (Bi et al. 2003), (Chen et al. 2006), exceeds those of already established traditional models. In fact, SVM originally formulated for twoclass classification problems, have been successfully extended for multiclass classification problems, and become in a very short period of time the standard state of-the-art tool. Also in the SVM-multiclass, original input space is mapped into a higher dimensional feature space in which an optimal separating hyper-plane is constructed on the basis of Structural Risk Minimization (SRM) to maximize the margin between the different clusters, i.e., generalization ability.

4.1 Support Vector Machines (SVM)

This method of supervised classification was published by (Vapnik, 1998). An SVM is a supervised learning algorithm allows to learning a separator. We have a finite set of separated vectors of R^n, as part of a binary classification into two groups, or two classes. Learning to a group is defined by an associated label to each vector, which is inscribed group 1 or group 2. Find a separator is equivalent to reconstruct a function that takes a vector from a set of samples and can say what group it belongs. However, it expects the SVM good generalization properties, ie if a new vector is introduced and it was not in the set of database, the SVM will say which group it is probably belongs.

The linear separator is the lower part of the SVM, that is a relatively simple problem, and that allows later on to the SVM providing some well more powerful separators. We saw that we started from a finite set of labelled vectors. We say that all the labelled vectors that we give are the set of samples noted S, which contains p elements.

$$S = (\vec{x}_l, y_l)_{1 \le l \le p} \, with \forall l, y_l \in \{-1,1\}$$

The scalar product of two vectors is noted where is $\langle \vec{x}, \vec{y} \rangle$, we can then define the linear separator $f_{\vec{w},b}$ by the following equation:

$$f_{\vec{w},b}(\vec{x}) = \langle \vec{w}, \vec{x} \rangle + b$$

This separator does not provide valid values only -1 or 1, but we assume that the result when $f_{\vec{w},b}(\vec{x})$ is positive, then the vector is of the same class as samples of label 1, and the result when $f_{\vec{w},b}(\vec{x})$ is negative, the vector belongs to the class of examples labelled -1. The equation $f_{\vec{w},b}(\vec{x}) = 0$ defines the border between the two classes. This boundary is an affine hyper-plane in the case of linear separator.

4.1.1 Separability and margin

We suppose that S is linearly separable. The basic idea of SVM is to separate samples of each class, but it is also necessary that the hyper plane passes as possible between the two classes. It is to define this notion that we introduce the margin. The margin is the minimum distances of each sample to the hyper plane (figure 7).

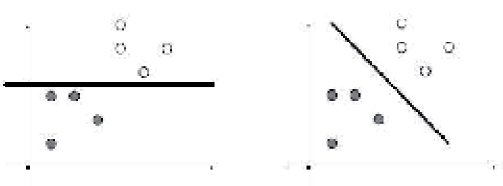


Fig. 7. A separator that the margin is not maximized, have less good generalization properties than that which the margin is maximized.

The aim of the SVM, in case where S is separable, is to give separator S whose margin is maximal, while ensuring that properly separates the samples with label -1 and the samples with label 1.

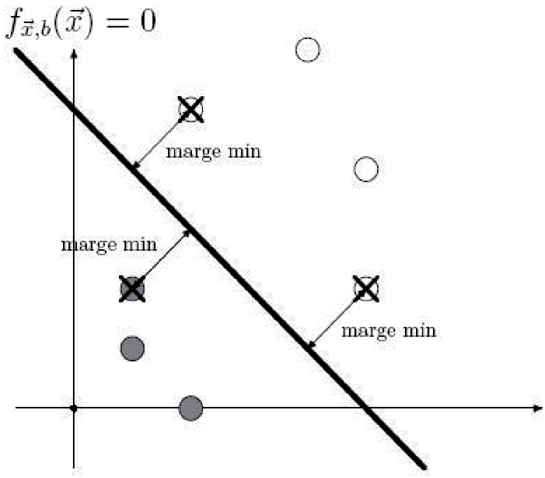


Fig. 8. The separator which should maximize the margin.

The maximum margin separator is such that, the smaller sample has a margin wider than the sample of the smallest margin of the other possible separators. In fact, there are really at least two samples of smaller margin, a class 1 and class -1. They force this margin, and the border of separation passes between them (figure 8). These are the only samples that force the margin, and remove all other samples of the database does not change the separator. These samples are called *support vectors*, hence the name of the method.

4.1.2 General case

For the general case where S is not separable, the solution is to allow some samples to have a lower margin than the margin chosen as the smallest margin or even negative. However, the solution of the problem may be very bad if too many samples are allowed to have a small margin. The idea is to add value margins lower than the maximum margin in the expression to minimize. This avoids that the margins are too low, which limits the samples that do not respect the separability through a separator solution of optimization problem. This is a problem of quadratic convex optimization, i.e. an optimization problem that admits no local optimum, but only one optimum, thus overall. This is crucial because the convexity of the problem is a guarantee of convergence to the SVM solution.

The interest of the kernel functions is that they allow using what we just presented on the linear separation to the non-linear separations. Let S a set of samples labelled by 1 or -1 depending on the class to which they belong, which is not at all linearly separable. The method we have seen works in this case but may give poor results, and many samples became support vectors. The idea of using kernels comes from the assumption that if a set is not linearly separable in the descriptors space, it can be in a space of higher dimension. A better way to separate the samples is to project them into a different space, and perform a linear separation in this space, where this time it should be more adapted. The kernel functions can achieve this projection, and must check a number of properties to ensure the effectiveness of this technique, so you do not have to make calculations in very large dimensions. With the kernel functions, we can work in very large dimensions. However, a linear separation, and a linear regression is facilitated by the projection of data in a space of high dimension. Projecting in the space of descriptors and using an algorithm to maximize the margin, SVM managed to get a severability retaining good generalization capacity, is the central idea of SVM.

For more details on SVMs, we refer interested readers to (Cristianini & Taylor, 2000). A comparison between SVM-multiclass, as supervised classification and Euclidian distance based k-means, as unsupervised classification, is presented in (Kachouri et al., 2008b). The obtained results prove that SVM classifier outperforms the use of similarity measures, chiefly to classify heterogeneous image database. Therefore, we integrate SVM classifier in our proposed image retrieval systems in this chapter.

5. Image recognition and retrieval results through relevant features selection

To ensure a good feature selection during image retrieval, we present and discuss the effectiveness of the different feature kind and aggregation. Since heterogeneous image database contains various images, presenting big content difference. The idea to introduce a system optimization tool was essential when one realized during the carried out tests that the use of all extracted features could be heavy to manage. Indeed, more features vectors dimensions are significant more the classifier has difficulties for their classification. The traditional way that one followed in (Djouak et al., 2005a) and that one finds in many CBIR systems is a diagram which consists of the use of all extracted features in the classification step. Unfortunately, this method presents a great disadvantage, by using all features the classifier manages a great dimensions number. That involves a consequent computing time what creates a real handicap for great images databases. In fact, this problem which is the direct result of the high dimensionality problem was the subject of several works which led to cure it.

Feature (content) extraction is the basis of CBIR. Recent CBIR systems retrieve images based on visual properties. As we use an heterogeneous image database, images are various categories, and we can find a big difference between their visual properties. So a unique feature or a unique feature kind, cannot be relevant to describe the whole image database.

Moreover, while SVM is a powerful classifier, in case of heterogeneous images, given the complexity of their content, some limitations arise, it is that many features may be redundant or irrelevant because some of them might not be responsible for the observed image classification or might be similar to each other. In addition when there are too many irrelevant features in the index dataset, the generalization performance tends to suffer. Consequently, it becomes essential and indispensable to select a feature subset that is most relevant to the interest classification problem. Hence the birth of a new issue, other than image description, it is relevant feature selection. Subsequently, to guarantee a best classification performance, good content image recognition system must be, mainly, able to determine the most relevant feature set, then to well discretize correspond spaces. Feature selection for classification purposes is a well-studied topic (Blum & Langley 1997), with some recent work related specifically to feature selection for SVMs. Proposed algorithms in this regard, shall be ample literature for several years (Guyon & Elisseeff 2003). Although proposed selection methods, are quite varied, two main branches are distinguished, wrappers and filters (John et al. 1994), (Yu & Liu 2004). Filtres are very fast, they rely on theoretical considerations, which allow, generally, to better understanding variable relationships. Linear filters, as PCA (Principal Component Analysis), or FLD (Fisher's Linear Discriminant) (Meng et al. 2002) are very used, but these methods are satisfactory, only if there is a starting data redundancy. (Daphne & Mehran 1996) propose markov blanket algorithms, which allow to found for a given variable xi, a set of variables not including xi that render xi un-necessary. Once a Markov blanket is found, xi can safely be eliminated. But this is only a summary approximation, because this idea is not implementable in practice. However, as it does not take into account the used classifier in generalization stage, all filters kind selection methods still, generally, unable to guarantee high recognition rate. Although conceptually simpler than filters, wrappers are recently introduced by (John et al. 1994). This selection kind uses the classifier as an integral part of the selection process. Indeed, the principle of a feature subset selection is based on its success to classify test images. Therefore, the selected feature subset is well suited to the classification algorithm, in other words, high recognition rates are obtained because selection takes into account the intrinsic bias of classification algorithm. Some specifically related works on feature selection using SVM classifier are recorded in literature (Guyon et al. 2002), (Zhu & Hastie 2003), (Bi et al. 2003), (Chen et al. 2006). The major inconvenient of this selection technique is the need for expensive computation, especially when the variable number grows. More details, are accommodated in (Guyon & Elisseeff 2003) and references therein. To take advantage, of both of these selection kinds, filters speed and selected feature subset adaptability with the used classifier in wrappers, new selection methods ensuring that compromise is always looking. Recently, (Bi et al. 2003) have proposed the use of 1-norm SVM, as a linear classifier for feature selection, so computational cost will not be an issue, then non linear SVM is used for generalization. Other methods combining filters and wrappers are presented in (Guyon & Elisseeff 2003). It is within this framework that we propose in this section, the modular statistical optimization (section 5.1) and the best features type selection (section 5.2) methods.

5.1 Modular statistical optimization

The proposed modular statistical architecture in figure 9 is based on a feedback loop procedure. The principal idea (Djouak et al., 2006) of this architecture is that instead of using all features in the classification step, one categorizes them on several blocks or modules and after one tries to obtain the optimal precision with the minimum of blocks. The introduced modular features database includes all presented features in section 3.

Using all these features one formed four features modules which one can describe as follows: The first module (b1) gathers the all shape features, the second module (b2) gathers the color features, the third module (b3) the texture features and finally the fourth module (b4) the Daubeshies features.

Features Blocs	B1	B2	B3	B4	B5	B6
Concerned Modules	b1	b1 b2	b3	b1 b2 b3	b4	b1 b2 b3 b4

Table. 1. Used features blocks table

The table (table.1) summarizes the obtained features blocks (B1 to B6) by combining the exposed features modules (b1 to b4).

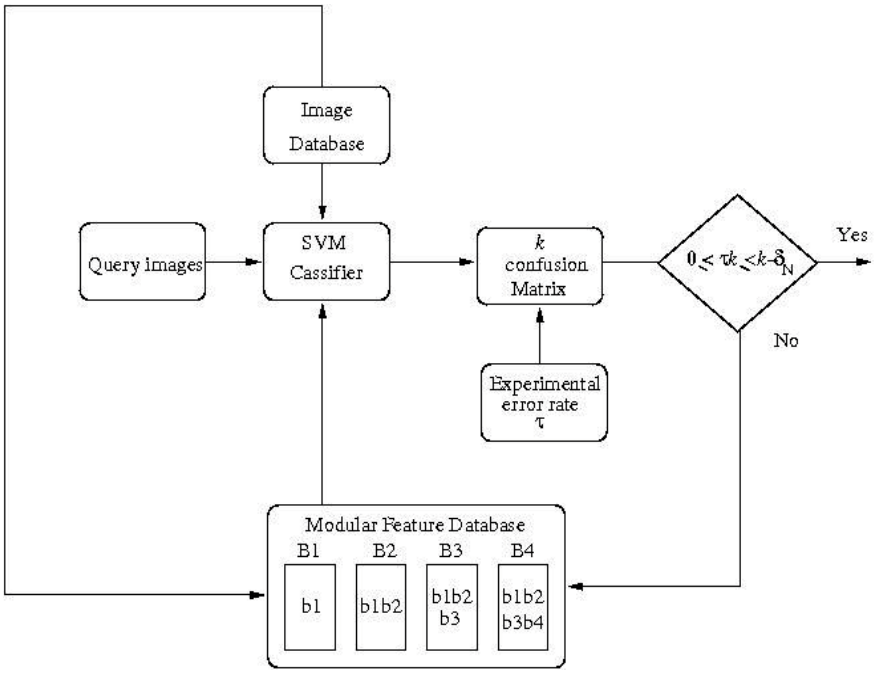


Fig. 9. Modular Statistical optimization architecture.

We can remark in figure 10, for the request (query) image number 4 that the classification rate error is very important for bloc B1. However, the rate error decrease progressively when the others features bloc are used. The presented modular architecture presents some disadvantages, the query images must be included in the database, the experimental rate error is used as prior information. To solve this problem, we propose in the next section the classification procedure based on hierarchical method using the best feature type selection.

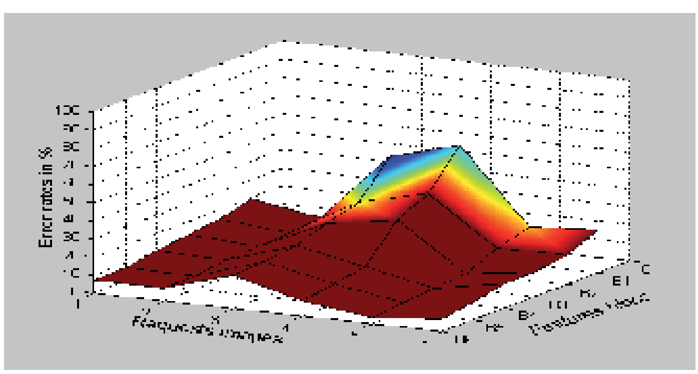


Fig. 10. Average of the classification rate error obtained for different feature blocs.

5.2 Best feature type selection method

The hierarchical feature model is proposed to replace the classical employment of aggregated features (Djouak et al., 2005a), (Djouak et al., 2005b). This method is able to select features and organize them automatically through their kinds and the image database contents. In the off-line stage, due to feature extraction step, we obtain from an image database correspond feature dataset. Then, we start, first of all by the training step, using, at every turn, one group of same feature kind separately, and based on the training rate criterion computed through used classifier, we select hierarchically the best same feature kind. In the on-line stage, we classify each image from the test image database, using separately the different same feature kinds. So, for each image, we will have different clusters as a retrieval result. Then To decide between these various outputs, we treat each two same feature kind outputs together, according to the hierarchical feature selection model, as described in figure 11. We start process within the two latest same feature kind outputs, until reaching the best one. Each time, according to the examined two group of same feature kind outputs, a comparison block, will decide the use of Nearest Cluster Center (NCC) block or not. The NCC block ensure the computation of Euclidian distance between the candidate image and the two cluster centers (clusters used are the two group of same feature kind outputs).

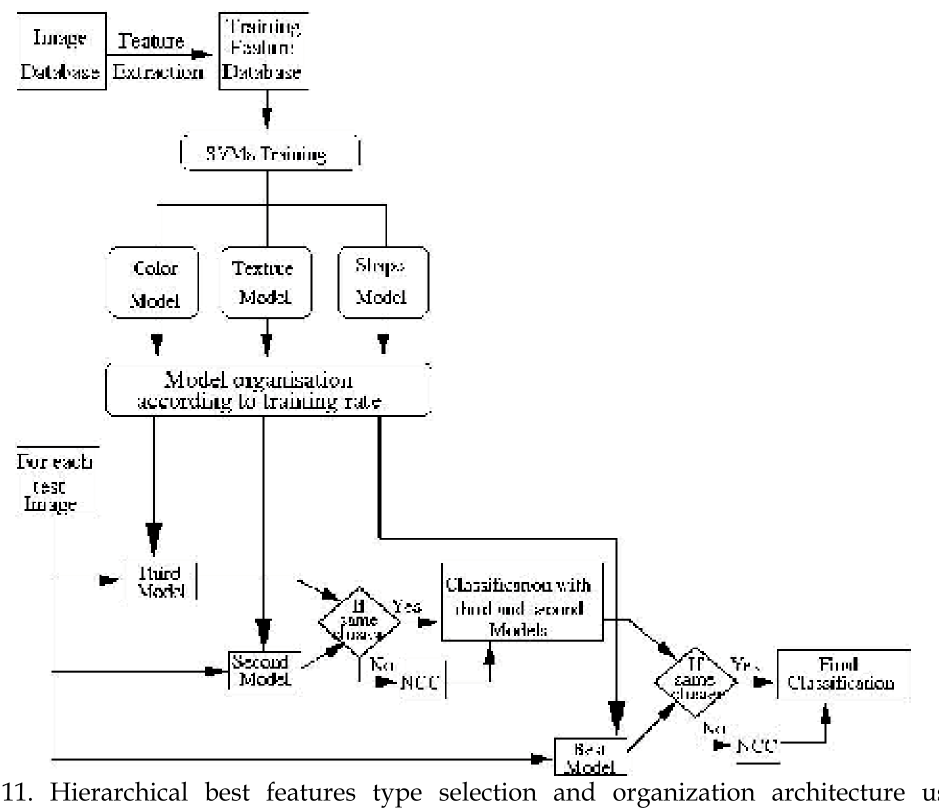


Fig. 11. Hierarchical best features type selection and organization architecture using different SVM models.

A comparison between classical mixed features and the proposed hierarchical feature model is applied. Hierarchical feature model (figure 11) outperforms the use of aggregated features (several feature kind combination) simply by mixing them all together (*color* + *texture* + *shape*). We present, in Figure 12 and Figure 13, the first 15 images retrieved for a query image, using respectively aggregated features and hierarchical features. In these two figures, the first image is the request image. We observe, obviously, that the retrieval accuracy of hierarchical feature model is more efficient than that of aggregated feature use. However, we demonstrate in this section that the feature aggregation is not enough efficient, if we just mix various feature kind. Indeed, each descriptor kind range is different than those of the other descriptor kinds.

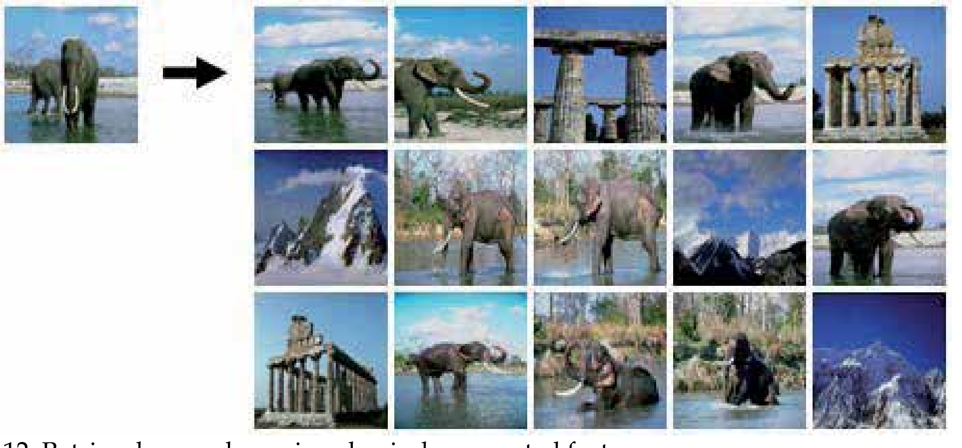


Fig. 12. Retrieval examples, using classical aggregated features.

So, each feature vector extracts signature which is not uniform with the other feature signature extracted from images.

Fig. 13. Retrieval examples, using hierarchical feature model.

Consequently, we prove that using proposed hierarchical feature model is more efficient than using aggregated features in an heterogeneous image retrieval system.
Figure 14 proves that using the hierarchical feature model is more efficient than using aggregated features in an image retrieval system. Indeed, we obtain with hierarchical features model 0,815 % representing the good classification results and 0,68 % with aggregated features method.

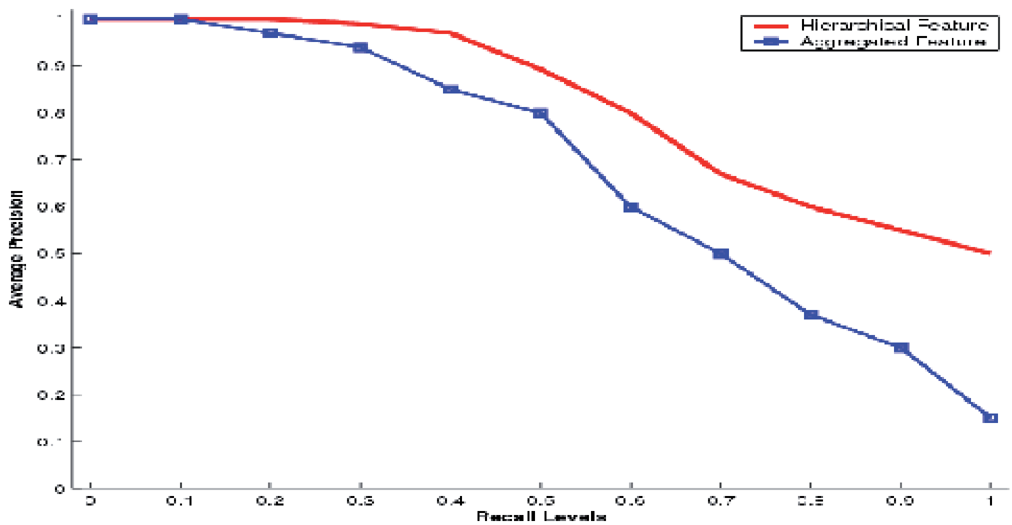


Fig. 14. Precision-recall graph comparing hierarchical features and Aggregated Features.

6. Conclusion

In this chapter, we have presented the different stages of image recognition and retrieval system dedicated to different applications based computer vision domain. The image description and classification constitute the two important steps of an image recognition system in large heterogeneous databases. We have detailed the principles of the features extraction, image description contained in large database and the importance of robustness.
After presenting the features extraction and some improvement we have detailed the importance of the classification task and presented the supervised SVM classifier.
To ensure a good feature selection during image retrieval, we have presented and discussed the effectiveness of the different feature kind and aggregation. We have detailed the need of the optimization methods in CBIR systems and we have proposed two architectures, the modular statistical optimization and the hierarchical features model. The satisfactory obtained results show the importance of optimization and the features selection in this domain.
Searching CBIR systems remain a challenges problem. Indeed, the different domains has been unable to take advantage of image retrieval and recognition methods and systems in spite of their acknowledged importance in the face of growing use of image databases in mobile robotics, research, and education. The challenging type of images to be treated and the lacking of suitable systems have hindered their acceptance. While it is difficult to develop a single comprehensive system, it may be possible to take advantage of the growing

research interest and several successful systems with developed techniques for image recognition in large databases.

7. References

Antania, S., Kasturi, R. & Jain, R. (2002). A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video, *Pattern recognition,* 35(4), pages: 945-965.

Bi J., Bennett K., Embrechts M., Breneman C., & Song M., (2003), Dimensionality reduction via sparse support vector machines, *J. Machine Learning Research (JMLR)*, 3, 1229–1243.

Bimbo A. D., Visual Information Retrieval, (2001), *Morgan Kaufmann Publishers, San Francisco, USA*.

Blum A.L. & Langley P., (1997), Selection of Relevant Features and Examples *in Machine Learning, Artificial Intelligence*, 97(1- 2), 245–271.

Carson, C., Belongie, Se., Greenpan, H. & Jitendra, M. (2002). Blobworld: Image segmentation using Expectation-Maximization and its Application to Image Querying, *IEEE Transaction on Pattern Analysis and Machine Intelligence,* Vol. 24, NO. 8.

Chen Y. & Wang J.Z., (2004), Image Categorization by Learning and Reasoning with Regions, J. Machine Learning Research, vol. 5, pp. 913–939.

Chen Y., Bi J. & Wang J.Z., (2006), MILES: Multiple-Instance Learning via Embedded Instance Selection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1931–1947.

Cristianini N. & Taylor J. S., (2000), An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, *Cambridge University Press.*

Daphne K. and Mehran S., (1996), Toward optimal feature selection, *In International Conference on Machine Learning*, 284–292.

Delingette H. & Montagnat J., (2001), Shape and topology constraints on parametric active contours, *Computer Vision and Image Understanding*, vol. 83, no. 2, 140–171.

Djouak, A., Djemal, K. & Maaref, H. (2007). Image Recognition based on features extraction and RBF classifier, *Journal Transactions on Signals, Systems and Devices, Issues on Comminucation and Signal Processing,* Shaker Verlag, Vol. 2, N°. 3, pp: 235-253.

Djouak, A., Djemal, K. & Maaref, H., (2005a), Image retrieval based on features extraction and RBF classifiers, *IEEE International Conference on Signals Systems and Devices, SSD 05*, Sousse. Tunisia.

Djouak, A., Djemal, K. & Maaref, H.k, (2006). Modular statistical optimization and VQ method for images recognition, International Conference on Artificial Neural Networks and Intelligent Information Processing, pp: 13-24, ISBN: 978-972-8865-68-9, Setúbal, Portugal, August.

Djouak, A., Djemal K. & Maaref, H., (2005b), Features extraction and supervised classification intended to image retrieval. *In IMACS World Congress: Scientific Computation, Applied Mathematics and Simulation,* Paris, France.

Egmont-Petersen, M., de Ridder & Handels, D. H. (2002). Image processing with neural networks-a review. *Pattern Recognition,* 35(10):2279–2301.

Faloutsos C., Equitz W., Flickner M., Niblack W., Petkovic D. & Barber R., (1994), Efficient and Effective Querying by Image Content, *Journal of Intelligent Information Systems,* vol. 3, No 3/4, 231– 262.

Guyon I.,Weston J., Barnhill S., & Vapnik V., (2002), Gene selection for cancer classifcation using support vector machines, *Machine Learning* 46, 389–422.

Guyon I., & Elisseeff A., (2003), An introduction to feature and variable selection, *Journal of Machine Learning Research*, vol. 3, 1157–1182.

Hafner J., Sawhney H.S., Equitz W., Flickner M., & Niblack W., (1995), Efficient color histogram indexing for quadratic form distance function, IEEE Trans. *Pattern Anal. Mach. Intell.,* 17, 7, 729–736.

Haralick R.M.S.K., & Dinstein I., (1973), Textural Features for Image Classification, *IEEE Transactions on Systems, Man and Cybernetics*, 3(6), 610-621.

Hu M.K., (1962), Visual pattern recognition by moment invariants, *IEEE Transactions information Theory*, 8, 179–187.

Huang J., Kumar S.R., Mitra M., & Zhu W.J., Spatial color indexing and applications, (1999), *Intl. Conf. on Computer Vision*, 35, 3, 245–268.

Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J. & Zabih, R., (1997). Image indexing using color correlograms. *In Proc. IEEE Comp. Soc. Conf. Comp. Vis. and Patt. Rec.,* pages 762–768.

Jacobs, C., Finkelstein, A. & Salesin, D. (1995). Fast multiresolution image querying. *In Proc. SIGGRAPH.*

Julezs. B., (1975), Experiments in the visual perception of texture. *Scientific American*, 232(4):2–11.

John G.H., Kohavi R., & Pfleger K., Irrelevant features and the subset selection problem, 1994, *In International Conference on Machine Learning, pages 121–129. Journal version in AIJ*, available at http ://citeseer. nj.nec.com/13663.html.

Kachouri R., Djemal K., Maaref H., Sellami Masmoudi D., & Derbel N., (2008b), Heterogeneous Image Retrieval System Based On Features Extraction and SVM Classifier, *International Conference on Informatics in Control, Automation and Robotics, ICINCO 08*, Funchal, Madeira, Portugal.

Kachouri R., Djemal K., Sellami Masmoudi D., & Derbel N., (2008c), Content Based Image Recognition based on QUIP-tree Model, *IEEE International Conference on Signals Systems and Devices, SSD 08*, Amman, Jordan.

Kachouri R., Djemal K., Maaref H., Masmoudi, D.S., Derbel N., (2008), Content description and classification for Image recognition system, *3rd IEEE International Conference on Information and Communication Technologies: From Theory to Applications, ICTTA08* 1–4.

Kadyrov, A. & Petrou, M. (2001). The Trace transform and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, Vol:811–828.

Lin, Chun-Yi., Yin, Jun-Xun., Gao, X., Chen, Jian-Y. & Qin, P. (2006). A Semantic Modeling Approach for Medical Image Semantic Retrieval Using Hybrid Bayesian Networks, *Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)*, vol. 2, pp.482-487.

Lipson, P., Grimson, E. & Sinha, P. (1997). Configuration based scene classification and image indexing. In *Proc. IEEE Comp. Soc. Conf. Comp. Vis. and Patt. Rec.,* pages 1007–1013.

Manjunath B.S., Ohm J.-R., Vasudevan V.V., & Yamada A., (2001), Color and texture descriptors, *IEEE transaction on circuits and systems for video technology*, 11, 703–715.

Meng J.E., Shiqian W., Juwei L., & Hock L.T., (2002), Face Recognition With Radial Basis Function (RBF) Neural Network, *IEEE Transaction on Neural Networks*, 13, 3.

Press W.H., Flannery B.P., Teukolsky S.A., & Vetterling W.T., (1987), Numerical Recipes, The Art of Scientific Computing, *Cambrigde, U.K.*: Cambridge Univ.

Rezai-Rad G., & Aghababaie M., (2006), Comparison of SUSAN and Sobel Edge Detection in MRI Images for Feature Extraction, *Information and Communication Technologies, ICTTA 06*. 1, 1103–1107.

Ramesh J. R., Murthy S.N.J., Chen P.L.-J., & Chatterjee S., (1995), Similarity Measures for Image Databases, *Proceedings of IEEE International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium*, 3, 1247-1254.

Sastry, Ch. S., Pujari, Arun K., Deekshatulu, B. L. & Bhagvati, C. (2004). A wavelet based multiresolution algorithm for rotation invariant feature extraction, *Pattern Recognition Letters,* 25:1845–1855.

Sclaroff, S. & Pentland, A. (1995). Modal Matching for Correspondence and Recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence,* 17(6):545–561.

Serrano N., Savakisb A.E., & Luoc J., (2004), Improved scene classification using efficient low-level features and semantic cues, *Pattern Recognition*, 37 , 1773–1784.

Smith, J. R. & Chang, S.-F. (1996). Tools and techniques for color image retrieval. *In SPIE Proc. Storage and Retrieval for Image and Video Databases*, volume 2670, pages 426–437.

Stricker, M. & Dimai, A. (1997). Spectral covariance and fuzzy regions for image indexing. *Machine Vision and Applications,* 10(2):66–73.

Stricker, M., & Swain, M. (1994). Capacity and the sensitivity of color histogram indexing, *Technical Report*, 94-05, University of Chicago.

Swain, M. & Ballard, D. (1991). Color indexing, *International Journal of Computer Vision,* 7(1), pages:11–32.

Takashi, I. & Masafumi, H. (2000). Content-based image retrieval system using neural networks. *International Journal of Neural Systems,* 10(5):417–424.

Vapnik, V., (1998). Statistical learning theory, *Wiley-Interscience.*

Wu, J., (2003), Rotation Invariant Classification of 3D Surface Texture Using Photometric Stereo. *PhD Thesis, Heriot-Watt University*.

Yu L. & Liu H., (2004), Efficient Feature Selection via Analysis of Relevance and Redundancy, *J. Machine Learning Research*, vol. 5, pp. 1205–1224.

Zhu J., & Hastie T., (2003), Classication of gene microarrays by penalized logistic regression, *Biostatistics*.

Edited by A D Rodić

The present edited book is a collection of 18 chapters written by internationally recognized experts and well-known professionals of the field. Chapters contribute to diverse facets of automation and control. The volume is organized in four parts according to the main subjects, regarding the recent advances in this field of engineering. The first thematic part of the book ¬†is devoted to automation. This includes solving of assembly line balancing problem and design of software architecture for cognitive assembling in production systems. The second part of the book concerns different aspects of modelling and control. This includes a study on modelling pollutant emission of diesel engine, development of a PLC program obtained from DEVS model, control networks for digital home, automatic control of temperature and flow in heat exchanger, and non-linear analysis and design of phase locked loops. The third part addresses issues of parameter estimation and filter design, including methods for parameters estimation, control and design of the wave digital filters. The fourth part presents new results in the intelligent control. This includes building a neural PDF strategy for hydroelectric satation simulator, intelligent network system for process control, neural generalized predictive control for industrial processes, intelligent system for forecasting, diagnosis and decision making based on neural networks and self-organizing maps, development of a smart semantic middleware for the Internet , development of appropriate AI methods in fault-tollerant control, building expert system in rotary railcar dumpers, expert system for plant asset management, and building of a image retrieval system in heterogeneous database. The content of this thematic book admirably reflects the complementary aspects of theory and practice which have taken place in the last years. Certainly, the content of this book will serve as a valuable overview of theoretical and practical methods in control and automation to those who deal with engineering and research in this field of activities.

ISBN 978-953-51-5858-5

Photo by Vladimir_Timofeev / iStock

IntechOpen