INTERNATIONAL JOURNAL

of

COMPUTERS, COMMUNICATIONS & CONTROL

Year: 2006 Volume: I Number: 2

CCC Publications

www.journal.univagora.ro

ISSN 1841-9836 (print version), 1841-9844 (online version)

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

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

CCC Publications, powered by AGORA University Publishing House, currently publishes the "International Journal of Computers, Communications & Control" and its scope is to publish scientific literature (journals, books, monographs and conference proceedings) in the field of Computers, Communications and Control.

Contents

International Journal of Computers, Communications & Control
Vol. I (2006), No. 2, pp. 5-5

Preface

Gastón Lefranc Hernández

It is an honor for me to introduce you a list of papers selected from the XVI Congress of ACCA (Asociación Chilena de Control Automático, Chilean Association of Automatic Control), done in Santiago of Chile at the end of 2004, in its 30 years of existence.

Since 1974, ACCA organize Congress (every 2 years), Tutorial Courses for continuing education, Seminars, Workshops, Exposition of Systems and Equipments and our Magazine Automatica and Innovation. All of these activities have results: important influence in all ambits, consolidating to ACCA as a point of meeting of people from industries, from private and public institutions, from the academics world, from suppliers of systems and equipments, and professional from all Latin America . In 1976, ACCA had its first Program Committee to select papers, and its first Proceedings.

This Congress was organized by ACCA, IEEE Chilean Chapter on Control, Robotics and Cybernetics, IFAC and its Technical Committees, and the Universidad de Las Américas, Chile. ACCA is NMO of IFAC since 1984.

ACCA Congress had Keynotes Speakers for Plenary: Dr. François Vernadat from France, Dr. Florin Filip from Romania, Dr. Shimon Nof from USA and Dr. Philippe Dupont, from France. A Round Table was organised to discuss about productivity, interested in Chile and Latin American professionals.

The 46 accepted papers selected by the International Program Committee, were presented in Technical Sessions where new ideas, critical comments, and the beginning of cooperation leading to future projects will take place. The papers come from Argentina, Brazil, Chile, Cuba and México. The International Program Committee selected 10 papers as the best of this Congress, to be published in "International Journal of Computers, Communications & Control". The papers are in areas of artificial intelligence, vision, manufacturing, pattern recognition and robotics.

ACCA wants to say thanks to Dr. Florin Filip to permit to show the activities in Chile and Latin America, oriented to improve and to solve our own problems.

It is pleasure to introduce you to this Special Issue Conference dedicated to ACCA in its 30 years.

Gastón Lefranc Hernández
Chairman of International Program Committee
XVI Congress of ACCA
Guest Editor of this Issue

International Journal of Computers, Communications & Control
Vol. I (2006), No. 2, pp. 7-14

Development of a Matlab® Toolbox for the Design of Grey-Box Neural Models

Gonzalo Acuña, Erika Pinto

Abstract: A Matlab Toolbox is developed for the design, construction and validation of grey-box neural network models. This toolbox, available in www.diinf.usach.cl/gacuna has been tested in simulations with a continuously stirred reactor process. The grey-box model performs well for validation data with 5% additive gaussian noise for one-step-ahead (OSA) and model-predictive-output (MPO) estimations.
Keywords: Grey-Box Model, Neural Networks, One-Step-Ahead estimation, Multiple Prediction Output, Time variant parameter identification.

1 Introduction

In the development of dynamic system models it is best to take advantage of the a priori knowledge of a process, generally expressed in terms of a set of ordinary differential equations which represent mass or energy balances. In complex biotechnological processes, the most difficult task is the modeling of the time-varying parameters, such as specific kinetics. In order to address this problem [1], proposed the use of grey-box models which combine a priori knowledge expressed in terms of a phenomenological, or white-box model, with a black-box model such as neural network. These models have proved to be satisfactory for dynamic systems, they have better generalization characteristics, and they can be identified with a smaller amount of data [1]. [2] classified these grey-box models into two principal categories: those which deliver intermediate values (of parameters or variables) for use in phenomenological models (serial grey-box models), or those in parallel with the dynamic model, adjusted to compensate for modeling errors (parallel grey-box models). [3] showed that the series strategy resulted in grey-box models with superior results. More recently [4] and [5] have employed and analyzed this type of model demonstrating their performance and their use in complex processes.

Matlab® is well known as a completely integrated application development environment, oriented to projects that involve complex mathematical calculations and graphic visualizations. This software has a large variety of toolboxes which are specialized packets which can carry out different functions according to the area of development, for example; optimization, image processing, neural networks, simulations and statistics among others. Thus, the present work deals with the creation of a group of functions integrated in a Matlab® Toolbox which allows the development of an Grey-Box Neural Model (GBNM) for complex systems in general.

The present document is organized as follows: Section 2 details some of the aspects of the GBNMs used, Section 3 examines the design and construction of the Toolbox; Section 4 looks at the applications of the Toolbox; and Section 5 outlines the conclusions reached.

2 Grey-Box Neuronal Models

As mentioned previously GBNMs take advantage of the combination of the a priori knowledge of a given process -expressed in terms of a set of differential equations that represent the first principles that govern the process- with neural networks. The latter are responsible for modeling the interaction between variables that are relevant to the system, and certain time-varying. It is a well established fact that neural networks are capable of approximating non-linear functions. In particular, it has been demonstrated that perceptrons, with only one hidden layer and an adequate number of neurons in their internal layer, are

universal approximators [6].

For the purposes of the present work it is important to distinguish between two training modes for neural networks included in GBNMs. The first type, also known as the direct learning mode [5], uses the error generated at the output of the neural network for the correct determination of its weights (Figure 1).

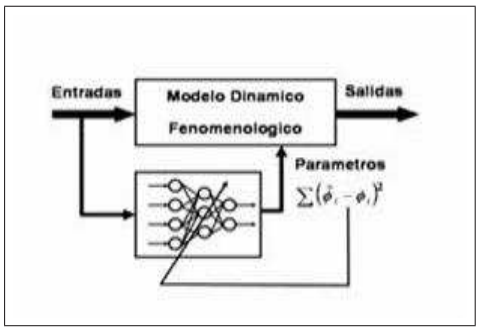


Figure 1: Grey-Box Neural Model in its direct learning mode.

The second type corresponds to an indirect mode by which the error generated at the output of the GBNM is used for the training of the neural network [5] (Figure 2).

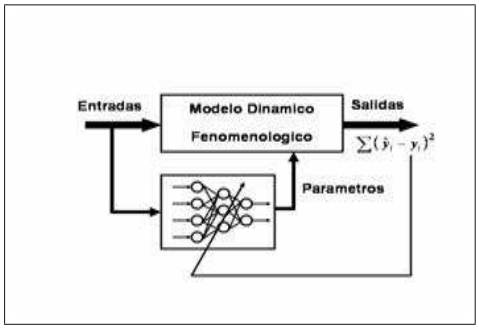


Figure 2: Grey-Box Neural Model in its indirect learning mode.

The Toolbox developed in the present work is based on the direct learning mode of the neural network. The neural networks used are multi-layered perceptrons with only one hidden layer. The training algorithm is error retropropagation combined with a Levenberg-Marquardt optimization.

The validation of the results obtained is carried out with tests that consist in evaluating the error produced when using the GBNM for a One Step Ahead (OSA) prediction, and for Model Predictive Output (MPO) prediction, [7]. The error indices used are three: the Root Mean Square (RMS) error, the Relative Standard Deviation (RSD) and the Adecuation Index (IA), which are presented below [7]:

$$RMS = \sqrt{\frac{\sum_{i=1}^{n}(o_i - p_i)^2}{\sum_{i=1}^{n} o_i^2}}; \qquad RSD = \sqrt{\frac{\sum_{i=1}^{n}(o_i - p_i)^2}{N}}; \qquad IA = 1 - \frac{\sum_{i=1}^{n}(o_i - p_i)^2}{\sum_{i=1}^{n}(|o_i'| + |p_i'|)^2}$$

Where o_i and p_i are the observed and predicted values respectively, in time i, and N is the total number of data. $p_i' = p_i - o_m$ and $o_i' = o_i - o_m$, where o_m is the median value of the observations.

3 Development of the Matlab Toolbox

This Toolbox consists of two principal parts: a Neural Network and serial Grey Box Model, either with OSA or MPO estimates.

The methodology used corresponds to the Modified Cascade Model which has the following stages:

- Definition of requirements.
- System Design.
- Implementation and testing of units.
- Integration and testing of system.

The details of the methodology can be found in [8].

3.1 Neural Network

This is the first part of the Toolbox, and its principal function is to create and train the network that the serial Grey-Box Model uses for estimating the unknown parameter or parameters which are then combined with the phenomenological part of the model. For this reason it is necessary to run this part of the program before running the serial Grey-Box Model.
The principal functions of the neural network are the following:

- Verify the existence of data: This function's role is to verify the existence of the data necessary for the functioning of the neural network.
- Verify dimensions: This function verifies that the quantity of input output data is equal.
- Verify layers: This function verifies that the total number of neurons in hidden layers is correct.
- Divide data: This function divides input and output data that have been entered by the user into training and validation data, (70% and 30% respectively).
- Create and train the network: This function creates and trains the neural retropropagation network based on the data input by the user. This is carry out using the Matlab® neural network Toolbox functions. For the creation of the neural network the "trainlm" function is used which applies the Levenberg-Marquardt method for optimizing weights.
- Simulate network outputs: This function simulates network outputs for the purpose of validation.
- Graph network inputs: This allows the visualization of the behavior of the entered input.

3.2 Serial Grey Box Model

This is the second part of the Toolbox which allows the development of general serial Grey-Box Models either with OSA or MPO estimations.

The principal functions of the Serial Grey-Box Model applied either to OSA or MPO estimations are the following:

- Verify the existence of the files: This function verifies the existence of the files necessary for the functioning of the serial Grey-Box Model.
- Verify the existence of data: Among these data are the number of iterations, the initial values for the status vector, the value of delta t, and where applicable to input u and the real values of the status vector variables for each instant of time used.

- Verify dimensions: This function verifies that the dimensions of the input vector be the same as the number of iterations. For estimating OSA the quantity of real values of the status variables must be equal to the number of iterations. For MPO it only returns the results for the output of the system, but cannot evaluate said output with real data.
- Develop Model: This function is in charge of developing the Grey-Box Model which consists of estimating the unknown parameter or parameters by using the neural network created and trained previously in order to develop the different equations entered.
- Return different graphs: Graphs the inputs of the system, each of the components of the status vector for the total number of iterations, the validation of the unknown parameter or parameters, and the output of the system together with its validations.
- Calculate the error indices: Calculates RSD, RMS and IA for the output.
- Display results: returns the last iteration both for the status vector and the output vector, and for the error indices that correspond to the unknown parameter or parameters or the output or outputs of the system.

4 Aplications and Results

4.1 Description of the process

In the simulation of a CSTR process a first order exothermic reaction is used [9], in which the inputs to the system correspond to the temperature of the cooling sleeve and the system output corresponds to the degree of completion of the reaction. The systems is described by the following status equations:

$$x_1' = -x_1 - D_a \cdot (1-x_1) \cdot e^{\left(\frac{x_2}{1+\frac{x_2}{\gamma}}\right)} \tag{4.1}$$

$$x_2' = -x_2 - B.D_a \cdot (1-x_1) \cdot e^{\left(\frac{x_2}{1+\frac{x_2}{\gamma}}\right)} + \beta.(u-x_2) \tag{4.2}$$

$$y = x_1 \tag{4.3}$$

Where: x_1 : Degree of completion of the reaction; x_2 : Adimensional temperature of the reactor contents; u : Input that corresponds to the adimensional flow rate of the heat-transfer fluid through the cooling sleeve, and y is the output of the system.

For the purposes of this simulation the following values are used for the model's constants:

$$D_a = 0,072; \qquad B = 8,0; \qquad \beta = 0,3; \qquad \gamma = 20,0$$

Because Grey-Box Models are made up of two parts, one of which is the phenomenological model represented by the different differential equations, and the other of which corresponds to the empirical model represented by the neural network, the equations that represent the phenomenological model of the serial Grey-Box Model are the following:

$$x_1' = -x_1 + 0,072.(1-x_1).\rho \tag{4.4}$$

$$x_2' = -x_2 + 8.0,072.(1-x_1).\rho + 0,3.(u-x_2) \tag{4.5}$$

$$y = x_1 \tag{4.6}$$

Where ρ is the parameter that is difficult to obtain:

$$\rho = e^{\left(\frac{x_2}{1+\frac{x_2}{20}}\right)}$$

When equations 4.4 through 4.6 are discretized in the interval between t and t+1, the following equations are obtained:

$$x_{1(t+1)} = x_{1(t)} + (-x_{1(t)} + 0,072.(1-x_{1(t)}).\rho).\Delta t \tag{4.7}$$

$$x_{2(t+1)} = x_{2(t)} + (-x_{2(t)} + 8.0,072.(1-x_{1(t)}).\rho + 0,3.(u-x_{2(t)})).\Delta t \tag{4.8}$$

Thus the equations that constitute the phenomenological part of the serial Grey-Box Model, either with OSA or MPO estimations, are the following:

$$x_{1(t+1)} = x_{1(t)} + (-x_{1(t)} + 0,072.(1-x_{1(t)}).\rho).\Delta t \tag{4.9}$$

$$x_{2(t+1)} = x_{2t} + (-x_{2(t)} + 8.0,072.(1-x_{1(t)}).\rho + 0,3.(u-x_{2(t)})).\Delta t \tag{4.10}$$

$$y = x_{1(t+1)} \tag{4.11}$$

4.2 OSA Estimation with 5% noise

The validation obtained for the unknown parameter with 5% noise can be seen in Figure 3.

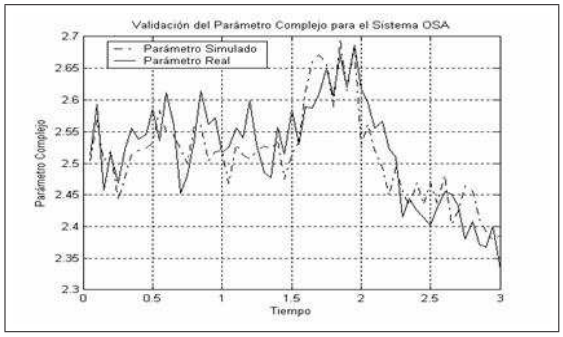


Figure 3: Validation of the unknown parameter with OSA estimation and 5% noise.

The validation obtained from the output of the system can be seen in Figure 4.

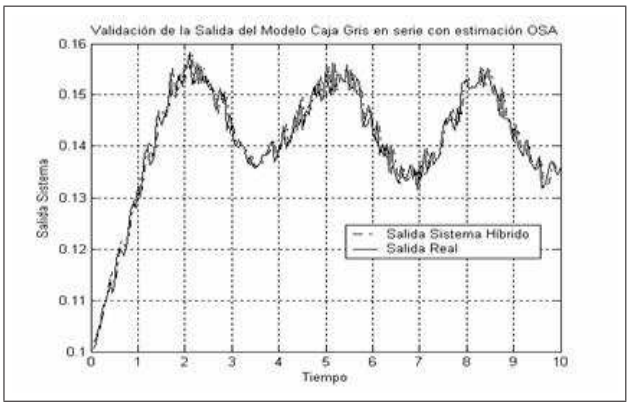


Figure 4: Validation of the output of the system with OSA estimation and 5% noise.

The error indices obtained from the serial Grey-Box Model with OSA estimation and 5% noise, for parameter ρ and the output of the system are presented in Table 4.1.

Table 4.1. Error indices for parameter and the output of the system with an OSA estimation and 5% noise.

	Parameter ρ	System Output
RSD	4.368098E-2	2.756748E-3
RMS	1.733075E-2	1.944067E-2
IA	9.111975E-1	9.846065E-1

The error indices obtained are quite acceptable given that the acceptability values are $RSD < 0.1$, $RMS < 0,1$ and $IA > 0.9$, as explained in Section 2.6 This indicates that the serial Grey-Box Model developed for the simulation of a CSTR process, with OSA estimation and 5% noise, is quite good.

4.3 MPO Estimation with 5% noise

The validation obtained for the unknown parameter with 5% noise can be seen in Figure 5.

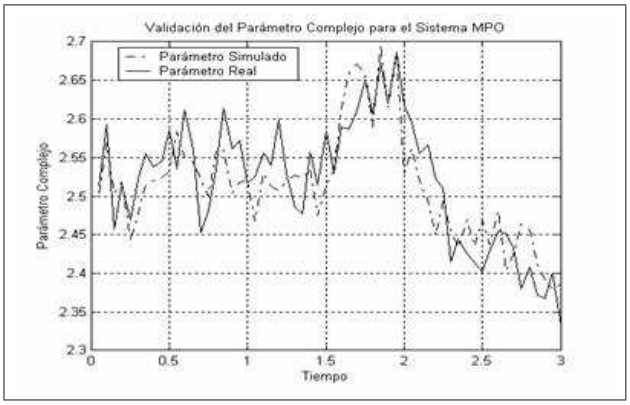


Figure 5: Validation of the unknown parameter with MPO estimation and 5% noise.

The validation obtained from the output of the system can be seen in Figure 6.

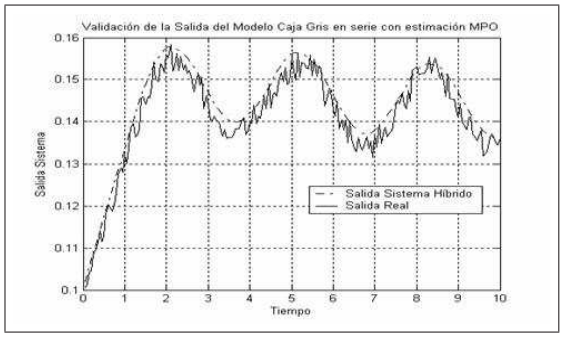


Figure 6: Validation of the output of the system with MPO estimation and 5% noise.

The error indices obtained from the serial Grey-Box Model with MPO estimation and 5% noise, for parameter ρ and the output of the system are presented in Table 4.2.

Table 4.2. Error indices for parameter ρ and the output of the system with an MPO estimation and 5% noise.

	Parameter ρ	System Output
RSD	4.368098E-2	3.441815E-3
RMS	1.733075E-2	2.427178E-2
IA	9.111975E-1	9.764686E-1

The error indices obtained are quite acceptable given that the acceptability values are $RSD < 0.1$, $RMS < 0,1$ and $IA > 0.9$. This indicates that the serial Grey-Box Model developed for the simulation of a CSTR process, with MPO estimation and 5% noise, is quite good.

5 Conclusions

Grey-Box Models constitute a real alternative for those real world processes for which the available a priori knowledge is incomplete, for example in a variety of industrial processes. As in Grey-Box Models only some of the physical and/or chemical laws that represent the model are known, and there are unknown parameters that must somehow be estimated, multi-layered perceptron neural networks have been employed for their notable capacity to approximate complex functions on the basis of observed data.

One of the advantages of this Toolbox, with reference to the creation and training of neural networks, is that neuron transfer functions for each of the hidden and output layers that make up the neural network, can be determined by the user, who can then create and train the neural network in a manner that is in keeping with the specific problem that is to be modeled.

Concerning the estimation of unknown system parameters, the developed Toolbox has the capacity to estimate as many parameters as necessary as long as the necessary data are available.

The developed Toolbox (available at www.diinf.usach.cl/gacuna), allows the creation of serial Grey-Box Models, either with OSA (One Step Ahead) or MPO (Model Predictive Output) estimations, depending on what is to be modeled. With the creation of this Matlab® Toolbox, the end user has access to a simple, trustworthy and fast tool which allows the development and subsequent manipulation of different serial Grey-Box Models, either with OSA or MPO estimations for solving a particular problem. This tool will become part of the Matlab® Toolboxes such that the user can access it at will simply by downloading Matlab®. Thus this new tool becomes a real and simple alternative for modeling those real work processes for which a priori knowledge available is incomplete.

The error indices obtained are good, both for the unknown parameters and for each of the outputs of the system, even when the level of noise applied to the various input data was 5%, which indicates that the models developed are of good quality.

Acknowledgements

The authors wish to thank partial financing provided by Fondecyt Project 1040208.

References

[1] Psichogios, D.; Ungar, L. (1992). “A Hybrid Neural Network-First Principles Approach to Process Modeling”, Computers & Chemical Engineering, 38(10): 1499-1511.

[2] Thompson, M.; Kramer, M. (1994). “Modeling Chemical Processes Using Prior Knowledge and Neural Networks”, Computers & Chemical Engineering , 40(8):1328-1340.

[3] Van Can, H; Hellinga, C.; Luyben, K.; Heijnen, J. (1996). “Strategy for Dynamic Process Modeling Based on Neural Network in Macroscopic Balances”, AIChE Journal, 42:3403-3418.

[4] Thibault, J, Acuña, G., Pérez-Correa, R., Jorquera, H., Molin, P., Agosin, E., (2000) “A hybrid representation approach for modelling complex dynamic bioprocesses” Bioprocess Engineering, 22(6):547-556.

[5] Acuña G.; Cubillos, F.; Thibault, J.; Latrille, E. (1999). “Comparison of Methods for Training Grey-Box Neural Network Models”, Computers & Chemicals Engineering Supplement, 23:561-564.

[6] Hornik K. Stinchcombe M., White H., (1989) “Multilayer Feedforward Networks Are Universal Approximators”, Neural Networks, 2:359-366.

[7] Billings S.A., Jamaluddin H. B. y Chen S., (1992) “Properties of Neural Network with Applications to Modeling Non-linear Dynamical System”, Int. J. Control, 55(1):193-224.

[8] Pinto Armijo, Erika (2004), “A Matlab based application for developing grey-box models”, Memoria de Título de Ingeniería Civil Informática, Universidad de Santiago de Chile (in Spanish).

[9] Hernández E.; Arkun, Y. (1992). “Study of the Control-Relevant Properties of Backpropagation Neural Network Models of Nonlinear Dynamical Systems”, Computers & Chemical Engineering, 16(4):227-240.

Gonzalo Acuña, Erika Pinto
Universidad de Santiago de Chile
Departamento de Ingeniería Informática
Avda. Ecuador No 3659 - Casilla 10233; Santiago, Chile
E-mail: gacuna@usach.cl

International Journal of Computers, Communications & Control
Vol. I (2006), No. 2, pp. 15-22

An Automatic Grading System for Panels Surfaces Using Artificial Vision

Cristhian Aguilera, Mario Ramos, Gabriel Roa

Abstract: This work describes an automatic grading system using artificial vision to improve the quality of wood panels surfaces. The objective is to control stains on the surface. Artificial Vision techniques like Thresholding and Transformed Watershed methods are applied. Defects quantitative measures found on the surface are also presented, in particular quantity, area, intensity and distribution.
Keywords: Vision, Image Processing, Quality Control, Plywood

1 Introduction

The need of manufacturing companies to maintain high quality product requires an exhaustive production control during and at the end of the process. When control is human visual inspection product control is not completely reliable and it is not guaranty of a total quality control.

However, the development of new technologies, and especially image analysis systems (Artificial Vision), has ostensibly improved the quality control process. Since it is a noninvasive technology and is capable to inspect the 100% of the production, Artificial Vision offers an advantage when compared with the manufactured product quality control techniques like human inspection and Sampling.

The wood industry and specifically in the panels industry has produced efforts to improve each, every one of the production stages. Two main characteristics determine the panel quality: mechanical resistance and appearance. Artificial Vision (Garrido, 2003) is a solution for the appearance control quality. Panels like plywood are used in the carpentry industry. Decorative paneling is the reason why surface quality must be controlled. Since the objective of these products is both functional and esthetic.

The main objective of this paper is to propose an analytical Artificial Vision based method to classify panels according to the Quantity, Area, Intensity, and Stains Distribution on the surface. It includes among these characteristics quantitative methods to measure the stain distribution on the panels. Factors to permit the quantification of these parameters and especially of the stain distribution in this paper are proposed.

2 Panels

2.1 Definition

Panels can be defined (Corma, 2003).

Structural Panels this term refers to those panels employed as structural elements in the construction and packing industry, included plywood.

Non-structural panels this term refers to those panels employed in the furniture industry, included decorative plywood, hard panels, and Medium Density Fiberboard, MDF.

2.2 Classification

Traditionally, wood, and consequently wood-derived products, has been classified considering two fundamental criteria: resistance and appearance.

In relation to the resistance, the objective is to control the mechanical resistance degree. There are standards that these products should satisfy. Tests are related with the mechanical quality evaluation like thickness, density, weight, flexion, traction, elasticity, and humidity.

In the case of appearance, the objective is the visual appearance control, considering esthetic more than structural values. These criteria are important in the furniture and paneling industry. Defects affecting the surface quality, especially in MDF and fiber panels, are the stains on the surface.

As an example, a summary of the characteristics considered by a Chilean company to grade surface quality of MDF and fiber panels are presented in Table 1. It can be observed that panel grading is divided in three types.

- First quality (I).
- Second quality (II).
- Third Quality (III).

3 Industrial vision system

Industrial Vision Systems are rapidly becoming a key factor in the development of total quality procedures in an industrial automation processes context. Industrial Vision Systems permit to inspect production processes without fatigue or distraction, facilitating the quantification of quality variables and contributing to a continuous improvement (Silvén and Matti, 2003). The figure 1 presents the outline of the stages of an Industrial Vision System.

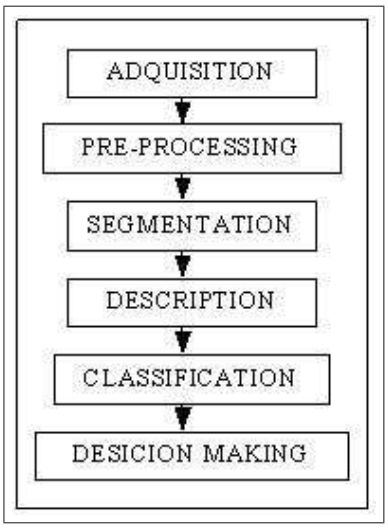


Figure 1: Industrial Vision System stages

3.1 Image Acquisitions and Digitalization

This is the stage in which sensors and the capacity to digitalize the signal produced by the sensor are used. The sensor is either color or monochrome camera that produces a complete image. After capturing the image, this information is sent to the computer to be analyzed (Mery, 2003).

3.2 Pre-processing

In this process, the acquired image is modified in order to improve it according to the parameters to analyze considering:

- Noise elimination.

	MDF	FIBER
QUAL	Surface	Surface
I	Both faces free of porosity, cracks, semi-inflated and dark stains	Both faces free of porosity, cracks, un-sanded holes, semi-inflated, and stains falling from the surface
I	Minor stains are permitted when there are no more than 2 per face and they are smaller than 2 cm	Longer strips that do not surpass 0.5 cm^2 in area are permitted. The quantity should be less than 10 per face and presenting a separation greater than 5 cms. between them
I	Good behavior during milling and coating application.	Minor stains are permitted if they are not greater than 5 cm^2 of area. The quantity should be less than 5 per face and with a separation greater than 5 cm
II	Cracks, thick particles, dust, adhesive, or other agent stains, and semi-inflated	Porosity, cracks, un-sanded holes, thick particles on the surface, adhesive, dust, or other stains
III	Without quality requirements	Without quality requirements

Table 1: Characteristics for panel grading according to the surface defects

- Accentuate or profile image characteristics (borders, limits, etc).
- Improve the quality of some parts of the image.

3.3 Segmentation

The segmentation process divides the digital image in unconnected regions in order to separate region of interest from the rest of the scene. In the last years, diverse segmentation techniques have been developed, which can be grouped in three techniques: pixel-oriented, border-oriented, and region-oriented techniques.

3.4 Description

This process labels the objects considering information supplied by inspection that can be:

Quantitative: Measuring of areas, lengths, perimeters, etc.

Qualitative: Verification of correct task performance (assembling, bottling, labeling, etc.).

3.5 Classification

Classification orders the segmented regions in classes, assigning to each region a group of many pre-established groups that represent all the possible types of regions that are expected to exist in the image. In this stage, a statistical study is performed on the characteristics that are extracted from the objects whose defects are known a priori.

3.6 Decision Making

Frequently, computer vision systems control a mechanical apparatus that manipulate the products after classification.

4 Proposed techniques

To grade the panel surface defects, image analysis techniques need to be used. This stage differentiates one vision system from another. The analysis is performed on images using a gray scale where the intensity values for black are 0 and 255 for white. One of the techniques to be used is thresholding (Guindos, 2001). Panel surface stains are considered regions. Another segmentation technique is the one based in growth of the regions applying functions under morphological operators, specifically the Transformed Watershed.

4.1 Thresholding

The thresholding technique (Gonzales and Goods, 1996) classifies each pixel in two groups, depending on whether or not the gray level exceeds the given threshold. If it does not exceed, then a lower gray level (color black) is assigned; in the opposite, the upper level (white color) is assigned. The objective is to obtain an image segmentation, creating black stains on a white background in a form that is similar to the objects seen from an upper plane, as can be observed in Figure 2.

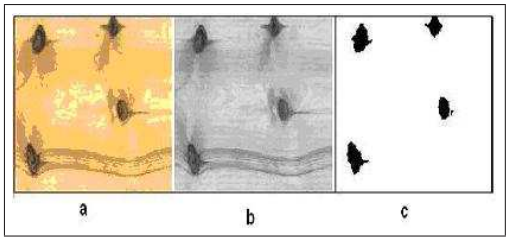


Figure 2: (a) original image, (b) I. Gray-scaled image (c) Thresholded Image

4.2 Transformed Watershed

This technique is part of a series of algorithms emerging to analyze images as a topographic surface where height is equal to the intensity. The highest pixel values correspond to the highest land areas and vice versa.

The Transformed Watershed simulates, based in the minimum values of the image of water filling the image, figure 3. Also, these points can be randomly selected, called Markers. The point where the water comes from two regions is called the Watershed and is the point that separates the basins corresponding to two minimums.

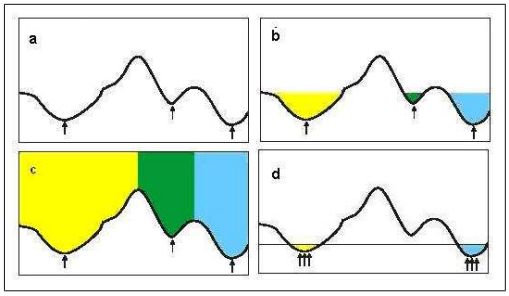


Figure 3: (a) Marker selection, (b) Flooding process from the markers, (c) Finalized Segmentation, (d) Selection of markers using thresholding

The key to segment using the Transformed Watershed is the markers selection. If each local minimum is taken, it could produce an over-segmentation. In order to reduce over-segmentation, the original image

is frequently softened before performing segmentation.

With the thresholding method, a set of markers can be obtained in which a pre-determined filling level is fixed. Figure 4 is an example of the over-segmentation when the Transformed Watershed is applied to the original image without establishing local minimums.

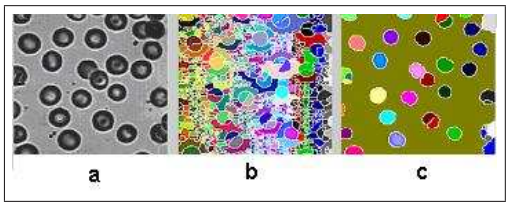


Figure 4: Watershed Results: (a) original image, (b) over-segmented image, (c) segmented image using local minimums

5 Extraction of panel characteristics

When the stains (defects) are found numeric properties are extracted. These characteristics provide information to grade the panel. The characteristic are Quantity, Area, Intensity and Distribution.

5.1 Quantity

This characteristic indicates the number of stains on the panel surface and whose factor is defined by the quantity of stains.

$$FC = C_T \tag{1}$$

where: FC is the quantity factor, C_T is the number of stains present on the panels surface.

5.2 Area

This characteristic indicates the stained surface of the panel. Its value is defined by the ratio between the sum of stain areas on the panel surface and the total panel surface area. This characteristic is called *"Area Factor"* and this is represented by:

$$FA = \frac{1}{AT}\sum_{1}^{k} A_n \tag{2}$$

where: FA is the area factor, AT is the panel area, An is the stain area and k is the number of stains (C_T).

In figure 5, it can be observed that the Area Factor increases proportionally with the stained area of the panel surface, indicating in this way a lower quality.

5.3 Intensity

This characteristic indicates the intensity level (gray scale, 0-255) of the stains. This property provides information of the stain type on the panel surface. For example, the humidity stain will have a different intensity than an oil stain. To represent this factor, the stain intensity average value and the stain-free panel intensity average are considered.

This characteristic will be called "Intensity Factor" and is represented by:

$$FI = \frac{XI - XI_m}{XI} \tag{3}$$

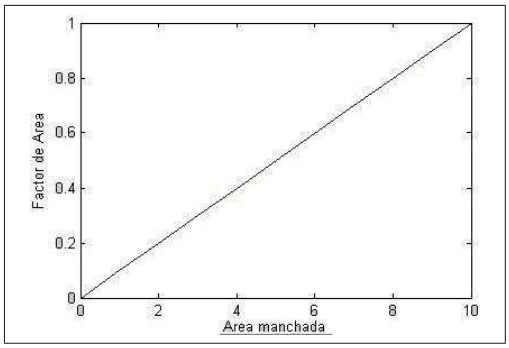


Figure 5: Area Factor v/s Stained Area

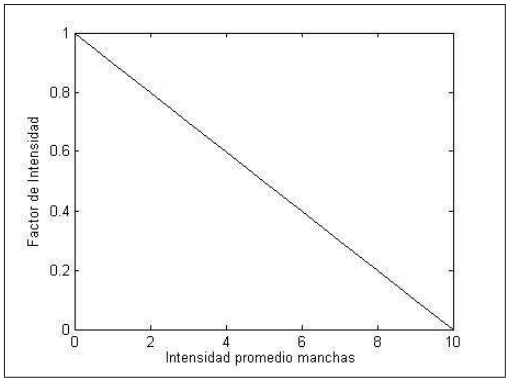


Figure 6: Intensity Factor v/s Average Stain Intensity

where: XI is the stain-free panel intensity average and XI_m is the stain intensity average.

Consequently, like can be appreciated in figure 6, the Intensity Factor will increase when stain intensity average value decrease, indicating in this a lower quality.

5.4 Distribution

This characteristic determines the spatial location of the stains on the surface. This shows if the stains are grouped in only one panel sector or rather are distributed throughout its surface.

The position of the objects is established through Cartesian coordinates. The method uses the average of the Euclidean distances to the objects geometric centers. The factor is defined by the ratio between the distances average and a reference value. The reference value is the distance from the diagonal to the sample. Expressed in percentage:

$$FD(\%) = 100 * \left[\frac{\mu(D)}{DT} \right] \tag{4}$$

Where: $\mu(D)$ is the average of Euclidean distances, D is the set of objects, DT is the distance from the diagonal of the sample.

The average of distances $\mu(D)$ depend how distant these object are from the geometric center. The farther they are, the greater is the value of (D) and consequently, FD (%) also be greater, indicating that the object distribution is greater. In figure 7, three examples of distributions are showed.

As an example, table 2 presents numerical values of the factors proposed in this work, in relation to the grading criteria presented in table 1. The panel format was 1.5×2.3 meters.

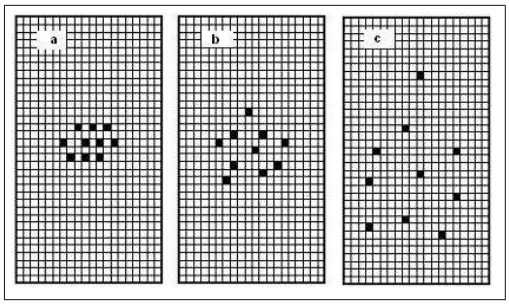


Figure 7: (a) FD=6.8%, (b) FD=9.05%, (c)FD=17.32%

	MDF			FIBER		
QUAL	FC	FA (%)	FD	FC	FA (%)	FD (%)
I	< 2	< 0,01	NE	< 5	< 0.07	1,8
II	- < 10	NE	-	< 10	7	
III	SEC	SEC	SEC	SEC	SEC	SEC

Table 2: Example of Factors (NE: Not Specified, SEC: Without Quality Requirements.)

6 Comments and conclusions

This work is focused towards Image Analysis by segmentation of stains present on wood panel surfaces and in the definition of factors to quantify defects properties (Quantity, Area, Intensity and Distribution).

Techniques like thresholding and Transformed Watershed showed goods results to segment stains in gray scales. Thresholding is a useful and simple technique, but when it is used only at one threshold level it is not capable of separating (labeling) the distinct stain intensity values that are below this value.

The Transformed Watershed method offers the capacity of separating regions (stains) in function of the intensity value presented by each defect. However, this technique presents over-segmentation, problem that is solved through the image pre-processing to increase the contrast for the objects of interest and the assignment of local minimums (Markers).

Factors proposed in this work provide the necessary information to define a Stain Quality Index to on -line control of based-wood panels.

Acknowledgements

This work was supported by the Project INCON-032008 3/R of the Research Secretary, University of Bio-Bio.

References

[1] Garrido R., *Adquisición y Tratamiento Digital de Señales para Visión Artificial*, Reporte Final Ingeniería Electrónica, Universidad del Bío-Bio, 2003.

[2] Corma, *Corporación Chilena de la Madera*, www.corma.cl, 2004.

[3] Mery D., *Inspección Visual Automática*, Informe Interno, Departamento de Ingeniería en Computación, Universidad de Santiago de Chile, 2003.

[4] Guindos F., Fernandez J., and Peralta M., *Visión Artificial con IMtdi*, Almería, 2001.

[5] González R.,Woods R., *Tratamiento Digital de Imágenes*, Addison-Wesley, 1996.

[6] Olli Silvén and Matti Niskanen, *Framework For Industrial Visual Surface Inspections*, 6th International Conference on Quality Control by Artificial Vision, Machine Vision Group, University of Oulu, Finland, 2003.

Cristhian Aguilera, Mario Ramos, Gabriel Roa
Universidad del Bío-Bío, Concepción-Chile
Departamento de Ingeniería Eléctrica y Electrónica
Departamento de Ingeniería en Maderas
Laboratorio de Sistemas Automatizados
E-mail: cristhia@ubiobio.cl, mramos@pegasus.dci.ubiobio.cl

International Journal of Computers, Communications & Control
Vol. I (2006), No. 2, pp. 23-33

Medium Term Electric Load Forecasting Using TLFN Neural Networks

Danilo Bassi, Oscar Olivares

Abstract: This paper develops medium term electric load forecasting using neural networks, based on historical series of electric load, economic and demographic variables. The neural network chosen for this work is the Time Lagged Feedforward Network (TLFN), which combines conventional network topology (multilayer perceptron) with good handling of time dependencies by means of gamma memory. This is a versatile mechanism that generalizes the short term structures of memory, based on delays and recurrences. This scheme allows smaller adjustments without requiring changes in the general network structure. The neural model gave satisfactory results exceeding those obtained by classical statistical models like multiple linear regression.
Keywords: Neural network model, forecasting, gamma memory, electric load

1 Introduction

Electric load is one of the key variables for electric power companies, since it determines its main source of income, particularly in the case of distributors. According to the foreseen load, the company makes investments and decisions on buying energy from the generating companies, and planning for maintenance and expansion. It is therefore absolutely necessary to have some knowledge of future power load: electric power distributors require a tool that allows them to predict the load in order to support its management and make more efficient its planning formulation. Accurate prediction of electric load is difficult, because it is determined largely by variables that involve "uncertainty" and whose relation with the final load is not deduced directly. The load is also characterized as a nonlinear and nonstationary process that can undergo rapid changes due to weather, seasonal and macroeconomic variations. A large number of the classical prediction models are inappropriate for this modelling because of their requirement of linearity and seasonality. On the other hand, an important contribution of Artificial Neuronal Networks (ANN) is their elegant capacity for approaching arbitrary nonlinear functions, which makes them adequate for solving this kind of problem. This paper proposes a solution to the problem of predicting monthly electric loads (medium term) of low voltage and low installed power (BT1) customers of distributing companies in Chile, based on both historical load values and on the country's economic and demographic growth data. To solve this forecasting problem various neuronal network models were analyzed and applied to a real case with information from four electric distribution companies in the north of Chile. Different methodologies for forecasting time series in various areas were considered, particularly financial.

2 Background of the forecasting problem

2.1 Theoretical aspects of the forecast

In general, one problem of forecasting time series consists in predicting future values for a time series based on past information. Formally, the forecasting can be defined as the search for or synthesis of a function F, such that (Moro 2002)

$$\vec{x}(t+1) = \vec{F}(\vec{x}(t), \vec{x}(t-1), \ldots, \pi_1, \ldots, \pi_k) \tag{1}$$

where, $\pi_1, \ldots, \pi_k$ are a set of independent parameters of the time variable. Other exogenous time-dependent variables can also be included in equation 1. In practice, the forecasting becomes a problem

of approximating a given function as precisely as possible (Bengio et. al. 1995). To be able to quantify the performance there are various metrics for the forecasting error (Khotanzad & Abaye 1997).

Traditionally, conventional (parametric) statistical models have been used to carry out the forecasting tasks, although Artificial Neuronal Networks (ANN) can be far more adequate for problems having nonlinearities, in addition to their capacity for extrapolating and approximating multivariable data and generalizing from examples (Costa & Ribeiro 1999).

2.2 Application of neuronal networks to forecasting problems

Since the forecasting problem consists in predicting the future value of one or a set of variable based on past patterns, interest must be focused on temporal processing with ANNs. Basically, the ANNs carry out input/output mapping, and to that end they have the ability of representing nonlinear functions arbitrarily (Martín del Brío & Sanz 2001). To carry out the temporal handling of the variables there are two main schemes, one that uses the external handling of the temporal part (delay line) plus a static conventional network (multilayer perceptron (MLP)), and another one that makes use of some properly dynamic network (with internal handling of the temporal variable). Diverse forecasting works are known using MLPs, but they have been limited in temporal processing because they occupy as many input neurons as present and past samples are needed, which is inefficient in terms of network design because of the large number of parameters. On the other hand, dynamic networks, whose output also depends on previous inputs or the system's history, are based on storing a state of the system, either in local internal memories (neuronal) or in the same network recurrence (with feedback connections). Recurrent systems are based on their connection topology which incorporates delays. There are spatially recurrent networks, with connections between neurons, and locally recurrent networks, with internal connections at one neuron. There are various elemental memory mechanisms for temporal neuronal processing (Principe et. al. 2000). The simplest is the "Delay Line PE", which implements in memory a window of length D samples, including the current one. After the limit of samples that the time window can hold (D), the stored values are lost. Other memory mechanisms alter the value of what is stored, but are much more flexible in the temporal aspect, because they can change their effect, modifying parameters without having to alter the topology. For example, there is the "Context PE," which has a feedback connection whose value weights the previous y(n-1) output and produces an exponential memory. Also, there are more advanced memory mechanisms such as the "Gamma Memory PE" (Principe et. al. 2000), which combine the basic memory mechanisms (delays) with an exponential filter type feedback.

Normally, the local dynamic capacities (neuronal) are integrated within a network with more classical topologies, such as the MLP. This is the case of the "time lagged feedforward networks" (TLFN), which are locally recurrent dynamic ANNs that integrate linear filter structures within a feedforward ANN to extend the network's nonlinear capacity with the representation of time. This paper details solutions using TLFN topologies, which are easier to train than spatially recurrent networks and handle the temporal dependence flexibly.

2.3 Forecasting electric load

In the case of the forecasting of electric load, some typologies must be considered, depending on the time interval involved (Gavrilas 2002): long term (intervals of years, systems development planning); medium term (months to one year, maintenance and expansion planning); short term (one day to one week; for a balance between energy load and generation as an efficient way of controlling the system's operation and the electric market); and very short term (for example, ten minutes, for a balance between load and generation during shorter time intervals).

2.4 Methods of analysis of input variables

Carrying out an analysis of input variables consists in studying the contribution of each variable to the result of the forecasting model. This is always useful in practice because it makes it possible to eliminate inputs with little information to describe the output or redundancy between the variables (Bishop, 1995). The steps to be followed are:

- Determine all the candidate variables: based on what an expert in the area of the problem to be solved indicates or on what is stated in the literature.
- Determine linear correlations and statistical analysis: relations between the candidate variables and the output variable, and between themselves, and the behavior of the variable to be forecast is visualized. If there is no linear correlation between two variables, it does not mean that that there are no nonlinear correlations between them (Spiegel, 1997).

Following that, what is mentioned by (Refenes et. al. 1997) can be used, which consists in:

- Building a model that captures the influence of time on the variable to be forecast and the most significant linear dependencies on exogenous variables: it implies identifying the temporal structure using univariate analysis of time series (Hagan & Behr 1997) (García & García 2002) and identifying exogenous influences using multiple linear regression.
- Capturing residual nonlinear dependencies: it is verified whether added value is produced by including additional variables which, even though they may have been rejected by previous linearity criteria, have nonlinear influence. It uses iterative search strategies (forward, backward) and selection criteria, using ANN as a tool.

2.5 Conceptual aspects of preprocessing

Although in theory all ANNs have arbitrary mapping capacities between sets of variables, it is convenient to normalize the data before carrying out the training, to compensate for the inevitable scaling and variability differences between the variables. The above can have a significant impact on the system's performance, so it has to be considered (Bishop 1995).

3 Neuronal modelling of the problem

3.1 Selection of candidate variables

The a priori knowledge of the models determined the following candidate variables: historical electric loads, meteorological (temperature), macroeconomic (GDP, CPI, HSI (hourly salary index)) (Torche 1998) (Gavrilas 2002), demographic (population, housing) (Instituto Nacional de Estadística (INE) 2002b), and referential (month of the year, kind of month) (Karady 2001).

3.2 Selection of input variables

The first phase of the selection method was applied to the candidate variables of the four companies by means of a linear model. An initial analysis was carried out (linear correlation and basic statistics to inspect the series). To determine the past load samples having the greatest influence on the present sample, autocorrelation and partial autocorrelation were used Based on the model obtained, exogenous variables were added (with a forward search strategy). Multiple linear regression was used, measuring the performance with the corrected determination coefficient, ($\overline{R}^2$) (Pérez 2001), which the bigger it is, the

better the quality of the regression. The exogenous variable that yields the largest $\overline{R}^2$ in the corresponding iteration is chosen, provided the improvement achieved is equal to or better than 5% (improvement and value added criterion). The linear models obtained for each company are (with linear function g):

- Company A:
$$C(t) = g(C(t-1), C(t-2), C(t-12), HSI(t)) \tag{2}$$
- Company B:
$$C(t) = g(C(t-1), C(t-2), Popul(t)) \tag{3}$$
- Company C:
$$C(t) = g(C(t-1), C(t-13), Popul(t), Temp(t) \tag{4}$$
- Company C:
$$C(t) = g(C(t-1), C(t-2), C(t-10), Hous(t)) \tag{5}$$

Where C (Load) corresponds to the electric load, and the other variables were defined in section 3.1. As can be seen, for the candidate variables to generate a single "universal and portable" model for the four companies that are studied, they have variable incidence in each case. A general model is developed that has as inputs the past loads and the exogenous variables having the largest linear impact.

- The "CPI" variable was excluded because it does not appear in the linear models. In spite of not appearing in the linear models, the "GDP" variable was not excluded because of a priori knowledge. The "Temperature" variable was excluded because even though it appears in one of the linear models, in general it has a low linear correlation with the load.
- For the input values, gamma memory was used in the input layer to reflect delay.
- In view of the delay with which BT1 customers react to changes in the macroeconomic and demographic variables, the Initial Neuronal Model assumes that the present load has at least one month of delay with respect to the exogenous variables.

3.3 Definition of training and test sets

The following sets of data are defined:

- Learning Set. Inputs: historical load data and exogenous variables (January 1999 through November 2001). Outputs: historical load data from February 1999 through December 2001.
- Cross-Validation Set. Inputs: historical load data and exogenous variables (December 2001 through May 2002). Outputs: historical load data from January 2002 through June 2002.
- Test Set. Inputs: historical load data and exogenous variables (June 2002 through November 2002). Outputs: historical load data from July 2002 through December 2002.

3.4 Data preprocessing

This preprocessing consisted in depurating the available information both for training and for testing the ANN. It was applied to the defined candidate variables (section 3.1).

3.5 Construction of neuronal network for electric load forecasting by low voltage clients

Based on the selection of variables made, the network was modeled with short term memory mechanisms. A training and test were made separately for each company, but applying a general model to the four companies. It consisted in a focalized TLFN (FTLFN) with gamma memory mechanisms in the input layer, in view of its versatility. Since gamma is a unifying system of the two basic memory mechanisms (delay line and context PE), the use of their respective potential is not discarded a priori, since both can be seen as special cases of gamma memories. A hiden layer was used. It will be trained with Backpropagation (static BP with optimization method of descent along the gradient) (Freeman & Skapura 1993) (Martín del Brío & Sanz 2001), since it is a locally recurrent and focalized topology, and also because the μ values (for the gamma memory) will be fixed in an arbitrary but convenient manner, i.e. they will not be adapted, using the criterion of (Principe et al. 2000). The training parametrs are: a momentum of value 0.7 and a "path size" of value 0.1 for the weights that connect the input layer with the hidden layer, and 0.01 for the weights that connect the hidden layer with the output layer (chosen on the basis of the initial tests). In the input layer there will be as many input neurons as there are input variables. The hidden layer is nonlinear, with a "hyperbolic tangent" activation function. The number of hidden neurons will be adjusted based on successive tests to optimize the model. The output layer was chosen to be linear. This layer has one neuron (since there is only one output, the load at instant t) and it will be connected to the hidden layer. The output (still normalized) will be postprocessed. Cross-validation was used to test the quality of learning and avoid overtraining. In this case the training is stopped if the mean square error (MSE) in the cross-validation set does not improve after 100 time-periods (Martín del Brío & Sanz 2001). If the condition is not fulfilled, every component will be submitted to the ANN a maximum of 1000 time-periods. The cross-validation was applied every five training time-periods to measure the corresponding error with a certain stability and avoid sudden oscillation biases. Since the initial weights are random, the results of the training can differ when carried out more than once. For that reason, the network will be trained five times, saving the best weight combination (minimum MSE). To assess the model's viability, a final evaluation is made with the test set. The forecast horizon will be one month (forecasting the following month noniteratively, so as not to carry over probable forecasting errors that may distort the results). Thus, for the training set, a forecast will be made for thirty five months. For the test set the forecast will be made for six months, all this with the forecast horizon mentioned earlier. The forecast is always for the following month. For the training set with BP, the MSE will be calculated successively, defined as follows:

$$MSE = (\sum_{j=0}^{P}\sum_{i=0}^{N}(desired_{ij} - forecast_{ij})^2)/(N*P) \tag{6}$$

where P is the number of output neurons; N is the number of data set components; $forecast_{ij}$ is component i output in output neuron j; $desired_{ij}$ is desired component i output in output neuron j.

Together with a good approximation, it is desirable for the ANN to give as a result an adequate correlation (absolute value between 0.5 and 1.0), thus reflecting the similarity in the shapes of the desired and forecast output curves. In this way, MSE, MAPE, MAE and correlation (r) will be measured for the cross-validation and test sets (Khotanzad & Abaye 1997). The single Initial Neuronal Model proposed for the four companies is (with nonlinear h function (includes operations performed by gamma memory and ANN on inputs)):

$$C(t) = h(C(t-1), GDP(t-1), HSI(t-1), Popul(t-1), Hous(t-1)) \tag{7}$$

In the initial model, greater memory depth M (greater number of samples) will be given to past loads and to the exogenous variable of highest linear impact (linear correlation) on the load. For the four companies, this variable was the population. Based on this and on the tests made, the values of for the gamma memories of the corresponding input neurons were fixed as shown in Table 1. After testing

the performance of this model, adding the rest of the candidate variables to the network is evaluated, considering that nonlinear relations with the load could still be captured.

Variable	μ
Historical Load	0.8
HSI	1
GDP	1
Population	0.8
Housing	1

Table 1: Values of μ for the gamma memories of the Initial Neuronal Model

As definition of the model's requirements we have: cross-validation MSE and test less than 0.1; training MSE less than that of cross-validation; for the cross-validation and test the correlation must be greater than 0.5; for cross-validation and test MAPE must be less than 5%. The number of hidden layer neurons was adjusted making successive tests, getting an optimum number of five neurons.

The initial neuronal model is illustrated in figure 1. When three output taps were used in the gamma memories, for the "resolution vs. memory depth" compromise, the optimum values shown in table 2 were obtained.

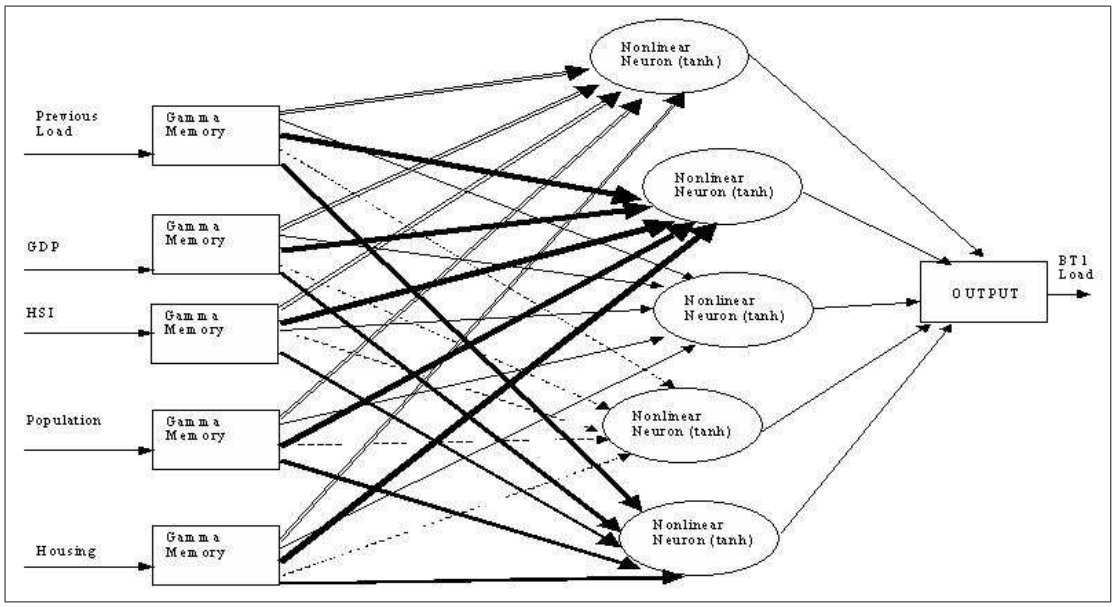


Figure 1: Initial Neuronal Model

The results were satisfactory. Also, the model (single) was portable in topology and μ values (fixed) for the four companies. The only difference is that of the final synaptic weights.

To depurate the initial neuronal model and get residual nonlinear relations of load with other candidate input variables not included so far, the following alternatives were tested: initial model adding "CPI;" initial model adding "Temperature;" initial model adding referential variable "Month of the Year;" initial model adding referential variable "Kind of Month;" and alteration of initial model using as basis the current samples of the exogenous variables. In each case, different combinations of μ were tried, but the results showed no improvement. This depuration attempt was made, in the first place, with a forward search strategy (see section 2.4). No improvement was obtained when either the backward strategy (Refenes et. al. 1997) was used or when variables were removed from the initial model. Therefore, the Final Neuronal Model is the same as the Initial Neuronal Model of equation 7.

Variable	Resolution (R =μ)	Taps (D)	Depth ($M = D/R$)
Historical Load	0.8	3	3.75
HSI	1	3	3
GDP	1	3	3
Population	0.8	3	3.75
Housing	1	3	3

Table 2: Resolution vs. memory depth compromise Initial Neuronal Model

3.6 Multiple linear regression model for the case studied

This model makes an estimation of the dependent variable Y, given the values for a set of independent variables x_i. The general model is given by (Pérez 2001) (Spiegel 1969):

$$Y = b_0 + b_1 * x_1 + b_2 * x_2 + \ldots + b_k * x_k + u \tag{8}$$

where the coefficients b_i represent the effect of the explanatory variables x_k on the independent variable; b_0 is a constant (or independent) term of the model, and u is the residue or error term of the model. When T observations are available in time, equation 9 can be written as follows:

$$Y = b_0 + b_1 * x_{1t} + b_2 * x_{2t} + \ldots + b_k * x_{kt} + u_t \tag{9}$$

where, $t = 1, 2, 3, \ldots, T$.

A multiple linear regression model was constructed using as input the same variables of equation 7. That is, for this case the regression model is defined formally with a dependent variable: $load(t)$; independent variables: $load(t-1)$, $GDP(t-1)$, $HSI(t-1)$, $population(t-1)$, $housing(t-1)$. When the regression is carried out, the coefficients of table 3 are obtained.

Variable	Company A	Company B	Company C	Company D
Constant b	-105058393.5	-27666403.6	-29892228	-28500210.8
Hist. Load	-9.02E-02	-0.119	7.65E-02	-0.19651476
GDP	0.803	-0.18	2.036	-1.29619841
HSI	-97417.432	207055.599	-97524.597	83108.4594
Population	1068.58	0	0	0
Housing	-1718.429	211.062	447.505	455.258917

Table 3: Regression Coefficients by Company

4 Comparative analysis with linear regression model

The results obtained with the neuronal model are compared below with those of the multiple linear regression model already defined. The mean absolute percent error (MAPE) of the sets representative of the ANN model and the regression model were compared. In this way, for each company:

- The MAPE of the regression model was calculated based on all the existing samples so as not to bias (avoid harming or favoring the results of this model) the results in particular samples.
- The MAPE of the neuronal model was calculated for the samples of the cross-validation and for the samples of the test set.

- The MAPE of the neuronal model does not consider the training set, because the network, due to the learning reinforcement itself and the constant error minimization, will produce an almost exact forecast, introducing considerable bias in the results and distorting the comparison.

Table 4 shows a comparison of the results by company:

	Neuronal Cross-Validation	Neuronal Test	Regression
Company A	2.71	2.54	2.77
Company B	1.74	3.03	3.17
Company C	6.95	3.08	3.69
Company D	3.18	5.38	2.71

Table 4: Comparison of Results (MAPE) for the Companies Studied

The previous comparison and its results show that the neuronal model developed was in general superior in accuracy to the regression model, except in specific cases in which the ANN had slight forecasting problems (Company C cross-validation and Company D test). The cross-validation of Company D shows a larger error, but with a slight difference compared to the regression. In the remaining cases the ANN approximation was better than that of the regression model. It may be concluded that the ANN, by using nonlinear relations that affect the electric load, can decrease the forecasting error, giving a better approximation of the variable that is being studied than the statistical model used.

5 Conclusions

This paper and its results show that the ANN represent a powerful tool for decision making in electric distribution companies. This is based on a methodological selection of variables, a prior study of the problem to be solved, and the processing that the ANN can make in the temporal aspect. In this respect, it was decided to construct the neuronal model based on a typology that potentiates the nonlinear static mapping capacity of conventional neuronal topologies (such as the multilayer perceptron), with the handling of time series by means of the "Time Lagged Feedforward Network" (TLFN). To implement it, a versatile memory mechanism (gamma memory) was used, which allows minor adjustments to be made without making changes in the network's general structure, and which generalizes the basic short term memory structures. As future development, forecasting the load of high voltage (AT) customers can be attempted. This is more sensitive to price changes and is more exposed to unexpected functional changes (for example, industry breakdowns) and changes in the macroeconomic variables. The results obtained fulfilled the main and specific objectives of the work, which implied having a reliable ANN model for predicting the monthly electric load of BT1 customers. Also, the ANN model had better accuracy than the classical statistical model used. All this means that the model constructed makes it possible to determine the medium term load with acceptable accuracy, as required by the electric distribution companies for the system's planning and expansion. The gamma networks are more versatile for modelling nonstationary time series. Handling of the feedback parameter gives the developer an additional element for adjusting the network, disregarding the increase or decrease of the neurons of the hidden layers, which often is ambiguous. Support by linear statistical techniques and time series is an important tool for selecting variables, making it possible to detect linear dependencies prior to the construction of the ANN, leading to focusing the test/error of such network on finding nonlinear dependencies, with a large part of the linear modelling already solved. The electric customer reacts with delay or inertia to changes in the macroeconomic variables, and the load reacts with inertia to changes in the demographic variables. It was shown that the present electric load has a delay of at least one month in its "reaction" to exogenous variables. From the above it is seen that if short term memory mechanisms were used for the neuronal

modelling, the delays that must be considered should be handled carefully. Otherwise the training would not be useful.

References

[1] Bengio, S., Fesant, F., Collobert D., *A Connectionist System for Medium - Term Horizon Time Series Prediction*, International Workshop on Applications of Neural Networks to Telecommunications, IWANNT, Stockholm, Sweden, 1995.

[2] Bishop, C., *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[3] Costa, N., Ribeiro, B., *A neural prediction model for monitoring and fault diagnosis of a plastic injection moulding process*, Centro de Informática e Sistemas da Universidade de Coimbra, 1999.

[4] Freeman, J. and Skapura D., *Redes Neuronales: Algoritmos, Aplicaciones y Técnicas de Programación*, Versión en español de García-Bermejo, R. Joyanes, L. Addison Wesley / Díaz de Santos, Wilmington, Delaware, EUA, 1993.

[5] García, J. and García, F., *Econometría. Tema 5. Autocorrelación*, Curso 2002/2003. Universidad de Huelva, 2002.

[6] Gavrilas, M., *Neural Network based Forecasting for Electricity Markets*, Technical University of Iasi, Romania, 2002.

[7] Hagan, M. and Behr, S., "The Time Series approach to short term load forecasting", *IEEE Transactions on Power Systems*, Vol. PWRS-2, NO. 3, 1997.

[8] Instituto Nacional de Estadísticas (INE 2002b), *Censo 2002. Síntesis de Resultados*.

[9] Karady, G., *Short - Term Load Forecasting using Neural Networks and Fuzzy Logic*, Arizona State University. Power Zone, 2001.

[10] Khotanzad, A., Abaye A., "ANNSTLF - A Neural Network Based Electric Load Forecasting System". *IEEE Trans. on Neural Networks*, Vol. 8, No. 4, 1997.

[11] Martín del Brío, B., Sanz Molina A., *Redes Neuronales y Sistemas Borrosos*, 2da. Edición, Ra-Ma, Madrid, España, 2001.

[12] Moro, Q., *Series Temporales y Redes Neuronales Artificiales*, Departamento de Informática Universidad de Valladolid. Pérez, C. (2001), *Técnicas Estadísticas con SPSS*, Prentice Hall, Madrid, España, 2002.

[13] Principe J., Euliano N., Lefebvre W., *Neural and Adaptive Systems: Fundamentals Through Simulations*, John Wiley and Sons, New York, 2000.

[14] Refenes, A., Burgess, A., Bentz, Y., "Neural Networks in Financial Engineering: A Study in Methodology". *IEEE Transactions in Neural Networks*, Vol 8; No. 6, 1997.

[15] Spiegel, M., *Estadística*, Serie de Compendios Schaum. Libros McGraw-Hill, Colombia, 1969.

[16] Torche, A., *Contabilidad Nacional. NúmerosIndices. Desestacionalización y Trimestralización*, Trabajo Estadístico Número 63. Pontificia Universidad Católica de Chile. Insituto de Economía. Santiago, 1998.

Danilo Bassi and Oscar Olivares
Universidad de Santiago de Chile
Departamento de Ingeniería Informática
Santiago, CHILE
E-mail: dbassi@ieee.org, oolivare@yahoo.com

International Journal of Computers, Communications & Control
Vol. I (2006), No. 2, pp. 33-40

The Challenge of Designing Nervous and Endocrine Systems in Robots

Felisa M. Córdova, Lucio R. Cañete

Abstract: We discuss in conceptual terms the feasibility of designing a Nervous System and an Endocrine System in a robot and to reflect upon the bionic issues associated with such highly complex automatons. The emulation of biological phenomenon in artificial systems, both nervous and endocrine imitation in a mechatronic automaton is an attractive proposition as a mechanism of organic integration. The ability of living organisms to maintain their internal organization in spite of external changes, encourages attempts to imitate such achievements in devices. The complexity of the network sensors-integrators-effectors and the proportion of internal fluids must be increased when seeking the homeostasis of the robot from mechatronic design to bionic design.
Keywords: bionics, integration, robots.

1 Introduction

Biology has developed in a very accelerated way over the last three decades with the 21st Century, widely predicted to be the "Biology Age" (Ovchinnikov, 1987). In this context, a certain interest in emulating biological characteristics in appliances can be seen, for the purpose of recreating the successful behavior of living organisms in artificial systems.

Sensors such as hair and skin, effectors such as feet and fins and integrators such as artificial brains are some examples of bionic projects (Cañete, 2002). However, in order to come close to achieving the behavior exhibited by animals as inspiratory beings, then by necessity (Ofek, 2001), it must be recognized the pertinence of Nervous System and Endocrine System.

Of course, when the Nervous System and the Endocrine System work in unison in animals, they could generate synergic effects in the behavior of these alive beings. The use and integration of both systems in a human being can be seen for example when a person is attacked by a swarm of wasps. The electromagnetic receptors (the eyes), acoustics (the hearing) and those related to pain (the skin) receive information which is interpreted as threatening, so the association neurons and effectors not only order the locomotors effectors (the leg's skeletal muscles) to take flight but also order the glands to release adrenaline and other hormones that designate resources to the task of escaping.

No doubt, animal characteristics like these are interesting to emulate in automatons (Cañete and Córdova, 2003). So, the objects of this work are to conceptually design a Nervous System and Endocrine System in robots as a biological emulation, and to reflect upon the bionic design of those systems.

2 Why a nervous system and a endocrine system?

On examining any animal, one observes that the digestive, circulatory, respiratory and excretory systems have specific structures and functions, but none of them can carry out their functions independently of the others; rather they work together in harmony to meet the metabolic needs of the body. Any living being, not only an animal, is an organized entity that behaves as a unit and this unity is the result of the participation of a larger coordinated and integrated system. As an automaton, a robot must also possess such a system, and select animal inhalation as a perfect example of how it must fit together with its environments.

Vertebrates and other multicellular beings whose structures are the result of the evolution (Paccault, 1991), the source of the emulation, posses two coordination systems of varying form and complexity

that are closely related: Nervous and Endocrine. Both serve to make the body a unique active entity, individually regulating the functions of it parts and enabling it to adjust to changes.

In biology texts, robots designers are accustomed to seeing both systems compared with each other with the emphasis on the considerable difference that exist between them: in the Nervous System the message is an electrochemical disturbance (a nerve impulse) that travels through a nerve fiber, whereas in the Endocrine System the message is a chemical substance (a hormone) that travels in fluid, like blood.

If a robot had a sort of these system, it behavior in aggressive environments would be better. So, as it shown in Figure 1, the immediate task is to emulate the Nervous System and the Endocrine System from natural stage to artificial stage (animal to robot).

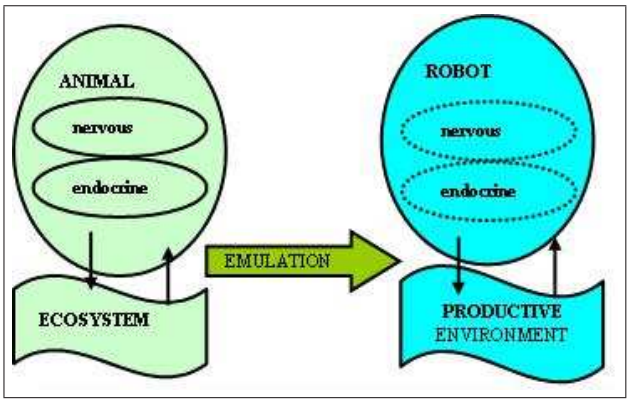


Figure 1: From a scene of origin towards another one of destiny. Considering the contribution of the Nervous System and Endocrine System to the successful performance of the animal facing its ecosystem, such systems conceptually may be copied in a robot to face its productive environment

3 The nervous system

3.1 The necessity of sensors

It is possible to face the design of basic a Nervous System that allows to make the sensomotor coordination to guide a Load-Haul-Dump (LHD) vehicle inside a street within a tunnel in an underground mine (Córdova, 1996). The vehicle loads mineral from a pit and it dump it in a ore-pass. Then, a set of sensors could be in charge to acquire the relevant data from the tunnel. Among them, is possible to use some of the following ones:

- Distance sensors, that can be an adjustment of ultrasonic sensors, or infrared sensors.
- Laser sensors to measure depth.
- Tact and of proximity sensors.
- Vision sensors, that uses TV cameras and processors of stereoscopic images, that can detect a painted line in the ceiling of the gallery or the profile of the tunnel.
- Angle sensors.
- Torque and acceleration sensors.
- Sensors of virtual surroundings, that can detect them holes of ditches and resentments.

- Sensors of the operational conditions of the vehicle, among them sensorial of humidity, temperature, pressure.
- Sensors of environmental conditions of the tunnel, such as fire detectors.
- Tags, that allows to detect figures of bar codes and it the location of the vehicle inside the gallery.

3.2 A sensomotor coordination design

In this particular design, an array of six located ultrasonic sensors in the flanks of the vehicle is used. The navigation criteria is of homeostatic type, that means that the vehicle maintains a equidistant distance to the walls of the tunnel. In this design, a neuronal network of sensomotor coordination can be used that coordinates the sensors and the actuators. The strategy of sensomotor coordination makes use of a propioceptive network presented in Figure 2.

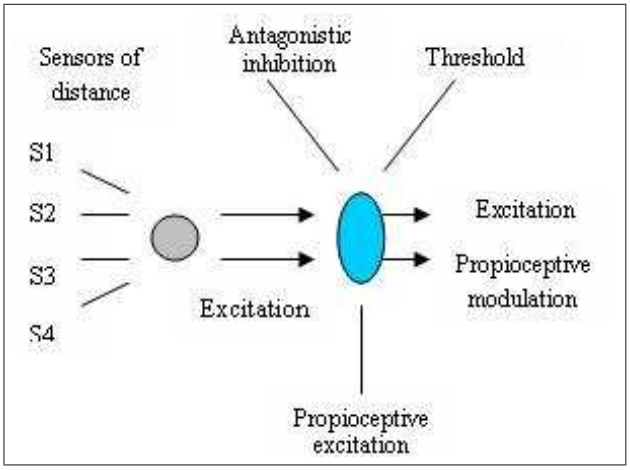


Figure 2: Propioceptive network

In the reflective network of senso-motor coordination of Figure 3, from the general and previous design of a fuzzy neuron, defined the characteristic of design for each neuron in individual, obtaining itself eight types of neurons. In the case of the sensorial neuron a not-linear excitation (sigmoidal) and with different weight for each sensor from the respective sub-network is required (the Agonist-Antagonist sub-networks are symmetrical in their construction).

Modulating Neuron **m** dynamically varies the characteristic of the Excitatory Neuron **e1** and the Threshold Neuron. This activity of modulation is function of the speed of the vehicle and the ratio of turn **Rg**. The rest of the neurons has excitatory and/or inhibitory activities according to its connectivity in the network.

The rank of activity of all the neurons varies between 0 and 1, minimum activity and maximun activity for all the variables of the neurons. The signals of activity are Excitation, Inhibition, Modulation and a Threshold of adjustment for each neuron.

- Excitatory Neuron e1: this neuron receives the activity of the distance sensors whose weight decreases and this is modulated as well by the activity of the Modulating Neuron depending on inverse on the turning ratio and the speed of vehicle v at low speeds and v2 at high speeds (over v2 march).
- Excitatory Neuron e2: this neuron actives the turn actuator, receiving a modulated activity of the previous activity of its sensors, inhibitions of the antagonistic network that are against to him and of its own actuator that restrains it and stabilizes as the networks arrive at an activity balance.

- Inhibitory Neuron i1: this neuron allows the incorporation of the activity from one sub-network to the other, and its action is strong in high ranks of internal activity to allow to a fast compensation forehead to strong stimulus of the network that originates it when the networks are very far from the balance.

- Inhibitory Neuron i2: this neuron restrains the activity of the turn actuator when it is in conditions near the balance, or allows a fast turn (depending on the Threshold Neuron) when a strong sensorial activity of distance exists and the vehicle goes fast.

- Modulating Neuron m: this neuron modulates the activity of sensors of distance in agreement with the speed and the turning ratio of the vehicle, in addition stimulates to the neuron threshold when its own activity is high along with the activity of the distance sensors.

- Threshold Neuron: When this neuron is stimulated by the Modulating Neuron, it inhibits to inhibitor of the actuator to correct in fast form the direction of the vehicle. This is required in the case that the vehicle has a small turning radius or when the speed is high.

- Sensor Neuron: The activity received by each sensor of the sub-network is heavy in decreasing form while more moved away it is of the advance direction, in addition the sensation one to the activity is not-linear for each sensor.

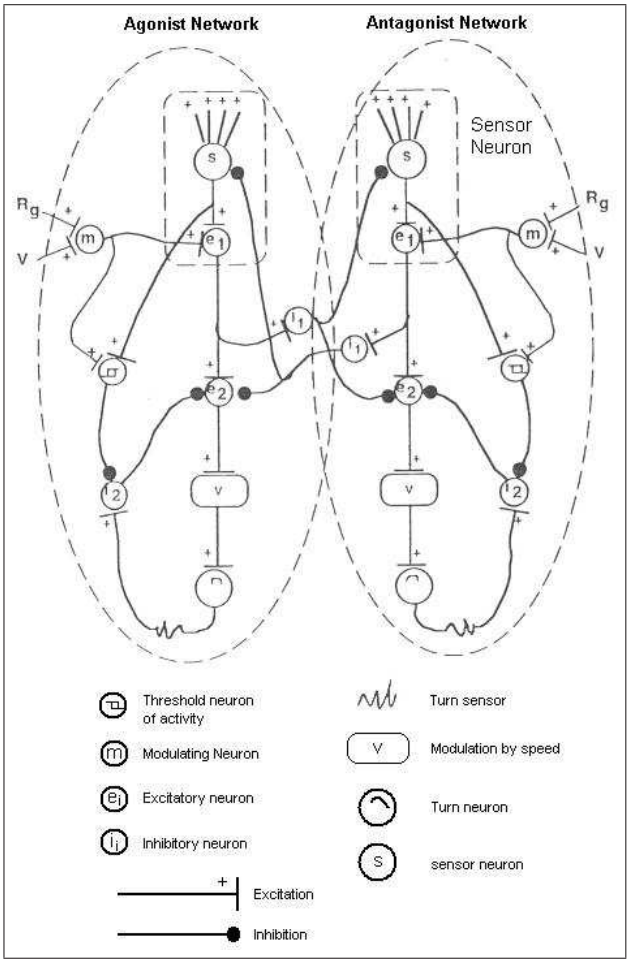


Figure 3: Sensomotor network

Structurally speaking, nerves can be emulated and in fact they have been, by way of electrical conductors specifically in mechatronic automaton like LHD.

4 The endocrine system

4.1 The necessity of an internal fluid

In the case of blood, it has not been emulated in robots and the absence of internal fluid in an automaton was one of the reasons why work was halted on the emulation of other animal features dependent on internal chemical transportation.

Of course, the existence of fluids in the body of the robot would not only facilitate the emulation of the Endocrine Systems' own hormones but also the emulation of antibodies for immunity, of bradicins (when animal tissue is broken, enzymes are released that convert certain plasmatic proteins into a substance called bradicin), for nociception (perception of damage) and of solvents for quimosensitivity (to stimulate the sense of taste, the chemical substances must be dissolved).

Furthermore, a fluid not only facilitates the movement of intra-corporal messengers but could also act as a lubricant, combustible and a thermo-regulator.

A classic mechatronic robot contains in percentage terms of volume approximately one sixth part of fluids which are principally lubricants, and substances for hydraulic devices and combustion. An arthropod on the other hand, contains at least three quarter parts of fluid (Cañete and Córdova, 2003).

Therefore, the first challenge in this respect consists of the proportion of internal fluids in the robot. Such an increase would lead to an increase in homeostasis which would bring with it an increase in the design complexity as shown in Figure 4.

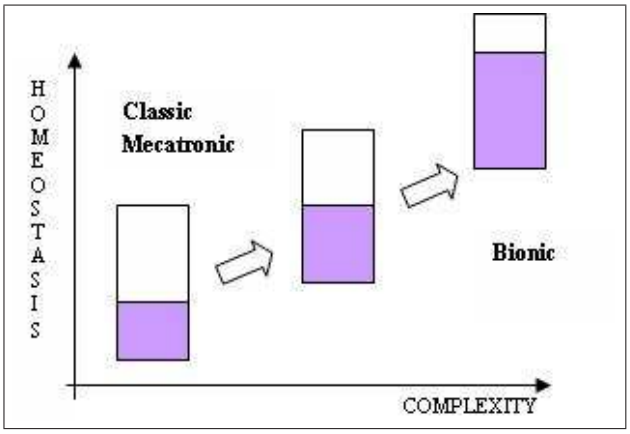


Figure 4: Complexity versus homeostasis. An increase in complexity for Endocrine System, comes with increasing the proportion of internal fluids when seeking the homeostasis of the robot from mechatronic design to bionic design

4.2 A simple design

When a robot works in changing environments, some parts of the surroundings without its control would become dangerous. In such situations, the robot will have to react in urgent form to fit its internal structures and to assure its viability. One of such urgencies can be the repair or reinforcing a part of the body that is being exposed to an external requesting. In order to face such situation it is possible to be resorted to a endocrine emulation as it is explained next and presented in Figure 5.

The internal part of the sensible zone of the body exposed to harmful emboss and ruptures are designed rough and it is covered with little smooth laminas.

The last ones are conductive elements with bad plastic behavior; of such form get fractures and can release them when happens a damage of the outside.

The internal part in addition is bathed by a fluid and slightly energized with electrical current.

An internal sensor to the circuit of the fluid exists somewhere in addition that is able to capture fragments of little laminas conductors by magnetic action. Whenever it captures some piece of laminas, releases a plastic and rough substance in form of grume (small semisolid mass) and it orders to a pump to increase the speed of the fluid circulation. The rough grumes slide by the smooth surface and they are crowded in the rough surface that is being dangerous. Finally the grumes reinforce the exposed zone.

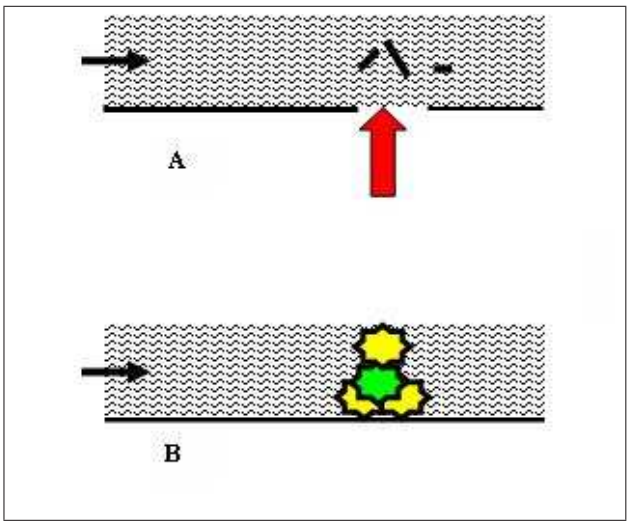


Figure 5: Cross section of the bodywork with phenomenon of endocrine emulation.
A: An external agent breaks or damages the interior where a fluid circulates, conductive fragments are freed in the open leaving one rough surface. The fragments when traveling by the fluid are captured by a magnetic receiver that releases rough grumes.
B: The rough grumes travel by the current and they lodge in the naked zone, cushioning the external action. The effect on the speed flow caused by the grumes is insignificant due to the extension planar of the conduit and to the precise location of the grumes.

Why not to design a hard bodywork and thus to avoid a complex Endocrine System? For the same reason for which the nature through million years of evolution has not done it (David and Samadi, 2000). Comparatively (Massé and Thibault, 2000), to maintain a hard bodywork is more expensive than to maintain a light bodywork with repair capacity.

5 Organic integration in the robot

Once the Endocrine and Nervous Systems have been conceived, both systems may be connected to each other and at the same time joined to the functions which they control. It is then that an organization is configured, which for the purpose of the robot being proposed in the current research, would have a internal communication structure as shown in Figure 6.

In animal kingdom, the levels are recursive and thus subordinates: that is to say the Supreme Level contains and governs the Homeostasis Level (Nervous and Endocrine) which in turn does the same with the Sensomotor Level.

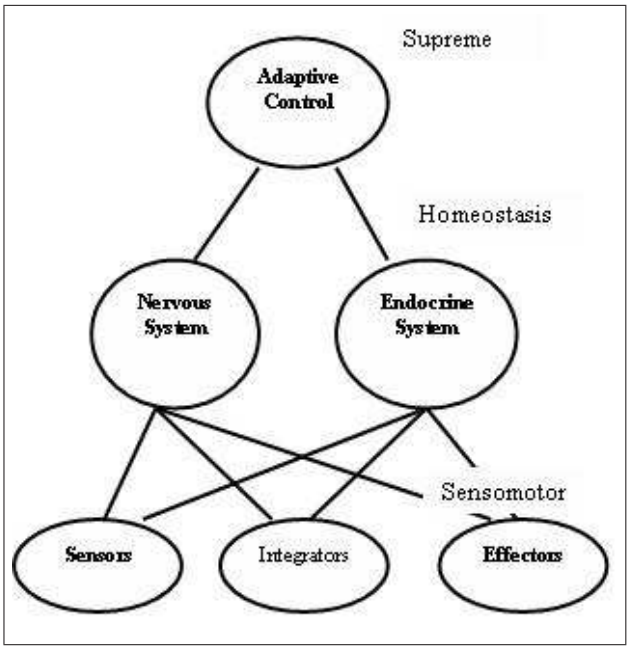


Figure 6: Three levels in organic integration of bionic robot. The nervous and endocrine reaction corresponds at a Homeostatic Level. Such reactions are subordinated to Supreme Level, but they govern at Sensomotor Level.

Adaptive Control, as the name implies, is the higher level of coordination and functional integration in a robot. Its structure can be a managed by a system that knows the mission the robot and links both information of the inner state of the automata and his surroundings. A proposed future challenge in the present work, is the conceptual design of this Adaptive Control.

6 Conclusions

Although the robot has neither nerves nor blood where it may send messages in charge of functionally integrating and coordinating all its parts, they can be emulated.

There is evidence that the emulation with certain restrictions of both Nervous and Endocrine System is feasible. However such an emulation would require the convergence of two sciences: Biology and Robotics.

The convergence of Biology with its different branches (Ecology, Genetics, Evolution among others) and Robotics is no accidental; rather it constitutes an essential issue in as far as abstract terms are concerned, that is: there is little to distinguish automatons and living beings. In reality both beings show behavior that is autopoietic (self produced), telenomic (self initiating) y homeostatic (self regulating with feedback function). Put more precisely: many of the features of a living being are desirable in an automaton.

Aspiring to have such features comes from the observations that man has carried out in the biological world of he himself is a part. The ability of living organisms to maintain their internal organization in spite of external changes, encourages attempts to imitate such achievements in devices, whose internal compositions are quite modest when compared with those of plants and animals.

This motivation, which has further increased on the verification of real successes achieved by man more than three decades ago (in sonar, prosthesis and artificial neuronal networks among others) provides a particular incentive to continue with the current research.

References

[1] Cañete, L., *Ecología cognitiva en robots terrenos para el desierto de Atacama*, Tesis Doctoral, Facultad de Ingeniería de la Universidad de Santiago de Chile, 2002.

[2] Córdova, F., *Guiado autónomo de equipos cargadores frontales LHD en una mina subterránea*, Informe Proyecto FONDEF. CONICYT. Chile, 1996.

[3] Cañete, L. and Córdova, F., Ecología cognitiva en robots terrenos, *Proceedings of the First IEEE Latin American Conference on Robotics and Automation LCRA*, Santiago de Chile, 2003.

[4] Cañete, L. and Córdova, F., A robot with immunity and perception of damage, *Proceedings of the First IEEE Latin American Conference on Robotics and Automation LCRA*, Santiago de Chile, 2003.

[5] David, P. and Samadi, S., *La théorie de l'évolution*, Paris: Champus Université Flammarion, 2000.

[6] Glávic, N. and Ferrada, C., *Biología. Tercer año de Educación Media*, Santiago: Ministerio de Educación, 1985.

[7] Ofek, H., *Second Nature: economics origins of human evolution*, Cambridge: University Press, 2001.

[8] Ovchinnikov, Y., *Basic Tendencies in Physico-Chemical Biology*, Moscow: MIR Publisher, 1987.

[9] Massé, G., and Thibault, F., *Intelligence économique*, Bruxelles: De Boeck Université, 2001.

[10] Paccault, I. (1991). La terre et la vie. Paris: Larousse.

Felisa M. Córdova, Lucio R. Cañete
Universidad de Santiago de Chile
Departamento de Ingeniería Industrial
Laboratorio de Concepción e Innovación de Productos, LACIP
Santiago de Chile
E-mail: fcordova@usach.cl

International Journal of Computers, Communications & Control
Vol. I (2006), No. 2, pp. 41-51

A New Technique For Texture Classification Using Markov Random Fields

Mauricio Gomez, Renato A. Salinas

Abstract: This paper proposes, applies and evaluates a new technique for texture classification in digital images. The work describes, as far as possible in a quantitative way, the concept of texture in digital images. Furthermore, we developed an innovative model that allows classifying and characterizing texture in digital images, to be used as a useful tool in noninvasive inspection of visual surfaces. The proposed methodology extracts the statistical order from an image of texture. The extraction of the high statistical order has been made using as a tool Markov Random Fields. The Backpropagation neural net is used for designing a classification module that will serve to test the performance of the configuration histograms, which are based on the statistical order. Furthermore, the research suggests the evaluation of the proposed technique from a qualitative perspective.
Keywords: Texture, backpropagation, configuration histograms, classification, Markov Random Fields

1 Introduction

Digital image processing is a current subject of research that has being developed for a long time. Of all the invested efforts, multitude of applications have aroused in quite diverse fields such as industrial, medical, spatial, etc. Where very different tasks are made, such as: codification, edge detection, color processing, etc. We are therefore, in front of a subject in constant growth and expansion, that it is giving rise to a high number of projects and publications.

One of the characteristics that are almost always present in an image is the texture, which generally can be associated to different elements from the scene.

In order to be able to apply the concept of texture to the digital analysis of images a quantitative characterization of the texture is required. Nevertheless, it does not exist a precise definition nor a mathematical expression for the quantitative description of this property.

For the analysis of the texture, there exist three widely used formulations in image processing, they are: statistical [1], structural and spectral [2]. The statistical formulation indicates if an image is smooth, rustic, granulated, etc. [1]. The structural technique, on the other hand, indicates the primitive features that exist in the image, such as regularity of parallel lines; and the spectral technique is based on the energy properties and it is used mainly to detect global regularity in an image, indicating small peaks of high energy in its spectrum [2]. These techniques can used separately or jointly to detect different textures.

The detection of textures has wide spectra of applications, mainly in the following areas:

- Medicine: in several types of images (x-rays, ultrasound, magnetic resonance, etc.); the properties of the texture are important for the diagnosis. Cancer frequently is characterized by analysis of texture in different types of medical images; also techniques of texture have been applied satisfactorily for the detection of abnormalities in mammographic images [3].

- Remote sensors: numerous formulations for recognition of textures in remote sensors have been proposed. Their applications include: land classification, cloud classification, recognitions of seismic patterns, etc.

- Industrial inspection: in industrial processes, the detection of texture in defective manufactured products or natural materials is of crucial importance. The manual inspection frequently is tedious and laborious, for that reason, the automation is very useful [4].

However, none of these classic descriptions leads to a quantitative measurement universally accepted of texture. Due to this difficulty, research in this field continues being interesting until a method be obtained that is satisfactory for most of the applications.

This research is an attempt to describe in a quantitative way, the concept of texture in digitized images. Furthermore, it develops an innovative model that allows to classify and to characterize textures in digitized images, and it can be used as a useful tool in noninvasive visual inspection.

2 Models of the texture

In the image processing literature, texture is often defined as the spatial interaction between the values of pixels. That is to say, the texture analysis attempts capturing its visual characteristics in an analytical way, in order to be able to model mathematically those spatial interactions. Modeling successfully the visual characteristics, we will be able to associate and to discriminate different textures analytically. This type of analysis will require of a number and type of textures previously known. This model only needs to capture sufficient textural characteristics on a training set to classify the texture, however, the procedure is best adapted for textures that belong to the training set.

Of these models of understanding of texture, there are four of them that stand out in the literature of digital image processing.

Statistical models of texture: It is based on a set of features to represent the characteristics of the texture of an image. Those features are contrast, correlation, entropy, etc. They are usually derived from “measurement of the gray level of the image”, “it differentiates from the values of gray” or “cooccurrence matrix” [1]. The characteristics are selected in heuristic form; nevertheless, an image similar to the analyzed one cannot be recreated using some measurement of the set of features.

Structural models of the texture: Some textures can be seen as two-dimensional patterns, composed of a set of primitives or sub-patterns, which are organized according to a certain rule of positioning. Examples of such textures are: brick walls and mosaics; the primitives used are areas of constant gray level, lines, curves and polygons. The correct identification of those primitives is quite difficult. However, if the textures primitives are identified completely, then it is possible to recreate the texture from the primitives. A work using a structural model is indicated in [4].

Models of signal processing for the texture: psychological researchers have given evidence that the human brain analyzes the frequency of the images. The texture is specially fit for this type of analysis due to its properties. Most of the techniques try to process certain characteristics from filtered images, which are used in tasks of classification and segmentation. Among the methods we can mention: filters in the spatial domain, Fourier analysis, wavelets and Gabor filters. Works using this type of formulation are given in [5].

Stochastics models of the texture: a texture is assumed to be the realization of a stochastic process, which is governed by some parameters. The analysis is executed, defining a model and considering the parameters. This way, the stochastic processes can be reproduced from the model and associated to the parameters. The estimation of the parameters can serve to classify and to segment textures. This type of model offers a good possibility to recreate realistic examples of natural textures. Stochastic formulation of the texture is used in [6].

3 Our model of texture

As seen previously, there are several problems associated with modeling texture. A key factor in our model of texture analysis is that it be able to distinguish between different textures. Conventional techniques have been based on prior knowledge of the number and types of textures that are desired to analyze. In our model we shall use the same philosophy of analysis. Therefore, we need to construct a model that can take a texture and capture the totality of the characteristics that distinguish that group of textures. But, in which way we can capture the real characteristics that distinguish a group of textures? And, which will be our approach in the direction of constructing such characteristics? We believe that an understanding of the statistical order applied to images of digitized textures will allow us to capture the real characteristics that distinguish natural textures. We know that the statistical understanding of the texture was proposed by Julesz as early as 1962. The statistical order in the understanding of the texture, applied to digital processing of images has been used in [1], where second order statistics are used to define 14 types of textural features from the cooccurrence matrix. In the texture models that use the auto-models and multilevel logistic models also use second order statistics; nevertheless, they have not provided a base of understanding applicable to natural textures, although in [7] modest results were reached using a Gaussian model of Markov Random Field that use precisely second order statistics. Models that have managed to recreate natural textures using several multiresolution filters do not use third or higher statistical orders, and even more, they become indefinite (ill-defined) if the election of the filters is globally optimal for all the analyzed textures. Julesz in 1962 had already indicated the existence of textural information in the high statistical order.

In such a way, starting from the knowledge already generated and using the approach of the human visual perception proposed by Julesz, is that we have raised our hypothesis applied to textures in digitized images, which establishes that to model textures, models of different statistical orders are required and for it we have chosen the Markov Random Fields as the basis of our model of texture due to its flexibility in defining the statistical dimension and the order.

4 Markov Random Fields

The theory of the Markov Random Fields (MRF) provides suitable tools for us to model context dependent entities such as pixels or other characteristics of spatial correlation [8], [9]. Markov Random Fields are defined with respect to a system of neighborhoods. The mathematical interpretation of the model is defined with respect to the corresponding set of cliques. In this research, a systematic method for the total extraction of the set of local cliques has been used from any system of neighborhood in which the Markov Random Field can be defined.

The fundamental property of a MRF is that given a point j on lattice, the probability of that point, given the values of all the other points of the lattice, is the same as the probability of that point j, given only the values of the neighboring points to j [8, 9]. In other words, the MRFs are characterized by a conditional probability function defined with respect to a system of neighborhoods.

We denote a set of sites on a lattice by S and the system of vicinities on S as $G = \{G_s, s \in S\}$, where G is the set of neighbors for s so that $Gs\ G_s \subset S, s \notin G_s$.

Given the variable X_s in site s with valuex_s the function of conditional probability of a Markov Random Field with its respective system of vicinity G is defined by the theorem of Hammersley and Clifford [9] as:

$$P(X_s = x_s | X_r = x_r, r \in G_s) = \frac{1}{Z_s} \exp\{-\sum_c V_c(x)\} \tag{1}$$

Where Z_s is a constant and Vc is a potential function defined on clique C. The sum is over all cliques in the local set of cliques Cs.

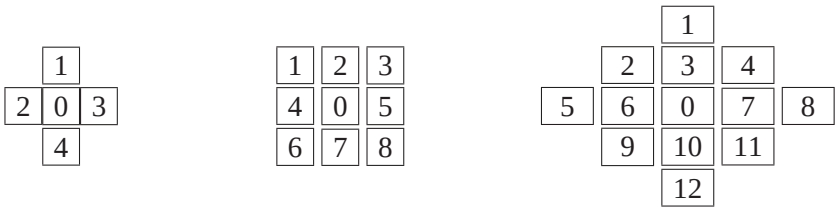


Figure 1: Three different neighborhoods.

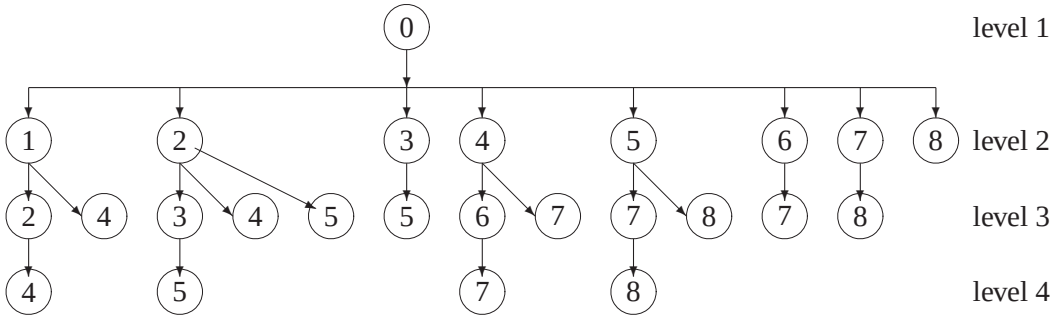


Figure 2: Tree diagram for the 3x3 neighborhood.

The theorem of Hammersley and Clifford, indicates that for a system of neighborhoods is necessary that $s \in G_r \iff r \in G_s$. Three different neighborhoods are shown in figure 1.

Given a system of neighborhoods G, a clique is a set $C \subseteq S$ so that $s,r \in C, s \neq r$, then $s \in G_r$. That is, all the pairs of different sites in a clique are neighbors. In such a way that the local set of cliques for site s is defined as:

$$C_s = \{C \subseteq S : s \in C\} \tag{2}$$

The method used for the extraction of the local set of cliques from a system of neighborhoods of a MRF, is based on a structure of a tree diagram [10]. The root of the tree represents only a site, whereas the branches in the first level represent all the pairs of connections between the site and each one of the components of the neighborhoods. The other branches of the highest levels represent connections of high order that form more complex cliques . Each level in the tree diagram will be represented by a node, which is indicated by a number within the neighborhood previously defined. For example, the tree diagram for the neighborhood of the figure 1.b is shown in figure 2.

5 Characterization of texture using order statistics

In order to characterize the texture of a digitized image, we will use a vector called of directional homogeneity [10], which considers the relation between the pixel corresponding to a site s and all the remaining pixels of neighborhood G_s. The relation to be established between the site and the other pixels of the neighborhood will depend on the type of clique and the statistical order used. Figure 1 shows different types from neighborhood with site s.

Let us suppose that X is a random field of the class described in [8, 9], with a space of discrete range $Z = \{z_k, k = 1, ..., M\}$. Let us also suppose that we have an instance x of this one random field that

V1	U2	V2
U1	S	U3
V4	U4	V3

Figure 3: Site s and its neighborhood.

can be used for the estimation of parameters of the distribution. Figure 3 shows a site (i, j) and its system of neighborhood with the notation employed.

Furthermore, we define as homogeneity function the function I such that:

$$I(z_1,...,z_k) = \{-1, \text{ if } z_1 = z_2 = ... = z_k; 1, \text{ otherwise} \tag{3}$$

This homogeneity function, for the case of cliques of first order will be defined as:

$$I(s) = \ \{-1 \text{ if } s = z_m; 0, \text{ otherwise} \tag{4}$$

Using the terms defined until now, we can express the potential functions of a Gibbs distribution,

$$V(s,E,\theta) = \phi(s,E)^T\theta \tag{5}$$

Where is the vector of parameters and the vector that we will call of directional homogeneity. For example, for the neighborhood of figure 3 the component of the vector , using the tree of cliques (see figure 2) is defined as:

$$\phi_3 = \left\{ \begin{array}{c} (I(s,u_2,v_2)+I(s,u_4,u_3)+I(s,u_1,v_4)), \\ (I(s,u_4,v_3)+I(s,u_2,u_3)+I(s,u_1,v_1)), \\ (I(s,u_2,v_1)+I(s,u_1,u_4)+I(s,u_3,v_3)), \\ (I(s,u_1,u_2)+I(s,u_4,v_4)+I(s,u_3,v_2)) \end{array} \right\} \tag{6}$$

The set of components ϕ_k of vector ϕ represents the different statistical order extracted from the image, where k is the statistical order with $k \in \{1,2,3,4\}$ for a neighborhood of second order like the one shown in figure 3.

6 Direct measurement of the statistical order

This section presents our proposal to characterize digitized images of texture that makes use of the properties of vector ϕ in each point of the digitized image.

Taking the expressions 3, 4and 5 for a second order neighborhood (figure 3) and applying them to the calculation of the components of vector ϕ , we see that the values that can take components are the following:

$$\phi_1 \in \{0,1\} \tag{7}$$

$$\phi_2 \in \{-2,0,2\} \tag{8}$$

$$\phi_3 \in \{-3,-1,1,3\} \tag{9}$$

$$\phi_4 \in \{-1,1\} \tag{10}$$

Of the possible values that can take different components ϕ_k , we establish equivalence between the value and its respective binary, ternary or tetranary value, as corresponds. Table 1 illustrates some examples of equivalences.

Table 1: Equivalence between the value and ternary number.

Value	Ternary Number
-2	0
0	1
2	2

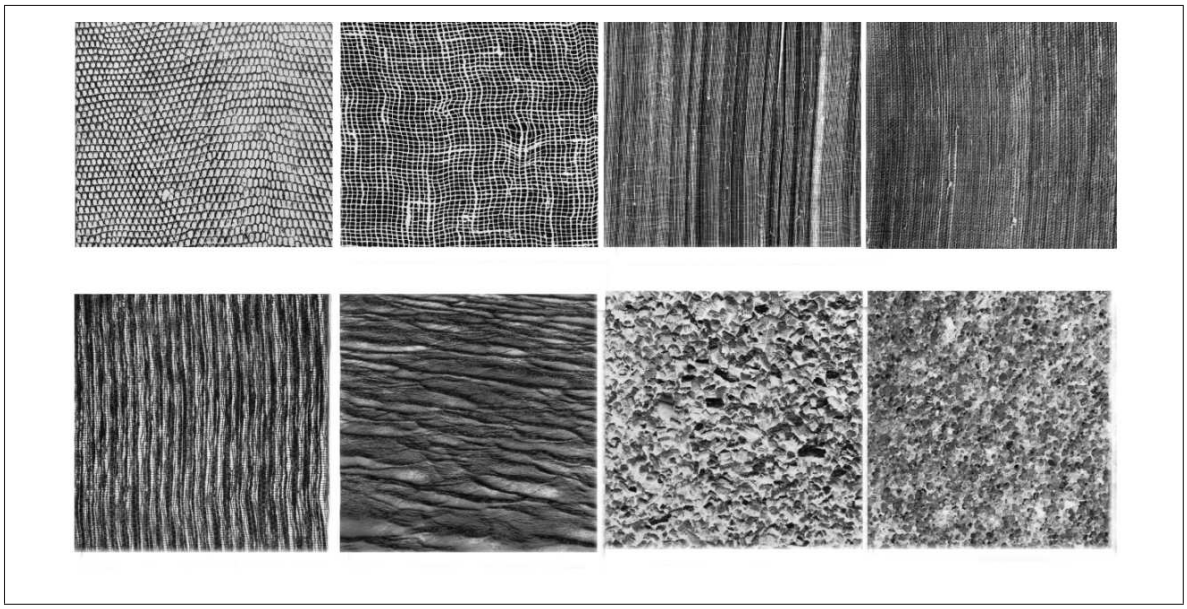


Figure 4: Brodatz textures used in our experiments. Top row: D22, D103, D105 and D79; Bottom row: D76, D37, D5 and D28.

Now, our interest is only in the values that take the components of vector ϕ, that is why we will work with a decimal number, instead of considering its vector array.

This way, given a digitized image of texture and applying the different statistical orders through the different cliques used, we will obtain histograms of frequency versus directionality, which will be different and particular for each of the analyzed textures.

7 Classification of textures using histograms of configurations

The process of texture classification will be based on the histograms of configurations proposed in the previous section. In the present work we have used a set of 8 classes of textures that have been obtained from the album of textures of Brodatz [11]. Each image has a dimension of 640x640 with 256 gray levels (8 bits/pixel). The images of textures used are shown in Figure 4.

For each of these images, 32 sub-images of 64x64 pixels have been extracted. For each sub-image, the histogram of configurations has been computed. In figures 6, 7 and 8, can be seen some of those histograms.

The form in which the 32 sub-images have been extracted is explained in a scheme (figure 5) that represents the 32 sub-images taken from one of the images shown in figure 4. The white squares represent the 32 sub-images and the dark background represents the greater image of the texture (figure 5). Images with an asterisk have been taken to construct a characteristic vector, which will be used to train the

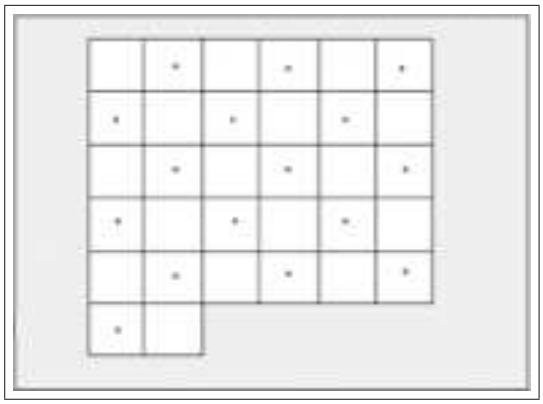


Figure 5: Layout employed to select the training and validation subimages. Each cell represents a subimage, and every cell containing an asterisk is used for training.

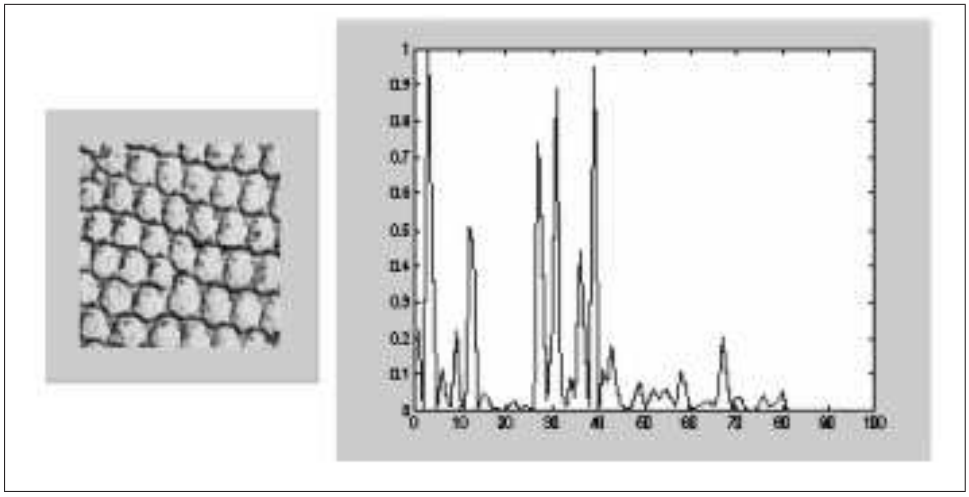


Figure 6: Texture D22 and its corresponding configuration histogram.

classification module. This procedure has been performed with the eight images of textures shown in figure 4. The sub-images without asterisk in figure 5 are used to test the performance of the texture classification module.

The architecture that has been developed for classification of the texture consists of four basic blocks (see figure 9): a) Extraction of characteristic vector. The purpose of this block is to construct a vector made of the frequencies of the histogram of configurations used; b) Classification: the objective of this block is to classify the different characteristic vectors, that is to say, the information contained in the digitized input image (texture). For this block it has been proposed using a backpropagation (BP) neural network; c) Decision: this block makes the final decision from data generated by the classification block; d) Data Base: it stores parameters and models to control adaptively the operation of the various blocks.

Once determined the configuration histograms for different statistical orders for the 32 sub images and for each texture category, each configuration histogram is evaluated with respect to performance.

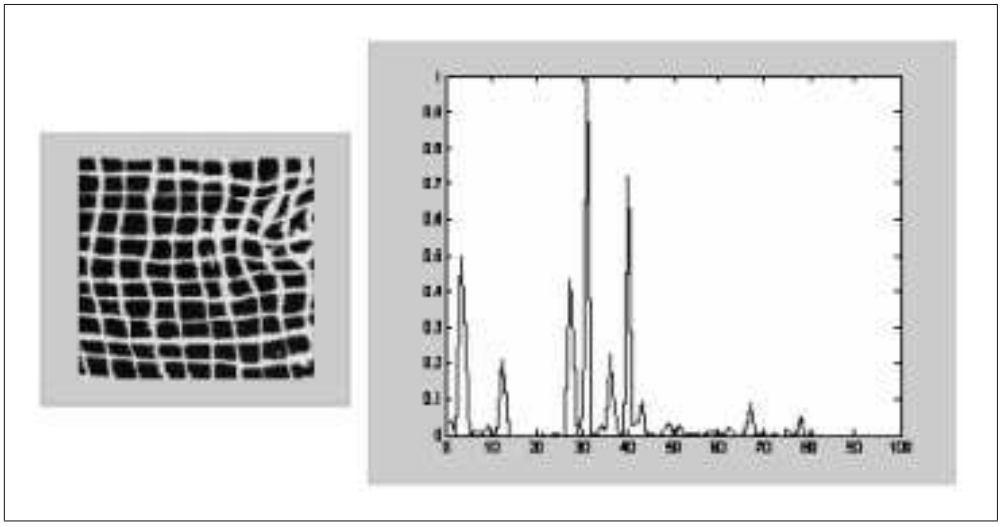


Figure 7: Texture D103 and its corresponding configuration histogram.

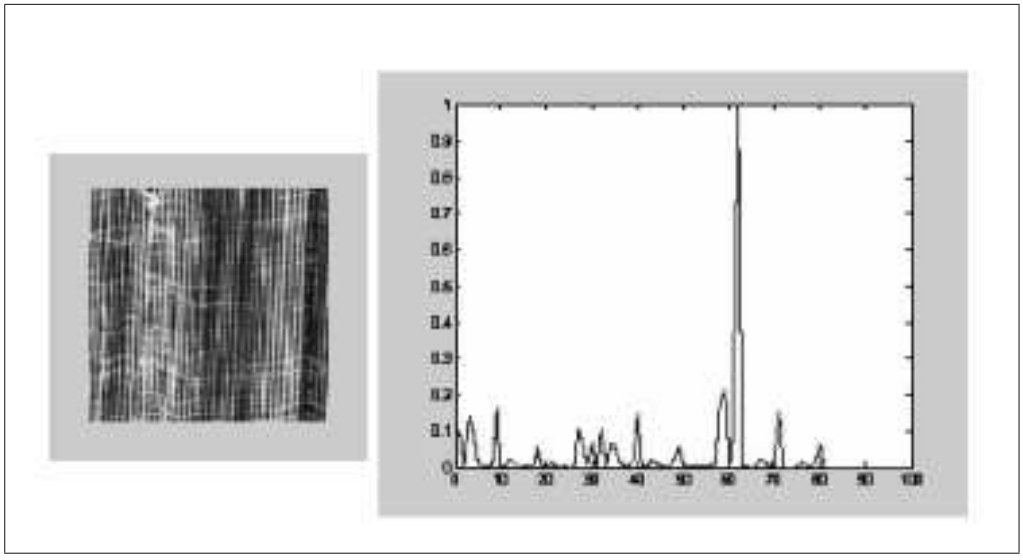


Figure 8: Texture D105 and its corresponding configuration histogram.

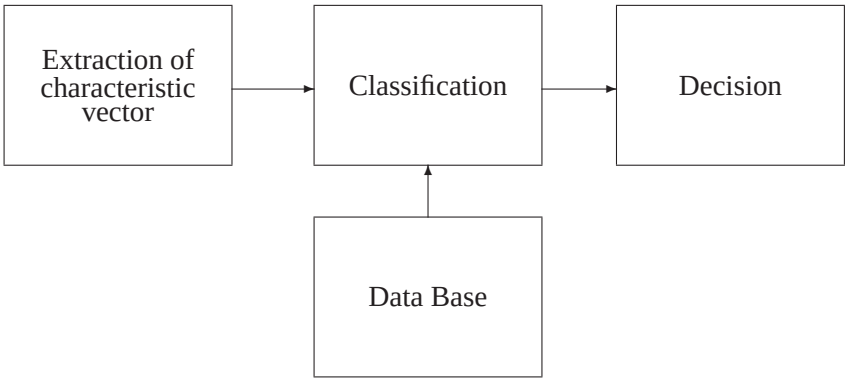


Figure 9: Block diagram of the proposed system.

Table 2: Proposed binary codes for the seven output nodes.

Texture	**1st**	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**
D22	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	1
D79	0	0	0	0	0	1	0
D76	0	0	0	0	1	0	0
D28	0	0	0	1	0	0	0
D37	0	0	1	0	0	0	0
D105	0	1	0	0	0	0	0
D103	1	0	0	0	0	0	0

The classification module, is built using a BP neural net that is trained with 16 sub images and validated with the remaining 16 unused sub images.

A BP neural net is trained with a predefined set of input-output pairs. Each input pattern (vector) is applied to the input layer, and it propagates through the successive layers until it generates a given output. This current output is compared against the target output to obtain the so called error vector. These errors are back propagated, beginning at the output layer and then propagated to the hidden layers until the input layer is reached. Based upon an error minimization procedure, the connection weights of each neuron are modified, in such a way as to reduce the final error the next time the same input pattern be presented to the network.

For this research, a BP neural net with 3 layers has been used. The number of nodes of the input layer is the same as the number of components of the characteristic vector. For the hidden layer, from 12 to 16 nodes have been used, and finally, for the output layer, 7 neurons with proper codification have been employed. Table 2 shows the proposed code for the seven output nodes.

The BP neural net has been programmed and trained using Matlab®. In particular, two algorithms have been evaluated for weight modification: gradient descent (traingd) and Levenberg-Marquardt (trainlm).

Table 3: Results using a 3x3 neighborhood and second order histograms.

Texture	D22	D103	D105	D79	D76	D37	D5	D28	Total
D22	**16**								
D103		**14**							
D105		2	**11**		2		1		
D79			3	**12**				1	
D76			2		**13**		2	2	
D37				3		**16**			
D5					1		**11**	3	
D28				1			2	**10**	
%	100	87.5	68.8	75	81.3	100	68.8	62.5	**80.47**

Table 4: Results using a 5x5 neighborhood and third order histograms.

Texture	D22	D103	D105	D79	D76	D37	D5	D28	Total
D22	**16**			1					
D103		**14**							
D105		1	**13**		1				
D79				**12**					
D76			2		**13**			1	
D37				3		**16**			
D5		1			2		**13**	1	
D28			1				3	**14**	
%	100	87.5	81.3	75	81.3	100	81.3	87.5	**86.72**

8 Experimental results using histograms of configurations

Table 3 shows the classification results obtained using a 3x3 neighborhood and histograms of configurations of second order statistics. In this case the classification module was built with 16 hidden layer neurons, and trained for 500 epochs.

Later on, the performance was evaluated with 5x5 neighborhoods and third order histograms. Table 4 shows the results of this experiment.

From the above results, we can confirm that the histograms of configurations are sufficiently representative of the textures analyzed using different statistical orders. Furthermore, we achieve a classification rate of 86.72% while employing third statistical order.

9 Summary and Conclusions

A new texture classification method has been developed, based on the extraction of the statistical order from a digitized texture image. This later operation has been performed using the set of local cliques from a neighborhood.

The use of histograms of configurations in the process of texture classification has been quite successful, achieving a classification rate of around 87%.

The decision of employing a BP neural network was also adequate for the classification. Best results

were reached when training the neural net with the Levenberg-Marquardt method.

The proposed classification architecture requires of just a few features, thus achieving a reduction in training time and smaller computational efforts during the classification stage.

The work developed so far constitutes the initial stage of a series of applications and works related to texture and non-destructive visual inspection of materials.

References

[1] Haralick R., Shanmugam K. and Dinstein I., Textural Features for Image Classification, *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-3, N^o. 6, pp. 610-621, 1973.

[2] Hsin H., Texture Segmentation Using Modulated Wavelet Transform, *IEEE Trans. on Image Processing*, Vol 8. N^o. 7, 2000.

[3] Hernández L., Torrealba V. and Reigosa A., Clasificación Automática del carcinoma de la mama, mediante un sistema de reconocimiento basado en redes neuronales, *Memorias II Congreso Latinoamericano de Ingeniería Biomédica*, La Habana, Cuba, 2001.

[4] Mery D., Da Silva R., Calôva L. and Rebello J., Detección de fallas en piezas fundidas usando metodología de reconocimiento de patrones, *3rd Panamerican Conference for Nondestructive Testing*, Rio de Janeiro, Brazil, 2003.

[5] Bader D., JáJá J. and Chellapa R., Scalable Data Parallel Algorithms for Texture Synthesis using Gibbs Ramdon Fields, *IEEE Transactions on Image Processing*, vol. 4 N^o. 10 , pp. 1456-1460, 1995.

[6] Pun C. and Lee M., Rotation Invariant Texture Classification Using a Two Stage Wavelet Packet Features Approach, *IEE Proc. Vis. Image Signal Process*, vol. N^{o}148, N^{o}6, pp 422-428, 2001.

[7] Manjunath B., Simchony T. and Chellappa R., Stochastic and Deterministic Networks for Texture Segmentation, *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. N^{o}6, pp. 1039-1047, 1990.

[8] Li S., Modeling Image Analysis Problems Using Markov Random Fields, Handbook of Statistics, vol. N^o 20, 2000.

[9] Li S., Markov Random Fields Models in Computer Vision, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp 866-869, 1994.

[10] Gómez M., Modelación y clasificación de texturas utilizando Campos Aleatorios de Markov, Master Thesis, Electrical Engineering, Universidad de Santiago de Chile, Santiago, Chile, 2004.

[11] Brodatz P., *Textures*. New York: Dover, 1966.

Renato Salinas, Mauricio Gomez
Universidad de Santiago de Chile
Electrical Engineering Department
Ave. Ecuador 3519, Santiago, Chile
E-mail: rsalinas@lauca.usach.cl, mauriciogomez@esfera.cl

International Journal of Computers, Communications & Control
Vol. I (2006), No. 2, pp. 53-60

Importance of Flexibility in Manufacturing Systems

Héctor Kaschel C., Luis Manuel Sánchez y Bernal

Abstract: Flexibility refers to the ability of a manufacturing system to respond cost effectively and rapidly to changing production needs and requirements. This capability is becoming increasingly important for the design and operation of manufacturing systems, as these systems do operate in highly variable and unpredictable environments.
It is important to system designers and managers to know of different levels of flexibility and / or determine the amount of flexibility required to achieve a certain level of performance. This paper shows several representations schemes for product flexibility and discusses the usefulness and limitations of each.
Keywords: Flexibility, Fuzzy Logic, Disjunctive Graph, Constraints Temporal Network, Complexity in Manufacturing, Non Lineal Mathematic Programming

1 Introduction

Present market demands require that Manufacturing Systems develop their activities under a dynamic and uncertain production environment (Calinescu et al., 2003). To understand the problems affecting Flexible Manufacturing Systems (FMS), it is necessary to incorporate the concept of flexibility in the scheduling (Deshmukh et al. 2002). Flexibility constitutes a strategic topic in decision making to give quick and efficient answers to the demands of the national and international markets (Pelaez, and Ruiz, 2004).

In order to define the flexibility in FMS, the scientific literature reports a wide variety of concepts. For example, flexibility for Sethi and Sethi (Sethi and Sethi, 1992) represents the capacity of FMS to modify manufacturing resources to produce different products efficiently maintaining an acceptable quality.

For Gupta (Gupta, 2004), it is necessary to make the difference between flexibility inherent to the FMS and flexibility required during the production cycle. The first one is defined as the capacity of resources and human operators to implement decisions of the factory company. While required flexibility is defined by the changes in the environment that surrounds the FMS and for the implementation of new manufacturing strategies. Starting from these differences, it is possible to determine the flexibility type and measure of the FMS quantitatively.

For (Benjaafar and Ramakrishman, 1996), it is important to differentiate the types of flexibility in the FMS. They define the flexibility of the FMS as product flexibility and process flexibility. The first one refers to the variety of factory options for a certain product. While process flexibility is defined as a characteristic of an industrial process to operate under diverse dynamic conditions of operation. Both general flexibilities can be grouped hierarchically as shown in the figure 1.

The classification proposed by (Benjaafar and Ramakrishman, 1996) shows flexibility as a consequence of the physical and logical characteristics that the manufacturing system presents To quantify the operative flexibility of job i, a function is defined starting from the time of dependent configuration of the sequence between operations:

$$\Phi(p_i) = \frac{-\sum_{j=1}^{N_i}\sum_{l=1}^{k_{ij}}\left(\frac{\frac{1}{s_{ijl}}}{\sum_{l=1}^{k_{ij}}\frac{1}{s_{ijl}}}\right) log\left(\frac{\frac{1}{s_{ijl}}}{\sum_{l=1}^{k_{ij}}\frac{1}{s_{ijl}}}\right)}{N_i} \quad (1)$$

where:

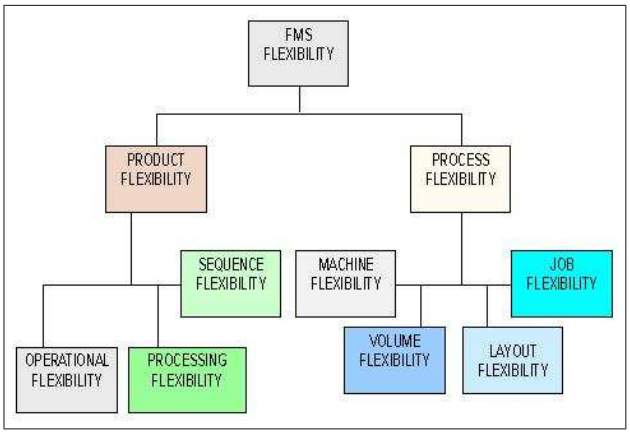


Figure 1: Classification of the FMS Flexibility

- s_{ijl} represents the time of configuration of job i during the commutation of the operation j to the operation l.

- N_i represents the total number of operations associated to job i.

In the present paper different models were analyzed as described in several studies dealing with product flexibility in FMS.

2 Product flexibility

In all FMS, each job or jobs group is associated to a certain scheduling, consisting in an orderly sequence of operations to be executed in a group of machines. An example of scheduling of the FMS is illustrated in figure 2(a).

If we incorporate conditions of flexibility to the scheduling (operational flexibility), an operation can be carried out in other machines. These, in general, can be different or they can have different operational levels (figure 2 (b)).

The classification of the operations can also be varied (sequence flexibility), where some operations should be carried out in sequence while others don't have this restriction. An example of this type of flexibility is shown in figure 2(c), where operations 2 and 3 can be carried out in any order after executing operation 1.

Finally, in a highly flexible FMS, a job can have a group of schedulings (processing flexibility). This means that the group of operations required to manufacture a job can be variable and depending on the operation conditions of the FMS. For example, in figure 2(d), operations 1 and 2 can be replaced by the operation 5 while operation 3 can be divided in operations 6 and 7.

3 Analysis models for product flexibility

The product flexibility constitutes an important aspect in the functionality of the FMS, where the variability in the scheduling and the operability of the machines allow several alternatives to process the jobs. In what follows some models of analysis of the product flexibility are presented.

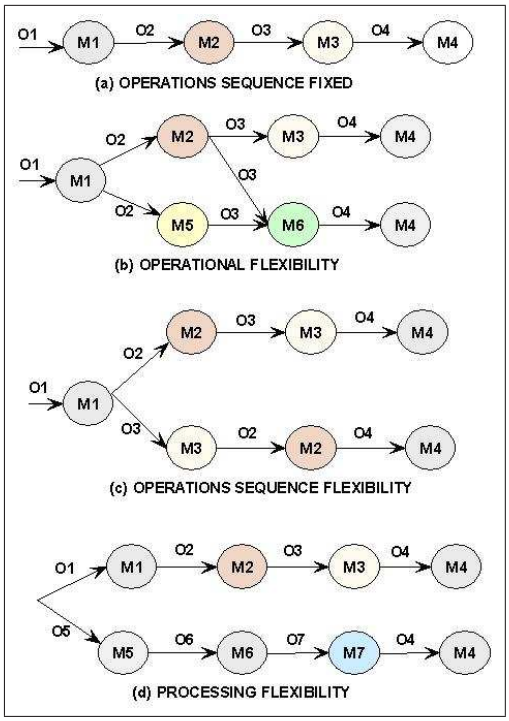


Figure 2: Types of Product Flexibility

3.1 Disjunctive Graph

It is a model of directed graph, whose nodes have a certain weight assigned. Mathematically it is expressed as 3 - tupla (Benjaafar and Ramakrishman, 1996):

$$G = (N, A, E) \tag{2}$$

where:

- N: represents the group of nodes that represent the operations. In this model are defined two particular nodes, called initial and final. The positive weight assigned to each node represents the processing time of the corresponding operation. The initial node is connected with the first operation of each job, and in the same manner the last operation of each job is connected with the final node.

- A: represents the group of conjunctive directed arcs that indicate the precedence restrictions among the operations of each job.

- E: represents the group of disjunctive directed arcs that indicate the restrictions of machines operative capacity.

To illustrate the application of the disjunctive graph, an example of scheduling of 4 different jobs is presented (J_1, J_2, J_3, J_4) in three types of machines (M_1, M_2, M_3). The distribution of processing times and assignment of machines is shown in table 1, starting from this information we proceed to generate the disjunctive graph of figure 3, that shows the precedence restrictions between the operations (conjunctive arcs) and the flexibility in the machines operative sequence (disjunctive arcs).

	Processing time			Machines assignment		
	M_1	M_2	M_3	1	2	3
J_1	5	8	2	M_1	M_2	M_3
J_2	3	9	7	M_3	M_1	M_2
J_3	1	10	7	M_1	M_3	M_2
J_4	7	4	11	M_2	M_3	M_1

Table 1: Machines assignment and processing times

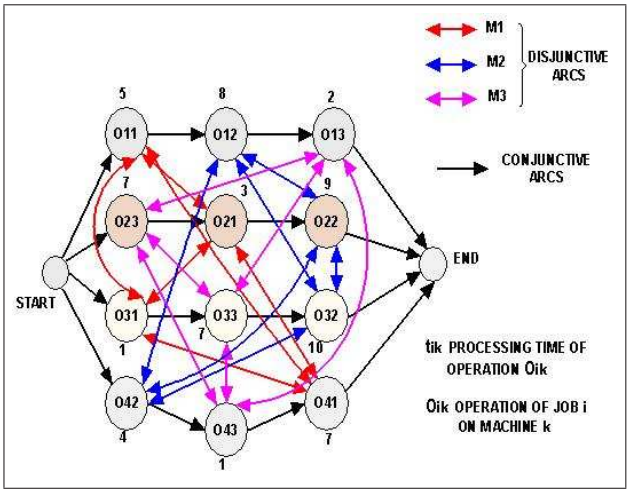


Figure 3: Disjunctive Graph generated by table 1

3.2 Temporal Constraints Network

It is a model of directed graph that allows the representation of temporary constraints assigned to the precedence relationships between operations that are part of a scheduling of a FMS. The nodes of the graph represent the operations and the arcs have associated temporary intervals to denote the temporary constraints that exist between two nodes. All the times are calculated starting from of the scheduling initial state. The development of the network is based mathematically from Disjunctive Metric Temporal Algebra (Alfonso, 2001).

This methodology is used to solve scheduling problems characterized from a group of temporary constraints, consequently, the techniques developed in areas like Temporary Reasoning and Constraints Satisfaction can be applied. In general, two techniques are used: Closure and Constraint Satisfaction Problem (CSP) (Barber, 2000).

Closure is a deductive process by means of which new constraints are inferred, starting from existing constraints. It also allows to detect possible inconsistencies (value of the variables that don't lead to any solution) that can be eliminated. The main advantage of the closure techniques consists in the reduction of the search space. Therefore, the closure is used fundamentally as a previous step to a process of search of solutions.

Techniques of Constraints Satisfaction Problem constitutes a process of search of solutions by means of the successive assignment of values to the variables that are part of the constraints of the problem. In a process of Constraints Satisfaction Problem, it is important to use different heuristics to make the search process more efficient (Beck and Fox, 2000).

To illustrate the use of the Temporary Constraints Network, a distribution of temporary constraints is presented among four operations of a FMS (see table 2).

Starting from the information in table 2, we proceed to establish the group of operations (O), the

	O_1	O_2	O_3	O_4
O_0 initial state	(10, 20)	(10, 20)	(20, 50)	(60, 70)
O_1		(30,40), (60,60)	(10,30), (40,40)	(40, 60)
O_2				(0, 30)
O_3		(10, 20)		(20,30), (40,50)

Table 2: Precedence Temporary Constraints between operations

group of precedence relationships between operations (P) and the group of temporal relationships between operations (R). Finally, starting from the definitions of the groups we proceed to generate the corresponding Graph of Temporal Constraints (see figure 4):

$$G = (O, P) \tag{3}$$

$$O = (O_0, O_1, O_2, O_3, O_4) \tag{4}$$

$$P = \left\{ \begin{array}{c} (0_0, O_1), (0_0, O_2), (0_0, O_3), (0_0, O_4) \\ (0_1, O_2), (0_1, O_3), (0_1, O_4), (0_2, O_4) \\ (0_3, O_2), (0_3, O_4) \end{array} \right\} \tag{5}$$

$$R = \left\{ \begin{array}{c} (O_0(10,20)O_1), (O_0(10,20)O_2), \\ O_0(20,50)O_3), (O_0(60,70)O_4, \\ (O_1(30,40)(60,60)O_2), \\ (O_1(10,30)(40,40)O_3), (O_1(40,60)O_4), \\ (O_2(0,30)O_4), (O_3(10,20)O_2) \\ (O_3(20,30)(40,50)O_4) \end{array} \right\} \tag{6}$$

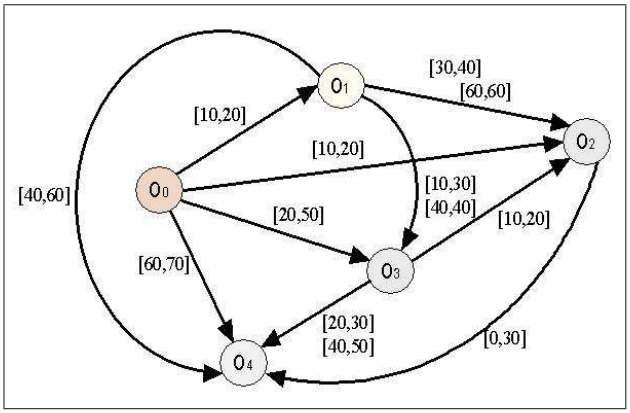


Figure 4: Temporary Constraints Network generated by Table 2

3.3 Model based on Mathematical Programming

Chandra and Tombak (Chandra and Tombak, 1993) propose a operations sequence flexibility model, based on the theory of lineal programming. The model incorporates the concepts of machines reliability and machines processing capacity. Machine reliability is defined as the probability of an operation to be executed by a machine in a certain interval of time. Machine processing capacity, represents the total number of jobs that a machine can execute in a interval of time.

The FMS base for the development of the model consists of a finite group of machines that can present independent fault events. The model is formulated considering the jobs flow on a fixed scheduling and it is expressed mathematically as:

$$RF = max \sum_{i} \sum_{h} c_{ih} x_{ih} \tag{7}$$

subject to the following constraints:

$$\sum_{i} \sum_{h|b_{ihk}=1} \frac{t_{ik}}{P_k} x_{ih} \leq T_k, \forall k \tag{8}$$

$$\sum_{h} x_{ih} > d_i, \forall i \tag{9}$$

$$x_{ih} \geq 0, \forall i, h \tag{10}$$

where,

- t_{ik} represents the processing time of job i in machine k;
- T_k represents the total processing time of machine k;
- P_k represents the probability that machine k is operative in a certain time;
- b_{ihk} represents the possibility that job i is processed in machine k along route h;
- C_h represents the factor of cost of job i when processed in route h. This factor depends on the reliability machines located along route h;
- d_i represents the minimum demand of job i and,
- x_{ih} represents the flow of job i on route h.

The model represents the processing capacity of each machine considering its reliability, where the factor $\frac{t_{ik}}{P_k}$ represents the time required to process job i in machine k. The model considered $n+m$ constraints and $m(1+\sum_{k=1}^{n-2} k! C_k^{n-2})$ variables in case of full machines connections.

3.4 Model based on Fuzzy Logic

In (Tsourveloudis and Phillis, 1998) and (Fortemps, 2000), analyze the flexibility starting from a methodology based on the knowledge that the human expert has on the structural and dynamic behavior of the FMS. For (Tsourveloudis and Phillis, 1998) the flexibility is represented by means of a group of diffuse linguistic rules of the type:

$$\text{IF } \langle \text{antecedent fuzzy} \rangle \text{ THEN } \langle \text{consequent fuzzy} \rangle \tag{11}$$

This proposal is based on the fuzzy behavior that the FMS presents, in cases of concurrence and synchronization of operations or fault machines events. For (Tsourveloudis and Phillis, 1998), a FMS presents seven different types of flexibility:

1. Materials Manipulation System Flexibility: It measures the capacity of the system to transfer several jobs types efficiently between two neighboring points.
2. Product Flexibility: It measures the capacity of the system to produce mixed jobs.

3. Operative flexibility: It measures the capacity to modify the sequence of operations to produce a job.

4. Process Flexibility: It measures the capacity of the system to produce different jobs without modifying the FMS structure.

5. Volume Flexibility: It measures the operative capacity of the FMS for different levels of productivity.

6. Scalability Flexibility: It measures the cost or time required to enlarge a FMS.

7. Labor Flexibility: It measures the operator's capacity to develop different operative tasks of FMS.

Each one of these types of flexibility is expressed by means of a fuzzy rule that represents the expert's knowledge about this flexibility. The fuzzy rule is expressed mathematically as:

$$\text{IF } F_1 \text{ is } A_1 \text{ AND} \ldots \text{ AND } F_N \text{ is } A_N \text{ THEN } F_{MF} \text{ is MF} \tag{12}$$

where: $A_i = \{Low, About\_Low, Average, About\_High, High\}$ represents the group of linguistic values for the flexibility parameter. For example, the following fuzzy rule expresses the Operative Flexibility of the FMS:

$$\text{IF } C_0 \text{ is } T_{C_0} \text{ AND } S_B \text{ is } T_{S_B} \text{ THEN } F_R \text{ is } T_{F_R} \tag{13}$$

For this rule, C_0 represents the number of operations common to machines group and S_B represents the FMS capacity to route a jobs group under conditions of machines faults.

4 Conclusions

The models described in this paper define a set of basic principles that should satisfy all parameters associated to the FMS flexibility, such as:

- They should be specified and derived according to the methodology proposed for the analysis of the flexibility.

- They should incorporate structural and operational aspects of the FMS.

- They should incorporate and accumulate the human expert's knowledge.

Of the models, we can conclude that the problem of the Flexibility is analyzed starting from one of the three big current problems affecting the FMS [CALINESCU et al 2003]:

- Making Decisions Problem, analyzed by the methods based on Mathematical Programming and Fuzzy Logic.

- Structural Problem, analyzed by the method of the Disjunctive Graph and

- Behavior Problem, analyzed by the method of the Temporary Constraints Network.

References

[1] Alfonso, M., *An integration model of Temporary Constraints based on methodology CLOSURE and CSP: Application to Scheduling Problems*, Doctoral Thesis presented at University of Alicante, Spain, 2001.

[2] Barber, F., "Reasoning on interval and point-based disjunctive metric constraints in temporal contexts". *Journal of Artificial Intelligence Research*, pp:35-86, 2000.

[3] Beck, J. and Fox M., "Constraint directed techniques for scheduling alternative activities". *Journal of Artificial Intelligence*, pp: 211-250, 2000.

[4] Benjaafar, S. and Ramakrishman, R., "Modeling, measurement and evaluation of sequencing flexibility in manufacturing systems", *International Journal of Production Research*, Vol. 34, pp:1195-1220, 1996

[5] Chandra, P. and Tombak, M., *Models for the evaluation of routing and machine flexibility*, Technical Report of Decision Craft Analytics, www.decisioncraft.com, 1993.

[6] Deshmukh, A., Talavage, J. and Barash, M., "Complexity in Manufacturing Systems", *IIE Transaction on Manufacturing Systems*, Vol 30, pp: 645 - 655, 2002.

[7] Fortemps, P., "Introducing Flexibility in Scheduling: The Preference Approach", *Advances in Scheduling and Sequencing under Fuzziness*, Ed. Springer Verlag, pp:61-79, 2000.

[8] Gupta, A., "Approach to Characterize Manufacturing Flexibility", *Second World Conference on Production & Operations Management Society*, Cancun - México, pp:100-120, 2004.

[9] Calinescu, A., Sivadasan, S., Schirn, J. and Huaccho, L., *Complexity in Manufacturing: An Information Theoretic Approach*, Technical Report of Manufacturing System Research Group, Department of Engineering Science, University of Oxford, England, 2003.

[10] Pelaez, J. and Ruiz, J., *Measuring Operational Flexibility*, Technical Report, University of Murcia, Faculty of Economy and Enterprise, Spain, 2004.

[11] Sethi, A. and Sethi, S., "Flexibility in Manufacturing: A survey", *International Journal of Operations and Production Management*, pp:35-45, 1992.

[12] Tsourveloudis, N. and Phillis, Y., "Fuzzy Assessment of Machine Flexibility", *IEEE Transaction on Engineering Management*, Vol. 45, pp:78-87, 1998.

Héctor Kaschel C. and Luis Manuel Sánchez y Bernal
Depto. de Ingeniería Eléctrica and
Depto. de Matemática y Ciencia de la Computación
Universidad de Santiago de Chile
Avda. Ecuador ♯ 3519 Estación Central. Santiago - Chile
E-mail: hkaschel@lauca.usach.cl o msanchez@lauca.usach.cl

International Journal of Computers, Communications & Control
Vol. I (2006), No. 2, pp. 61-71

Solution to the Reliability Problem of Radial Electric Distribution Systems Through Artificial Intelligence

E. López, J. Campos, C. Tardon, F. Salgado, J. Tardon, R. López

Abstract: In this paper the distribution electric system reliability is recognized like an artificial intelligence problem. How this idea is applied in evaluation of reliability is detailed. Concepts as Intelligence Matrix and Inter-feeder Route are defined. From the last one a reliability prediction strategy for medium voltage networks is proposed and tested.
Keywords: Artificial intelligence, agent, searching, primary route, secondary route

1 Introduction

It has been established that within an electric system, nearly 80% of the faults occur in the distribution system, (Billinton and Allan,1998). Legal aspects protect the costumers from faults in the electric system. So it's necessary, for electrical companies, to guarantee a high level of security, quality and reliability in the service. Companies are not only affected by the demanding norms but also by high financial lost due to energy non-sell and penalties. The reliability is evaluated using the markovian models (Brown and Gupta, 1998; Asgapoor and Mathine, 1997 Brown, et al., 1997 Brown and Ochoa, 1998) or different analytic methods (Billinton y Wagn, 1999), that, according to the past behavior, they make an extrapolation of its future one. This paper is based in developing a method that makes possible the prediction of distribution system's behavior in terms of reliability, addressing it to the Artificial Intelligence (AI). This is capable of giving information in terms of representative indexes in the points of loading and overall system, including alternative supply in case of the principal feeding's failure. When considering the AI, we intent to model the distribution system's reliability and developing concepts to be used in the solution of a generic problem that requires topological analysis. Therefore, emphasis is made in searching problems as AIt's tasks.

Within the AI's wide world we find the searching's problems (Russell and Norvig, 1996). Once the problem is defined (this action is called the "abstraction") the process starts from an initial state. Then, operators are applied to the present state, resulting in a new set of states. This process is called "expansion" state. The essence of the search is to choose one option and to put all the others aside to be expanded afterwards, just in case the first election does not reach solution. The dilemma of which state to be expanded first is determined by the "strategy of search". The process of searching can be included as the construction of a search tree that is superposed on the space of states where the node of roots is the initial state whereas the nodes leaves are states that do not have successor in the tree. (because they have not been expanded yet or because they have already been expanded but they generate an empty state). An efficient strategy of search is named "depth-first search". This strategy always expands one of the nodes at the deepest level of the tree and only when the search finds its end (a goal state or one without expansion) the search continues expanding nodes of less deeper level.

2 Proposed method

The algorithm has the mission or purpose of finding the failure modes of the system. The hypotheses are: The elements of protection are a 100% reliable. The fault is determined by first order cut. The first step of the method is to make a protocol of nodal assignation. It consists in labelling a bus or node of distribution with number 1 and the branches with ascendant (rising) numbers from the departure node to full stop of load, in the form shown by the arrows in figure 1.

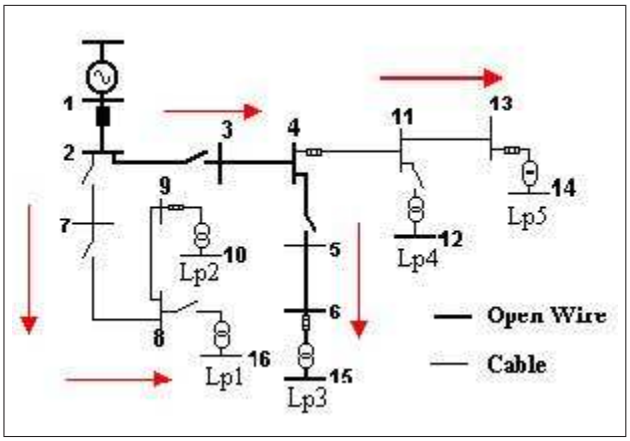


Figure 1: Basic System

Next the space of state is defined, the one that corresponds to the environment in which the agent can move to find the solution. According to the above for the basic system of test (BS), figure 1, the states are defined by each one of the buses of the system. The BS consists of 16 buses and 5 points of load (Lp). There are 16 possible states by which the agent can "travel" with the purpose of satisfying the test of goal. The information related with the topology of the BS is in the so called "Intelligence Matrix"(IM) which forms the base of the algorithm of search. Part of the IM corresponding to the BS is shown in figure 2

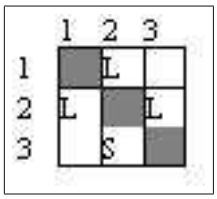


Figure 2: Part of IM for the Basic System

The numbers for the rows and columns are the nodes or buses. The procedure to enter the data consists of entering the components between two consecutive nodes, identifying it with a letter and associating it with the respective element of the IM. For BS, between the node 1 and 2 there is a line, indicated by an L in the position of the element [1,2] of the matrix. The same for the element [2,1]. When there is a switching or protection device (as between nodes 2 and 3) or when there is a line and a disconnects, the element, [2,3], is defined with an L (line). In the position [3,2] you enter an S (disconnects) indicating that the switching device is closer to bus 3. This directional quality or differentiation of the common elements of the system with those of switching and/or protection is conclusive in the optima resolution of the problem, this is the reason of the name "Intelligence Matrix". Specially, for the evaluation of the indicator of future reliability, en the BS are identified:

Initial State: The search begins from the point of load to evaluate.

Operators: The agent can "jump" or "travel" from a particular state towards the next closer, i.e., from [3,2] to [2,1]. In direction to the node of generation (ascending form) or towards the other point of load (descending form).

State Space: The states extended by the agent from the load point (initial state).

Route: The route from one state to another within the space of states.

Goal Test: It corresponds to have extended all the possible states in which any eventual fault may produce a lost in supply energy to the load point evaluated.

Cost of Route: Unit cost in order to travel from one state to the next nearest one.

Solution: It corresponds to the result of the algorithm of search once the goal is fulfilled.

In accordance to the above the complete IM for the BS is shown in figure 3. The non-existence of elements in the system is indicated with blank spaces in figure 3. In order to find the failure modes of system, the algorithm must identify the primary route and the secondary route (López et al, 1998). The IM of the BS shows that the diagonal divides the information of the different routes. The last one, due to the arranged assignation of the number of nodes. On the other, the algorithm uses the "best first search", then, the best expanding option occurs at first. In this case, it corresponds to the primary route. It leaves, in "waiting queue", the secondary routes for the next expansion.

2.1 Getting the Primary Route

The primary route is the minimal cost one from the load point to the feeding node. The information about the primary route is contained under the principal diagonal of the IM. The primary route of each load point is stored in a matrix called " matrix of primary route" (MPR). Though each row will contain the primary route corresponding to a load point. The MPR is of order m x n (load point x nodes) and initially is a null matrix. Its formation is ordered and it is starting with the agent in the first load point (Lp1). In the intelligence matrix the row corresponding to the node number, that has the load point, is checked. Then the elements contained under the diagonal are examined. When finding an element within the search sector, in the row of the matrix of primary routes, the number of the next node that corresponds is entered. The number of the column that corresponds to the found element gives this node. Then, is verified the row corresponding to the next node of the primary route and so on, until arriving at node 1 of the feeding. The figure 4 outlines the way of how to find the primary route for the load point Lp1. It corresponds to the node 16. Figure 5 shows the BS's matrix of the primary route (MPR of BS). Regarding to the agent like an "problem-solving agent" (Russell and Norvig, 1996) and one state as a structure of simple data, the agent is placed in a problem of determining the route. The agent corresponds to an algorithm of resolution that uses a sequence of actions to follow in order to obtain its goal. The "resolutor-agent" of the simplified problem, in order to find the primary route, corresponds to figure 6. This agent is executed for each load point (Lp) and ends or gets to the goal, when finding the feeding node (node 1). The route of the agent corresponds to the one shown above in figure 4. It is observed in figure 6 that the agent has two rules (if condition, then action).They take step from a basic and incipient system to an expert one that has the particularity to provide a great flexibility by the time of incorporating new knowledge. The incorporation of a new rule is enough for it, which don't need to change the algorithm's behavior.

2.2 Secondary Routes

At the moment the "resolutor agent" of the primary route crosses the system and full the matrix of primary route, MPR. A vector is added that stores the quantity of nodes including each primary route This vector is called "cont". By knowing the dimensions of each primary rout (cont) the agent can track the secondary routes 'looking' for it in the superior diagonal of the IM. The last one for each one of the nodes that the primary route contains. Extending the first secondary routes nearer to the feeding and leaving in delay the rest of the routes in the nearest nodes to the load point does this. For this, the MPR in inverse form is crossed. For example, for the load point Lp1 the first secondary routes that are originated in node 1, then those of node 2, 7, and 8 are extended, to arrive to node 16 which corresponds to the node that contains the load point. The search of these routes is similar to which is made in the primary routes, but with some restrictions.

One of them is that the agent must finish the search at the moment a fuse is find and must begin to extend the following one that is in delay. As secondary routes are subdivided in different branches, it is

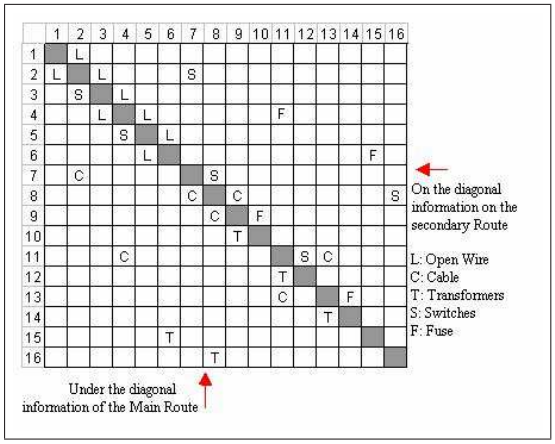


Figure 3: Intelligence Matrix corresponding to BS

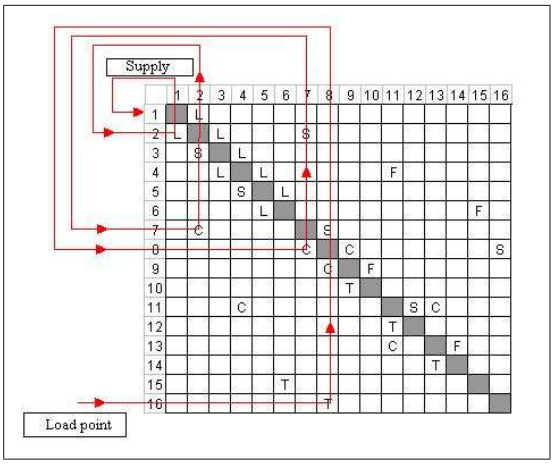


Figure 4: Searching the primary route for Lp1 (node 16)

necessary to store the nodes that at the moment of the distance traveled (on the diagonal of the matrix) will have to wait until the previous one in its deeper level extends (point of neighboring or fusible load). For this, it is strictly necessary to store four data of the node to expand:

Lp1 →	16	8	7	2	1	0	0	0	...	0
Lp2 →	10	9	8	7	2	1	0	0	...	0
Lp3 →	15	6	5	4	3	2	1	0	...	0
Lp4 →	12	11	4	3	2	1	0	0	...	0
Lp5 →	14	13	11	4	3	2	1	0	...	0

Figure 5: Matrix of primary route (MPR) for the BS

- The preceding node is required in case that there is a switching or protection device in the arrival of the node. These elements can involve the end of the route searching or a change in the repairing time to switching time.
- The node to expand, is the node that indicates in which row of the intelligence matrix, to look for the elements that form the secondary route and if there are more elements that will be needed to expand more ahead.
- The node towards which the agent travels, gives the direction in which the search takes place.
- Condition in which it is found, considering the time of repairing or switching, essential condition for the calculus of the unavailability.

These data are stored in a matrix of 4 rows and a undetermined number of columns. It is filled during the route and it is an uncertain number at first. This matrix has the name of "extend" and determines the end of the process of searching as the last element in delay expands. The simplified agent to find the secondary routes is summarized in figure 7. This agent is executed for each primary route, i.e., for each load point it finds all the corresponding secondary routes. It is also observed that it conditions on switching time the rest of the route in case of finding a disconnects (S) and ends the route when finding a fuse (F). The end of this agent corresponds to the extension of the last secondary route (on the primary route) and that there aren't any nodes in delay waiting to be extended.

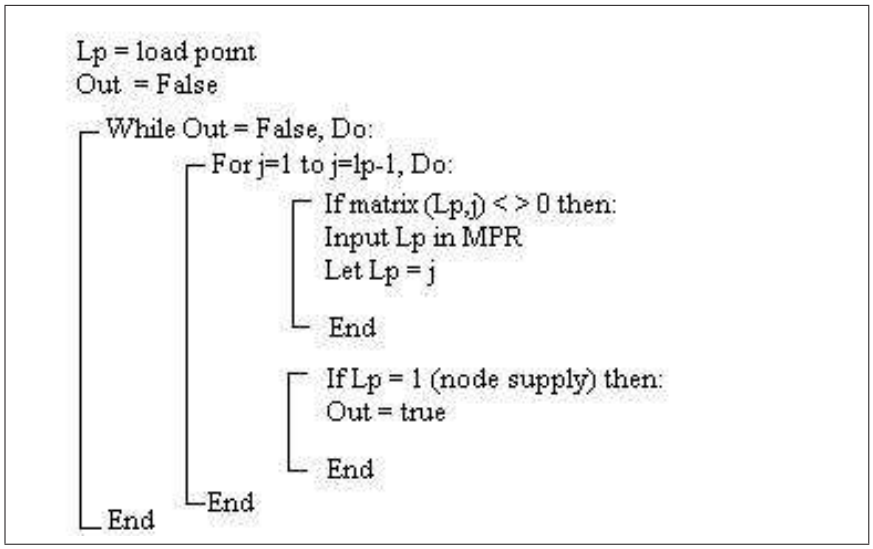


Figure 6: Simplified agent to find the primary route

2.3 Rate of failure and Unavailability

For a load point it is possible to calculate its rate of failure and its unavailability by adding those that correspond to primary or secondary routes. The time of shut down is the quotient between the total unavailability of the load point and its corresponding rate of failure.

$$Rate\ of\ failure:\ \lambda(lp) = \lambda_{pr}(lp) + \lambda_{sr}(lp) \tag{1}$$

$$Unavailability:\ U(lp) = U_{pr}(lp) + U_{sr}(lp) \tag{2}$$

$$Outage\ time:\ r(lp) = \frac{U(lp)}{\lambda(lp)} \tag{3}$$

In the primary route: the information stored is checked in the MPR. A rate of failure is assigned to each element. This information has been entered in a routine of parallel data entrance, to the IM. The sum of the rates individual failures provide the rate of failure of the primary route for the load point $\lambda_{pr}(lp)$. The unavailability of the primary route is the sum of individual unavailability of the elements involved in it, without considering the presence of the switching devices.

In the secondary routes: while the search is made the rates of individual failures of the elements in this route are added. At the same time, the unavailability of the secondary routes is being calculated. At the beginning, the unavailability is related to the repair time of the element, when a switching device is found the unavailability is associated to the switching time and this is kept from this node to the end of the corresponding secondary route. As long as one advances in the search the individual unavailability's are added. At the end of the process the results are the unavailability, $U_{sr}(lp)$ and the rate of failure $\lambda_{sr}(lp)$ of the secondary routes by load point. The way in which the algorithm is worked is shown in figure 9 (getting the mode of failure for the Lp1). With a different color the point where the disconnects is indicated, which conditions to use the time of switching for the rest of the route.

2.4 Considering the Alternative Feeding

A system with alternative feeding diminishes the times of unavailability to the customer. Before making any calculus, it necessary to know where the alternative feeding is. For the previous, the *"Inter-*

feeding route" that joins the main feeding with the alternative feeding is defined. The corresponding agent is similar to the *"resolutor agent"* that finds the primary route (under the diagonal of the matrix). But instead of beginning at the load point (Lp), it starts its nodal search where the alternative feeding is. This route is stored in a vector called *"ralt"*. In the first place, the inter-feeding route is located. Then, it is compared with the primary routes for each load point settling down the intersection point among them. Then, a pursuit between the main feeding and the intersection point is done, where the agent has to realize if there is a switching device in this sector. If not in this route, a calculus is made in a traditional way, without affecting the alternative feeding. If there is a switching device between the main feeding and the intersection point of the routes, this conditions all the elements that are found between the switching device and the main feeding, in case of a possible fault to only consider the time of switching, since this one is isolated, connecting the alternative supplies. This lightens the search, since the agent leaves the search of other switching devices in that sector to assign times. The rest of the system is analyzed as if there is no alternative feeding. Considering the BS with alternative feeding in the node 11, nodes-1 - 2 - 3 - 4 - 11 form the inter-feeding route for that configuration. For the LP3 (node 15), the intersection of the primary routes, the Inter-feeding and the switching device in [3,2] are shown in figure 11 The existing switching device conditions a great part of the system to a time of switching. This same example is considered, in terms of evaluating the Lp1 (node 16). The primary route of this load point only intersects in node 2 with the inter-feeding route and there is no switching device between node 2 and the main feeding, in which the alternative supply does not influence the index calculus.

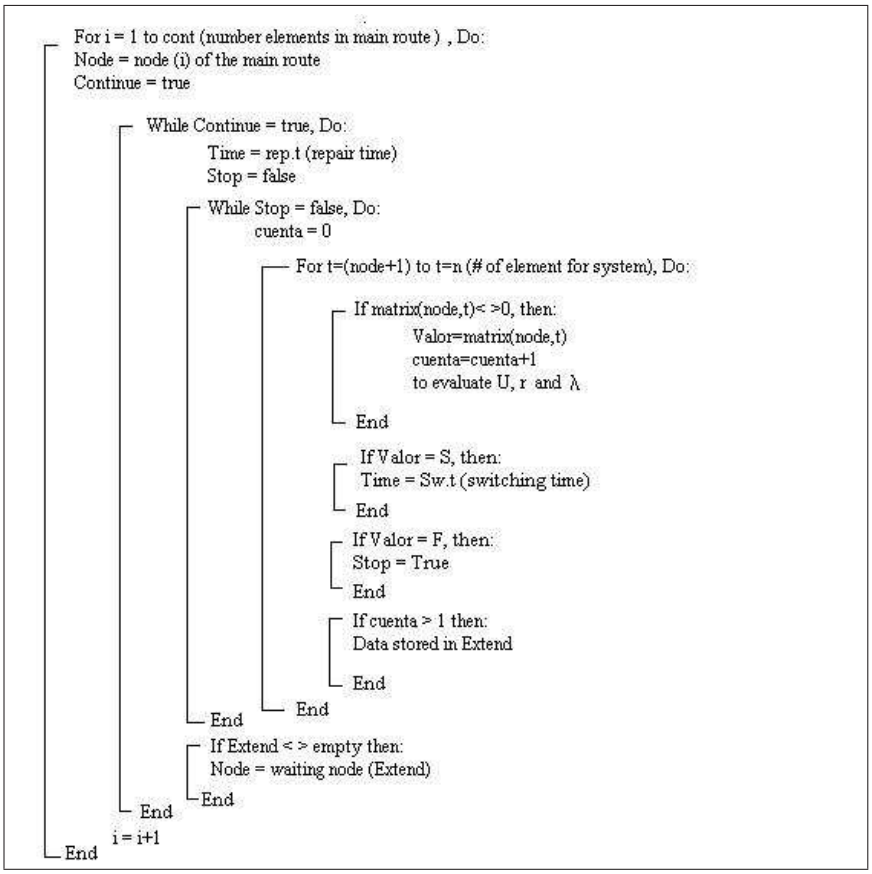


Figure 7: Simplified agent finding the secondary routes

3 Applications

The algorithm was applied to several distribution systems. In this document 4 of them are presented: the basic system (BS), two RBTS feeders (Billinton and Kumar, 1989; Allan et al., 1991) and a real system.

3.1 Basic system

This system was created to replace the necessity of having a small distribution system (DS) that had the complexity and diversity of a real one. The system has 5 load points and a primary feeding in node 1. The elements corresponding to a conventional DS are also present, such as: line, cables, fuses, transformers, switching devices, and so on. The modes of failure, the times of switching and repairing used correspond to those indicated in the IEEE St. 493 (1990). The results are indicated in Table 1.

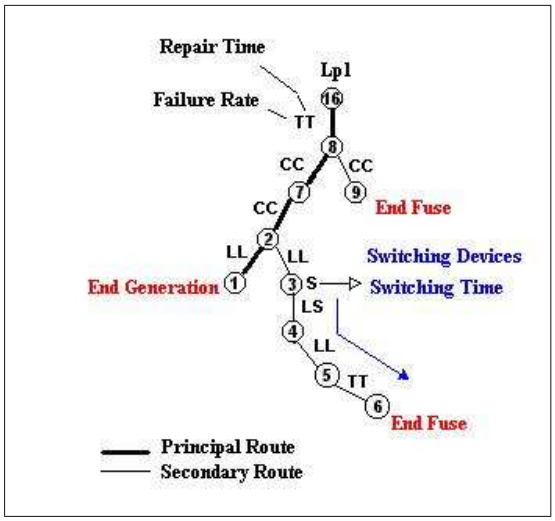


Figure 8: Mode of failure Lp1 (node 16)

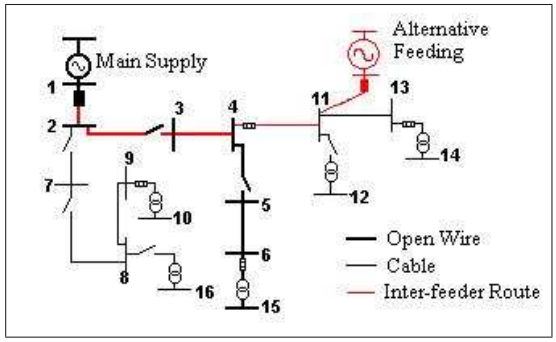


Figure 9: Inter-feeding Route. Alternative supplies in 11

3.2 RBTS System (Roy Billinton Test System)

This is a test system that is divided en 6 buses. Bar 1 is the feeding of the system, whereas bus 2 to 6 correspond to the load buses. The whole system shows different level of tension which it works 220, 138, 33, and 11 kV. With the purpose of proving the algorithm, the reliability of the feeder 1 of buses 2 and 4 are evaluated, corresponding to a level of 11 kV. Table 2 gives the result of the reliability indexes.

3.3 The CGE's Feeder

This feeder is inserted in the CGE's distribution system. This has 70 elements and 33 load points. For its evaluation the same considerations of the experimental feeder are applied. Three evaluations to the system were made. Case 1: Future Reliability Evaluation. Case 2: Future Reliability Evaluation considering the Alternative Feeding in node 27. Case 3: Future Reliability Evaluation considering the Alternative Feeding in node 55. Table 3, show the results of the reliability evaluation for the three different cases given.

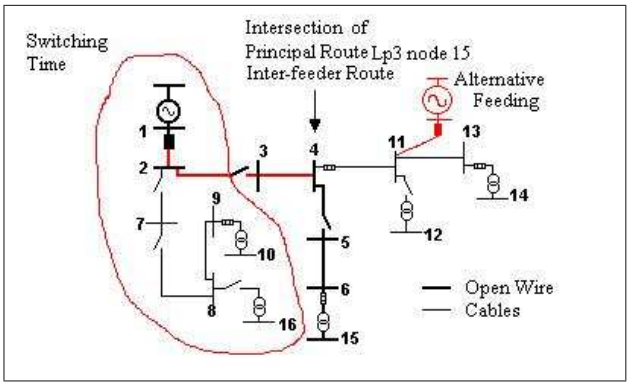


Figure 10: Switching zone and Lp3-alternative supplies

			Alt Alt	Node 5 Node 11
NERC indexes				
SAIFI		0.3963	0.3963	0.3963
SAIDI		4.4847	4.0171	4.1588
CAIDI		11.3159	10.1359	10.4935
ENS	MWh/year	21.396	19.102	19.759
CIER indexes				
F		0.3964	0.3964	0.3964
T		4.4576	3.9616	4.1166
D		11.2442	9.9931	10.3841
END		21.396	19.102	19.759

(CIER: Regional electric inter-american council)
(NERC: National electric reliability council)

Table 1: Reliability indexes of the basic system

4 Analysis of Results

The index of future reliability for each system as the NERC presents was obtained, those that are associated to the client. The index corresponding to the CIER; that referring to the power (kVA) were calculated. These last ones are of great importance due to the fact that these are the authorized by the CNE (national energy council) in Chile through the General Law's Regulation of the Electric Service (1980). Table 1 shows the difference between the alternative feeding in some of the BS's nodes in respect to the same system with only a main feeding. An important improvement is observed in the reliability's index when considering the alternative feeding. If the analysis is centered in one of the index as the average time of fault is, when applying the alternative feeding in node 5 a diminish in time of 11%, can be proved. By applying the same alternative feeding in point 11 the diminishing time of fault was around 7.6%. This is justified whereas node 5 is in the biggest point load of the feeder and there is a switching device immediately before the connection of the new feeding. The RBTS system was useful in confirming the good behavior of the proposed algorithm. This system, though being small, was very easy to follow with a hand calculator, in a addition to the complete results base already given in the Brown et al., (1997). Referring to the results (Table 2) the most important is to see the effect that is produced in the reliability the connection to the client from the main one with a disconnects (bus 2) or with two disconnects isolating the client in both sides (bus 4). A this difference is only remarked by the fact that both bus's feeders consider an alternative feeding. With this specific fact and in similar conditions the

	RBTS 2.1	RBTS 4.1
NERC indexes		
SAIFI	0.2488	0.3021
SAIDI	3.6193	3.4694
CAIDI	14.5443	11.4849
ENS	13.155	12.196
CIER indexes		
F	0.2477	0.3031
T	3.6090	3.4746
D	14.5696	11.4630
END	13.155	12.196

Table 2: Reliability indexes of the RBTS systems

best reliability is obtained in the bus 4's feeder. This is clear since with the possibility of having an alternative feeding from both ends, the most advisable it is to maintain the client with the switching. This means that before a possible fault, the client can isolate himself of the extreme in which the fault is produced and can keep connected with the other end in which the feeding remains, diminishing therefore the times of unavailability and improving the reliability of the system. For the CGE'S feeder just like the previous cases, got good results being within the established limits for F and T (Ministry of Mining, 1980). In Table 3 it is possible to see that the unavailability index decreases when considering alternative feeding. For the alternative feeding en node 27, the algorithm obtained a decrease of 15.6% en the unavailability. In as much as if the alternative feeding is in node 55, the decrease in the unavailability is of 24,5%. The differences between values is directly related with the amount of the existing switching devices. All results were checked with theoretical or manual analysis to assure the correct execution of the developed algorithm.

	Case 1	Case 2	Case 3
NERC indexes			
SAIFI	0.3222	0.3222	0.3222
SAIDI	2.8079	2.3575	2.1948
CAIDI	8.7139	7.3162	6.8112
ENS	12.410	10.470	9.3700
CIER indexes			
F	0.3198	0.3198	0.3198
T	3.2547	2.7458	2.4573
D	10.176	8.5849	7.6830
END	12.410	10.470	9.3700

Table 3: Reliability indexes of CGE System

5 Conclusions

This paper deals with the problem of the future reliability of a distribution system through AI. The solution considers the process of "searching" placed in the field of the AI. The approach presented in this paper is a powerful tool in distribution system reliability prediction. Its kindness was demonstrated in many evaluations that were made in systems with completely different characteristics. As it was expected, the results obtained were consequent with the theoretical and practical considerations of the

problem. The model proposes the concepts of "Intelligence Matrix" and "Agent". A very remarkable aspect of the conjunction matrix-agent is the facility with which it deals with the elements of protection and switching devices to value the importance of the strategic location of this elements. Furthermore the "Intelligence Matrix" gathers a condition so that the "Agent" works in an efficient way within the topological search. His connections make the run in an efficient and rapid way to complete the layout of the routes that involves the distribution reliability's calculus. This point is the clue of success in the search tree-failure modes.

A second significant contribution corresponds to the route concept "Inter-feeding" helping to harness the model in which it says to be related with the option of alternative feeding. With the help of this a decrease in the unnecessary expenses of time and computer memory was obtained. This, as well, becomes a potential tool of evaluation of the ideal positioning of an alternative point of feeding in a real system of distribution.

From a more general perspective, we can conclude that the use of this model prevents important economical measures, in which the electric companies could commit or incur when not having a suitable control.

Finally, the investigation's development resulted in the necessity to deepen in ordaining the switching and protection devices, that can lead to obtain the best reliability of the system.

References

[1] Asgarpoor S. y M. Mathine, Reliability Evaluation of Distribution Systems with Non-Exponential Down Times. *IEEE Transactions on Power Systems*. Vol. 12, No. 2, 1997.

[2] Allan R., R. Billinton, I. Sjarief, L. Goel and K. S. So, A Reliability Test System for Educational Purposes - Basic Distribution System Data and Results, *IEEE Transactions on Power system*, Vol. 6, No. 2, 1991.

[3] Billinton, R. y Allan, R., Reliability Assessment Of a Large Electric Power Systems. *Kluwer Academic Publishers*, Boston, USA, 1988.

[4] Billinton R y S. Kumar, A Reliability Test System for Educational Purposes - *Basic Data IEEE Transactions on Power Systems*, Vol. 4, No. 3, 1989.

[5] Billinton R., y S. Jonnavithula, A Test System for Teaching Overall Power System Reliability Assessment, *IEEE Transactions on Power Systems*, Vol. 11, No. 4, 1996.

[6] Billinton R., y P. Wang, Teaching Distribution System Reliability Evaluation Using Monte Carlo Simulation, *IEEE Transactions on Power Systems*, Vol.14, No. 2, 1999.

[7] Brown R., S. Gupta, R. D. Christie, S. S. Venkata y R. Fletcher, Distribution System Reliability Assessment Using Hierarchical Markov Modeling, *IEEE Transactions on Power Delivery*, Vol. 11, No. 4, 1996.

[8] Brown R., S. Gupta, R. D. Christie, S. S. Venkata y R. Fletcher, Distribution System Reliability Assessment Momentary Interruptions and Storms, *IEEE Transaction on Power Delivery*, Vol. 12, No. 4, 1997.

[9] Brown R., y J. Ochoa, Distribution System Reliability: Default Data and Model Validation, *IEEE Transactions on Power Systems*, Vol. 13 No. 2, 1998.

[10] IEEE St 493, *IEEE Recommended Practice for the Design of Reliable Industrial and Commercial Power Systems*, 1990.

[11] López E., Tardon C., Contreras V., Reliability evaluation of primary distribution systems via search tree, *Procc. of International Congress of ACCA/IFAC Automatic Control Association 2000*, Santiago of Chile, 1998.

[12] Ministry of Mining, *General Law of Electric Services. Official Newspaper of the Republic of Chile*, 1998.

[13] Russell S y P. Norvig, *Artificial Intelligence. A Modern Approach*, Editorial Prentice Hall & Hispanoamericana S.A., 1996.

E. López, J. Campos, C. Tardon, F. Salgado,
University of Concepción, Chile
Department of Electrical Engineering
E-mail: elopez@die.udec.cl

J. Tardon
Sts-Saesa-Frontel
E-mail: jtardon@saesa.cl

R. López
Supelec. LEP-University of Paris XI.
E-mail: rodrigo.lopez@supelec.fr

International Journal of Computers, Communications & Control
Vol. I (2006), No. 2, pp. 73-83

Inteligent Line Follower Mini-Robot System

Román Osorio C., José A. Romero, Mario Peña C., Ismael López-Juárez

Abstract: This paper shows a prototype development of an intelligent line follower mini-robot system, the objective is to recognize, understand and modify the actual performance of the movements of the robot during its pathway by way of getting information in real time from different magnetic sensors implemented in the system and based in a V2X digital compass, microcontroller and odometric measurements. The paper shows as well, the system characterization of the V2X sensor (digital compas) and the cost-benefit of the prototype implementation and performance. The programming techniques and easy operation is detailed too.
Keywords: autonomous, mobil-robot, inteligent system

1 Introduction

It is shown in the article the design and implementation of an "autonomous intelligent line follower navigation system" which has been designed in the DISCA-SEA IIMAS-UNAM, it is used an inductive-magnetic based sensor to measure magnetic fields which shows the orientation and position of the pathway of the mobil-robot to get real-time navigation information. This efficient potential tool has been used since the 90's in industrial applications with a lot of success because of its good performance, small size and compatibility with different electronic systems. The follower system (FS) uses a continuous black line on a white background in its pathway during its travel trajectory, the line has to be 18 millimeters width in order to have an acceptable functioning. To analyze the inductive sensor, test experimentation was done within a roadway, capturing data where possible conflicts could be appears during the test, this data was used as inputs to the algorithm to achieve corrections on the decision speed and control movements. A microcontroller assigns the speeds to optimize the performance according to time movements in order to get a closer behavior as humans could do it without a vision system to catch its environment. The task of this work has been to study the V2X compass real performance to develop the navigation algorithms and implement them in a AT90S1200 Atmel microcontroller.

2 System Development

The elements involved during the physical design are:

2.1 Transmission

The mobil prototype was implemented with a transmission system conformed with two rear wheels, coupled to two direct current motors, each wheel however is independent functioning for gears and reduction box system.

In front of the prototype there is a "crazy wheel" system making possible to achieve free turns up to 360 degrees. Wheels are made of heavy duty plastic with a rugged polymer plastic made friction-avoid central line, this implementation allows heavy weight carry performance (between 0.8 and 1.5 kilograms) with minimal current consumption (typically 300 milliamps). Allowed movements are:

- Forward direction
- Backward direction

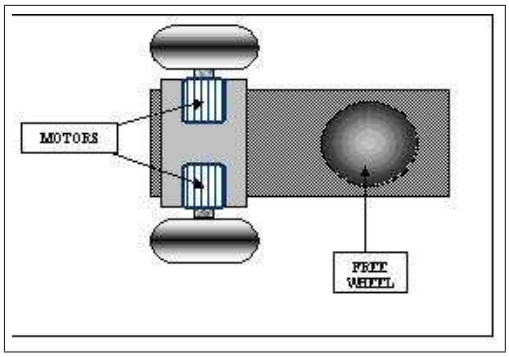


Figure 1: Transmission system and wheels location

- Left and Right turns
- 360 degrees rotation

2.2 Direct current motor control

Control for the two motors in the system is carried out by using the L293D integrated circuit H bridge, a microcontroller enable and disable the motor excitation elements using the internal H bridges in the circuit. The easy operation of the L293D circuit and considering the only one movement in a single direction, it it possible to us the following operation movements codification as shows in following table:

MOTOR 1		MOTOR 2		ACTION
IN 1	IN 2	IN 3	IN 4	
0	0	0	0	High "Z"
1	0	1	0	Rotate right
0	1	0	1	Rotate left

Table 1: L293D operation

And because of the direct control using pulse width modulation (PWM) to on/off and speed control for motors, we use the table 2 input/output information to enable the motors, as it is showed next:

INPUT	PIN ENABLE	OUT
1	1	1
0	1	0
1	0	High "Z"
0	0	High "Z"

Table 2: Motors enable input/output signal sequence

In order to get microcontrollers electrical protection as short circuit and high energy current peaks because of continuous motor switching activity, an isolated purpose interface was implemented between micrcontrollers and L293D circuit by using non-inverted buffers (74HC4050 high speed CMOS circuits) as it is showed in figure 2.

2.3 Line Follower Prototype Architecture

Figure 3 shows the line follower architecture, where microcontroller, sensors and motors interaction can be observed.

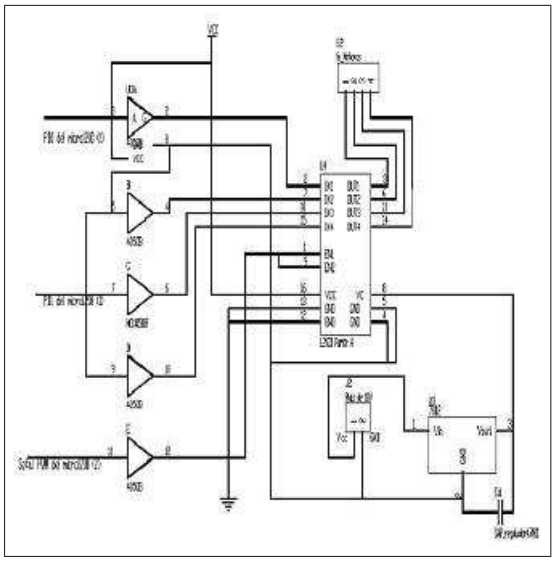


Figure 2: Motor isolated interface

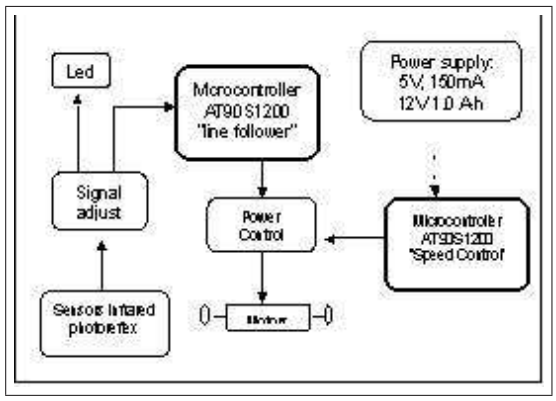


Figure 3: Prototype Architecture

2.4 Sensors Distribution

Sensor distribution within the system is showed in figures 4 and 5.

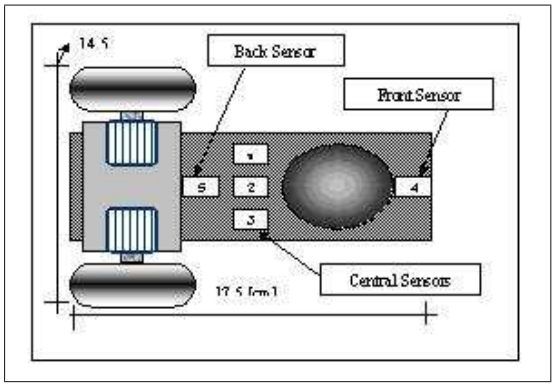


Figure 4: Sensors location within the prototype

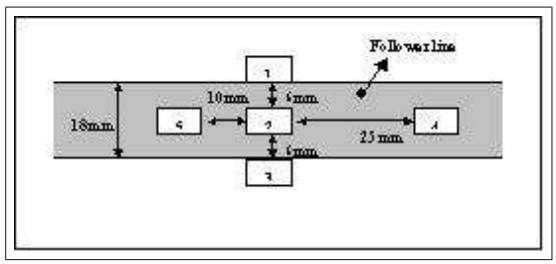


Figure 5: Distance distribution of sensors within the layout platform prototype

2.5 Line following programming description

To implement this section of prototype, a microcontroller AT90S1200 was used, because of its simplicity for programming purposes, the program achive five sensor monitoring and direct current motor control, using the available 32 registers and 2 communication ports of the chip. The microcontroller is good situated for such a task, even still can be expanded if a more complex implementation is required for some other application. First approximation in the design, showed the possibility of using only 3 central sensors for line follower control, but different experiments showed there was a problem when very close curves where in the robot pathway, so adding some more sensors fixed this problem and the line follower system showed good performance for all pathways in the experiments, methodology analysis for sensors and prototype reaction was done as it is showed in next section.

Line drawing analysis

Microcontroller achieves following analysis criteria:

- When two side-central lateral sensors (1 and 3) are out of pathway line and central sensor is on pathway line, the decision is the prototype is following a straight line drawing, then the 2 motors have to be activated instantly. (Figure 4 and 5).
- If prototype is following a line and the front sensor is not ON, there is no curve line and two motors must be still ON.
- If a curve line has not been detected but the right sensor (3) is within the line, the microcontroler activates the left motor to turn left until the signal sensor indicates is already out of the line, then the two motors are activated again at the same time to go fordward.
- If a curve line has not been detected but the left sensor (1) is within the line, the microcontroler activates the right motor to turn right until the signal sensor indicates is already out of the line, then the two motors are activated again at the same time to go fordward.
- When a curve line is detected (fornt sensor signal is activated), the operation process jumps to the algorithm for curve lines reaction.

Curve line analysis

When the front sensor in the prototype finds a curve line (front sensor 4 is ON) the microcontroller achieves next analysis:

- If central and side sensors are out of the line, and the middle sensor is within the line, the microcontroller will activate the right motor until the central right sensor is within the line, this action is named "zigzagder". When right sensor is already within the line, it jumps immediately to "zigzagizq" action.
- ZIGZAGIZQ: if the middle and right central sensors are within the line (left sensor does not care), the microcontroller must activate the left motor until the right sensor is out of the line drawing. When the action gets the right sensor out of the line, the operation jumps to the "zigzagder" action.
- ZIGZAGDER: the middle central sensor is within the line and the right sensor is out of the line, the microcontroller will activate the right motor until the right sensor is within the line. When this is done it jumps immediately to he "zigzagizq" action.

While a curve line is been followed, the front sensor is out of the line (figures 6B,6C and 6D) and the rear sensor is within the line in about first half of the curve line pathway,(figures 6D and 6E). This action guarantees the prototype been in a straight line again, this action is graphically explained in figure 6.

While mentioned actions "zigzagder" and "zigzagizq" are taken placed, at the same time the front sensor signal state is been continuosly monitored, the inf the fornt sensor is OFF (logical "0"), the algorithm jumps to verify the rear signal sensor, and having next two possible decisions:

1. If it is ON (logical "1") implies curve line still does not finish and then "zigzagder" and "zigzagizq" actions must still been taken place until rear sensor is OFF and then the straight line operation is guaranteed and takes place.
2. If it is OFF, implies the sensors are aligned and "zigzagder" and "zigzagizq" actions can still be operating and go to straight line operation mode.

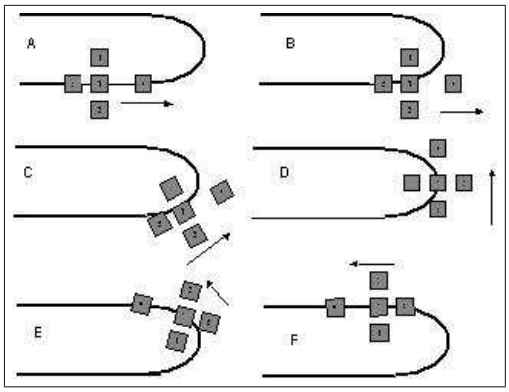


Figure 6: Curve line behavior of prototype sensors

Speed Control

This module in the system architecture allows to have proportional signals to achieve pulse width modulation control for motor speed. The input signal to change width pulse is acquired by way of a dip-switch and a microcontroller to enable and disable the H bridge in the L293D integrated circuit mentioned before.

This stage uses an Atmel AVR AT90S1200 microcontroller to achieve pulse width modulation, this chip was used because of its "cost-benefit" advantages as small size and cost, as well as, easy programming development facilities. Using a microcontroler gives the system flexibility and user capabilities to change frequency operation and optimize power consumption and performance of motors. Most microcontrollers are restricted to 4Khz. frequency to be used for PWM reducing motors performance.

One problem with microcontroller operation used to capture and process information, is the clock cycle lossed signaling as a result of speed changes, this generates a delay operation in the infrared sensors and magnetic-inductive sensor data acquisition signaling.

From experimental results, using different pathways and circuits, and by slope and curve analysis of line drawings used as the prototype road, 4 output driving signals were choosed to speed control of the mobil system. This signals are shown in figures 7, 8 and 9, in figures can be observed a 11.0 Khz. carrier frequency for PWM and ON/OFF timing.

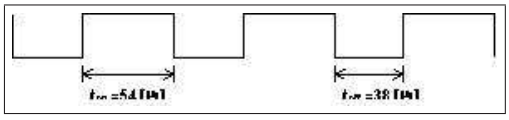


Figure 7: Speed 1 signal representation

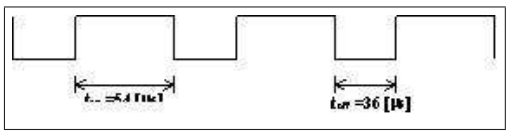


Figure 8: Speed 2 signal representation

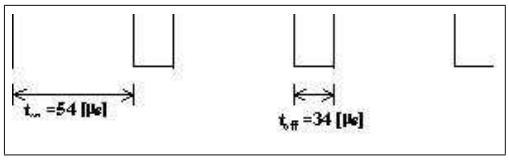


Figure 9: Speed 3 signal representation

2.6 V2X sensor implementation with mobil prototype

Capture and processing information

To capture data from line drawings, the V2X magnetic-inductive sensor was used. In the application an slave mode operation and a binary output format were used in order to make arithmetic operations easier for microcontroller programming, having directly angular data output within the range of 0 to 168ř in hexadecimal numerical representation. In this stage of system architecture, an AT90S8515 microcontroller was used to achieve following operations:

- Infrared sensor monitoring
- Data acquisition for V2X digital compass
- Data analysis to validation process
- Continuous turn recognition
- Data storage
- Data acquisition and comparative processing for continuous turn case operation.
- Data interpretation from mass storage.

V2X acquisition and interpretation programming method

First, data capture is taken place with troubled sections on road pathway, in order of having a best criteria for easier pathways sections as straight and small curve lines in a "learning and acknowledge" route in the pathway road. As it has mentioned before, to detect a line curve first the front infrared sensor numbered 4 (figure 4) is used, if the sensor is out of the line, the microcontroller capture an orientation data from digital compass and storage the information in flash memory. Every time this action takes place the microntroller actually capture two consecutive orientation data to get an acceptable data range to guarantee the data is correct (2 consecutive data has to be almost the same value) because the V2X sensor sometimes generates not-correct data because of prototype vibrations in the road pathway. Typical data from sensor in technical manual information is $\pm 1^{\circ}$, but for the present design $\pm 3^{\circ}$ were considered to achieve a robust acquisition.

After orientation of first point-set within the curve line has been detected, it has to be detected when the line finishes, this is done by using the rear sensor (5) (figure 4) as follows:

Because of the front sensor is out of the curve line (it gets a logical "1"), the microcontroller expects the infrared sensor number 4 gets inside the line again, when this is done, inmediatly the rear sensor (5) is monitored and following decisions are taken:

- Sensor is ON, prototype still inside the curve line.
- Sensor is OFF, it is guaranteed that prototype is aligned and curve line has finished.

Once the curve line is finished, the microcontroller acquires an orientation data and storage it in flash memory.

Orientation data storage method

To easy analysis and interpretation of captured data, 3 index registers "X,Y and Z" were used as follows:

1. Data list for orientation to indicate the beginning and end of curve lines, is stored using the X index register within the memory range $62 to $100 addresses.

2. Address $60 is defined to storage the orientation data for the very beginning of the prototype pathway.

3. For every orientation data to indicate the end of every curve line, data is stored in memory beginning in address $102.

4. In register "z" the "curve characterization" is stored, this data tells the system the size of the curve line (time duration) and is memorized beginning in address $1A4.

The method is graphically illustrated in figure 10, when generated and stored data structure is showed.

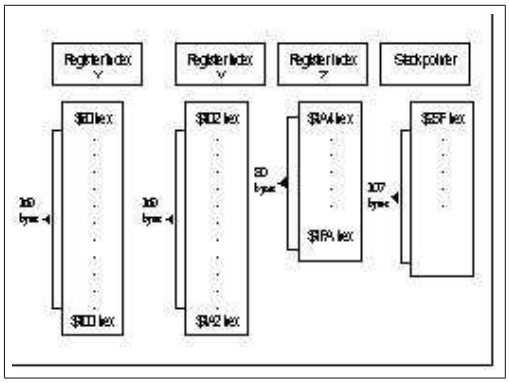


Figure 10: Generated data structure in Microcontroller

The used AT90S8515 microcontroller can storage 512 bytes of data memory, as it is shown in the figure the memory was sectioned in 4 parts to optimize data storage and access of information. Registers X and Y can storage 160 bytes, every curve is codified with 2 bytes so 60 curve lines can be stored which makes the system available to almost any type of curve lines within a line drawing road pathway real operation and with acceptable speed movements. In the other hand, register Z only storage 1 byte data which lets only 107 free bytes for stack pointer operation beginning in memory address $25F.

Data Interpretation method

For curve characterization, the magnitude difference of initial and final orientation values are used (Z_1), if X_1 is the initial orientation value and Y_1 the final value:

$$Z_1 = |X_1 - Y_1|$$

Z_1 is the curve representation in degrees with a correspondence speed value, this action can be seen in figure 11.

To evaluate and relate the real behavior of the mobil with this values, experimentation was done with different curves types to approximate the real operation of the system. Three experimental speeds were used with different pathways curves and straight lines and the following results are showed in figure 12 and 12 for two different curves.

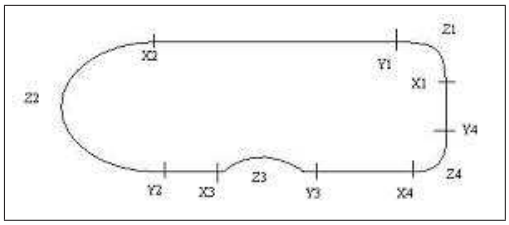
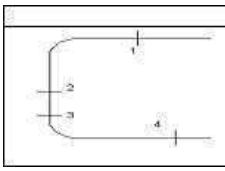


Figure 11: Orientation values X and Y, and difference value Z

3 Results

Curve case studies and speed reaction.

Figures from 12 to 17, show the different curve types case studies, to classify the difficult degree of road line drawing pathway, and so the times and behavior of the prototype in its travelling operation.



CURVE 1

Measurement (V2X)	Portion curve			
	1 +	2 *	3 +	4 *
Stable	341°	66°	75°	158°
Mobil	343°	56, 57°	65, 66°	158°

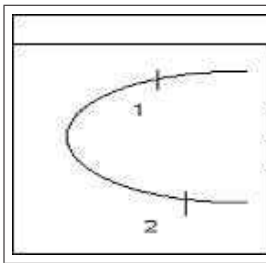
\+ Sensor ON front (begin curve)
* Sensor OFF front (end curve)

Figure 12: Curve 1 case

Curve case 1 with speed 1:
T_{14}= 4.29-5.44 [seg]
Curve case 1 with speed 2:
T_{14}=3.35-4.07[seg]

Times for travelling through pathway:
For this speeds, the sensor 2 is going inside the line, while sensors 1 and 3 are going out of the line, considering this values adequate for the curve type used.
Curve case 1 with speed 3:
T_{14}=3.18-3.45[seg]

This velocity shows a small time operation, however the mobil gets an unstable behavior because of is trying to follow the best way the curve line, and fast head movements are carried out in the mobil prototype (shaking head movements).



CURVE 2

Measurement (V2X)		
	1 +	2 *
Stable	340 °	170°
Mobil	344°	176, 177°

\+ Sensor ON front (begin curve)
* Sensor OFF front (end curve)

Figure 13: Curve 2, case study

Time of travelling pathway:
with speed 1:

t12=3.75-4.31 [seg]

with speed 2:
t12=2.75-3.31 [seg]

With this speeds, sensor 2 is going inside the line and sensor 1 and 3 are going out the line.
With speed 3:
t12=2.31-2.80 [seg]

with this speed the mobil makes shaking head movements 3 times before reaches part 2 of the curve, and never gets a stable state.

Figures 14 show results for 10 different curves and conclusions about decision on its use operation.

CURVE 1

Section Curve	Measure	Data to compare	Analysis	Range [°]	Speed assign
Section 1 at 2	Stable	341° –> 66°	360° - 341° = 19° 19° + 66° = 85°	73 - 74°	The suitable speed for this curve is the 2, for times and way to travel.
	Mobil	343° –> 56°	360° - 343° = 17° 17° + 56° = 73°		
		343° –> 57°	360° - 343° = 17° 17° + 57° = 74°		
Section 2 at 3	Stable	75° –> 66°	75° - 66° = 9°	8 - 10°	The suitable speed for this curve is the 3, because the range to assign is short
	Mobil	65° –> 56°	65° - 56° = 9°		
		65° –> 57°	65° - 57° = 8°		
		66° –> 56°	66° - 56° = 10°		
		66° –> 57°	66° - 57° = 9°		
Section 3 at 4	Stable	158° –> 75°	Speed assign 3	92 - 93°	The suitable speed for this curve is the 2, for times and way to travel to assign is short
	Mobil	158° –> 66°	158° - 66° = 92°		
		158° –> 65°	158° - 65° = 93°		

CURVE 2

Section Curve	Measure	Data to compare	Analysis	Range [°]	Speed assign
Section 1 to 2	Stable	340° –> 170°	360° - 340° = 20° 20° + 170° = 190°	192-193°	The suitable speed for this curve is the 1, for the very close curve (difficulty grade)
	Mobil	344° –> 176°	360° - 344° = 16° 16° + 176° = 192°		
		344° –> 177°	360° - 344° = 16° 16° + 177° = 193°		

Figure 14: Curve characterization format

4 Conclusions

Results and conclusions are showed in figure 16 and 15 for 10 different curves, fr values 0 to 17° speed number 3 was used, and the mobil makes an interpretation as to consider the pathway a straight line. For 18 to 109° speed used was number 2, within this range most curves are classified when experimental

results were done and the mobil prototype get fast movements without getting unstable behavior.

For values bigger than 110° experiments shows and agree to use speed number 1, because in this range the mobil make high degree difficult turns like a "U" turn, so the speed has to have a small value in order to get stable behavior.

Range [°]	Assign speed
0 - 17	speed 3
18 - 109	2
110 - 200	1

Figure 15: Speed assigned value ranges

Section Curve	Range [°]	Assign Speed
CURVE 1		
1 - 2	73 - 74°	speed 2
2 - 3	8 -10°	3
3 - 4	92 - 93°	2
CURVE 2		
1 - 2	192 - 193°	2
CURVE 3		
1 - 2	86 - 88°	2
CURVE 4		
1 - 2	167°	1
CURVE 5		
1 - 2	42 - 48°	2
2 - 3	46 -56°	2
CURVE 6		
1 - 2	81 -86°	2
2 - 3	9 -14°	3
3 - 4	81 - 92°	2
4 - 5	68 - 108°	2
CURVE 7		
1 - 2	35 - 42°	2
2 - 3	188 - 199°	1
3 - 4	23 -29°	2
CURVE 8		
1 - 2	17 - 25°	2
2 - 3	45 - 55°	2
3 - 4	26 - 39°	2
CURVE 9		
1 - 2	86 -90°	2
2 - 3	79 - 85°	2
3 - 4	11 - 20°	3
4 - 5	69 - 73°	2
CURVE 10		
1 - 2	81 - 89°	2

Figure 16: Ten different curves characterization

References

[1] Anderson H Peter and Foster Vance, “Interfacing with A Precision Navigation Vector 2X Compass Module”, *Dept of Electrical Engieering*, Morgan State University, http://www.phanderson.com/printer/compass/compass.htm.

[2] Cormen, C. E. Leiserson, R. L. Rivest, The MIT Press, McGraw-Hill, 1990.

[3] Osorio R, Romero J, Bautista A., “Estudio de un Mini Robot seguidor Autónomo Inteligente utilizando un compas V2X.”, Facultad de Ingeniería UNAM.

[4] M. de Berg, M. van Kreveld, M. Overmars and O.Schwarzkopf., *Introduction to Algorithms, Computational Geometry: Algorithms and Applications.*, Springer-Verlag, Berlin, 1997.

[5] K. Mehlhorn, *Data Structures and Algorithms*, Springer-Verlag, 1984.

[6] R. Sedgewick, *Algorithms*.

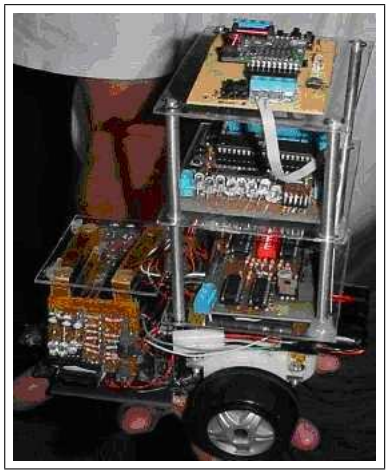
[7] Stefan Gotts, *Collision Queries using Oriented Bounding Boxes.* Tesis North Caroline University, U.S. 2000.

[8] Stolfi., *Oriented Projective Geometry: A Framework for Geometric Computations*, Academic Press, 1991.

[9] Thomas Möller et. al., *Fast, minimum storage Ray/Triangle Intersection.* Universidad Tecnológica de Chalmers.

[10] www.avrfreaks.com, *Microntroladores, Atmel*,10-septiembre-2001.

[11] www.atmel.com , septiembre-2001.



Román Osorio C., José A. Romero, Mario Peña C
Universidad Nacional Autónoma de México
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
Departamento de Ingeniería de Sistemas Computacionales y Automatización
Sec. Electrónica y Automatización Apdo. Postal 20-726, México D.F.
E-mail: roman@servidor.unam.mx

Mario Peña C
CIATEQ, A.C. Centro de Tecnología Avanzada
Grupo de Investigación en Mecatronica y Sistemas Inteligentes de Manufactura
Manantiales 23A. Parque Ind. B. Quintana, El Marques, Qro. CP 76256. MEXICO

International Journal of Computers, Communications & Control
Vol. I (2006), No. 2, pp. 85-94

Coordinated Control Of Mobile Robots Based On Artificial Vision

Carlos M. Soria, Ricardo Carelli, Rafael Kelly, Juan M. Ibarra Zannatha

Abstract: This work presents a control strategy for coordination of multiple robots based on artificial vision to measure the relative posture between them, in order to reach and maintain a specified formation. Given a leader robot that moves about an unknown trajectory with unknown velocity, a controller is designed to maintain the robots following the leader at a certain distance behind, by using visual information about the position of the leader robot. The control system is proved to be asymptotically stable at the equilibrium point, which corresponds to the accomplishment of the navigation objective. Experimental results with two robots, a leader and a follower, are included to show the performance of the vision-based control system.
Keywords: Mobile Robots, Coordinated Robots, Vision Based Control, Artificial Vision.

1 Introduction

During the last years, efforts have been made to give autonomy to single mobile robots by using different sensors, actuators and advanced control algorithms. This was mainly motivated by the necessity to develop complex tasks in an autonomous way, as demanded by service or production applications. Robots have thus become highly sophisticated systems. In some applications, a valid alternative (or even the mandatory solution) is the use of multiple simple robots which, operating in a coordinated way, can develop complex tasks ([1]; [2]; [7]). This alternative offers additional advantages, in terms of flexibility in operating the group of robots and failure tolerance due to redundancy in available mobile robots [6].

As the number of robots increases, the task of controlling the system becomes more complex. Control strategies can be classified as centralized and decentralized. In a centralized control system, all planning and control functions are developed in a single control unit. Each mobile robot has few simple sensors, the actuators and the communication system with the single control unit. Every motion and conflict between robots is solved by the control unit. Nevertheless, the system becomes vulnerable to any failure that may occur in the control unit. On the other hand, with the decentralized control approach, each robot is equipped with multiple sensors and a controller thus becoming capable of recognizing the environment and taking its own control actions ([3]; [5]). The function of the control center -if available- is to assign tasks to each robot and to govern the information flow in the system.

The coordinated control of robots allows a team of robots to perform missions not easy or viable to be achieved by a single robot. The guidance of a mobile robot requires its localization in the environment [4]. A precise localization is needed when multiple robots share a common environment. Odometric sensors, sonar sensors, gyros, laser and vision and its fusion are commonly used for robot localization and environment modeling. Vision sensors are increasingly applied because of its ever-growing capability to capture information.

This work proposes a strategy based on capturing vision information with single cameras mounted on the follower robots, to obtain the relative posture of a leader robot and to guide a specified robots formation. The controller design is then obtained using non linear control theory, and the overall control system is proved to be asymptotically stable. The paper is organized as follows. The concept of robot formation is briefly discussed in Section 2 and the robot kinematics model in Section 3. The visual measurement of the posture of a leader robot is presented in Section 4. The controller design and the stability analysis are given in Section 5. Some experimental results are discussed in Section 6. Finally, Section 7 presents some concluding comments.

2 Robots Formation

In this work the following formation strategy is considered. One of the robots is defined as the leader robot, and moves in an unknown trajectory to the other follower robots. The leader has a pattern mounted on its back, which is observed by the follower robots to obtain information about their relative posture to the leader. This information is used to control their position to reach the specified formation. The visual information is obtained by a looking ahead camera mounted on each follower robot. Figure 1 shows typical triangle and convoy configurations that can be defined for three robots by using this strategy.

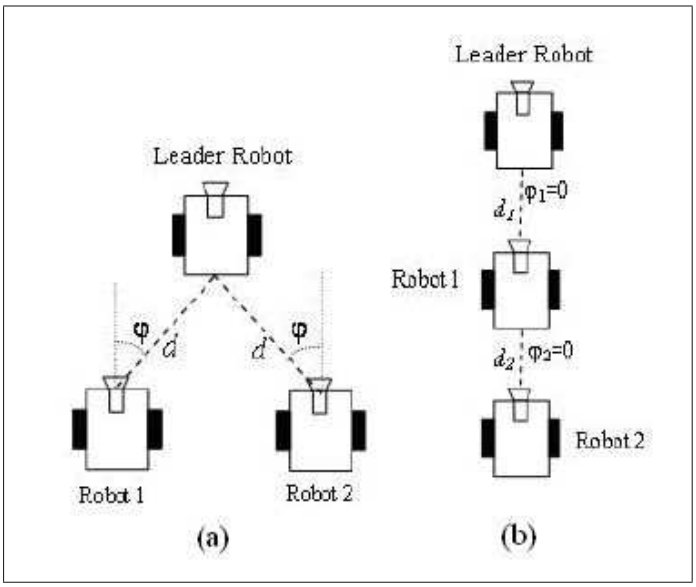


Figure 1: a) Triangle formation. b) Convoy formation.

3 Mobile Robot Model

In this work it is considered the unicycle-like mobile robot, with is described by the following kinematics equations:

$$\begin{aligned} \dot{x} &= v\cos\varphi \\ \dot{y} &= v\sin\varphi \\ \dot{\theta} &= \omega \end{aligned} \tag{1}$$

where (x,y) are the Cartesian coordinates of robot position, and φ the robot heading or orientation angle; v and ω are the robot's linear and angular velocities. The non-holonomic restriction for model (1) is

$$\dot{y}\cos\varphi - \dot{x}\sin\varphi = 0 \tag{2}$$

which specifies the tangent trajectory along any feasible trajectory for the robot. The reference point of the robot is assumed to be the middle point between the two driven wheels, Fig. 2. v_1 and v_2 denote linear speeds of the left and the right wheels, respectively. Linear and angular velocities of the robot can be expressed as $v = (v_1 + v_2)/2$ and $\omega = (v_1 - v_2)/L$ respectively, where L represents the distance between the two driven wheels.

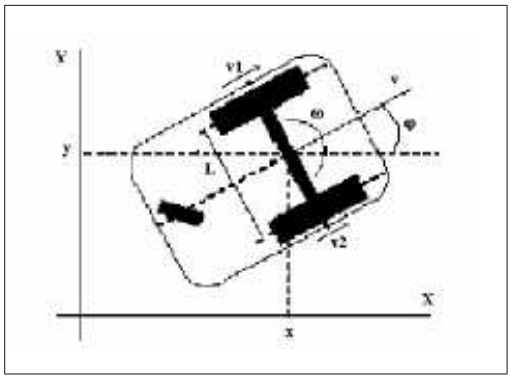


Figure 2: Geometric description of the mobile robot.

4 Visual Measurement of the Leader Posture

The control objective is that the follower robots follow the leading robot evolving with unknown motion in the working area. The robots have been equipped with a looking-ahead fixed vision camera. This camera captures the image of a pattern mounted on the leading vehicle that features four marks on a square of known side length $E[m]$. In order to ease the calculations, and without losing generality, the height of the horizontal median of this square is made to coincide with the height of the image's camera center. The projected pattern on the camera image will appear with a projection distortion as represented in Fig. 3. The positions of the pattern's marks on the image are expressed in pixels as (x_i, y_i) with $i = A, B, C$ and D. These variables are considered as the image features:

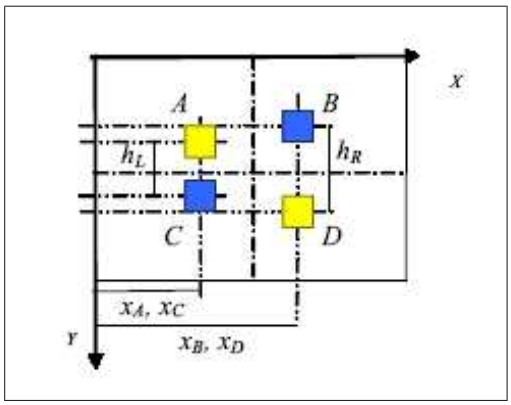


Figure 3: Image of the pattern's marks.

From these image features, as measured by the vision system, it is possible to compute the posture of the leading or target vehicle (x_T, z_T, φ_T), measured on a coordinate system associated to the camera (X_c, Z_c). Fig. 4 shows a diagram with the horizontal projection of the vision system, showing the posture of the leader or target vehicle on the camera's coordinates of a follower robot, from which the following expressions are obtained:

$$
\begin{aligned}
x_T &= \frac{x_R + x_L}{2} \\
z_T &= \frac{z_R + z_L}{2} \\
\varphi_T &= \cos^{-1}\left(\frac{x_R - x_L}{E}\right)
\end{aligned}
\tag{3}
$$

By resorting now to the model of reverse perspective, expressions are obtained to compute the leading vehicle's posture of (3) as a function of the measurements supplied by the vision system, Fig.4:

$$
\begin{aligned}
x_L &= \frac{z_L - f}{f} x_A, & x_R &= \frac{z_R - f}{f} x_B \\
z_L &= f\left(\frac{E}{h_L} + 1\right), & z_R &= f\left(\frac{E}{h_R} + 1\right)
\end{aligned}
\tag{4}
$$

Using the variables from (3) and (4), and assuming that the camera is mounted on the robot's center, the relative posture of the follower robot with respect to the leading robot (φ, θ, d) can then be calculated. The relative posture is defined as in Figs. 4 and 5. These variables are calculated by the following expressions:

$$
\begin{aligned}
\phi &= \tan^{-1}\left(\frac{z_R - z_L}{x_R - x_L}\right) \\
\varphi &= \tan -1\left(\frac{x_T}{z_T}\right) \\
\theta &= \varphi + \phi \\
d &= \sqrt{x_T^2 + z_T^2}
\end{aligned}
\tag{5}
$$

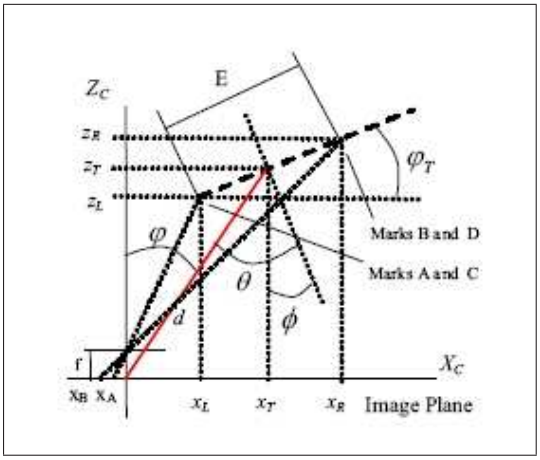


Figure 4: Location of the leader respecting the camera.

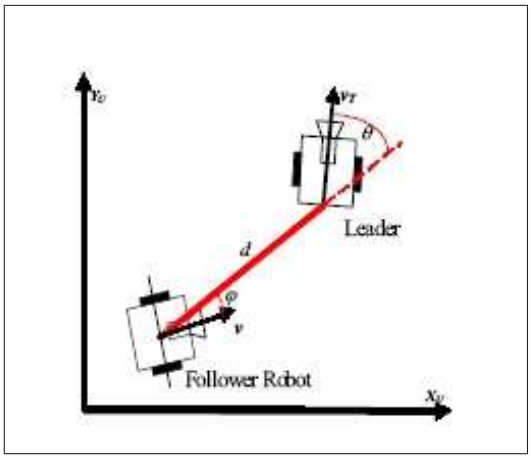


Figure 5: Relative position between the leader and a follower robot.

5 Servo Visual Control of the Mobile Robot

5.1 Controller

Figure 6 defines the control structure to be used in this work, where H represents the relationship between the variables defining the relative posture of the follower and the leading robot, and the image features ξ.

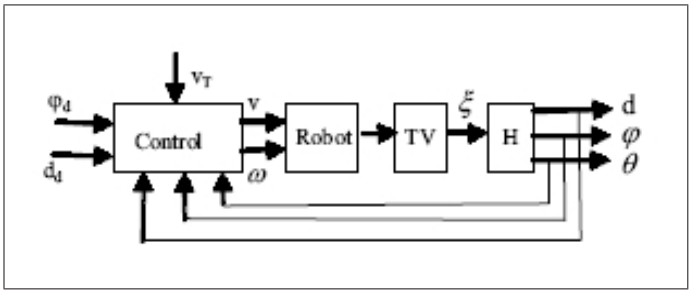


Figure 6: Control structure.

The control objective is defined as follows: Assuming a case where a leading robot moves along an unknown trajectory, with unknown velocity as well, make the follower robot keep a desired distance d_d to the leader and pointing to it (that is $\varphi_d = 0$), using only visual information, Fig. 5. More specifically, the control objective can be expressed as

$$\lim_{t\to\infty} e(t) = \lim_{t\to\infty}(d_d - d) = 0 \qquad (6)$$
$$\lim_{t\to\infty} \widetilde{\varphi}(t) = \lim_{t\to\infty}(\varphi_d - \varphi) = 0$$

The evolution of the posture of the follower robot relative to the leader will be stated by the time derivative of the two error variables. The variation of distance error is given by the difference between the projection of the leader's velocity and the follower robot velocity on the line connecting both vehicles, that is:

$$\dot{e} = -v_T \cos\theta + v\cos\widetilde{\varphi} \tag{7}$$

Likewise, the variation of angle error has three terms: the angular velocity of the follower robot, and the rotational effect of the linear velocities of both robots, which can be expressed as:

$$\dot{\widetilde{\varphi}} = \omega + v_T \frac{\sin\theta}{d} + v\frac{\sin\widetilde{\varphi}}{d} \tag{8}$$

For the system dynamics expressed by (7) and (8), the following nonlinear controller is proposed to achieve the control objective given in (6),

$$v = \frac{1}{\cos\widetilde{\varphi}}(v_T\cos\theta - f(e)) \tag{9}$$
$$\omega = -f(\widetilde{\varphi}) - v_T\frac{\sin\theta}{d} - v\frac{\sin\widetilde{\varphi}}{d}$$

In (9), $f(e),\ f(\widetilde{\varphi}) \in \Omega$, with Ω the set of functions that meet the following definition:

$$\Omega = \{f : \Re \to \Re / f(0) = 0 \text{ and } xf(x) > 0\ \forall x \in \Re\}$$

In particular, the following functions are considered: $f(e) = k_e \tanh(\lambda_e e)$ and $f(\widetilde{\varphi}) = k_{\widetilde{\varphi}} \tanh(\lambda_{\widetilde{\varphi}}\widetilde{\varphi})$.

These functions prevent the control actions becoming saturated. The variables used by this controller (θ, φ, d) as given by (5) are calculated from the image captured by the vision system.

By combining (7), (8) and (9), the closed-loop system is obtained:

$$\dot{e} = -f_e(e) \tag{10}$$
$$\dot{\widetilde{\varphi}} = -f_{\widetilde{\varphi}}(\widetilde{\varphi})$$

5.2 Stability Analysis

Considering the system of (10) with its single equilibrium point at the origin, the following Lyapunov candidate function,

$$V = \frac{e^2}{2} + \frac{\widetilde{\varphi}^2}{2} \tag{11}$$

has a time-derivative on the system trajectories given by

$$\dot{V} = -ef(e) - \widetilde{\varphi} f(\widetilde{\varphi}) \tag{12}$$

It is then concluded the asymptotic stability of the equilibrium, that is:

$$e(t) \to 0, \quad \widetilde{\varphi}(t) \to 0, \qquad \text{with } t \to \infty.$$

It should be noted that the controller of (9) requires knowing the linear velocity v_T of the leader vehicle. This variable should be estimated with the available visual information. By approximating the derivative of the position error as given in (7) by the discrete difference between successive positions, and considering a $0.1s$ sampling period, the objective velocity can be approximated as follows:

$$\widehat{v}_T = \frac{(d_k - d_{k-1})/0.1 + v\cos\widetilde{\varphi}}{\cos\theta}$$

6 Experiments

In order to evaluate the performance of the proposed coordination control algorithm, experiments were carried out with two Pioneer 2DX Mobile Robots (Fig. 7). Each robot has its own control system. The vision system includes a frame grabber PXC200 that allows capturing the images from a camera SONY EV-D30 mounted on the follower robot. These images are transmitted from the follower robot to a Pentium II-400 Mhz PC, in charge of processing the images and of calculating the corresponding control actions. Figure 8(a) shows the image captured by the camera; this image is processed to obtain the image shown in Fig. 8(b). From this image, the centroids of the four projected pattern's marks are calculated and used to compute the variables (θ, φ, d) needed by the controller. Finally, the computed control actions are sent by a transmitter to the follower robot.

For the experiences, the following parameter values were used: $k_e = 200$, $\lambda_e = 0.005$, $k_{\widetilde{\varphi}} = 10$ and $\lambda_{\widetilde{\varphi}} = 0.1$. The follower robot has to follow the leading robot by keeping a desired distance of $d_d = 0.50m$ and $\varphi_d = 0$. Figure 9 shows the evolution of the distance between both robots. Figure 10 shows the evolution of angles φ and θ. From these figures, the accomplishment of the control objective can be verified. Fig. 11 depicts the control actions that are calculated and sent to the follower robot. The estimation of the leader robot's velocity is shown in Fig. 12, and the trajectory of the follower robot is depicted in Fig. 13.

Figure 7: Robots used in the experiment.

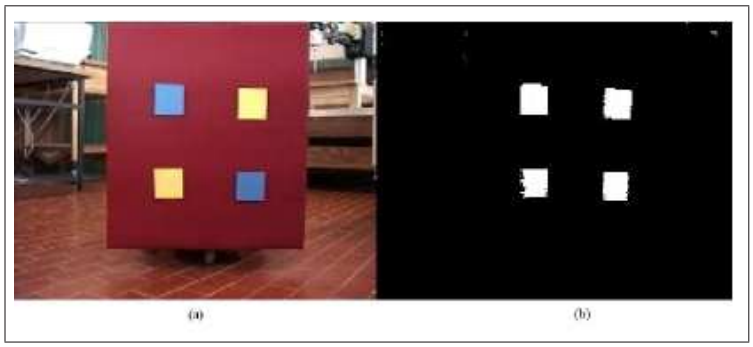


Figure 8: (a) Image captured by the robot's camera. (b) Image processed to find the centroid of each mark.

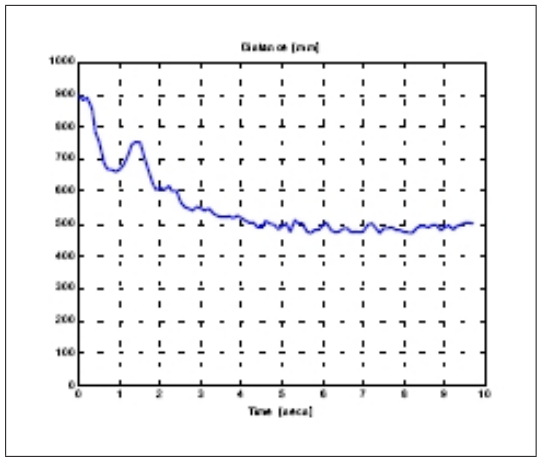


Figure 9: Evolution of the distance to the objective.

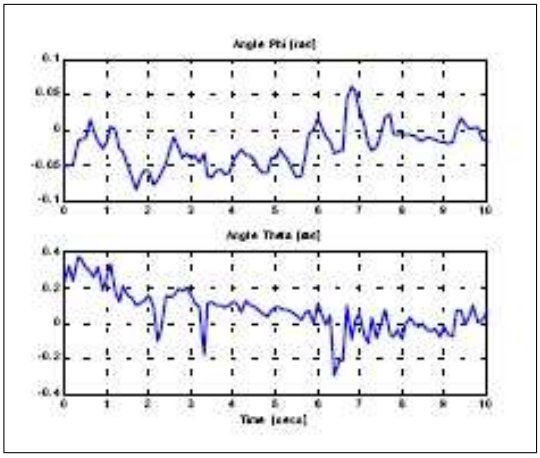


Figure 10: Evolution of angles φ and θ.

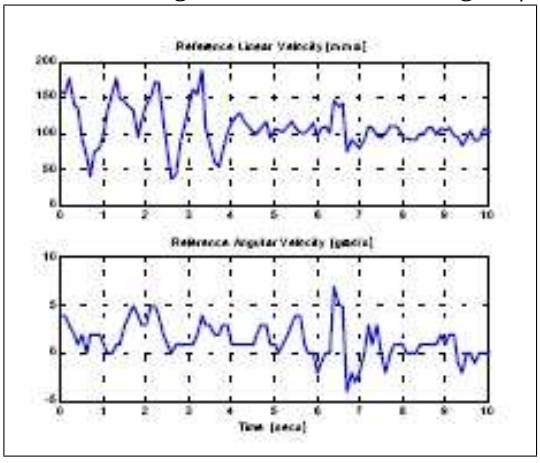


Figure 11: Calculated control actions.

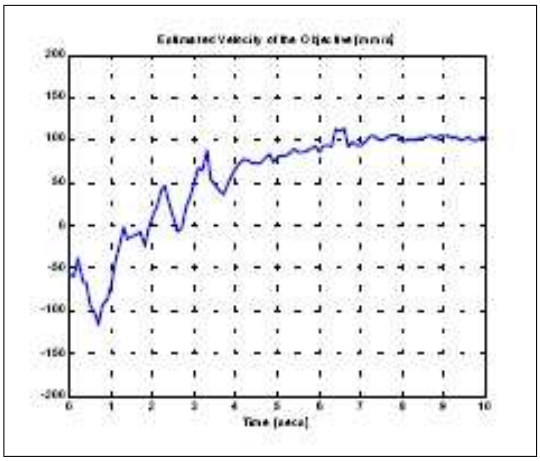


Figure 12: Estimated velocity of the objective.

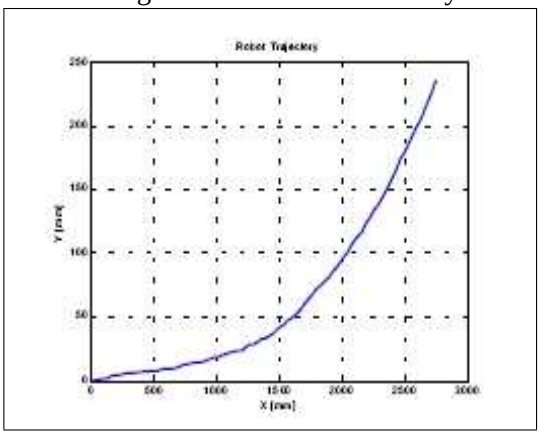


Figure 13: Trajectory followed by the follower robot.

7 Conclusions

In this work, a non-linear vision-based controller for coordinated motion of mobile robots following a leader robot has been presented. The nonlinear controller has been designed with state dependent control gains that allow avoiding the saturation of the control actions. By using the Lyapunov method, it has been proven that the resulting control system is asymptotically stable. Through experiences, it has been demonstrated that the proposed control system accomplishes the control objective with a good performance, leading to a specified formation in which the follower robots follow a leader robot by using visual information.

8 Acknowledgment

This work was partially supported by ANPCyT and CONICET, Argentina. The authors thank the Science and Technology for Development Program (CYTED) for promoting the research cooperation between their groups.

References

[1] R. Alami, Multirobot cooperation in the MARTHA project, *SIEEE Robotics and Automation Magazine*, Vol. 5, No. 1, pp. 36-47, 1998.

[2] R. C. Arkin, Integrating behavioural, perceptual and world knowledge in reactive navigation, *Journal of Robotics and Autonomous Systems,* Vol. 6, No. 1, pp. 36-47, 1990.

[3] H. Asama, Operation of cooperative multiple robots using communication in a decentralised robotic system, *Proc. Conf. From Perception to Action, Switzerland,* 5-7 Sept., 1994.

[4] J. Borestein, Navigating MobileRobots - Systems and Techniques, *A K Peters, Wesley, MA, USA,* 1996.

[5] R. A. Brooks, A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation,* Vol. 2, No. 1, pp. 14-23, 1986.

[6] Y. Ishida, Functional complement by co-operation of multiple autonomous robots, *Proc. IEEE Int. Conf. on Robotics and Automation,* pp. 2476-2481, 1994.

[7] M. J. Mataric, Learning in Multi-Robot Systems, *Lecture Notes in Artificial Intelligence – Adaptation and Learning in Multi-Agent Systems,* Vol. 10, 1996.

Carlos M. Soria, Ricardo Carelli
Universidad Nacional de San Juan
Instituto de Automática
Av. San Martín oeste 1109
5400, San Juan, Argentina
E-mail: csoria@inaut.unsj.edu.ar, rcarelli@inaut.unsj.edu.ar

Rafael Kelly
Centro de Investigación Científica y de Educación Superior de Ensenada
Ensenada, Baja California, 22800 México
E-mail: rkelly@cicese.mx

Juan M. Ibarra Zannatha
Laboratorio de Robótica del Departamento de Control Automático
Centro de Investigación y de Estudios Avanzados
Av. IPN N 2508, Lindavista, 07360 México, DF
E-mail: jibarra@ctrl.cinvestav.mx

International Journal of Computers, Communications & Control
Vol. I (2006), No. 2, pp. 95-101

Improvement and extension of Virtual Reality for flexible systems of manufacture

Flavio Véliz Vasconcelo, Gastón Lefranc Hernández

Abstract: In this work it presents the improvement and extension of the virtual reality software created by the Robotics, Artificial Intelligence and Automatization Outpost Laboratory in the school of electrical engineering of the Pontificia Universidad Católica de Valparaíso.
Keywords: Software, Virtual Reality, Java, FMS.

1 Introduction

The development of this program has as it bases the version 1,0 on which a virtual model is created prototype of a cartesian manipulator pertaining to a cell of storage AS/RS of a Flexible System of Manufacture. Version 2,0, version created in this project bases its modifications on three aspects:

- The 3D Model, is made using support tools that facilitate the work and allow a better design, in this one case 3Dstudio.
- Extends software adding to the virtual reality new objects from the flexible system of complete manufacture that is in the laboratory
- Settles down a control of the Scara manipulator and the virtual reality of the work imagines that is made in an assembled flexible cell of, constituted by a robotic manipulator, a vision system and a automated transport system.

The reason for this one project is to maintain the control of a process and also to make modifications in the cell doing the simulation in the virtual reality to optimize the work of the flexible system reducing to the died time of the cell produced by the halting of the process when making modifications and tests of the same ones.

Can be tested in the virtual reality then this account with all the factors that affect a process, for example, gravity and restriction of movement of the objects.

2 State of the art

The generation of interactive tridimensional graphs has advanced immensely in the last years, now we can simulate real worlds with a good degree of reality in real time. But the techniques of interface man-machineare still in investigation.

The first Virtual Reality (VR) systems appeared at the end of 80' and beginnings of 90', but the investigation in this field begun at end of the 60' decade, especially in the aerospace military industry, and in investigation center as the Massachusetts Institute of Technology, and de University of North of Carolina.

Between the 90' decade ant the beginning of the XXI century, the VR has experimented a big development united to the advances in communications, computers and all the digital technology.

Now the Virtual Reality is considered as a work tool for the industry, like the automotive industry, robotic, etc., in universities like Universidad Autónoma de México (UNAM), Pontificia Universidad Católica de Valparaíso and Universidad Politécnica de Cataluña. Even in the art universities like Pompeu Fabra and museum like the Gugenheim museum from New York.

In the PUCV the Virtual Reality began in 2001 with a prototype of this type of software. This was the representation of a manipulator in the space. This manipulator is part of a Flexible Manufacturer System (FMS) made in the Robotic, Automation and Artificial Intelligence Laboratory of the Engineering Electrical School. The first idea was represent a cell from the FMS, the software must be flexible and must work in any operative system so was made completely with java. In the V2.0 of this software, is represent all the system that exist in the laboratory, using 3D development tools. This version can control both manipulators in the FMS, the cartesian manipulator, part of the AS/RS cell, and the SCARA manipulator that work with artificial stereo vision and the assembled cell.

At this moment, the laboratory is working in the creation of the environment in real time, using the 3d vision, and the final idea is add this application to the Virtual Reality V2.0 software to make this more flexible.

3 Structure of the system

The represented cell is a manufacture cell of storage that is made up of three stages, the first in charge one of the administration of information (base it of data), the second interface for the administrator and third, the mechanism that will execute the indicated tasks. The storage system is conformed by a warehouse in rectangular form that denominated matrix and a cartesian manipulator of 4 degrees of freedom.

With respect to the design of the virtual world a virtual universe was structured so that it was possible to be added other devices of a SFM, integrating of this form in which are different cells from a flexible production system as it is in figure 1.

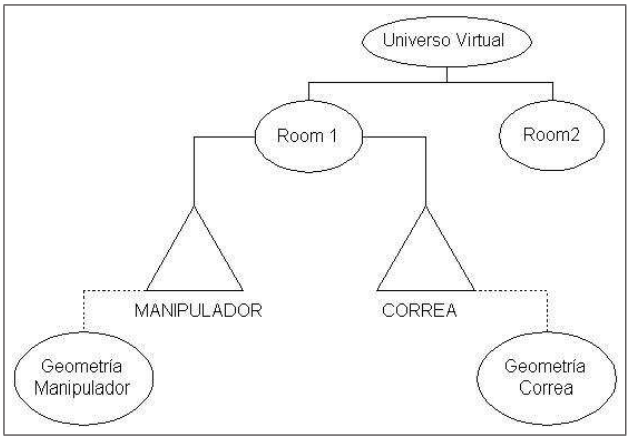


Figure 1:

4 Representation and control of the graphical devices

The established movements of the manipulator in the classes of cartesian manipulator settle down by means of the incorporation of the inverse kinematics of the manipulator, defined as a general characteristic of the manipulating class, which will inherit each manipulator I specify. For the cartesian manipulator simulated, the inverse kinematics the position given by the system of data base is applied directly, since or a user, will be carried out on the base of several final positions. Since the manipulator is cartesian, the final position and the position of each one of their joints are referred a single system of coordinates, unlike another type of manipulators.

The information that gives the data base working in automatic way, with respect to the task to make is the following one:

- Move Manipulator to one position (X_1, Y_1, Z_1)
- Storage the "pallet"
- Take out the "pallet"
- Take the "pallet" from the transport system (X, Y, Z)
- Put the "pallet" on the transport system (X, Y, Z)

The information entered by the user when working in manual way updates the data base so that this one issues the action order to the manipulator. The orders that can be sent are the same ones that gives the data base to him when working in automatic way, that is to say, to keep "pallet", to remove "pallet", etc. The real manipulator is implemented with sensors that confirm their position when executing a movement and this one information is realimentada to the program of virtual reality so that it makes sure that what it is happening in the virtual reality is just like it is happening in the true manipulator.

5 Interface with the reality

An interface with the data base must exist, whose the objective is to receive information of this one or to send information towards this to modify it. This interface is implemented by means of the program NET8, which allows to the connection between the program of virtual reality and the data base to be able to extract the information stored by means of order tables or position. The program must be able to execute the control of a new cell of manufacture, corresponding to the cell of assembled and vision stere. The cell is composed by a manipulator of the type Scara 7547, the work table, the conveyor belts, "pallets" that are used for the transport of products and the system of vision by computer.

This flexible system of production imagines in a virtual reality of with the intention of having the total control on the same one, being able to represent everything what it is happening in the reality, in a virtual reality, thus is possible to be detected any fault in the real system while the behavior of the virtual reality is being observed. The system of virtual reality has communication with the assembled system of through the port of Com1 communications of interface RS-232, through which the coordinates are transmitted towards which the manipulator is due to move when he is commanded from the assembled station of or to order a manual control of the Scara manipulator when controlling the interface of virtual reality.

6 Tools of programming and design

The described program previously was developed programming in Java language using the API of design 3D, Java3D. The Programming language was chosen by the following reasons: 1. - The possibility must exist of creating an interface between the graphical generator and the source program. Because Java3D is a API of Java, this one function already is built-in. 2. - The availability and facility of manual and tutorial obtaining about Java of these programs. 3. - The advantage to create a program multiplatform, since Java works on a virtual machine. 4. - As version 1,0 of the program of virtual reality exists, created with the tools of Java, it will use its code of source to reuse it in new versions.

The great advantage of Java is that it is a programming language OO and of neutral architecture, and the advantage of the Programming Oriented to Objetos (POO) is the capacity to reuse code. Another one of the great advantages of Java is that not they must creates applications different to use the application

in different platforms. With Java we can to develop an application that automatically can be used in diverse platforms, like Windows, UNIX or Macintosh systems. Using JBuilder, of the Borland company, it is possible to be made an express and easy joint of the elements of a graphical interface for Java applications. To construct user interfaces is not of great complexity because it is counted on several blocks of components, which are selected from a trowel that contains components such as: text bellboys, areas, lists, pictures of dialogues, etc. Soon the values of the properties of these components are selected and al is enclosed component event del the code that will treat this event, saying al program how to respond to an event in the user interface.

In order to obtain a better design of virtual reality that resembles more the real FMS it uses the design program 3Dstudio Max5 version in Spanish supported by the tool of Autocad design 2002 version in Spanish. To the support of these tools and using the class "Loader" of J3D it is possible to be related the archives of objects 3D of 3Dstudio and Java. The design is begun creating the objects in the Autocad program thus to be able to handle the measures of the objects and to give with greater facility the different forms from the objects, soon the archives of autocad are exported towards 3Dstudio and here the design is completed giving colors and textures to these.

7 3D Model

The great improvement that incorporates version 2,0 of the program of virtual reality bases on the generation of objects since in version 1,0 the manipulator was created in his totality having used the API of Java 3D, which implies to have to write great amount of lines of code for a simple object and without great definition. Although using J3D a scene can be created, also it is possible to be done generating the objects of the scene in a program of design like for example Autocad or 3Dstudio Max. This is a powerful form to bring objects towards the virtual world being concerned the information from a file.

To load the data from a file allows to accede to the data created in another application. The program 3DStudio Max5 provides the tools necessary to be able to create the objects that will compose the flexible system of manufacture. The simulation of a universe is generated using a modelador. The development of a modelador in 3-D includes three main phases, as it is in figure 2.

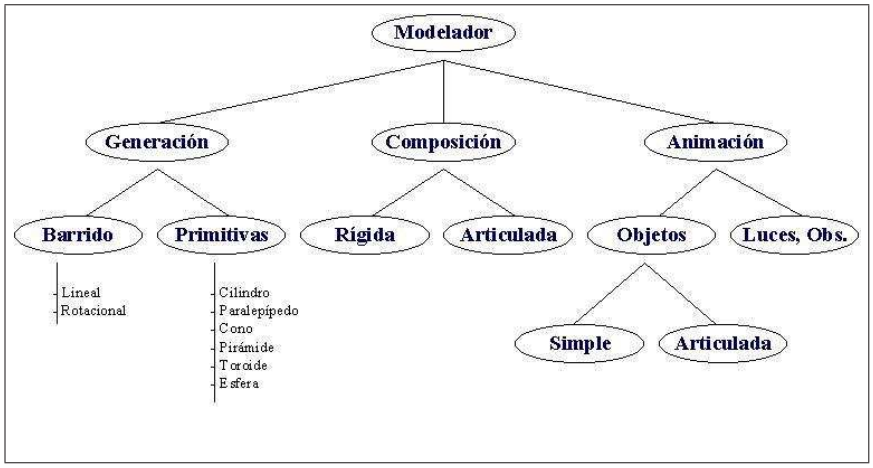


Figure 2:

Generation of primitive: The simple geometric objects are created, as it bases of any geometric model. The simple objects are created by primitive parametric of generation, or linear or radial sweeping. Composition: The compound objects are created assembling simple objects.

The composition generates more complex objects with rigid or articulated behavior. Animation: The animation consists of equipping with movement to the objects within the environment. This can be

applied to the objects composed in its totality or one of its joints. The animation includes movements of camera (point of view of the observer) and manipulation of the lights.

Through different techniques from design the images of figures 3 and 4 were obtained:

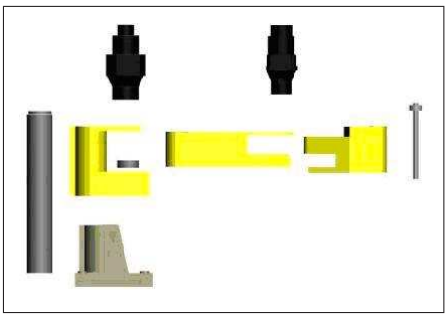


Figure 3: Simple Object

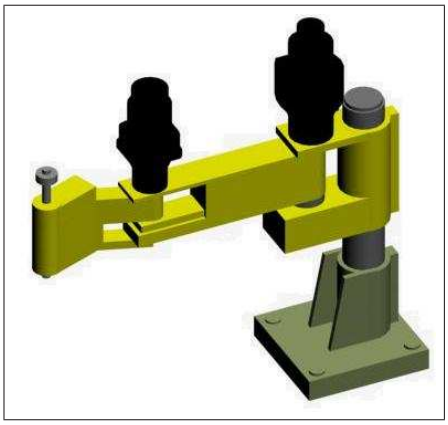


Figure 4: Compound Object

In order to be able to obtain these one better quality in the design the support for the design looks for using the programs Corel Draw and their application with Scanner to make OCR of vectorization. The data from measures of the manipulator are taken, that are in the manual of this one, where are two views that are sufficient to be able to make a design 3D of the manipulator. Similarly all the components are being created that comprise of the flexible system which they were mentioned in the previous points, arriving at a final design of the version 2,0 that can be seen in figure 5.

8 Inverse kinematics

When we select a position in the cartesian plane inside of the work space, the SCARA is instruct to move to the selected position. With this information we can calculate the inverse kinematics to generate the animation of the 3D manipulator, moving the articulations or joints to the set point.

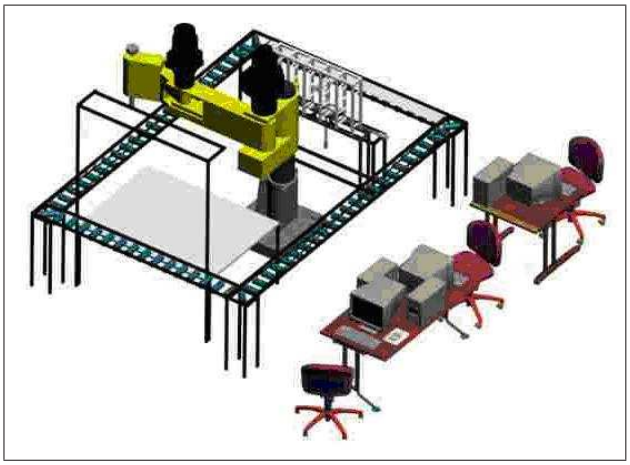


Figure 5: Virtual SFM

In every joint, the Z axis is selected as rotation axis, as X axis the junction of the joints, and the Y axis the third perpendicular axis.

To describe the position of the joints from the angles of joints we can describe a transformation matrix that it relates the frame assigned to a joint with the frame of the previous joint. To the SCARA manipulator, the transformation matrix between the ended frame and the base frame is:

$$T = \begin{bmatrix} cos\theta_{12} & -sin\theta_{12} & 0 & L1cos\theta_1 \\ sin\theta_{12} & cos\theta_{12} & 0 & L1sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where: $\theta_{12} = \theta_1 + \theta_2$

The problem of the inverse kinematic is obtain the angle from the position in the cartesian coordinates (x,y) of the greeper. This angles can be obtain in algebraic way or geometrically. In this case we chose the geometrical method, because is more simple.

In this way, we obtain:

$$cos\theta_2 = (r^2 - L_1^2 + L_2^2)/(2L_1L_2) \quad (2)$$

$$\theta_1 - \beta + \delta, if\ \theta_2 < 0 \quad (3)$$

$$\theta_1 = \beta - \delta, if\ \theta_2 > 0 \quad (4)$$

where,

$$r^2 = X^2 + Y^2 \quad (5)$$

$$tan\ \beta = Y/X \quad (6)$$

$$cos\delta = (r^2 + L_1^2 - L_2^2)/(2L_1r) \quad (7)$$

L_1 and L_2 are know values, because are the length of the two part of the arm of the SCARA manipulator, so is necessary to know the value of r to could calculate δ. Like the point of destiny of the greeper is know too, and know in Cartesian coordinates, the X and Y values are replaced in the equation number 5, then we can obtain the r value. X and Y value are data entered by the user, using the virtual reality software interface, or trough the database that command all the Flexible Manufacturer System.

Like we have the values of L_1, L_2, and r, we can obtain the β value working with equation 7.

With the X and Y values replaced in equation 6 we can obtain the value of β. With equation 2 we obtain the θ_2, that indicate the rotation angle of L_2 and that define the equation of rotation of the extremity L_1, equations 3 and 4 depending of the case.

This information is passed to the Animation class, class that have the capacity to move the 3D design using the behavior class of J3D.

9 Conclusions

The use of the virtual reality in the industrial area allows to increase the production and to keep us to the moved away personnel of the accomplishment of dangerous tasks. With the simulation of the flexible system tests in a simulated system are reduced to the production costs when doing that it counts on all the factors that affect to the real system, in such a way that the best option can be selected to make a task without having to stop the system for a long time, but so single the necessary thing to make the modification. The advantage to make the virtual reality in Java is to allow to implement the application in independent platforms without no problem then this runs on a virtual machine, and as Java were born oriented to the Internet, also it allows us to adapt software so that service to the production can be used in the Web rendering and allowing the control of the flexible system, from any part of the world.

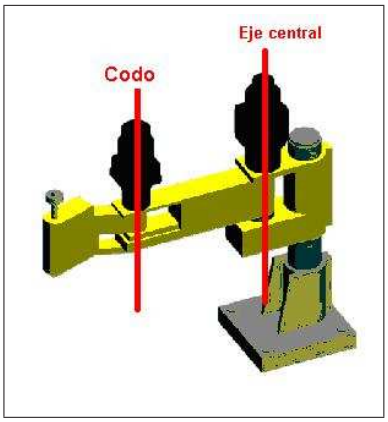


Figure 6: Virtual SCARA, Joints and rotation axis

Virtual Reality can join two communities very different as artist and scientist, because this projects need informatics and engineering experts and designers and expert in visual arts.

References

[1] Zepeda, Roberto (Lefranc, Gastón, P.G.), *Modelación en redes de petry y simulación en realidad virtual de sistemas flexibles de manufactura*, Informe final de proyecto de titulación, Pontificia Universidad Católica de Valparaíso. Agosto, 2001.

[2] Flores, Javier (Lefranc, Gastón, P.G.), *Realidad virtual a sistemas flexibles de manufactura*, Informe final de proyecto de titulación, Pontificia Universidad Católica de Valparaíso. Mayo, 2004.

[3] Aaron E. Walsh, Doug Gehringer, *Java 3D, API Jump Start*.

[4] Selman, Daniel, *Java 3D Programming*.

[5] Burgos, Daniel, *3Dstudio Max Práctico. Guía de aprendizaje*.

[6] Lemay Cadenhead, *Aprendiendo Java en 21 Días*.

Flavio Véliz Vasconcelo, Gastón Lefranc Hernández
Pontificia Universidad Católica de Valparaíso
Escuela de Ingeniería Eléctrica
Av. Brasil 2147, Valparaíso, Chile
E-mail: flavio.veliz@vtr.net, glefranc@ieee.org

Author index

Description

International Journal of Computers, Communications & Control (IJCCC) is published from 2006 and has 4 issues per year, edited by CCC Publications, powered by Agora University Editing House, Oradea, ROMANIA.

Every issue is published in online format (ISSN 1841-9844) and print format (ISSN 1841-9836). We offer free online access to the full content of the journal (http://journal.univagora.ro). The printed version of the journal should be ordered, by subscription, and will be delivered by regular mail.

IJCCC is directed to the international communities of scientific researchers from the universities, research units and industry.

IJCCC publishes original and recent scientific contributions in the following fields:

- Computing & Computational Mathematics,
- Information Technology & Communications,
- Computer-based Control

To differentiate from other similar journals, the editorial policy of IJCCC encourages especially the publishing of scientific papers that focus on the convergence of the 3 "C" (Computing, Communication, Control).

The articles submitted to IJCCC must be original and previously unpublished in other journals. The submissions will be revised independently by two reviewers.

IJCCC also publishes:

- papers dedicated to the works and life of some remarkable personalities;
- reviews of some recent important published books.

Also, IJCCC will publish as supplementary issues the proceedings of some international conferences or symposiums on Computers, Communications and Control, scientific events that have reviewers and program committee.

The authors are kindly asked to observe the rules for typesetting and submitting described in *Instructions for Authors*.

There are no fees for processing and publishing articles. The authors of the published articles will receive a hard copy of the journal.

Instructions for authors

Papers submitted to the International Journal of Computers, Communications & Control must be prepared using a LaTeX typesetting system. A template for preparing the papers is available on the journal website http://journal.univagora.ro. In the template file you will find instructions that will help you prepare the source file. Please, read carefully those instructions.
Any graphics or pictures must be saved in Encapsulated PostScript (.eps) format.
Papers must be submitted electronically to the following address: ccc@univagora.ro.
The papers must be written in English. The first page of the paper must contain title of the paper, name of author(s), an abstract of about 300 words and 3-5 keywords. The name, affiliation (institution and department), regular mailing address and email of the author(s) should be filled in at the end of the paper.
Manuscripts must be accompanied by a signed copyright transfer form. The copyright transfer form is available on the journal website.
The publishing policy of IJCCC encourages particularly the publishing of scientific papers that are focused on the convergence of the 3 "C" (Computing, Communications, Control).
Topics of interest include, but are not limited to the following:

- Applications of the Information Systems
- Artificial Intelligence
- Automata and Formal Languages
- Collaborative Working Environments
- Computational Mathematics
- Cryptography and Security
- E-Activities
- Fuzzy Systems
- Informatics in Control
- Information Society - Knowledge Society
- Natural Computing
- Network Design & Internet Services
- Multimedia & Communications
- Parallel and Distributed Computing

Order

If you are interested in having a subscription to “Journal of Computers, Communications and Control”, please fill in and send us the order form below:

ORDER FORM		
I wish to receive a subscription to “Journal of Computers, Communications and Control”		
NAME AND SURNAME:		
Company:		
Number of subscription:	Price Euro	for issues yearly (4 number/year)
ADDRESS: City: Zip code: Country: Fax: Telephone: E-mail:		
Notes for Editors (optional)		

1. **Standard Subscription Rates (2006)**
   - Individual: One Year (4 issues) 70 Euro
   - Institutional: One Year (4 issues) 90 Euro
2. **Single Issue Rates**
   - Individual: 20 Euro
   - Institutional: 25 Euro

For payment subscription rates please use following data:
HOLDER: Fundatia AGORA, CUI: 12613360
BANK: EUROM BANK ORADEA
BANK ADDRESS: Piata Unirii nr. 2-4, Oradea, ROMANIA
IBAN ACCOUNT for EURO: RO02DAFB1041041A4767EU01
IBAN ACCOUNT for LEI/ RON: RO45DAFB1041041A4767RO01,
Mention, please, on the payment form that the fee is “for IJCCC”.

EDITORIAL ADDRESS:
CCC Publications
Piata Tineretului nr. 8
ORADEA, jud. BIHOR
ROMANIA
Zip Code 410526
Tel.: +40 259 427 398
Fax: +40 259 434 925
E-mail: ccc@univagora.ro, Website: www.journal.univagora.ro

Copyright Transfer Form

To The Publisher of the International Journal of Computers, Communications & Control

This form refers to the manuscript of the paper having the title and the authors as below:

The Title of Paper (hereinafter, "Paper"):

.....

The Author(s):

.....

.....

.....

.....

The undersigned Author(s) of the above mentioned Paper here by transfer any and all copyright-rights in and to The Paper to The Publisher. The Author(s) warrants that The Paper is based on their original work and that the undersigned has the power and authority to make and execute this assignment. It is the author's responsibility to obtain written permission to quote material that has been previously published in any form. The Publisher recognizes the retained rights noted below and grants to the above authors and employers for whom the work performed royalty-free permission to reuse their materials below. Authors may reuse all or portions of the above Paper in other works, excepting the publication of the paper in the same form. Authors may reproduce or authorize others to reproduce the above Paper for the Author's personal use or for internal company use, provided that the source and The Publisher copyright notice are mentioned, that the copies are not used in any way that implies The Publisher endorsement of a product or service of an employer, and that the copies are not offered for sale as such. Authors are permitted to grant third party requests for reprinting, republishing or other types of reuse. The Authors may make limited distribution of all or portions of the above Paper prior to publication if they inform The Publisher of the nature and extent of such limited distribution prior there to. Authors retain all proprietary rights in any process, procedure, or article of manufacture described in The Paper. This agreement becomes null and void if and only if the above paper is not accepted and published by The Publisher, or is withdrawn by the author(s) before acceptance by the Publisher.

Authorized Signature (or representative, for ALL AUTHORS): .....

Signature of the Employer for whom work was done, if any: .....

Date: .....

Third Party(ies) Signature(s) (if necessary): .....