Special Issue: Computer Engineering

Research in Computing Science

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 30,** Noviembre, 2007. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. 04-2004-062613250000-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor Responsable: *Juan Humberto Sossa Azuela, SOAJ 560723*

Research in Computing Science *is published by the Center for Computing Research of IPN.* ***Volume 30,*** *November, 2007. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, October, 2007, in the IPN Graphic Workshop – Publication Office.*

Volume 30
Volumen 30

Special Issue: Computer Engineering

Volume Editors:
Editores del Volumen

Amadeo Argüelles Cruz
José Luis Oropeza Rodríguez
Oscar Camacho Nieto
Osvaldo Espinosa Sosa

Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2007

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http.//www.ipn.mx
http://www.cic.ipn.mx

The editors and the Publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

Indexed in LATINDEX and PERIODICA
Indexada en LATINDEX y PERIODICA

Printing: 500
Tiraje: 500

Printed in Mexico
Impreso en México

Preface

Computer engineering is one of the most rewarding areas of computer science and it has one of the most extraordinary advance of the last decades. In this occasion they have been included in this volume of the Research in Computing Science magazine the contributions of specialists in areas such as: digital signal processing, image processing, digital systems design and computer architecture, disciplines that have a high impact in the present technology.

The selection process has been very rigorous in such a way that only the best articles of the many sent by researchers of several parts of the world have been accepted, guaranteeing that the content is of the highest quality.

We wished to sincerely express acknowledge to all those people who contributed of some form in the accomplishment of this magazine. We are specially in debt with all the authors, reviewers and scientific committee by its extraordinary work and the advice offered with the only aim to obtain the best product. Thanks also to the National Polytechnic Institute for all the offered facilities to be able to reach the primary objective: this special number of this magazine that we believe is of much importance for the academic and scientific community of any part of the world.

November, 2007

Amadeo Argüelle Cruz
José Luis Oropeza Rodríguez
Oscar Camacho Nieto
Osvaldo Espinosa Sosa

Scientific Committee

Table of Contents

Digit recognition using Continuous Density Hidden Markov Models for different speakers

José Luis Oropeza Rodríguez[1] and Sergio Suárez Guerra[1]

[1] Center for Computing Research, National Polytechnic Institute,
Juan de Dios Batiz esq Miguel Othon de Mendizabal s/n, P.O. 07038, Mexico
joropeza@cic.ipn.mx, ssuarez@cic.ipn.mx

Abstract. The goal of automatic speech recognition (ASR) is to develop techniques and systems that enable computer to accept speech input. The digit task has been so much employed to contribute the effort in ASR. Since 1950's, Automatic Speech Recognition Systems (ASRS) has been studied by so much researchers. In our country, universities have inverted a lot of time creating speech recognition systems (UNAM, IPN, ITESM, UAM, UDLA, among others). In this paper we present the results obtained when we trained a digit corpus with 5 people (3 men and 2 women). Firstly, we trained our system with 5 speakers, and we obtained a 94.13% of recognition rate. Secondly, we probed the ASR individually for each speaker and we obtained a recognition rate above of 98.5%. Thirdly, we employed a digit corpus with men only and another with women only, the results obtained were above of 95% for the men corpus and 98% for women corpus. Finally, we report that we used 300 sentences of speech signal (100 for training task and 200 for recognition task) from each speaker, then we processed for the complete corpus (5 speakers), a total of 1500 sentences were utilized. The results were obtained using Hidden Markov Model Toolkit (HTK), and personal software.

Keywords. Automatic Speech Recognition, speaker recognition, Continuous Density Hidden Markov Models, Viterbi Trainning, and dependent-independent speaker automatic speech recognition systems.

1 Introduction

Speech and language are perhaps the most evident expression of human thought and intelligence –the creation of machines that fully emulate this ability poses challenge that reach far beyond the present state of the art.

© A. Argüelles, J. L. Oropeza, O. Camacho, O. Espinosa (Eds.)
Computer Engineering.
Research in Computing Science 30, 2007, pp. 1-12

The speech recognition field has been fruitfully and productively benefited from sciences as diverse as computer science, electrical engineering, biology, psychology, linguistics, statistics, philosophy, physics and mathematics among others. The interplay between different intellectual concerns, scientific approaches, and models, and its potential impact in society make speech recognition one of the most challenging, stimulating, and exciting fields today.

As early as 1950s, simple recognizers have been built, yielding credible performance. But it was soon found that the techniques used in these systems were not easily extensible to more sophisticated systems. In particular, several dimensions emerged that introduce serious design difficulties or significantly degrade recognition performance.

Most notably, these dimensions include:

- Isolated, connected, and continuous speech
- Vocabulary size
- Task and language constraints
- Speaker dependence or independence
- Acoustic ambiguity, confusability
- Environmental noise

The first question one should ask about a recognizer or a task is: is the speech connected or spoken one word at a time? Continuous Speech Recognition (CSR) is considerably more difficult than isolated word recognition (IWR) that is because at first, word boundaries are typically not detectable in continuous speech, at second, there is much greater variability in continuous speech due to stronger coarticulation (or interphoneme effects) and poorer articulation (“El ave es grande” becomes “la vez”).

A second dimension is the size of the vocabulary. Exhaustive search in very large vocabularies is typically unmanageable. Instead, one must turn to smaller sub-word units (phonemes, syllables, triphonemes, etc.), which may be more ambiguous and harder to detect and recognize.

A system with a semantic component may eliminate such sentences from consideration. A system with a probabilistic language model can effectively use this knowledge to rank sentences.

These knowledge sources or language models can reduce an impossible task to a trivial one. The challenge in language modeling is to derive a language model that provides maximum constraint while allowing maximum freedom of input. The constraining power of a model language can be measured by perplexity, roughly the average number of words that can occur at any decision point.

The different sources of variability that can affect speech determine most of difficulties of speech recognition. During speech production the movements of different articulators overlap in time for consecutive phonetic segments and interact with each other. As a consequence, the vocal tract configuration at any time is influenced by more than one phonetic segment. This phenomenon is known as coarticulation mentioned above. The principal effect of the coarticulation is that the same phoneme can have very different acoustic characteristics depending on the context in which it is uttered [Farnetani 97].

The most prominent issue is that of Speaker dependence as opposed to speaker independence. A speaker dependent system uses speech from the target speaker to learn its model parameters. On the other hand, a speaker-independent system is trained once and for all, and must model a variety of speaker's voice.

Speech recognition-system performance is also significantly affected by the acoustic confusability or ambiguity of the vocabulary to be recognized. A confusable vocabulary requires detailed high performance acoustic pattern analysis. Another source of recognition-system performance degradation can be described as variability and noise.

Finally, the applications of the ASR are vast, for example: Credit-card numbers, telephone numbers, and zip codes, require only a small vocabulary.

2 Characteristics and Generalities

"The schools of thought in speech recognition" describe four different approach researched at today, they are [Kirschning 1998]:

- template-based approach
- knowledge-based approach
- stochastic approach and,
- connectionist approach

Before continuing described the characteristics of them, we must to say that ASR has implemented one stage called "speech analysis". The applications that need voice processing (such as coding, synthesis, and recognition) require specific representations of speech information. For instance, the main requirement for speech recognition is the extraction of voice features, which may distinguish different phonemes of a language.

To decrease vocal message ambiguity, speech is therefore filtered before is arrives at the automatic recognizer. Hence, the filtering procedure can be considered as the first stage of speech analysis. Filtering is performed on discrete time quantized speech signals. Hence, the first procedure consists of a conversion analog to digital signal. Then, the extraction procedure of the significance features of speech signal is performed.

In the template-based approach, the units of speech (usually words, like in this work), are represented by templates in the same form as the speech input itself. Distance metrics are used to compare templates to find the best match, and dynamic programming is used to resolve the problem of temporal variability. Template-based approaches have been successful, particularly for simple applications requiring minimal overhead.

In the knowledge-based approach, proposed in the 1970s and early 1980s. The pure knowledge-based approach emulates human speech knowledge using expert systems. Rule-based systems have had only limited success. The addition of knowledge was found to improve other approaches substantially. Recently, in the Spanish language a new approach using the rules of the syllabic units has showed the utility of these units in the ASR.

The stochastic approach, which is similar to the template-based approach has been using in the recent developments of ASR. One major difference is that the probabilistic models (typically Hidden Markov Models –HMM-) are used. HMM are based on a sound probabilistic framework, which can model the uncertainty inherent in speech recognition. HMM have an integral framework for simultaneously solving the segmentation and the classification problem, which makes them particularly suitable for continuous-speech recognition. One characteristic of HMM is that they make certain assumptions about the structure of speech recognition, and then estimate system parameters as though the structures were correct.

The connectionist approach use distributed representations of many simple nodes, whose connections are trained to recognize speech. Connectionist approach is a most recent development in speech recognition. While no fully integrated large-scale connectionist systems have been demonstrated yet, recent research efforts have shown considerable promise. Some of the problems that remain to be overcome include reducing time training and better modelling of sequential constraints.

3 Automatic Speech Recognition Systems

The frequency bandwidth of a speech signal is about 16 KHz. However, most of speech energy is under 7 KHz. Speech bandwidth is generally reduced in recording. A speech signal is called orthophonic if all the spectral components over 16 KHz are discarded. A telephonic lower quality signal is obtained when ever a signal does not have energy out of the band 300-3400 Hz. Therefore, digital speech processing is usually performed by a frequency sampling ranging between 8000 samples/sec and 32000 samples/sec. These values correspond to a bandwidth of 4 kHz and 16 kHz respectively. In this work, we use a frequency sampling 11025 samples/sec [Bechetti and Prina 1999].

The excitation signal is assume periodic with a period equal to the pitch for vowels and other voice sounds, while for unvoiced consonants, the excitation is assumed white noise, i.e. a random signal without dominant frequencies. The excitation signal is subject to spectral modifications while it passes through the vocal tract that has an acoustic effect equivalent to linear time invariant filtering. These modifications give to the final sound the characteristic features of the different phonemes of a language. The model is relevant, for each type of excitation; a phoneme is identified mainly by considering the shape of the vocal tract configuration can be estimated by identifying the filtering performed by the vocal tract on the excitation. Introducing the power spectrum of the signal $P_x(\omega)$, of the excitation $P_y(\omega)$ and the spectrum of the vocal tract filter $P_h(\omega)$, we have:

$$P_x(\omega) = P_y(\omega)P_h(\omega) \qquad \textbf{[1]}$$

Where ω is the frequency of the discrete time signal. The spectrum of the filter can be obtained from the power spectrum of the speech $P_x(\omega)$ the contribution of the excitation power $P_y(\omega)$.

3.1 Signal preprocessing

The characteristics of the vocal tract define the current uttered phoneme. Such characteristics are evidenced in the frequency domain by the location of the formants, i.e. the peaks given the resonances of the vocal tract. Although possessing relevant information, high frequency formants have smaller amplitude with respect to low frequency formants. A preemphasis of high frequencies is therefore required to obtain similar amplitude for all formants. Such processing is usually obtained by filtering the speech signal with a first order FIR filter whose transfer function in the z-domain is [Oppenheim 89]:

$$H(z) = 1 - az^{-1} \qquad \textbf{[2]}$$

a being the preemphasis parameter. In essence, in the time domain, the preemphasized signal is related to the input signal by the relation:

$$x'(n) = x(n) - ax(n-1) \qquad \textbf{[3]}$$

A typical value for ***a*** is 0.95, which gives rise to a more than 20 dB amplification of the high frequency spectrum.

3.2 Windowing

Traditional methods for spectral evaluation are reliable in the case of a stationary signal (i.e. a signal whose statistical characteristics are invariant with respect to time). For voice, this holds only within the short time intervals of articulatory stability, during which a short time analysis can be performed by "windowing" a signal $x'(n)$ into a sequence of windowed sequences $x_t(n), t = 1,2,....,T$ called frames, which are then individually processed:

$$x_t'(n) \equiv x'(n - t \cdot Q), \quad 0 \le n < N, \quad 1 \le t \le T \qquad \textbf{[4]}$$

$$x_t(n) \equiv w(n) \cdot x_t'(n) \qquad \textbf{[5]}$$

Where $w(n)$ is the impulse response of the window. Each frame is shifted by a temporal length *Q*. If *Q*=*N*, frames do not temporally overlap while if *Q*<*N*, *N*-*Q* samples at the end of a frame $x_t'(n)$ are duplicated at the beginning of the following frame .

In ASR, the most-used window shape is the Hamming window, whose impulse response is a raised cosine impulse [DeFatta et al. 1988]:

$$w(n) = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{N-1}) & n = 0,..,N-1 \\ 0 & otherwise \end{cases} \qquad \textbf{[6]}$$

4 Hidden Markov Models and Experimental Methodology

Now, we are going to show the algorithms employed for Automatic Speech Recognition using Hidden Markov Models (HMMs). Like we know, HMMs mathematical tool applied for speech recognition presents three basic problems [Rabiner and Biing-Hwang, 1993] y [Zhang 1999]:

Problem 1. Given the observation sequence $O = O_1 O_2 ....O_T$, and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model?

1. Initialization

$$\alpha_1(i) = \pi_i b_i(O_1) \quad 1 \le \text{i} \le \text{N} \qquad \textbf{[7]}$$

2. Induction

$$\alpha_{t+1}(j) = b_j(O_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{ij} \quad 1<\text{j}\le\text{N}, 1\le\text{t}\le\text{T-1} \qquad \textbf{[8]}$$

3. Termination

$$\mathrm{P}(\mathrm{O} \mid \lambda) = \sum^{\mathrm{N}} \alpha_{\mathrm{T}}(\mathrm{i}) \qquad \textbf{[9]}$$

Problem 2. Given the observation sequence $O = O_1 O_2 ....O_T$ and the model λ, how do we choose a corresponding state sequence $Q = q_1 q_2 ...q_T$ which is optimal in some meaningful sense?

1. Initialization

$$\delta_1(i) = \pi_i b_i(O_1) \quad 1 \le \mathrm{i} \le \mathrm{N} \qquad \textbf{[10]}$$

2. Recursion

$$\delta_{t+1}(j) = b_j(O_{t+1})\left[\max_{1 \le i \le N} \delta_t(i) a_{ij}\right] 1 \le \mathrm{j} \le \mathrm{N}, 1 \le \mathrm{t} \le \mathrm{T}\text{-}1 \qquad \textbf{[11]}$$

3. Termination

$$p^* = \max[\delta_T(i)] \qquad 1 \le \mathrm{i} \le \mathrm{N} \qquad \textbf{[12]}$$

$$q^* = \arg\max[\delta_T(i)] \quad 1 \le \mathrm{i} \le \mathrm{N} \qquad \textbf{[13]}$$

Problem 3. How do we adjust the model parameters
$\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$?

$$a_{ij} = \frac{\exp\mathit{ected} \quad \mathit{number} \;\; \mathit{of} \;\; \mathit{times} \;\; \mathit{from} \;\; \mathit{state} \;\; s_1 \mathit{to} \;\; \mathit{state} \;\; s_j}{\exp\mathit{ected} \;\; \mathit{number} \;\; \mathit{of} \;\; \mathit{transition} \;\; s \;\; \mathit{from} \;\; \mathit{state} \;\; s_i} \qquad \textbf{[14]}$$

$$b_{jk} = \frac{\exp\mathit{ected\ number\ number\ of\ times\ in\ } s_j \mathit{and\ observatin\,g\ } v_k}{\exp\mathit{ected\ number\ of\ times\ in\ state\ } j} \qquad \textbf{[15]}$$

Then, HMMs algorithms must to solve efficiently the problems mentioned above. For each state, the HMMs can use since one to five Gaussian mixtures both to reach high recognition rate and modelling vocal tract configuration in the Automatic Speech Recognition.

Gaussian mixtures

Gaussian Mixture Models are a type of density model which comprise a number of component functions, usually Gaussian. These component functions are combined to provide a multimodal density. They can be employed to model the colours of an object in order to perform tasks such as real-time colour-based tracking and segmentation. In speech recognition, the Gaussian mixture is of the form [Bilmes 98] [Resch, 2001a], [Resch, 2001b], [Kamakshi et al., 2002] and [Mermelstein, 1975].

:

$$g(\mu, \Sigma)(x) = \frac{1}{\sqrt{2\pi^d}\sqrt{\mathbf{det}(\Sigma)}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \qquad \textbf{[16]}$$

Equation 12 shows a set of Gaussian mixtures:

$$gm(x) = \sum_{k=1}^{K} w_k * g(\mu_k, \Sigma_k)(x) \qquad \textbf{[17]}$$

In 12, the summarize of the weights give us

$$\sum_{i=1}^{K} w_i = 1 \quad \forall \quad i \in \{1,........, K\} \quad : w_i \geq 0 \qquad \textbf{[18]}$$

Viterbi Trainning

We used Viterbi training, in this a set of training observations O^r , $1 \leq r \leq R$ is used to estimate the parameters of a single HMM by iteratively computing Viterbi alignments. When used to initialise a new HMM, the Viterbi segmentation is replaced by a uniform segmentation (i. e. each training observation is divided into N equal segments) for the first iteration.

Apart from the first iteration on a new model, each training sequence O is segmented using a state alignment procedure which results from maximising

$$\phi_N(T) = \max_i \phi_i(T) a_{iN} \qquad \textbf{[19]}$$

For *1<i<N* where

$$\phi_j(t) = \left\lfloor \max_i \phi_i(t-1) a_{ij} \right\rfloor b_j(o_t) \qquad \textbf{[20]}$$

With initial conditions given by

$$\begin{aligned} &\phi_1(1) = 1 \\ &\phi_j(t) = a_{1j} b_j(o_1) \end{aligned} \qquad \textbf{[21]}$$

For *i<j<N*. In this and all subsequent cases, the output probability $b_j(.)$ is as defined in the following equations:

$$b_j(o_t) = \prod_{s=1}^{S} \left[\sum_{m=!}^{M_{js}} c_{jsm} \aleph(o_{st}; \mu_{jsm}, \sum{}_{jsm} \right]^{\gamma_s} \qquad \textbf{[22]}$$

If A_{ij} represents the total number of transitions from state i to state j in performing the above maximisations, then the transition probabilities can be estimated from the relative frequencies

$$\hat{a}_{ij} = \frac{A_{ij}}{\sum_{k=2}^{N} A_{ik}} \qquad \textbf{[23]}$$

The sequence of states which maximises $\phi_N(T)$ implies an alignment of training data observations with states. Within each state, a further alignment of observations to mixture components is made.

We can use two methods for each state and each stream

1. use clustering to allocate each observation o_{st} with the mixture component with the highest probability
2. associate each observation o_{st} with the mixture component with the highest probability

In either case, the net result is that every observation is associated with a single unique mixture component. This association can be represented by the indicator function $\psi^r_{jsm}(t)$ which is 1 if o^r_{st} is associated with mixture component m of stream s of state j and zero otherwise.

The means and variances are then estimated via simple averages

$$\hat{\mu}_{jsm} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \psi^r_{jsm}(t) o^r_{st}}{\sum_{r=!}^{R} \sum_{t=!}^{r} \psi^r_{jsm}(t)}$$

$$\hat{\sum}_{jsm} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \psi^r_{jsm}(t)(o^r_{st} - \hat{\mu}_{jsm})(o^r_{st} - \hat{\mu}_{jsm})'}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{t=1}^{M_s} \psi^r_{jst}(t)} \qquad \textbf{[24]}$$

5 Experiments and Results

The evaluation of the experiment proposed involved 5 people (3 men and 2 women) with 300 speech sentences to recognize for each one. Speech signals were recorded at 11200 frequency sample, with 8 bits mono stereo (one channel) in laboratory environment, that significance that they were clean speech without noise. After that, speech signals were processed to eliminate information not useful that is the speech sentences were processed to find start and end points of speech signal, using software own. This software used the energy parameter to find it the interesting region.

Firstly, we used 1500 speech sentences extracted from 5 speakers individually (we used 100 for training task and 200 for recognition task), and we trained the Automatic Speech Recognition using Hidden Markov Models with 6 states (4 states with information and 2 dummies to connection with another chain). Also, we employed one Gaussian Mixture for each state in the chain Markov. The parameters extracted of the speech signal were 39 (13 MFCC, 13 delta and 13 energy coefficients), they are used to training the Hidden Markov Model.

The results obtained in this experiment are resumed in table 1.

Table 1 Recognition rate for 5 speakers individually

	users				
Recognition rate	Speaker 1	Speaker 2	Speaker 3	Speaker 4	Speaker 5
	99.5	99.5	98.5	98.52	98.5

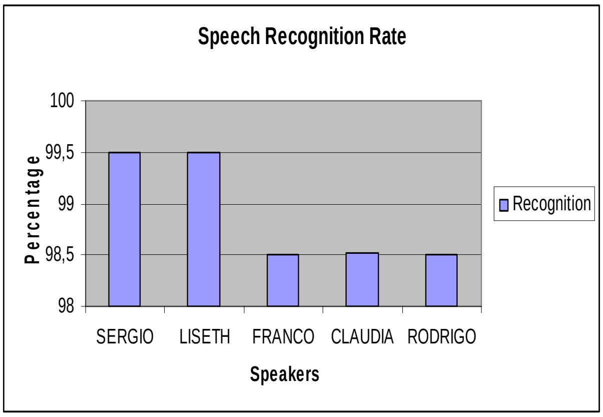


Fig 1 Graphical representation of the recognition rate reported in table 1

Secondly, we divided in two clusters our corpus, separating speech men of the women speech. The speech signals were the same that we used before. Finally, we integrated all speech signals (500 speech sentences, 100 for each speaker) in a corpus that we labelled as 'todos'. We trained the system newly and we probe this new corpus with the 1200 remaining. The results obtained in these two experiments are reported in table 2.

Table 2 Recognition rate for all, men and women speakers

	users		
Recognition rate	all	men	women
	94.13	95	98

Figure 2 shows a graphical representation of the results in table 2.

At respect, we obtain a 94.13% of successful recognition and 5.87% of error rate for all speakers. Given in this table are results for several speech sentences trained and recognized. The HTK (Hidden Markov Model Toolkit) was employed to obtain the results and it was based on complete words.

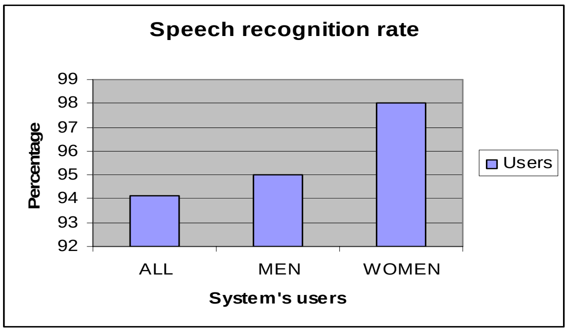


Fig 2 Graphical representation of the recognition rate reported in table 2

6 Conclusions and future works

The main purpose of this paper was to develop a fully ASR system using Hidden Markov Models. We included 5 people in our ASR system; it was because we constructed an ASR system independent of speaker. The results obtained demonstrated that ASR has a high performance independently of amount of speakers that it was included into it. Likewise, the Automatic Speech Recognition (ASR) for each speaker resulted to be satisfactory.

After the results obtained we revised the speech files that help us in this search. We must to say that we found problems with speech signals badly segmented in begin and end of the word. Overcoat for men speech files (especially third, fourth and fifth speakers). It obviously represented in bad results.

For future works we must to find an efficient algorithm that can split the speech signal in signal not necessary, useful signal and not useful signal. That is, we must to work in increase the number of the speakers and programming another splitting algorithm, and we will probably obtain better results. Though the results reported demonstrated a good performance.

References

[Bechetti and Prina 1999] Bechetti Claudio and Prina Ricoti Lucio, “Speech Recognition Theory and C++ Implementation”, Fundazione Ugo Bordón, Rome, Italy, John Wiley and Sons, Ltd, 1999.

[Bilmes 98] BILMES J.A., “A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models”, International Computer Science Institute, Berkeley, CA. 1998.

[DeFatta et al. 1988] DeFatta J. David, Lucas G. Joseph and Hodgkiss S. William, Digital Signal Processing, A system design approach, John Wiley & Sons, 1988.

[Farnetani 97] Farnetani E., “Coarticulation and connected speech processes”, in the Handbook of Phonetic Sciences, W. Hardcastle and J. Laver, Eds., Blackwell, pp. 371-404 (1997).

[Kamakshi et al. 2002] KAMAKSHI V. Prasad, Nagarajan T. and Murthy Hema A. “Continuous Speech Recognition Using Automatically Segmented Data at Syllabic Units“. Department of Computer Science and Engineering. Indian Institute of Technology. Madras, Chennai 600-636, 2002.

[Kirschning 1998] Kirschning Albers Ingrid, “Automatic Speech Recognition with the parallel Cacade Neural network”, PhD Thesis, Tokyo Japan, March 1998.

[Mermelstein 1975] MERMELSTEIN Paul "Automatic Segmentation of Speech into Syllabic Units". Haskins Laboratories, New Haven, Connecticut 06510, pp. 880-883,58 (4), June 1975.

[Oppenheim 89] Oppenheim A. V., Shafer R. W., Digital Dignal Processing, Prentice Hall (1989).

[Rabiner and Biing-Hwang 1993] RABINER Lawrence and Biing-Hwang Juang, “Fundamentals of Speech Recognition”, Prentice Hall, 1993.

[Resch 2001a] RESCH Barbara. "Gaussian Statistics and Unsupervised Learning". A tutorial for the Course Computational Intelligence Signal Processing and Speech Communication Laboratory. www.igi.turgaz.at/lehre/CI, November 15, 2001.

[Resch 2001b]. RESCH Barbara. "Hidden Markov Models". A Tutorial for the Course Computational Laboratory. Signal Processing and Speech Communication Laboratory. www.igi.turgaz.at/lehre/CI, November 15, 2001.

[Zhang 1999]. ZHANG Jialu, “On the syllable structures of Chinese relating to speech recognition”, Institute of Acoustics, Academia Sinica Beijing, China, 1999.

Using Recursive Least Squares Algorithm for Adaptive Filter and Wavelets to Increase Automatic Speech Recognition Rate in Noisy Environment

José Luis Oropeza Rodríguez[1]; Sergio Suárez Guerra[1]

[1] Center for Computing Research, National Polytechnic Institute,
Juan de Dios Batiz esq Miguel Othon de Mendizabal s/n, P.O. 07038, Mexico
joropeza@cic.ipn.mx, ssuarez@cic.ipn.mx

Abstract. This paper shows results obtained in the Automatic Speech Recognition (ASR) task for a corpus of digits speech files with a determinate noise level immerse. In the experiments, we used several speech files that contained Gaussian noise. We used HTK (Hidden Markov Model Toolkit) software of Cambridge University in the experiments. The noise level added to the speech signals was varying from fifteen to forty dB increased by a step of 5 units. We used the Recursive Least Squares Algorithm for an Adaptive Filtering (RLSAAF) to reduce the level noise and two different wavelets (Haar and Daubechies). With RLSAAF we obtained an error rate lower than if it was not present and it was better than wavelets employed for this experiment of Automatic Speech Recognition. For decreasing the error rate we trained with 50% of contaminated and originals signals to the ASR system. The results showed in this paper are focused to try analyses the ASR performance in a noisy environment and to demonstrate that if we are controlling the noise level and if we know the application where it is going to work, then we can obtain a better response in the ASR tasks. Is very interesting to count with these results because speech signal that we can find in a real experiment (extracted from an environment work, i.e.), could be treated with these technique and we can decrease the error rate obtained. Finally, we report a recognition rate of 99%, 97.5% 96%, 90.5%, 81% and 78.5% obtained from 15, 20, 25, 30, 35 and 40 noise levels, respectively when the corpus mentioned before was employed and RLSAAF algorithm was used. Haar wavelet level 1 reached up the most important results as an alternative to RLSAAF algorithm, but only when the noise level was 40 dB and using original corpus.

Keywords. Automatic Speech Recognition, Haar wavelets, Daubechies wavelet, Recursive Least Squares Algorithm for an Adaptive Filtering and noisy speech signal, noisy reduction.

1 Introduction

Speech recognition systems generally assume that the speech signal is a realisation of some message encoded as a sequence of one or more symbols. To effect the reverse operation of recognising the underlying symbol sequence given a spoken utterance, the continuous speech waveform is first converted to a sequence of equally spaced discrete parameter vectors. This sequence of parameter vectors is assumed to form an exact representation of the speech waveform on the basis that for the duration covered by a single vector (typically 10ms or so), the speech waveform can be regarded as being stationary. Although this is not strictly true, it is a reasonable approximation.

Typical parametric representations in common use are smoothed spectra or linear prediction coefficients plus various other representations derived from these.

The role of the recogniser is to effect a mapping between sequences of speech vectors and the wanted underlying symbol sequences. Two problems make this very difficult. Firstly, the mapping from symbols to speech is not one-to-one since different underlying symbols can give rise to similar speech sounds. Furthermore, there are large variations in the realised speech waveform due to speaker variability, mood, environment, etc. Secondly, the boundaries between symbols cannot be identified explicitly from the speech waveform. Hence, it is not possible to treat the speech waveform as a sequence of concatenated static patterns.

The second problem of not knowing the word boundary locations can be avoided by restricting the task to isolated word recognition. As shown in Fig. 1.2, this implies that the speech waveform corresponds to a single underlying symbol (e.g. word) chosen from a fixed vocabulary. Despite the fact that this simpler problem is somewhat artificial, it nevertheless has a wide range of practical applications. Furthermore, it serves as a good basis for introducing the basic ideas of HMM-based recognition before dealing with the more complex continuous speech case. Hence, isolated word recognition using HMMs will be dealt with first.

The different sources of variability that can affect speech determine most of difficulties of speech recognition. During speech production the movements of different articulators overlap in time for consecutive phonetic segments and interact with each other. As a consequence, the vocal tract configuration at any time is influenced by more than one phonetic segment. This phenomenon is known as coarticulation. The principal effect of the coarticulation is that the same phoneme can have very different acoustic characteristics depending on the context in which it is uttered [Farnetani 97].

Speech recognition-system performance is also significantly affected by the acoustic confusability or ambiguity of the vocabulary to be recognized. A confusable vocabulary requires detailed high performance acoustic pattern analysis. Another source of recognition-system performance degradation can be described as variability and noise.

State-of-the-art ASR systems work pretty well if the training and usage conditions are similar and reasonably benign. However, under the influence of noise, these systems begin to degrade and their accuracies may become unacceptably low in severe environments [Deng and Huang 2004]. To remedy this noise robustness issue in ASR due to the static nature of the HMM parameters once trained, various adaptive techniques have been proposed. A common theme of these techniques is the utilization of some form of compensation to account for the effects of noise on the speech characteristics. In general, a compensation technique can be applied in the signal, feature or model space to reduce mismatch between training and usage conditions [Huang at el. 2001].

2 Characteristics and Generalities

Speech recognition systems work reasonably well in quiet conditions but work poorly under noisy conditions or distorted channels. For example, the accuracy of a speech recognition system may be acceptable if you call from the phone in your quiet office, yet its performance can be unacceptable if you try to use your cellular phone in a shopping mall. The researchers in the speech group are working on algorithms to improve the robustness of speech recognition system to high noise levels channel conditions not present in the training data used to build the recognizer

Robustness in speech recognition refers to the need to maintain good recognition accuracy even when the quality of the input speech is degraded, or when the acoustical, articulatory, or phonetic characteristics of speech in the training and testing environments differ. Obstacles to robust recognition include acoustical degradations produced by additive noise, the effects of linear filtering, nonlinearities in transduction or transmission, as well as impulsive interfering sources, and diminished accuracy caused by changes in articulation produced by the presence of high-intensity noise sources. Some of these sources of variability are illustrated in Figure 1. Speaker-to-speaker differences impose a different type of variability, producing variations in speech rate, co-articulation, context, and dialect, even systems that are designed to be speaker independent exhibit dramatic degradations in recognition accuracy when training and testing conditions differ [Cole & Hirschman 92].

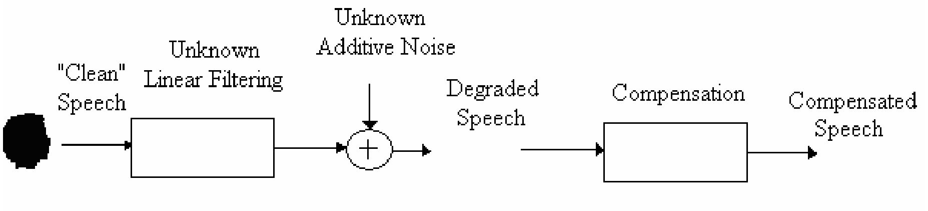


Fig. 1 Schematic representation of some of the sources of variability that can degrade speech recognition accuracy, along with compensation procedures that improve environmental robustness.

3 Automatic Speech Recognition Systems

Automatic Speech Recognition systems generally assume that the speech signal is a realization of some message encoded as a sequence of one or more symbols. The ASR is constitutive by: training and recognition stages. Voice is a static procedure that can to have a duration time between 80-200 ms. a simple but effective mathematical model of the physiological voice production process is the excitation and vocal tract model.

The excitation signal is assumed periodic with a period equal to the pitch for vowels and other voiced sounds, while for unvoiced consonants, the excitation is assumed white noise, i.e. a random signal without dominant frequencies. The excitation signal is subject to spectral modifications while it passes through the vocal tract that has an acoustic effect equivalent to linear time invariant filtering. The model is relevant because, for each type of excitation, a phoneme (or another structural linguistic) is identified mainly by considering the shape of the vocal tract. Therefore, the vocal tract configuration can be estimated by identifying the filtering performed by the tract vocal on the excitation. Introducing the power spectrum of the signal $P_x(\omega)$, of the excitation $P_v(\omega)$ and the spectrum of the vocal tract filter $P_h(\omega)$, we have:

$$P_x(\omega) = P_v(\omega)P_h(\omega) \qquad \textbf{[1]}$$

The speech signal (continuous, discontinuous or isolated) is first converted to a sequence of equally spaced discrete parameter vectors. This sequence of parameter vectors is assumed to form an exact representation of the speech waveform on the basis that for the duration covered by a single vector (typically 10-25 ms) the speech waveform can be regarded as being stationary. Although it is not strictly true, it is a reasonable approximation. Typical parametric representations in common use are smoothed spectra or linear predictive coefficients plus various other representations derived from these. The database employed consists of ten digits (0-9) for the Spanish language. Many of the operations performed by HTK (Hidden Markov Model Toolkit) which involve speech data assumes that the speech is divided into segments and each segment has a name or label. The set of labels associated with the speech data will be the same as corresponding speech file but a different extension.

4 Hidden Markov Models

As we know, HMMs mathematical tool applied for speech recognition presents three basic problems [Rabiner and Biing-Hwang, 1993] y [Zhang 1999]. For each state, the HMMs can use since one or more Gaussian mixtures both to reach high recognition rate and modeling vocal tract configuration in the Automatic Speech Recognition.

Gaussian mixtures

Gaussian Mixture Models are a type of density model which comprise a number of functions, usually Gaussian. These component functions are combined to provide a multimodal density. They can be employed to model the colors of an object in order to perform tasks such as real-time color-based tracking and segmentation. In speech recognition, the Gaussian mixture is of the form [Bilmes 98] [Resch, 2001a], [Resch, 2001b], [Kamakshi et al., 2002] and [Mermelstein, 1975].

:

$$g(\mu,\Sigma)(x) = \frac{1}{\sqrt{2\pi^{d}}\sqrt{\mathbf{det}(\Sigma)}} e^{-\frac{1}{2}(x-\mu)^{T}\Sigma^{-1}(x-\mu)} \qquad \textbf{[2]}$$

Equation 2-3 shows a set of Gaussian mixtures:

$$gm(x) = \sum_{k=1}^{K} w_k * g(\mu_k, \Sigma_k)(x) \qquad \textbf{[3]}$$

In 4, the summarize of the weights give us

$$\sum_{i=1}^{K} w_i = 1 \quad \forall \quad i \in \{1,\dots\dots\dots,K\} \;\; : w_i \geq 0 \qquad \textbf{[4]}$$

Viterbi Trainning

We used Viterbi training, in this a set of training observations O^r , $1 \leq r \leq R$ is used to estimate the parameters of a single HMM by iteratively computing Viterbi alignments. When used to initialise a new HMM, the Viterbi segmentation is replaced by a uniform segmentation (i. e. each training observation is divided into N equal segments) for the first iteration.

Wavelets Transform

This section shows an introductory description about wavelet analysis, includes a discussion of different wavelet functions

Windowed Fourier Transform

The WFT represents one analysis tool for extracting local-frequency information from a signal. The Fourier transform is performed on a sliding segment of length *T* from a time series of time step *δt* and total length *Nδt*, thus returning frequencies from

T−1 to *(2δt)−1* at each time step. The segments can be windowed with an arbitrary function such as a boxcar (no smoothing) or a Gaussian window.

As discussed by Kaiser (1994), the WFT represents an inaccurate and inefficient method of time–frequency localization, as it imposes a scale or "response interval" *T* into the analysis. The inaccuracy arises from the aliasing of high- and low-frequency components that do not fall within the frequency range of the window. The inefficiency comes from the *T/(2δt)* frequencies, which must be analyzed at each time step, regardless of the window size or the dominant frequencies present. In addition, several window lengths must usually be analyzed to determine the most appropriate choice. For analyses where a predetermined scaling may not be appropriate because of a wide range of dominant frequencies, a method of time–frequency localization that is scale independent, such as wavelet analysis, should be employed [Torrence Christopher and Compto Gilbert, 1198].

Wavelet Transform

The *wavelet transform* can be used to analyze time series that contain nonstationary power at many different frequencies (Daubechies 1990). Assume that one has a time series, x_n, with equal time spacing δt and $n = 0 \ldots N - 1$. Also assume that one has a *wavelet function,* $\psi_0(\eta)$, that depends on a nondimensional "time" parameter η. To be "admissible" as a wavelet, this function must have zero mean and be localized in both time and frequency space (Farge 1992). An example is the Morlet wavelet, consisting of a plane wave modulated by a Gaussian:

$$\psi_0(\eta) = \pi^{-1/4} e^{j\omega_0\eta} e^{-\eta^2/2} \qquad \textbf{[5]}$$

where ω_0 is the nondimensional frequency, here taken to be 6 to satisfy the admissi bility condition (Farge 1992). This wavelet is shown in Fig. 2a.

The term "wavelet function" is used generically to refer to either orthogonal or nonorthogonal wavelets. The term "wavelet basis" refers only to an orthogonal set of functions. The use of an orthogonal basis implies the use of the *discrete wavelet transform,* while a nonorthogonal wavelet function can be used

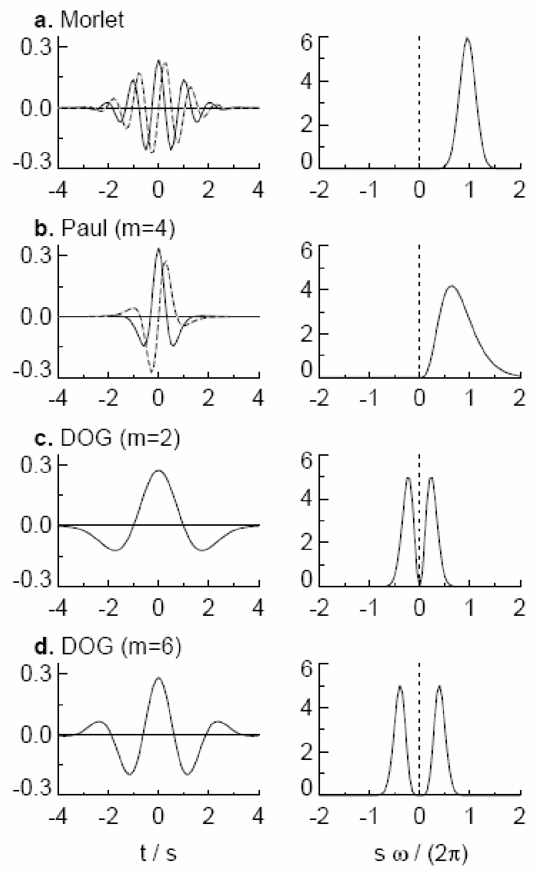


Fig.2. Four different wavelets bases. The plots on the left give the real part (solid) and imaginary part (dashed) for the wavelets in the time domain. The plots on the right give the corresponding wavelets in the frequency domain. (a) Morlet, (b) Paul (m=4), (c) Mexican Hat (m=2), and d) Mexican Hat (m=6)

The continuous wavelet transform of a discrete sequence *xn* is defined as the convolution of x_n with a scaled and translated version of $\psi_0(\eta)$:

$$W_n(s) = \sum_{n'=0}^{N-1} x_{n'} \psi^* \left[\frac{(n'-n)\delta t}{s} \right] \qquad \textbf{[6]}$$

where the (*) indicates the complex conjugate. By varying the *wavelet scale s* and translating along the *localized time index n*, one can construct a picture showing both the amplitude of any features versus the scale and how this amplitude varies with time. The subscript 0 on ψ has been dropped to indicate that this ψ has also been normalized (see next section). Although it is possible to calculate the wavelet transform using (6), it is considerably faster to do the calculations in Fourier space.

To approximate the continuous wavelet transform, the convolution (6) should be done *N* times for each scale, where *N* is the number of points in the time series (Kaiser 1994). (The choice of doing all *N* convolutions is arbitrary, and one could choose a smaller number, say by skipping every other point in *n.*) By choosing *N* points, the convolution theorem allows us do all *N* convolutions simultaneously in Fourier space using a discrete Fourier transform (DFT). The DFT of x_n is

$$\tilde{x}_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N} \qquad \textbf{[7]}$$

where $k = 0 \ldots N - 1$ is the frequency index. In the continuous limit, the Fourier transform of a function $\psi(t/s)$ is given by $\tilde{\psi}(s\omega)$. By the convolution theorem, the wavelet transform is the inverse Fourier transform of the product:

$$W_n(s) = \sum_{k=0}^{N-1} \tilde{x}_k \tilde{\psi}^*(s\omega_k)e^{j\omega_k n\delta t} \qquad \textbf{[8]}$$

where the angular frequency is defined as

$$\omega_k = \begin{cases} \dfrac{2\pi k}{N\delta t}: & k \le \dfrac{N}{2} \\ -\dfrac{2\pi k}{N\delta t}: & k > \dfrac{N}{2} \end{cases} \qquad \textbf{[9]}$$

Using (8) and a standard Fourier transform routine, one can calculate the continuous wavelet transform (for agiven *s*) at all *n* simultaneously and efficiently.

5 Experiments and Results

The evaluation of the algorithm proposed involved clustering a set of speech data consisting of 100 isolated patterns from a digits vocabulary. The training patterns (and a subsequent set of another 200 independent testing pattern) were recorded in a room free of noise. Only one speaker provided the training and testing data. All training and test recordings were made under identical conditions. The 200 independent testing patterns was addition with a level noise, we obtained a total of 1200 new sentences contaminated (200 per noise level, that is because we used 6 noise levels). After that, we used an adaptive filter to reduce that noise level and the results are shown below, then we obtained another 1200 sentences. Finally, we made experiments with a total of 2600 sentences (between noisy, filtered and clean sentences) of speech signal. Figure 3 shows the RLSAAF employed. For each corpus created, we used three databases test to recognition task: with same characteristics, noisy and filtered. All sentences were recorded at 16 kHz frequency rate, 16 bits and mono-channel. We use MFCCs (Mel Frequency Cepstral Coefficients) with 39 characteristics vectors (differential and energy components). A Hidden Markov Model with 5 states and 1 Gaussian Mixture per state.

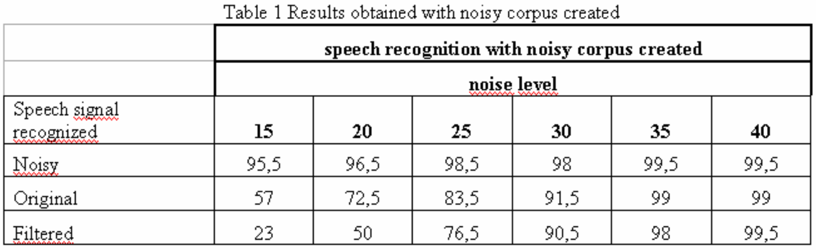
$$
\begin{aligned}
&W_k = W_{k-1} + G_k e_k \\
&P_k = \frac{1}{\gamma}\left[P_{k-1} - G_k x^T(k) P_{k-1}\right] \\
&where \\
&G_k = \frac{P_{k-1} x(k)}{\alpha_k} \\
&e_k = y_k - x^T(k) W_{k-1} \\
&\alpha_k = \gamma + x^T(k) P_{k-1} x(k) \\
&with \\
&x(k) \quad samples \\
&\gamma \quad forgetting \quad factor \quad \gamma = 0.98 \\
&P_k \quad recursive \quad way \quad to \quad \det er \min ate \quad \left[X_k^T X_k\right]^{-1}
\end{aligned}
$$

Fig. 3 Recursive Least Squares Algorithm for an Adaptive Filtering (RLSAAF)

This algorithm stop when the error is lest than 0.9%.

Table 1 shows the results obtained when we used a noisy corpus to training the ASR. A total of 600 speech sentences were analyzed.

Table 1 Results obtained with noisy corpus created



	speech recognition with noisy corpus created					
	noise level					
Speech signal recognized	**15**	**20**	**25**	**30**	**35**	**40**
Noisy	95,5	96,5	98,5	98	99,5	99,5
Original	57	72,5	83,5	91,5	99	99
Filtered	23	50	76,5	90,5	98	99,5

As we can see, when we used a noisy corpus like we hoped, recognition level with noisy database was adequately. When we used high S/N rate (25, 30, 35 and 40 dB), the recognition rate was increased. It is important because it significance that the noisy corpus is a good reference. Figure 4 shows a histogram related with the table contents.

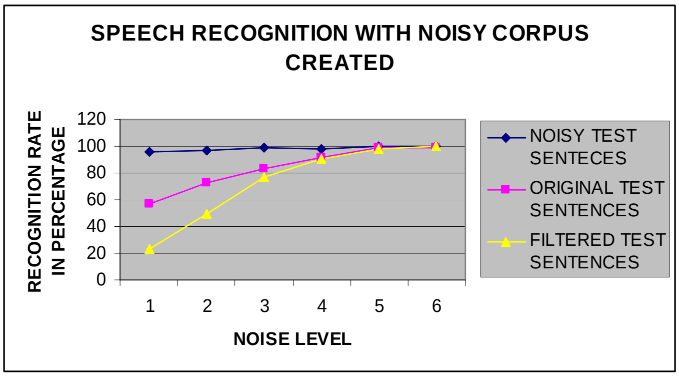
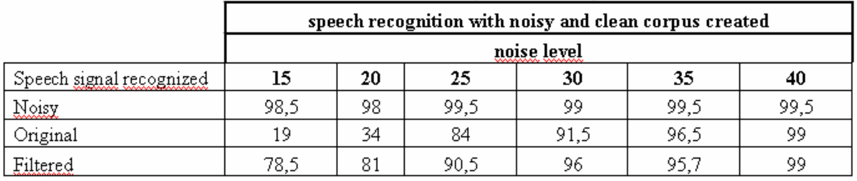


Fig. 4 Graphic representation using noisy corpus created

Table 2 shows the results obtained when we used a noisy and clean corpus to training the ASR. A total of 600 (300 noisy and 300 clean) speech sentences were analyzed.

Table 2 Results obtained with noisy and clean corpus created



	speech recognition with noisy and clean corpus created					
	noise level					
Speech signal recognized	15	20	25	30	35	40
Noisy	98,5	98	99,5	99	99,5	99,5
Original	19	34	84	91,5	96,5	99
Filtered	78,5	81	90,5	96	95,7	99

As we can see, when we used a corpus compound by noisy and original signals, the recognition rate for filtered speech signal was increased considerably. Figure 5 shows that.

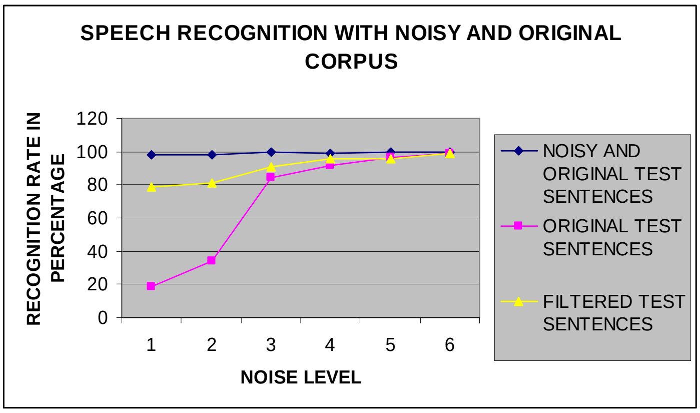
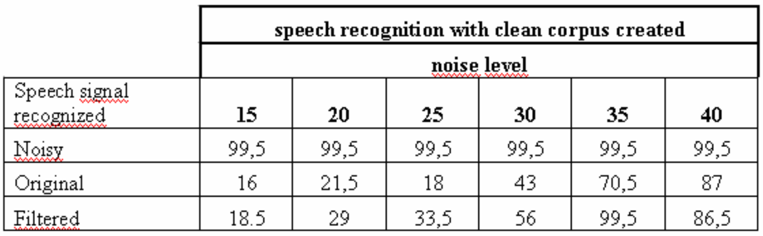


Fig. 5 Graphic representation using noisy and original corpus

Table 3 shows the results obtained when we used a clean corpus to training the ASR. A total of 600 speech sentences were analyzed.

Table 3 Results obtained with clean corpus created



	speech recognition with clean corpus created					
	noise level					
Speech signal recognized	15	20	25	30	35	40
Noisy	99,5	99,5	99,5	99,5	99,5	99,5
Original	16	21,5	18	43	70,5	87
Filtered	18.5	29	33,5	56	99,5	86,5

With the original corpus the results was not satisfactory, although the recognition rate with filtered signals was better than noisy signals, it was poor and not enough to be considered important as figure 6 shows.

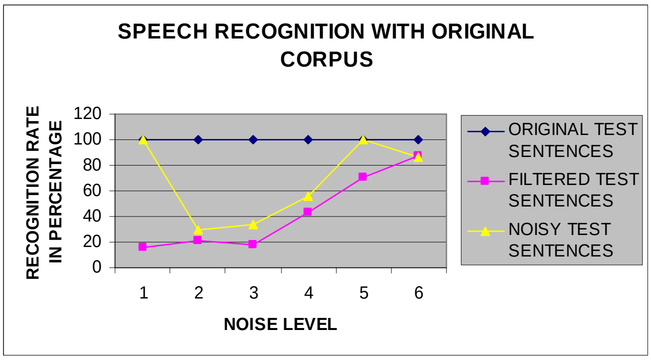
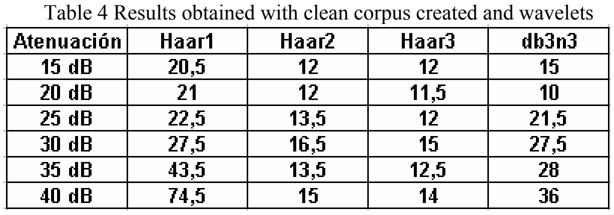


Fig. 6 Graphic representation using original corpus created

Finally, we probed different wavelets to try to determine better results than we obtained above. The results were not that we hoped.

Table 4 Results obtained with clean corpus created and wavelets



Atenuación	Haar1	Haar2	Haar3	db3n3
15 dB	20,5	12	12	15
20 dB	21	12	11,5	10
25 dB	22,5	13,5	12	21,5
30 dB	27,5	16,5	15	27,5
35 dB	43,5	13,5	12,5	28
40 dB	74,5	15	14	36

As we can see in figure 7, only Haar 1 wavelet at 40 dB had a high performance in ASR rate. We consider that results obtained were failed because noisy level selected before to apply wavelet transform must be changed. But we consider that it only can not help us so much.

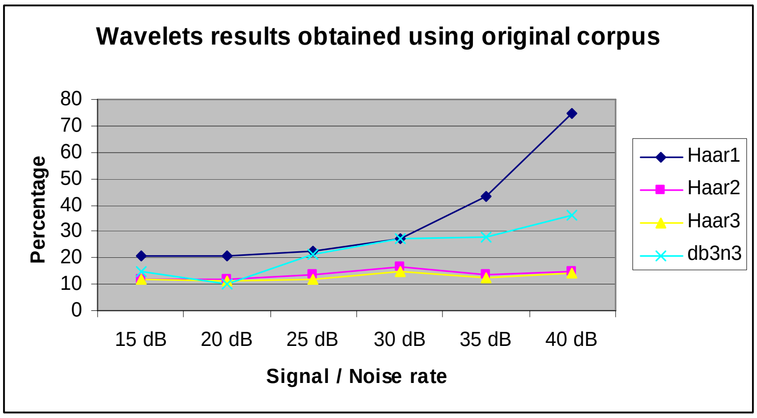


Fig. 7 Graphic representation for ASR using wavelets and original corpus

6 Conclusions and future works

The results shown in this paper demonstrate that we can use an adaptive filter to reduce the noise level in an automatic speech recognition system (ASRS) for the Spanish language. The use of this paradigm is not new but with this experiment we propose to reduce the problems find out when we tread with real speech signals. MFCCs and CDHMMs (Continuous Density Hidden Markov Models) were used for training and recognition, respectively. First, when we used database test with the same characteristics that corpus training a high performance was reached out, but when we used the clean speech database our recognition rate was poor. The most important results extracted of this experiment were when the clean speech was mixed with noisy speech, when we used filtered speech we obtained a high performance in our ASR.

For that, our conclusion is that if we want to construct an ASR immerse in a noisy environment, it is going to have a high performance if we included in our database training clean and noisy speech signal. So, if we known the Signal/Noise ratio and it are greater than 35%, we can use the filtered signal in an ASR without problems. For future works is recommendable try to probe the results obtained using another methods employed to reduce noise into signal (wavelets i. e.), and extract the results.

References

[Bilmes 98] BILMES J.A., "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models", International Computer Science Institute, Berkeley, CA. 1998.

[Cole & Hirschman 92] R. A. Cole, L. Hirschman, et al. Workshop on spoken language understanding. Technical Report CSE 92-014, Oregon Graduate Institute of Science & Technology, P.O.Box 91000, Portland, OR 97291-1000 USA, September 1992.

[Deng, Li. and Huang, X. (2004)] Challenges in Adopting Speech Recognition. Communications of the ACM, Vol. 47, No. 1, pp. 69-75.

[Farnetani 97] Farnetani E., "Coarticulation and connected speech processes", in the Handbook of Phonetic Sciences, W. Hardcastle and J. Laver, Eds., Blackwell, pp. 371-404 (1997).

[Huang, C., Wang, H. and Lee, C. (2001)] An ASR Incremental Stochastic Matching Algorithm for Noisy Speech Recognition. IEEE Trans. Speech and Audio Processing, Vol 9, No. 8, pp. 866-873.

[Kamakshi et al. 2002] KAMAKSHI V. Prasad, Nagarajan T. and Murthy Hema A. "Continuous Speech Recognition Using Automatically Segmented Data at Syllabic Units". Department of Computer Science and Engineering. Indian Institute of Technology. Madras, Chennai 600-636, 2002.

[Mermelstein 1975] MERMELSTEIN Paul "Automatic Segmentation of Speech into Syllabic Units". Haskins Laboratories, New Haven, Connecticut 06510, pp. 880-883,58 (4), June 1975.

[Rabiner and Biing-Hwang 1993] RABINER Lawrence and Biing-Hwang Juang, "Fundamentals of Speech Recognition", Prentice Hall, 1993.

[Torrence Christopher and Compto Gilbert, 1198] Torrence Christopher and Compo Gilbert P. Program in Atmospheric and Oceanic Sciences, University of Colorado, Boulder, Colorado. Bulletin of the American Meteorological Society, Vol. 79, No. 1, January 1998.

[Zhang 1999]. ZHANG Jialu, "On the syllable structures of Chinese relating to speech recognition", Institute of Acoustics, Academia Sinica Beijing, China, 1999.

FPGA Implementation of Alfa-Beta Associative Memories

Mario Aldape-Pérez, Cornelio Yáñez-Márquez, and
Amadeo José Argüelles-Cruz

Center for Computing Research, CIC
National Polytechnic Institute, IPN
Mexico City, Mexico
mario@aldape.org.mx; cyanez@cic.ipn.mx; jamadeo@cic.ipn.mx
WWW home page: http://www.aldape.org.mx

Abstract. Associative memories have a number of properties, including a rapid, compute efficient best-match and intrinsic noise tolerance, that make them ideal for many applications. However, a significant bottleneck to the use of associative memories in real-time systems is the amount of data that requires processing. Notwithstanding, Alfa-Beta Associative Memories have been widely used for color matching in industrial processes [1], text translation [2] and image retrieval applications [3]. The aim of this paper is to present the work that produced a dedicated hardware design, implemented on a field programmable gate array (FPGA) that applies the Alfa-Beta Associative Memories model for pattern recognition tasks. Along the experimental phase, performance of the proposed associative memory architecture is measured by learning large sequences of symbols and recalling them successfully. As a result, a simple but efficient embedded processing architecture that overcomes various challenges involved in pattern recognition tasks is implemented on a Xilinx Spartan3 FPGA.

Keywords: Associative Memories, FPGA, Pattern Recognition, Reconfigurable Logic.

1 Introduction

An associative memory M is a system that relates input patterns and output patterns as follows: $x \longrightarrow \boxed{\mathbf{M}} \longrightarrow y$ with x and y, respectively, the input and output pattern vectors. Each input vector forms an association with its corresponding output vector. For each k integer and positive, the corresponding association will be denoted as: (x^k, y^k). Associative memory M is represented by a matrix whose ij-th component is m_{ij} [4]. Memory M is generated from an *a priori* finite set of known associations, called the fundamental set of associations. If μ is an index, the fundamental set is represented as: $\{(x^\mu, y^\mu) \mid \mu = 1, 2, ..., p\}$ with p as the cardinality of the set. The patterns that form the fundamental set are called fundamental patterns. If it holds that $x^\mu = y^\mu$ $\forall \mu \in \{1, 2, ..., p\}$ M is

© A. Argüelles, J. L. Oropeza, O. Camacho, O. Espinosa (Eds.)
Computer Engineering.
Research in Computing Science 30, 2007, pp. 27-36

Table 1. Alfa and Beta Operators.

$\alpha : A \times A \longrightarrow B$

x	y	α(x,y)
0	0	01
0	1	00
1	0	10
1	1	01

$\beta : B \times A \longrightarrow A$

x	y	β(x,y)
00	0	0
00	1	0
01	0	0
01	1	1
10	0	1
10	1	1

auto-associative, otherwise it is heteroassociative; in this case, it is possible to establish that $\exists \mu \in \{1, 2, ..., p\}$ for which $x^\mu \neq y^\mu$. If we consider the fundamental set of patterns $\{(x^\mu, y^\mu) \mid \mu = 1, 2, ..., p\}$ where n and m are the dimensions of the input patterns and output patterns, respectively, it is said that $x^\mu \in A^n$, $A = \{0, 1\}$ and $y^\mu \in A^m$. Then the j-th component of an input pattern is $x_j^\mu \in A$. Analogously, the j-th component of an output pattern is represented as $y_j^\mu \in A$. A distorted version of a pattern x^k to be recuperated will be denoted as $\widetilde{x}^k$. If when feeding an unknown input pattern x^ω with $\omega \in \{1, 2, ..., k, ..., p\}$ to an associative memory M, it happens that the output corresponds exactly to the associated pattern y^ω, it is said that recuperation is perfect.

2 Alfa-Beta Associative Memories

Alfa-Beta Associative Memories mathematical foundations are based on two binary operators: α and β. Alfa operator is used during the learning phase while Beta operator is used during the recalling phase. The mathematical properties within these operators, allow the $\alpha\beta$ associative memories to exhibit similar characteristics to the binary version of the morphological associative memories, in the sense of: learning capacity, type and amount of noise against which the memory is robust, and the sufficient conditions for perfect recall [5]. First, we define set $A = \{0, 1\}$ and set $B = \{00, 01, 10\}$, so α and β operators can be defined as in Table 1.

These two binary operators along with maximum ($\vee$) and minimum ($\wedge$) operators establish the mathematical tools around the Alfa-Beta model. The definitions of α and β exposed in Table 1, imply that: α is increasing by the left and decreasing by the right, β is increasing by the left and right, β is the left inverse of α. According to the type of operator that is used during the learning phase, two kinds of Alfa-Beta Associative Memories are obtained. If maximum operator ($\vee$) is used, Alfa-Beta Associative Memory of type MAX will be obtained, denoted as M; analogously, if minimum operator ($\wedge$) is used, Alfa-Beta Associative Memory of type min will be obtained, denoted as W [6]. In any case, the fundamental input and output patterns are represented as follows:

$$x^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n \qquad y^\mu = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \in A^m$$

In order to understand how the learning and recalling phases are carried out, some matrix operations definitions are required.

α max Operation: $P_{mxr} \nabla_\alpha Q_{rxn} = \left[f_{ij}^\alpha\right]_{mxn}$, where $f_{ij}^\alpha = \vee_{k=1}^{r} \ \alpha(p_{ik}, q_{kj})$
β max Operation: $P_{mxr} \nabla_\beta Q_{rxn} = \left[f_{ij}^\beta\right]_{mxn}$, where $f_{ij}^\beta = \vee_{k=1}^{r} \ \beta(p_{ik}, q_{kj})$
α min Operation: $P_{mxr} \mathit{\Delta}_\alpha Q_{rxn} = \left[f_{ij}^\alpha\right]_{mxn}$, where $f_{ij}^\alpha = \wedge_{k=1}^{r} \ \alpha(p_{ik}, q_{kj})$
β min Operation: $P_{mxr} \mathit{\Delta}_\beta Q_{rxn} = \left[f_{ij}^\beta\right]_{mxn}$, where $f_{ij}^\beta = \wedge_{k=1}^{r} \ \beta(p_{ik}, q_{kj})$

Whenever a column vector of dimension m is operated with a row vector of dimension n, both operations ∇_α and $\mathit{\Delta}_\alpha$, are represented by $\oplus$; consequently, the following expression is valid:

$$y \nabla_\alpha x^t = y \oplus x^t = y \mathit{\Delta}_\alpha x^t.$$

If we consider the fundamental set of patterns $\{(x^\mu, y^\mu) \,| \mu = 1, 2, ..., p\}$ then the ij-th entry of the matrix $y^\mu \oplus (x^\mu)^t$ is expressed as follows:

$$\left[y^\mu \oplus (x^\mu)^t\right]_{ij} = \alpha(y_i^\mu, x_j^\mu).$$

2.1 Learning Phase

Find the adequate operators and a way to generate a matrix M that will store the p associations of the fundamental set $\left\{\left(x^1, y^1\right), \left(x^2, y^2\right), ..., (x^p, y^p)\right\}$, where $x^\mu \in A^n$ and $y^\mu \in A^m$ $\forall \mu \in \{1, 2, ..., p\}$.

Step 1. For each fundamental pattern association $\{(x^\mu, y^\mu) \,| \mu = 1, 2, ..., p\}$, generate p matrices according to the following rule:

$$\left[y^\mu \oplus (x^\mu)^t\right]_{mxn}$$

Step 2. In order to obtain an Alfa-Beta Associative Memory of type MAX, apply the binary MAX operator ($\vee$) according to the following rule:

$$M = \vee_{\mu=1}^{p} \left[y^\mu \oplus (x^\mu)^t\right]$$

Step 3. In order to obtain an Alfa-Beta Associative Memory of type min, apply the binary min operator ($\wedge$) according to the following rule:

$$W = \wedge_{\mu=1}^{p} \left[y^\mu \oplus (x^\mu)^t\right]$$

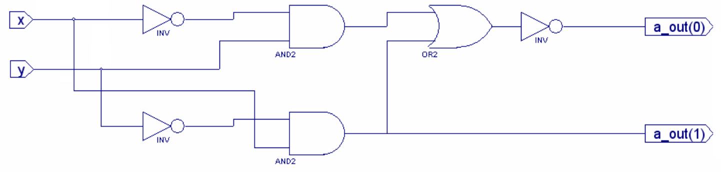


Fig. 1. Alfa Unit

Consequently, the ij-th entry of an Alfa-Beta Associative Memory of type MAX is given by the following expression:

$$\nu_{ij} = \vee_{\mu=1}^{p} \alpha(y_i^\mu, x_j^\mu)$$

Analogously, the ij-th entry of an Alfa-Beta Associative Memory of type min is given by the following expression:

$$\psi_{ij} = \wedge_{\mu=1}^{p} \alpha(y_i^\mu, x_j^\mu).$$

2.2 Recalling Phase

Find the adequate operators and sufficient conditions to obtain the fundamental output pattern y^μ, when either the memory M or the memory W is operated with the fundamental input pattern x^μ.

Step 1. A pattern x^ω, with $\omega \in \{1, 2, ..., p\}$, is presented to the Alfa-Beta Associative Memory, so x^ω is recalled according to one of the following rules.

Alfa-Beta Associative Memory of type MAX:

$$M \mathit{\Delta}_\beta x^\omega = \wedge_{j=1}^{n} \beta(\nu_{ij}, x_j^\omega) = \wedge_{j=1}^{n} \left\{ \left[\vee_{\mu=1}^{p} \alpha(y_i^\mu, x_j^\mu) \right], x_j^\omega \right\}$$

Alfa-Beta Associative Memory of type min:

$$W \bigtriangledown_\beta x^\omega = \vee_{j=1}^{n} \beta(\psi_{ij}, x_j^\omega) = \vee_{j=1}^{n} \left\{ \left[\wedge_{\mu=1}^{p} \alpha(y_i^\mu, x_j^\mu) \right], x_j^\omega \right\}$$

Without dependence on the Alfa-Beta Associative Memory type used throughout the recalling phase, a column vector of dimension n will be obtained.

3 Implementation Details

As previously mentioned, the main goal of this paper is to derive an efficient implementation of the Alfa-Beta Associative Memories targeted towards FPGAs.

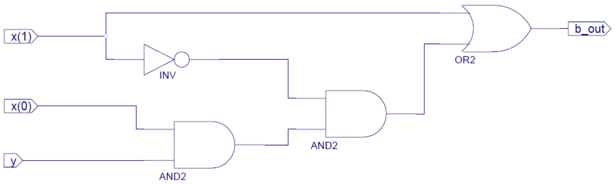


Fig. 2. Beta Unit

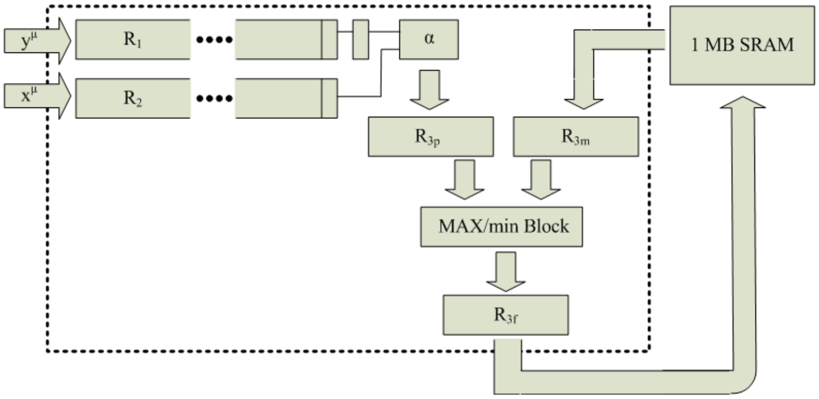


Fig. 3. Learning Phase Architecture

The Alfa operator implementation is shown in Figure 1, while the Beta operator implementation is shown in Figure 2.

The proposed architecture works with a 50 MHz master clock, which implies a 20ns period. As is it shown in Figure 3, the learning phase is implemented with 5 registers, 1 Alfa block, 1 MAX/min block and 2 external 10ns SRAM chips (mounted on the same board), that allow 1MB of data storage. There are two remarkable topics to be taken into consideration. The former concerns about the amount of logic resources that are needed to implement the two binary operators (Alfa and Beta). The latter results from the fact that most of the components that constitute the learning phase are combinatorial circuits. Hence, it is possible to read data from the external SRAM memory at the same time that a new bit is shifted to the Alfa block. Therefore, it is possible to write back the result of the MAX/min block to the external SRAM memory, during the same clock period.

As it is shown in Figure 4, the recalling phase is implemented with 4 registers, 1 Beta block, 1 min/MAX block and the same 2 external 10ns SRAM chips that were used to store the fundamental associations during the learning phase. The recalling phase is executed as follows. Firstly, R_{3m} receives one data word from

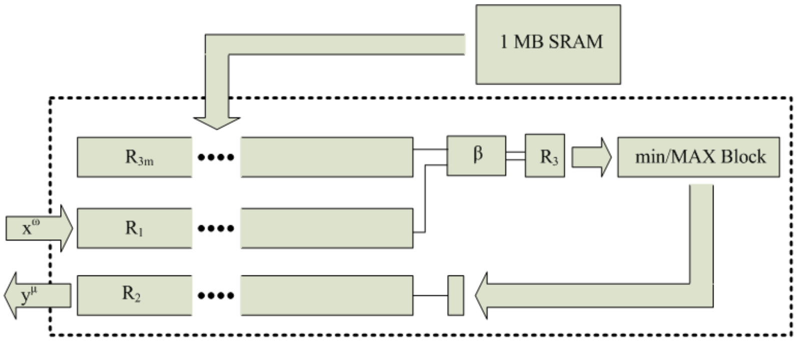


Fig. 4. Recalling Phase Architecture

the Alfa-Beta Associative Memory (stored in the 2 external 10ns SRAM chips). Then, R_1 receives the unknown input pattern. Finally, R_2 stores the recalled output pattern.

4 Numerical Results

Example 4.1. Let $p = 5$, $n = 4$, $m = 4$. Given the fundamental patterns $\{(x^\mu, y^\mu) \mid \mu = 1, 2, ..., p\}$, obtain an Alfa-Beta Associative Memory. The fundamental associations will be denoted as: $\{(x^1, y^1), (x^2, y^2), ..., (x^5, y^5)\}$.

$$x^1 = \begin{pmatrix}1\\1\\0\\1\end{pmatrix} \quad x^2 = \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \quad x^3 = \begin{pmatrix}0\\1\\1\\1\end{pmatrix} \quad x^4 = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \quad x^5 = \begin{pmatrix}1\\0\\1\\1\end{pmatrix}$$

$$y^1 = \begin{pmatrix}1\\1\\0\\1\end{pmatrix} \quad y^2 = \begin{pmatrix}1\\0\\0\\1\end{pmatrix} \quad y^3 = \begin{pmatrix}1\\1\\1\\1\end{pmatrix} \quad y^4 = \begin{pmatrix}1\\1\\0\\0\end{pmatrix} \quad y^5 = \begin{pmatrix}1\\0\\1\\1\end{pmatrix}$$

Learning phase. Obtain the corresponding matrices $M_1, M_2, \ldots, M_5$, according to step 1, indicated in section 2.1.

$$y^1 \oplus \left(x^1\right)^t = \begin{pmatrix}1\\1\\0\\1\end{pmatrix} \oplus \begin{pmatrix}1\;1\;0\;1\end{pmatrix} = \begin{pmatrix}1\;1\;2\;1\\1\;1\;2\;1\\0\;0\;1\;0\\1\;1\;2\;1\end{pmatrix}$$

$$y^2 \oplus \left(x^2\right)^t = \begin{pmatrix}1\\0\\0\\1\end{pmatrix} \oplus \begin{pmatrix}0\;0\;0\;1\end{pmatrix} = \begin{pmatrix}2\;2\;2\;1\\1\;1\;1\;0\\1\;1\;1\;0\\2\;2\;2\;1\end{pmatrix}$$

$$\vdots$$

$$y^5 \oplus \left(x^5\right)^t = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 1\,0\,1\,1 \end{pmatrix} = \begin{pmatrix} 1\,2\,1\,1 \\ 0\,1\,0\,0 \\ 1\,2\,1\,1 \\ 1\,2\,1\,1 \end{pmatrix}$$

According to step 2, an Alfa-Beta Associative Memory of type MAX denoted by M, is obtained. Analogously, according to step 3, an Alfa-Beta Associative Memory of type min denoted by W, is obtained.

$$M = \begin{pmatrix} 2\,2\,2\,2 \\ 2\,1\,2\,2 \\ 2\,2\,1\,1 \\ 2\,2\,2\,1 \end{pmatrix} \quad ; \quad W = \begin{pmatrix} 1\,1\,1\,1 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 1\,0\,1\,1 \end{pmatrix}$$

Recalling phase. Obtain the corresponding output patterns, by performing the operations $M \vartriangle_\beta x^\mu$, $\forall \mu \in \{1, 2, ..., p\}$ as stated in section 2.2. Due to paper space limitations, only the Alfa-Beta MAX type recalling phase results are shown.

$$M \vartriangle_\beta x^1 = \begin{pmatrix} 2\,2\,2\,2 \\ 2\,1\,2\,2 \\ 2\,2\,1\,1 \\ 2\,2\,2\,1 \end{pmatrix} \vartriangle_\beta \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = y^1$$

$$M \vartriangle_\beta x^2 = \begin{pmatrix} 2\,2\,2\,2 \\ 2\,1\,2\,2 \\ 2\,2\,1\,1 \\ 2\,2\,2\,1 \end{pmatrix} \vartriangle_\beta \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = y^2$$

$$M \vartriangle_\beta x^3 = \begin{pmatrix} 2\,2\,2\,2 \\ 2\,1\,2\,2 \\ 2\,2\,1\,1 \\ 2\,2\,2\,1 \end{pmatrix} \vartriangle_\beta \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = y^3$$

$$M \vartriangle_\beta x^4 = \begin{pmatrix} 2\,2\,2\,2 \\ 2\,1\,2\,2 \\ 2\,2\,1\,1 \\ 2\,2\,2\,1 \end{pmatrix} \vartriangle_\beta \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = y^4$$

$$M \vartriangle_\beta x^5 = \begin{pmatrix} 2\,2\,2\,2 \\ 2\,1\,2\,2 \\ 2\,2\,1\,1 \\ 2\,2\,2\,1 \end{pmatrix} \vartriangle_\beta \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = y^5$$

The reader can easily verify that the Alfa-Beta *min* type recalling phase also recalls the whole fundamental set of patterns perfectly.

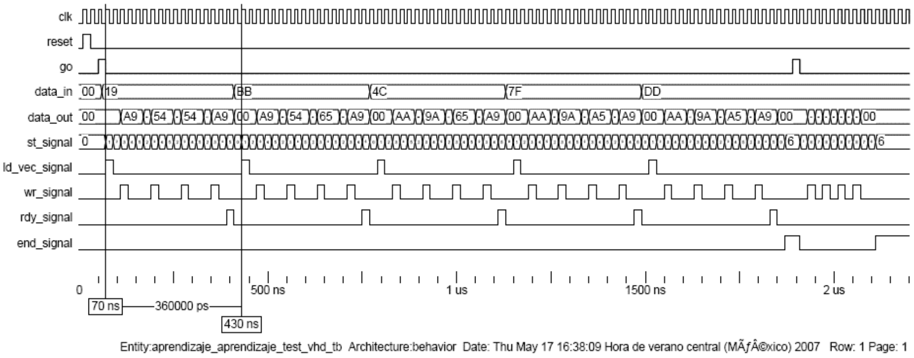


Fig. 5. Learning Phase

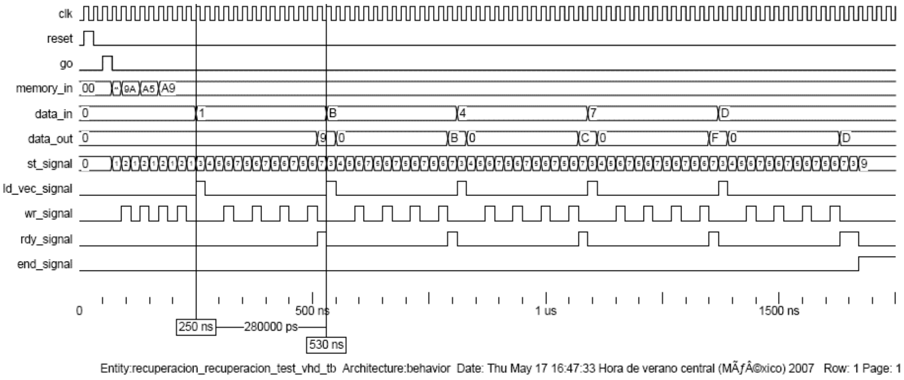


Fig. 6. Recalling Phase

5 Experimental Results

The experimental phase was carried out in two stages. In the first one, the same fundamental set of patterns that was presented in section 4, was downloaded to the proposed architecture. The performance results are shown in Figure 5 and Figure 6. The learning phase is executed in 2.5 μs and the recalling phase is executed in 2 μs. As expected, the entire fundamental set of patterns was perfectly recalled.

In order to estimate how the Alfa-Beta Associative Memory model performs with high dimensional data, 20 binary images (Figure 7 and Figure 8) were

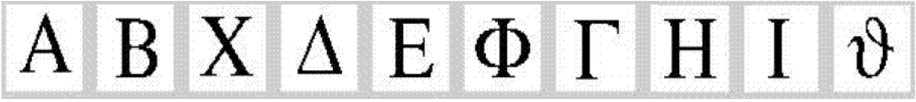


Fig. 7. Fundamental Input Patterns

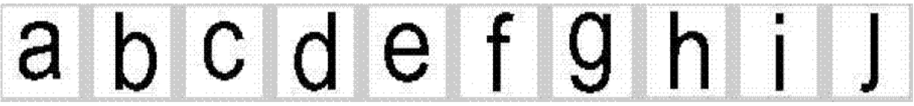


Fig. 8. Fundamental Output Patterns

used as fundamental patterns. Each one of these images is 40 by 40 pixels, which produces a 1600 bits pattern; accordingly, each pattern association results in a 640 Kbytes matrix. The experimental phase was carried out as follows: after the register initialization process was concluded, the first association was learned and recalled. Subsequently, the first and second associations were learned and recalled; after that, the same procedure continued in a consecutive manner until the fundamental set of patterns was completely learned and recalled. The above mentioned procedure was executed 100 times, each time changing the fundamental associations randomly. The averaged recalling results are shown in Table 2. A relevant thing to mention about the recalling criterion that was used along the experimental phase is that, in this case, perfect recall means that all of the 1600 bits were exactly recovered. Particularly, outstanding results were achieved by using the Alfa-Beta min type recall (the whole fundamental set of patterns was perfectly recalled).

6 Conclusions and Ongoing Research

In this paper, we introduced a simple but efficient implementation of the Alfa-Beta Associative Memories targeted towards FPGAs that overcomes a serious challenge in pattern recognition tasks (bottle-neck problems due to high dimensional data). A relevant thing to mention is that after a fundamental pattern is downloaded to the proposed architecture, each bit is learned in 90 ns, which fulfills one of the main purposes of this paper. Moreover, if the learning rate is known, it is possible to estimate the learning phase duration even with high dimensional fundamental patterns. Usually, this situation takes place when the fundamental patterns are RGB images. It is worth to mention that the proposed architecture can be easily adapted to work as an Alfa-Beta bi-directional associative memory.

Table 2. Fundamental set recalling results.

	1	2	3	4	5	6	7	8	9	10
$\alpha\beta$ MAX	1	2	3	4	4	4	4	3	1	1
$\alpha\beta$ min	1	2	3	4	5	6	7	8	9	10

Currently, we are investigating how to use the proposed architecture for feature selection in RGB images and mixed noise variants. We are also working towards a parallel implementation of the learning phase, based on recent mathematical results.

Acknowledgments The authors of the present paper would like to thank the following institutions for their support: National Polytechnic Institute, Mexico (CIC, CGPI, PIFI, COFAA) , CONACyT and SNI.

References

[1] Yáñez-Márquez, C., Felipe-Riverón, E. M., López-Yáñez, I., & Flores-Carapia, R. (2006). A Novel Approach to Automatic Color Matching. Lecture Notes in Computer Science (LNCS), 4225, 529-538.

[2] Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., & López-Yáñez, I. (2006). Alpha-Beta Bidirectional Associative Memories Based Translator. International Journal of Computer Science and Network Security (IJCSNS), 6, 190-194.

[3] Yáñez-Márquez, C., Sánchez-Fernández, L. P., & López-Yáñez, I. (2006). Alpha-Beta Associative Memories for Gray Level Patterns. Lecture Notes in Computer Science (LNCS), 3971, 818-823.

[4] Kohonen, T. (1972). Correlation Matrix Memories. IEEE Transactions on Computers, 21, 353-359.

[5] Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., & López-Yáñez, I. (2006). A New Model of BAM: Alpha-Beta Bidirectional Associative Memories. Lecture Notes in Computer Science (LNCS), 4263, 286-295.

[6] Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., & López-Yáñez, I. (2007). Alpha–Beta bidirectional associative memories: theory and applications. Neural Processing Letters, 26, 1-40.

A DWT Domain Video Watermarking Algorithm Using Chaotic Mixing Technique

Rogelio Reyes, Clara Cruz, Mariko Nakano, Héctor Pérez

SEPI ESIME Culhuacan, National Polytechnic Institute of México
Av. Santa Ana 1000 Col. San Francisco Culhuacan, CP 04430, México, City. MÉXICO
rogreyes@gmail.com, mariko@calmecac.esimecu.ipn.mx

Abstract. Based on the imperceptibility and robustness requirement for video content protection, a watermarking algorithm for digital video with blind extraction is presented in this paper. In practice, the robustness depends on the watermark embedding energy used, which is limited by the media degradation caused by the watermark. Reliable embedding algorithms must assure watermark persistence up to an extreme quality downgrading. The proposed algorithm embeds binary visually recognizable pattern, such as owner's logotype, in DWT domain of some randomly selected video shots. To increase security, watermark data is disordered by chaotic mixing method before its embedding and to increase watermark imperceptibility, blue channel of RGB color space is used its embedding. The experimental results demonstrate the watermark imperceptibility and robustness against several video degradations and attacks. Also we show that extracted watermark data from watermarked video sequence in blind manner is sufficiently clear after several attacks.

Key words: Video watermarking, Copyright protection, Blind detection, Discrete Wavelet Transform

1 Introduction

With the fast development of multimedia application and communication systems, including digital broadcasting, the growth of the internet and especially popular internet software based on a peer-to-peer architecture has been used to share movies, music and other digital materials. Due to that duplication and modification of digital content are realized in very easy manner using digital editor tools provided for personal computer, illegal copying and distribution of digital data have been serious problems for intellectual property protection. Technical solutions for copyright protection of multimedia data are actively being pursued; two typical technologies are encryption and watermarking. Digital watermarking is the technique which embeds the copyright information into digital content, unlike encryption, watermarking does not restrict access to the data. A watermark is designed to permanently reside in the

© A. Argüelles, J. L. Oropeza, O. Camacho, O. Espinosa (Eds.)
Computer Engineering.
Research in Computing Science 30, 2007, pp. 37-46

host data [1].While encrypted data once is decrypted, intellectual property rights are no longer protected.

A watermarking scheme for digital video should include the capability to support following requirements: the embedded watermark in video data should be imperceptible and difficult to remove without degrading the perceptual quality of video, it means if embedded watermark is removed from video sequence, the video sequence has lost commercial value any more. In addition to above two requirements, watermarking technique used for digital video should extract the watermark information without using original video data (blind detection). The current issue of the watermarking becomes to get robustness to the signal distortion and tampering caused by intentional or unintentional attacks and common signal processing, including frame averaging, frame dropping, frame swapping, etc. [2],[3],[4].

There are three embedding positions for video watermark, i.e. base band video stream, at the same time when video is being compressed and in the compressed video stream. The watermark signal can be a random sequence with one information bit or multiple bit meaningful information, such as logotype. The random sequence watermark is more robust in general; however, embedding meaningful watermark is more important in some application. Therefore in this paper, we propose a video watermarking scheme that embeds a meaningful watermark in a base band video stream based on Discrete Wavelet Transform (DWT) and the blind extraction process extract watermark clearly from watermarked video.

The paper is organized as follows. In Section 2 and 3, the watermark embedding and extraction process of the proposed algorithm are described, respectively. Experimental results are presented in Section 4 and finally in Section 5, some conclusions are drawn.

2 Watermark Embedding

2.1 Watermark Preprocessing

A binary watermark image W (with values -1 or 1) is disordered based on chaotic mixing method [5], this procedure makes a mapping $A_N(k)$ with $L_N \rightarrow L_N$, where L_N is a two-dimensional indexes set, and it is computed iteratively by (1).

$$W^d = W^{(i)} = A_N^i(k)W^{(0)}, \quad i = 1,2,\ldots,P-1,$$

$$A_N(k) = L_N \rightarrow L_N, \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} (\mathrm{mod}\, N). \tag{1}$$

where $W^{(i)}$ is an *i-th* disordered watermark image, $W^{(0)}$ is a original watermark image, $(x_n, y_n) \in L_N$, $k \in [1, N] \subset Z$, P is a recurrent time depended on k, and N is the size of $W^{(0)}$. Two keys are required to reconstruct the watermark image: k is a first key and the iterations number i is a second one. An example of watermark image disorder is given by figure 1.

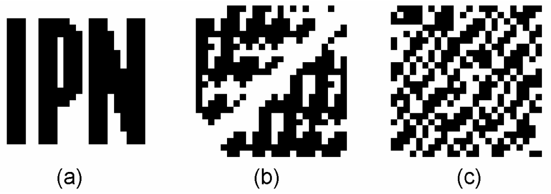


Fig. 1. (a) Original watermark, (b) Watermark processed with Chaotic Mixing using a key *k=5* and iteration *i=5* and (c) Watermark preprocessed with Chaotic Mixing using a key *k=5* and iteration *i=10*.

2.2 Video Watermark Embedding Algorithm

First the host video is segmented into video sequences, and then some of them are selected randomly for watermark embedding. For each selected sequence watermark is embedded. The watermark embedding scheme is shown by figure 2.

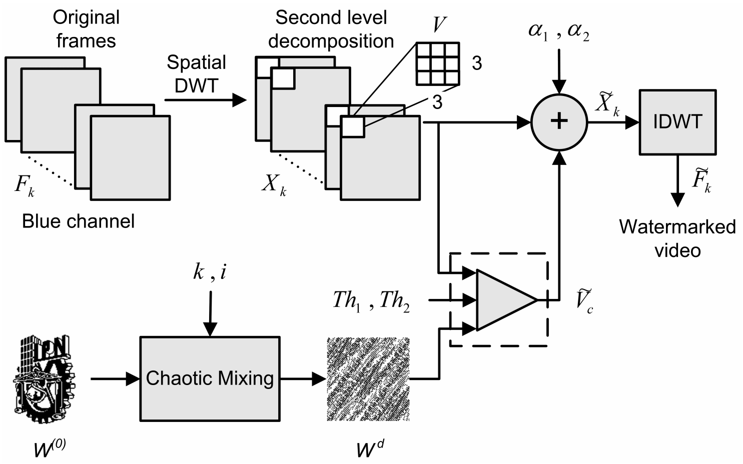


Fig. 2. Flowchart of video watermark embedding scheme.

The following steps are repeated for each frame of the selected video sequence and the process into the broken line frame in the figure 2 is described as follows.

A) Perform a two dimensional DWT based on the Daubechies-wavelet on each frame of the blue color channel, which is denoted by $F_k\ (k=1,2,\ldots,n)$. The multilevel decomposition is processed until the second level. Then the watermark is embedded into the LL_2 subband. Here wavelet coefficient frames of F_k are denoted by $X_k\ (k=1,2,\ldots,n)$.

B) Watermark embedding is based on magnitudes of the wavelet coefficients; disordered watermark image is adaptively spread spectrum and embedded into these coefficients. Wavelet coefficients X_k are divided into blocks with size 3x3 pixels, then a mean of the block V is computed and denoted as M. Watermark bit is embedded by changing the center coefficient value V_c of each block V with the corresponding modified value $\tilde{V}_c$, using the following rules shown below.

   i. Assign values to two thresholds Th_1, Th_2, and also to two intensity factors a_1 and a_2.

   ii. Calculate difference d_1 between V_c and M.

$$d_1 = \left|V_c - M\right| \tag{2}$$

   iii. Depending on d_1 value with respect to Th_1, Th_2 and the corresponding bit of the pixel $W^d(i,j)$, V_c is modified:

      1) If $d_1 > Th_1$, then watermark is not inserted.

      2) If $d_1 < Th_1$, then watermark is inserted according to the following cases:

         a) If $V_c > M$, $W^d(i,j)=1$ and $d_1 < Th_2$, then the center coefficient value V_c of the block is modified using (3).

$$\tilde{V}_c = V_c + \frac{9}{8}\left(Th_2 - d_1\right) + a_1 \tag{3}$$

         b) If $V_c > M$, $W^d(i,j)=-1$ and $d_1 > Th_2$, then the center coefficient value V_c of the block is modified using (4).

$$\tilde{V}_c = V_c - \frac{9}{8}\left(d_1 - Th_2\right) - a_2 \tag{4}$$

         c) If $V_c < M$, $W^d(i,j)=1$ and $d_1 < Th_2$, then the center coefficient value V_c of the block is modified using (5).

$$\tilde{V}_c = V_c - \frac{9}{8}\left(Th_2 - d_1\right) - a_1 \tag{5}$$

d) If $V_c < M$, $W^d\left(i, j\right) = -1$ and $d_1 > Th_2$, the center coefficient value V_c of the block is modified using (6).

$$\tilde{V}_c = V_c + \frac{9}{8}\left(d_1 - Th_2\right) + a_2 \tag{6}$$

C) The watermarked video is dbtained computing the inverse DWT of the modified wavelet coefficient frames in the blue color channel $\tilde{X}_k$ and recombining the color channels.

Watermark image is embedded into a video by adaptive spread spectrum based on the characteristics of video data, in which the watermark was preprocessed with Chaotic Mixing method; so each pixel of the disordered watermark is embedded into a 3x3 block of magnitudes of the wavelet coefficients for the corresponding frame in the blue color channel. As mentioned above each watermark data is related to the static and dynamic composition in the video sequence offering better robustness to malicious attacks.

3 Blind Watermark Extraction

To extract the watermark, the original unmarked video and original watermark data are not required. A watermark can be extracted easily according to scheme shown in figure 3.

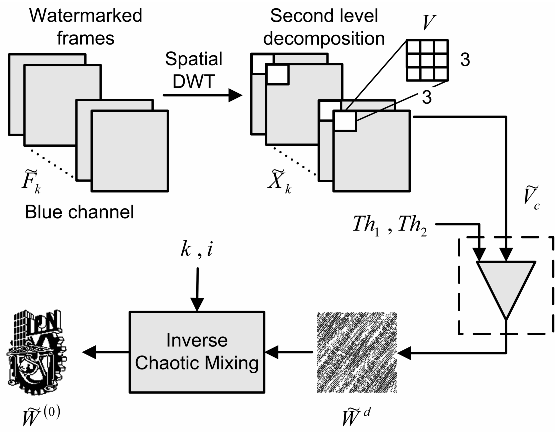


Fig. 3. Flowchart of video watermark blind detection.

The following steps are repeated for each frame of the selected sequence and the detail process into the broken line frame in the figure 3 is describe as follows.

A) Again, the process starts with the two dimensional wavelet decomposition of the blue color channel frames $\tilde{F}_k\,(k=1,2,\ldots,n)$ of each video sequence selected in embedding scheme. Applying DWT to $\tilde{F}_k$, we can retrieve the *LL₂* coefficients, these are denoted by $\tilde{X}_k\,(k=1,2,\ldots,n)$.

B) Wavelet coefficients $\tilde{X}_k$ are divided in blocks with size 3x3 pixels, and then a mean of the block $\tilde{V}$ is computed and denoted $\tilde{M}$, extract the center coefficient value $\tilde{V}_c$ of each block $\tilde{V}$, and to obtain the corresponding value of the pixel $\tilde{W}_k^d(i,j)$ in these block $\tilde{V}$ following the rules shown below.

   i. Thresholds values Th_1 and Th_2 are known.

   ii. Calculate difference $\tilde{d}_1$ between $\tilde{V}_c$ and $\tilde{M}$.

$$\tilde{d}_1 = \left|\tilde{V}_c - \tilde{M}\right| \tag{7}$$

   iii. Depending of $\tilde{d}_1$ value with respect to Th_1 and Th_2 the corresponding value of the pixel $\tilde{W}_k^d(i,j)$, is obtained according with:

      1) $\tilde{d}_1 > Th_1$, then $\tilde{W}_k^d(i,j)=0$.

      2) $\tilde{d}_1 < Th_1$ then a watermark bit is extracted according to the following cases:

         a) $\tilde{d}_1 \geq Th_2$, therefore $\tilde{W}_k^d(i,j)=1$.

         b) $\tilde{d}_1 < Th_2$, therefore $\tilde{W}_k^d(i,j)=-1$.

C) Calculate a mean of the disordered watermark embedded in each frame of the selected sequences using (8).

$$\hat{W}^d(i,j) = \frac{1}{k}\sum_{k=1}^{K}\tilde{W}_k^d(i,j) \tag{8}$$

Where K is the total number of frames used in the embedding process, $\tilde{W}_k^d(i,j)$ is the watermark bit in the *k-th* frame of video.

D) Disordered watermark image $\tilde{W}^d$ can be formed using (9).

$$\tilde{W}^d(i,j) = \begin{cases} 1 & \hat{W}^d(i,j) > 0 \\ -1 & \hat{W}^d(i,j) < 0 \end{cases} \tag{9}$$

E) The reconstructed watermark image $\tilde{W}^{(0)}$ starting from $\tilde{W}^{d}$, can be obtained through the two keys used in the watermarking preprocessing section k and i, by calculating the inverse matrix of $A_N(k)$ iteratively until the number of iterations i using (10) where $\tilde{W}^{(i)} = \tilde{W}^{d}$.

$$\tilde{W}^{(0)} = A_N^{-i}(k)\tilde{W}^{(i)} = A_N^{P-i}(k)\tilde{W}^{(i)} \quad \mathbf{(10)}$$

4 Experimental Results

The proposed algorithm has been evaluated using well-known video sequences "Foreman", "carphone" and "bus". These sequences are in the YUV color space in terms of one luma and two chrominance components, and their size are 288*352 (CIF format) [6]. The watermark is a binary image with size 24*24.

In the experiments we use 60 frames in these sequences to embed a watermark image and used the following values to the thresholds and intensity factors: $Th_1 = 70$, $Th_2 = \frac{1}{2}Th_1$, $a_1 = \frac{9}{16}(Th_1 - Th_2)$ and $a_2 = \frac{Th_2}{4}$.

We have conducted some experiments to evaluate the robustness of our proposed watermarking algorithm. For this purpose, we have performed some of the classical sequence manipulations including: noise attack, frame dropping, frame averaging, frame swapping. In the experiments, we have tested 10 times each video sequence with different frames (scenes), in which the results are average values of independent experiments.

Figure 4 demonstrates the invisibility of the embedded binary watermark image in the proposed algorithm. Similar results can be obtained with other video sequences.

4.1 Noise Contamination

The watermarked video is corrupted by adding impulsive noise and gaussian noise with different intensity. The performance of video watermark approach is calculated in terms of normalized correlation between original watermark and reconstructed watermark defined as (11). Simulation results are shown in figure 5.

$$C\left(\tilde{W}^{(0)}, W^{(0)}\right) = \sum_i \sum_j \left(\tilde{W}^{(0)}(i,j) \cdot W^{(0)}(i,j)\right) \Big/ \sum_i \sum_j W^{(0)}(i,j)^2 \quad \mathbf{(11)}$$

Fig. 4. Demonstration of invisibility. (a) The original “Foreman” frame, (b) Watermarked “Foreman” frame, (c) The original “bus” frame and (d) Watermarked “bus” frame.

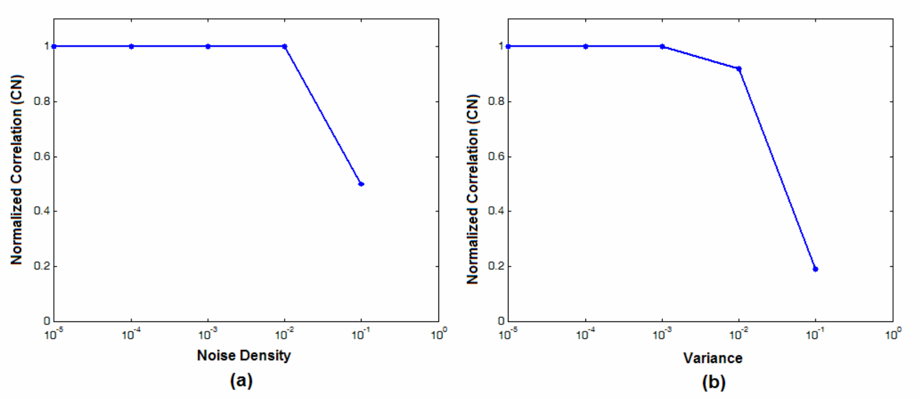


Fig. 5. Robustness to noise contamination, (a) robustness to impulsive noise with different noise densities, (b) robustness to Gaussian noise with different variances.

The results shown in figure 5 demonstrate the robustness of our video watermarking procedure against several noise contamination when up to 7% of the watermarked frame has been modified with impulsive or Gaussian noise, also video’s commercial value will be lost if its contaminated more than or closer to 10% due to lower quality.

4.2 Frame Dropping , Frame Averaging and Frame Swapping Attacks

Based on the existence of temporal redundancy in video data, frame dropping or frame cutting, which removed some frames from the video sequence, this is an effective video watermark attack since it doesn’t damage the video signal. Frame Dropping attack is given by (12).

$$V_{attacked} = V_{original} - \{F_{r1}, F_{r2}, \ldots\ldots F_{rn}\} \tag{12}$$

where $V_{attacked}, V_{original}$ are attacked and original video signals, and $F_{r1,} F_{r2}, \ldots.$ are some video frames.

Frame averaging attack compute the average of the two nearest neighbors' frames to replace the actual frame is calculated as (13).

$$F_k'(i,j) = \frac{1}{3}\left[F_{k-1}(i,j) + F_k(i,j) + F_{k+1}(i,j)\right] \tag{13}$$

Frame swapping attack can destroy some dynamic composition of the video signal and embedded watermark. This attack is formulated as (14), simulation results of the attacks are presented in figure 6.

$$F_k(i,j) \Leftrightarrow F_{k+1}(i,j), \quad k = 1,3,5,\ldots,n-1 \tag{14}$$

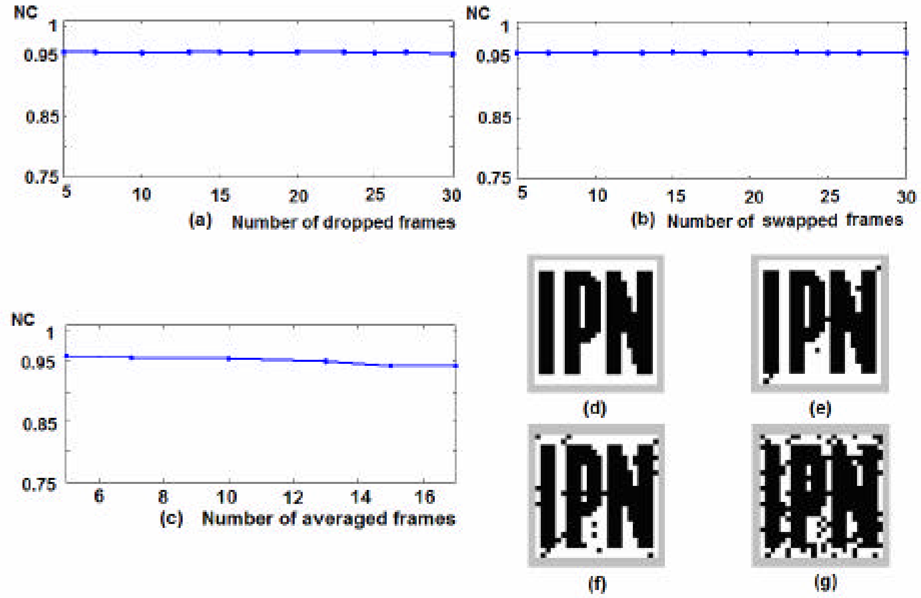


Fig. 6. Robustness to (a) frame dropping, (b) frame swapping, (c) frame averaging and extracted watermark with different normalized correlation respect to the original one: (d) original watermark, (e) extracted watermark with NC=0.9792, (f) extracted watermark with NC=0.9021, (g) extracted watermark with NC=0.7326.

Figure 6 demonstrates that our watermarking method is able to resist Frame Attacks, in all of these cases, the embedded watermark can be recovered with a little or no damage even when up to 50% of the watermarked frames has been modified.

In all analyzed attacks, normalized correlation values between original watermark and extracted one is higher than 0.95, which indicates that the extracted watermark image is very clear after watermarked video sequence is attacked.

We compare the proposed algorithm with different schemes [4], [7] and [8]. The results are shown in Table 1 'O' means that the embedded watermark is robust against attack and 'X' means that embedded watermark can not be recovered correctly.

Table 1. Performance comparison of the proposed scheme.

	Frame Attack			Noise Contamination	
	Averaging	Dropping	Swapping	Impulsive	Gaussian
Our scheme	O	O	O	O	O
Zhuang [4]	O	O	O	O	X
Zhao [7]	O	O	O	X	X
Zhang [8]	X	O	O	O	O

5 Conclusions

The video watermarking algorithm here presented is robust to various common attacks. This approach allows us to exploit the advantages of the DWT, HVS characteristic to improve invisibility and robustness of the visual recognizable watermark, such as logotype. The extraction process performs without original video sequence and original watermark data and extracted watermark image doesn't have any ambiguity.

Acknowledgments. This work is supported by the National Polytechnic Institute of México and National Council for Science and Technology of Mexico.

References

1. E. T. Lin, A. M. Eskicioglu, R. L. Lagendijk, and E. J. Delp.: Advances in Digital Video Content Protection. In: Proceedings of the IEEE, Vol. 93. No. 1, Jan. 2005. 171-183.
2. M. Maes, T. Kalker, J.-P. M. G. Linnartz, J. Talstra, G. F. G. Depovere, and J. Haitsma,: Digital watermarking for DVD video copy protection. In: IEEE Signal Processing Magazine, Vol. 17. Sep. 2000. 47–57.
3. M. D. Swanson, B. Zhu, and A. H. Tewfik. Multiresolution Scene-Based Video Watermarking Using Perceptual Models. In: IEEE Journal on Selected Areas in Communications. Vol. 16. May 1998. 540–550.
4. H. Zhuang, Y. Li and C. Wu.: A Blind Spatial-temporal Algorithm Based on 3D Wavelet for Video Watermarking. In: IEEE International Conference on Multimedia and Expo. (ICME) , vol. 3, 2004. 1727-1730.
5. G. Voyatzis and I. Pitas.: Embedding Robust Watermarks by Chaotic Mixing. In: Proceedings of Conf. Int. Digital Signal Processing, Vol. 1. 1997. 213-216.
6. K. N. Plataniotis and A. N. Venetsanopoulos.: Color Image Processing and Applications. Springer-Verlag Berlin. 2000. 20-39.
7. Z. Zhao, N. Yu and X. Li: A Novel Video Watermarking Scheme in Compressed Domain Based of Fast Motion Estimation. In: IEEE Int. Conf. on Communication Technology (ICCT). 2003. 1878-1882.
8. J. Zhang, J. Li, and L. Zhang.: Video Watermark Technique in Motion Vector. In: XIV Brazilian Symposium on Computer Graphics and Image Processing. 2001.

Implementation of a Biometric Iris Recognition System for Person Identification

Miguel Colores[1], Mireya García[1], Alejandro Ramírez[2],

[1] Instituto Politécnico Nacional-CITEDI, Av. Del Parque No.1310, Tijuana BC,
colores, mgarciav@citedi.mx
[2] Dpto R&D, PILIMTEC, Châteaugiron,Francia
alramirez10@yahoo.fr

Abstract. Iris recognition has received increasing attention in recent years because their biometrics features can be used on security systems applications, due to their great advantages, such as variability, stability and security. This paper presents a description of a biometric system for person identification based on the iris pattern. For the system implementation some known techniques are proposed for the segmentation, encoding and pattern comparison stages of the recognition**.**

Keywords: Biometrics, coding, identification, iris pattern.

1 Introduction

Biometric identification technologies, including facial recognition, fingerprint recognition, speaker verification and others, offer a new solution for personal identification, because they offer a more natural way for performing identification [1][2]. Technologies that use biometrics features have a potential application identifying individuals in order to control access to secured areas or services. Nowadays a lot of biometric techniques are being developed based on different features and algorithms. Each technique has its advantages and limitations, and it's not possible to determine which is the best, without considering first the application environment. Nevertheless, it is well known that, from all of these techniques, iris recognition is one of the most promising for high security applications. The possibility that the iris (Fig. 1.a) can be used as a kind of optical fingerprint for person identification was first suggested by ophthalmologists. Automated systems for person identification based on iris recognition are considered to be the most reliable among all biometric methods; this because a biometric identification system making many comparisons between different patterns of persons can give false results, but the probability of finding two people with identical iris pattern is almost zero [3]. For this reason, iris recognition technology is becoming an

© A. Argüelles, J. L. Oropeza, O. Camacho, O. Espinosa (Eds.)
Computer Engineering.
Research in Computing Science 30, 2007, pp. 47-56

important biometric solution for people identification in applications such as access control. Other applications include immigration controls using biometric passports [4].

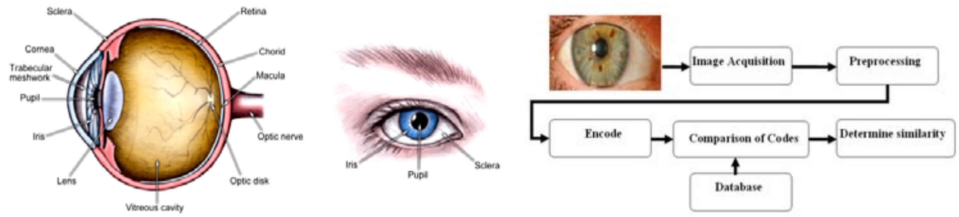


Fig. 1. **a**) Iris location: Lateral and frontal views. b) Iris Recognition System.

Only two iris recognition prototypes had been developed: one by J. Daugman [5], and the other by W. Wildes [6]. Most of the commercial iris recognition systems use the algorithms proposed by Daugman. One of the most successful commercial iris recognition systems is distributed by Iridian Technologies [7] and it uses algorithms patented by Daugman. That company has developed the entire recognition system (Fig. 1.b) which consists of the following steps: Firstly, an image of the user's eye is captured by the system (Images Acquisition). Then, the image is processed to normalize the scale and illumination of the iris. This region is localized and separated of the eye image (Preprocessing). Thirdly, features that represent the iris patterns are extracted and a code is generated (Encode). And finally, a decision is made by matching the code generated by the iris (comparison of codes and Determine Similarity).
The objective of this article is to present the implementation of an iris recognition system based in mathematical algorithms derived from the information found in open source literature. Since it is knows, the algorithms used in commercial system (Daugman's system) are not available. In this article the different concepts for every stage in an iris recognition system was developed by our-self implementing in C language in order to verify the system performance and to validate the chosen options for every stage. In section 2, we will introduce the iris acquisition device. Section 3 will discuss the preprocessing. Section 4 is about iris feature extraction. Section 5 will discuss the comparison of codes and how is made the decision when codes generates by iris are matching. The experiments and results will be presented in the section 6. Finally conclusions will be drawn in section 7.

2 Iris Image Acquisitions

This stage is important and difficult in an iris recognition system. Since iris is small in size and dark in color, is difficult to acquire good images for analysis using the standard CCD camera and ordinary lighting. It is necessary to use a special device for images acquisition which can deliver eyes image with sufficiently high quality while that remains as system noninvasive to the human operator. We need to considerate image acquisition with sufficient resolution and sharpness to support recognition, good contrast without resorting to a level of illumination that annoys the operator, and the optical aberrations and reflections should be eliminated as much as possible. One schematic diagram for image acquisition well-known because fulfills the requirements, was develop by John Daugman (Fig. 2.a).

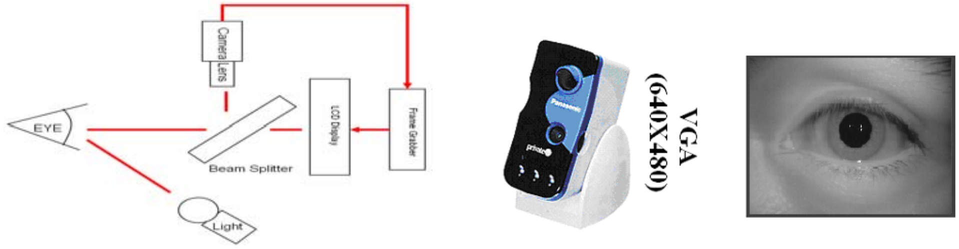


Fig. 2. a) Schematic diagram of an image acquisition system. b) Example of an iris image captured with a camera BMET100US.

The system don't use color information, use monochrome cameras with 8 bits gray-level resolution, because the color information provide additional discriminatory power. The positioning of the iris for image capture is concerned with framing the entire iris in the camera's field of view with good focus, so the system requires that the operator positions itself his eye region in front of the camera. For this reason the diagram provides a live video feedback via a miniature liquid crystal display placed in line with the camera's optics via a beam splitter. This allows the operator to see what the camera is capturing and to adjust his position accordingly [8]. In our experiments we work with a camera BMET100US of Panasonic (Fig. 2.b).

3 Preprocessing

An iris image contains some irrelevant parts (eg. eyelid, sclera, pupil, etc) (See Fig. 1.a). Even for the iris of the same eye, its size may vary depending on eye to camera distance as well as light brightness. Therefore, before matching the original image needs to be preprocessed to localize and normalize the iris.

3.1 Iris Localization

In this stage the iris is detected and isolated from the image, namely, finding both the inner (papillary) and the outer (sclera boundaries) (see Fig. 1.a). The captured image is a 2-D array and is described as I(x, y) where the point (x, y) is the gray-level. The first eye part to isolate is the pupil, knowing that the pupil region is generally darker than its surroundings (Fig. 3-c) that is used to fix a region threshold to binarize the image I(x, y) (Fig. 3-a) to obtain an image Ib(x, y) (Fig. 3-b), the image is scanning in the horizontal direction to find the longest chord which will be the diameter of the pupil.

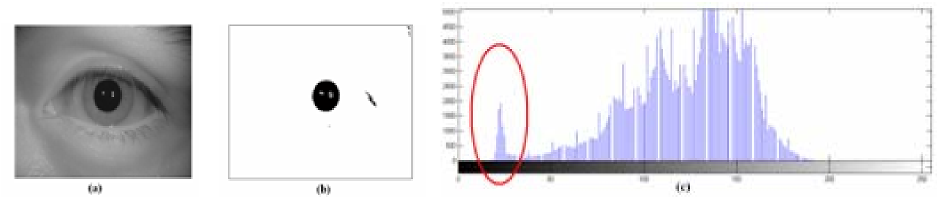


Fig. 3. (a) Original image I(x,y). (b)Binary image Ib(x,y) (c) Gray levels for the pupil region.

Thus the centroide coordinates of the pupil are gives by the followings equations:

$$x_p = x + \frac{D(x,y)}{2} \tag{1}$$

$$y_p = yy_p = y \tag{2}$$

$$r_p(ratio) = \frac{D(x,y)}{2} \tag{3}$$

Where D is the longest chord, and (x, y) is the start point of D.
The outer boundary of the iris is more difficult to detect because of the low contrast between the two sides of the boundary. Then Canny edge detection is performed to create an edge map to generate gradients information [9]. Circular Hough Transform which is employed by Wildes [10], is used to detecting the iris-sclera boundary to obtain a new centre (x_s,y_s) and radio r_s , also the linear Hough Transform [11] is used to detect and isolate eyelids and eyelashes.

3.2 Iris Normalization

Once the pupil and sclera boundaries are extracted and the iris ring is localized, the captured iris image which always varies in size is normalized into a rectangular block by a Cartesian to polar reference transform, suggested by J.Daugman [5]. In this way is compensate the stretching of the iris texture as the pupil changes in size, and is unfold the frequency information contained in the circular texture in order to facilitate next features extraction. Moreover, this new representation allows to separating the iris from the pupil.
Thus the polar transform is implementing by the following mapping (Fig. 4)

$$I(x(\rho,\theta), y(\rho,\theta)) \rightarrow I(\rho,\theta) \tag{4}$$

where

$$x(\rho,\theta) = (1-\rho) * x_p(\theta) + \rho * x_s(\theta) \tag{5}$$

$$y(\rho,\theta) = (1-\rho) * y_p(\theta) + \rho * y_s(\theta) \tag{6}$$

The θ ($\theta \in [0;2\pi]$) and ρ ($\rho \in [0;1]$) parameters dimensionless describe the polar coordinate system. I(x,y) is the region image,(x,y) are the original Cartesian coordinates (r, θ) are the corresponding normalised polar coordinates, $(x_p(\theta),y_p(\theta))$ and $(x_s(\theta),y_s(\theta))$ are the coordinates of the iris ring along the θ direction.

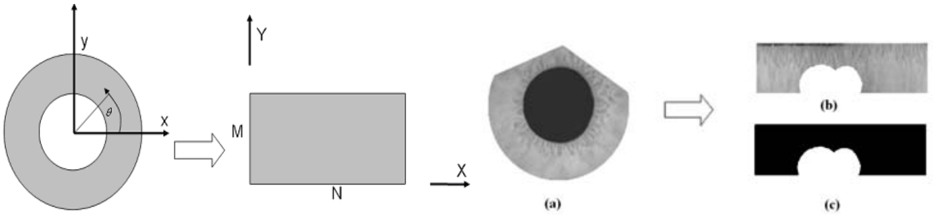


Fig. 4. Iris rectangular representation. a) Iris localization on an eye image. b) Iris mask. c) Noise mask.

Besides to generate the normalized iris image (Fig. 4.b) "iris mask" is necessary to generate another image called "noise mask" (Fig. 4.c). This mask indicates the regions of the normalized iris where the pattern of the iris is obstructed by the eyelids, eyelashes, etc. The noise mask has the same dimensions that the iris mask, its used in the comparison stage to avoid compares the obstructed regions.

4 Iris feature encode

This stage is to extract the information of the iris pattern and needed to be encoded for matching purposes. The iris features are obtained convolving the normalized iris pattern with a 1D Log-Gabor filters. Gabor Filters based methods have been widely used as feature extractor in computer vision, especially for texture analysis [12]. Daugman [5, 13] used multi-scale Gabor wavelets to extract phase structure information of the iris texture. However, Field [14] has examined that there is a disadvantage of the Gabor Filter in which the even symmetric filter will have a DC component whenever the bandwidth is larger than one octave. To overcome this disadvantage, another type of filter is used known as Log-Gabor, which is Gaussian on a logarithmic scale, can be used to produce zero DC components for any bandwidth. The Log-Gabor filters are obtained by multiplying the radial and angular components together where each even and odd symmetric pair of Log-Gabor filters comprises a complex Log-Gabor filter at one scale. The frequency response of a Log-Gabor Filters is given as:

$$G(f) = e^{\left(\frac{-(\log(f/f_0))^2}{2(\log(\beta/f_0))^2}\right)} \qquad (7)$$

Where f_0 represents the central frequency, and β bandwidth of the filters.

For the implementation of 1D Log-Gabor Filter is chosen to be the feature extractor of iris since 1D Log-Gabor Filters is an improved version of Gabor Filters. By applying 1D

Log-Gabor Filters, 2D normalized pattern is divided into a number of 1D signals, and these 1D signals are convolved with 1D Gabor wavelet. The rows of the 2D normalized pattern are taken as the 1D signal; each row corresponds to a circular ring on the iris region. The angular direction is taken rather than the radial one, which corresponds to columns of the normalized pattern, since maximum independence occurs in the angular direction. The row of the image is X = [P(1), P(2), ... ,P(N)]. First is applied a discrete Fourier transform (DFT) on the vector X to get vector Y (eq.8). Then multiply vector Y and a 1-D log-Gabor wavelet to get the vector Z (eq.9). The 1D Log-Gabor function is defined (eq.7), only one filter is used, with f_0 =12, and a bandwidth β = 0.5 [15]. Finally, using the inverse DFT on the vector Z to get the vector D (eq.10).

$$Y_k = \sum_{n=1}^{N} x_n e^{-j2\pi(k-1)(n-1)/N} \tag{8}$$

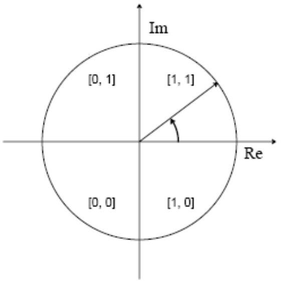
$$z_k = Y_k \times G_k \tag{9}$$

$$D_k = \frac{1}{N}\sum_{n=1}^{N} z_k e^{-j2\pi(k-1)(n-1)/N} \tag{10}$$

The output h of the Gabor filter is complex numbers $h=h_{Re}+h_{Im}$, where the phase angle at each output point is quantized to two bits depending on the quadrant where this each element in the complex plane (Fig. 5). Thus for an image normalized with size (MxN) the resulting template is of size (Mx2N). The template size with radial resolution of 128 pixels and angular resolution of 256 pixels was chosen. These parameters generate an iris template that contains 32768 bits of information.

5 Comparison of codes

For this process is used the Hamming distance (HD) that gives a measure of how many bits are the same between two bit patterns. Using the Hamming distance of two bit patterns, a decision can be made as to whether the two patterns were generated from different irises or from the same one. Since an individual iris region contains many features, each iris region will produce a bit-pattern which is independent to that produced by another iris, and two iris codes produced from the same iris will be highly correlated. If two bits patterns are completely independent, such as iris templates generated from different irises, the Hamming distance between the two patterns should equal 0.5. This occurs because independence implies the two bit patterns will be totally random, so there is 0.5 chance of setting any bit to 1, and vice versa. Therefore, half of the bits will agree and half will disagree between the two patterns. If two patterns are derived from the same iris, the Hamming distance between them will be close to 0.0, since they are highly correlated and the bits should agree between the two iris codes. For matching, the Hamming distance was chosen as a metric for recognition, since bit-wise comparisons were necessary. The Hamming distance algorithm employed also incorporates noise masking, so that only significant bits are used in calculating the Hamming distance between two iris templates (A,B).



$$h_{\Re e} = \begin{cases} \text{"1"} & if \quad \mathrm{Re} \geq 0 \\ \text{"0"} & if \quad \mathrm{Re} < 0 \end{cases}$$

$$h_{\mathrm{Im}} = \begin{cases} \text{"1"} & if \quad \mathrm{Im} \geq 0 \\ \text{"0"} & if \quad \mathrm{Im} < 0 \end{cases}$$

Fig. 5. How is quantize in 2 bits the filter output

Both iris patterns (maskA,maskB) are used in the calculation. The Hamming distance will be calculated using only the bits generated from the true iris region given by the equation:

$$HD(A,B) = \frac{1}{\sum_{i,j} C(i,j)} \sum_{i,j} (A(i,j) \; xor \; B(i,j) \; \& \; C(i,j) \tag{11}$$

where

$$C(i,j) = \begin{cases} 1 \; if \;\; maskA(i,j) = 0 \;\; and \quad maskB(i,j) = 0; \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad otherwise \end{cases} \tag{12}$$

Although, in theory, two iris templates generated from the same iris will have a Hamming distance of 0.0, in practice this will not occur. Normalization is not perfect, and also there will be some noise that goes undetected, so some variation will be present when comparing two intra-class iris templates. In order to account for rotational inconsistencies, when the Hamming distance of two templates is calculated, one template is shifted left and right bit-wise and a number of Hamming distance values are calculated from successive shifts. This bit-wise shifting in the horizontal direction corresponds to rotation of the original iris region by an angle given by the angular resolution used. To calculate HD between A and B, we fix the code A, and shift the code B from -15^{o} to $+15^{o}$ with an increment of 1.5^{o}. The minimum HD from these shift positions is used as the reported HD. The recognition is based on a test of statistical independence obtained comparing iris patterns (binary codes) generated by same and different eyes [16]. In the figure 6.a the curve “Intra-class” shows the HD between two iris templates generated by one eye, and the curve “Inter-class” shows the HD between the iris templates generated by different eyes, as a function of the rotation amount applied. The fig. 6.b shows a typical curves of probability distribution of the distance Hamming. (Intra-class) generated by comparing patterns generated by the same eye and (Inter-class) comparing patterns generated by comparing patterns by different eyes.

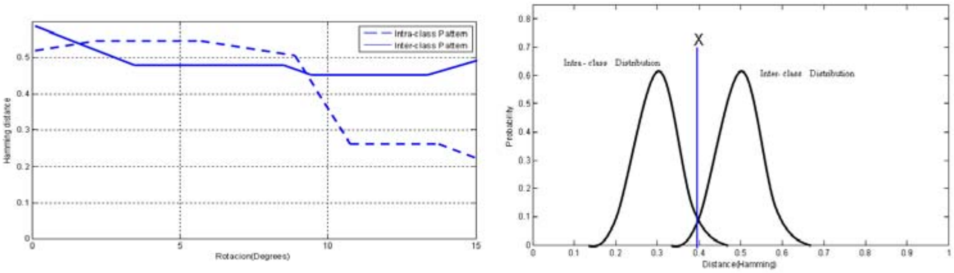


Fig. 6. a) Show an example of the reported HD for different rotations. b) Typical curves of probability distribution of the distance Hamming.

To make a recognition decision can be defined a threshold X. If the obtained distance of Hamming is minor who thresholds X, is decided that the compared codes were generated by same eye, otherwise is decided that they were generated by different eyes. Both distributions Inter-class and intra-class, generally, are overlapped. For that reason, the area under the distribution Inter-class to the left of threshold X, represents the probability of a false identification, whereas the area under the distribution intra-class to the right of threshold X, represents the probability of a false rejection. These measures are known as false acceptance rate (FAR) and false rejection rate (FRR). The distance for both distributions ("decidability") can be calculate by equation (13), this value can be used to optimize the parameters on log-Gabor filters

$$d = \frac{|\mu_I - \mu_D|}{\sqrt{{\sigma_I}^2 + {\sigma_D}^2 / 2}} \tag{13}$$

where

μ_I, μ_D = the mean of the probability distributions

σ_I, σ_D = standard deviation of the probability distributions

6 Results

This section describes the experiments using an iris image database "CASIA" [17] for evaluating matching performance and the effectiveness of the algorithms. This database contains images of 20 different irises and every iris has 3 images with 320 × 280 pixels in 256 gray levels. The experiments were obtained with verifications (one to one matching). Firstly, evaluating genuine matching scores for all the possible combinations (60 intra-class comparisons). Then evaluating the impostor matching scores for all the possible combinations (190 inter-class comparisons). There are a number of parameters required in processing feature extraction using Log-Gabor Filters. Optimum settings for these parameters are needed to attain the best verification rate. To select an optimal Log-Gabor filter parameters we probe different values, obtaining that features for iris can be extracted with bandwidth □= 0.5, one with center wavelength fo=14 pixels. In a second stage of experimentation, the values for the False Acceptance Rate (FAR) and the False Reject Rate (FRR) was estimate. Both distributions obtained were adjusted to a normal distribution with parameters $\mu_I = .182 \quad \sigma_I = .029$ for Intra-class distribution and $\mu_D = .4195 \quad \sigma_D = .0534$ for Inter-class Distribution. It was found that the optimal values for (FAR=.182516% and FRR=.185026%) are given with threshold fixed in X=.2645. The Fig. 7 shows the distributions of probability for matching distance for intra-class and inter-class. It was found that the distance between the intra-class and the inter-class distribution is large (d=5.52732), and the portion that overlaps between the intra-class and the inter-class is very small. This proves that the proposed features are highly discriminating. In addition, several tests were made to obtain the time of execution for each stage in the implemented system: Binarization (time:71.8 ms), Pupil Localization (135.2 ms), Iris Localization (362.8 ms), Normalization (64.2 ms), Extraction Features (12.65 ms), Iris Code Extraction (15.07 ms), Iris Code Matching (131.7 ms), Total: 793.42 ms. The time computation consumed by each stage of the system that was implemented on a P4 computer (1.8 GHz, 512 MB RAM, Windows XP). The average total execution time of a basic verification process not exceeds 800 ms, which is suitable for a recognition system.

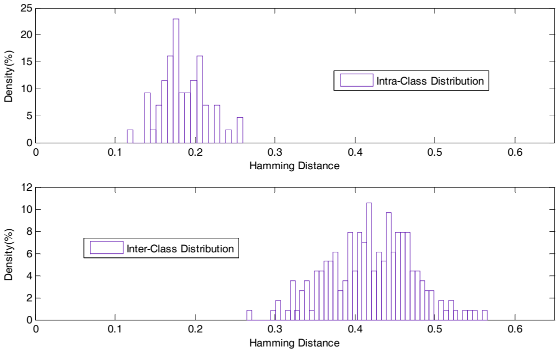


Fig. 7. Distributions of probability intra-class and inter-class in function to hamming distance**.**

This kind of information is not reported commonly in the literature. Therefore a comparison between the obtained results and another system cannot be done. This is to the lack of public datasets. It is highlighted that the algorithms operations were programmed with double precision floating-point to obtain greater exactitude in the results.

7 CONCLUSIONS

The objective of this article was to present the implementation of an iris recognition system based in mathematical algorithms derived from the information found in open source literature. Different concepts for every stage in a iris recognition system was developed by our-self implementing in C language in order to verify the system performance and to validate the chosen options for every stage. The selection of the parameters and the algorithms for every stage has been adopted to ensure the system maintains a good compromise between accuracy and speed. The evaluation of the system achieved high confidence identity verification based on iris texture; further efforts should be applied to be insensitive to variations in the conditions of iris images acquisition.

References

[1] Kresimir Delac,Mislav Grgic .”A survey of Biometric Recognition Methods ” 46th International Symposium Electronics in Marine, ELMAR-2004, 16-18 June 2004, Zadar, Croatia.

[2] R.Garcia,C.Lopez,O.Aghzout,J.ruiz, “biometric Identification Systems” Signal Processing , V.83 n.12 December 2003.

[3] Y.Belganoui,J.C Guézel and T. Mahé, “La biométrie,sesame absolu…”,Indistries et Techniques,france, no 817 ,July 2000.

[4] “Iris recognition border-crossing system in the UAE”. Reproduced from International Airport Review, Issue 2, 2004.

[5] John Daugman, “High confidence visual recognition of persons by a test of statical independence” ,IEEE Transactions On Pattern analysis and Machine Intelligence,vol.15,no 11 November 1993.

[6] W.W.Boles and B.Boashash, “a human identification technique using images of the iris and wavelet transform”,IEEE Transactions On Signal Processing,vol .45,no 4, April 1998.

[7 Iridian Technologies, “Moorestown,NJ”. http://www.iridiantech.com/

[8] Richard P . Wildes, “Iris recognition:An Emerging Biometric Technology” Proceedings of the IEEE , vol. 85,no. 9, september 1997.

[9] J.F.Canny , “Finding edges and lines in images ” M.S. thesis ,Mass.Inst .Technologies,1983.

[10] R.P.Wildes,J.C.Asmuth,G.L.Green “A System for Automated Recognition” 0-8186-6410-X/94 ,IEEE 1994.

[11] P.V.C.Hough, “Method and means for recognizing complex patterns ” . U.S. Patent 3 069 654,1962.

[12] Sánchez Avila,C Sánchez Reíllo “Two Different Approaches for Iris Recognition using Gabor Filters and Multiscale Zero-Crossing Representación” Pattern Recognition Vol:38 nº ,23 July 2004.

[13] John Daugman, “Uncertainly relation for resolution in space,spatial frecuency,ans orientation optimized by two dimensional visual cortical filters,” J Opt.Soc.Amer .A. ,vol.2.,pp.1160-1169,1985.

[14] D Field .”Relations between the statics of natural images and the response propierties of cortical cells ” Journal of the Optical Society of America,1987.

[15] Chong S,Andrew T,David N , “High security Iris verification system based on random secret integration” Computer Vision and Image Understanding, Vol 102, Issue 2, May2006, Pag.169-177.

[16] W.P. Tanner & J.a. Swets ,”A desicion-making theory of visual detection ,”Psychol.rev.vol 61 pp. 401-409, 1954.

[17] “CASIA” iris image database collected by institute of automation, Chinese academy of sciences. Available: www.sinobiometrics.com.

IMPROVEMENT OF ALGORITHM FOR MEASURE ANGULAR SPEED

López-Chau A.[1], Medel-Juárez J. J.[2], Quezada Quezada J. C.[3]

internal@terra.com[1], jjmedelj@yahoo.com.mx[2], quezada@hotmail.com[3]

[1, 3] Universidad Autónoma del Estado de Hidalgo, Campus Tizayuca México, Hidalgo.

[2] Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada - IPN
CICATA-IPN Legaria 694 Col. Irrigación., Deleg. Miguel Hidalgo. CP. 11850.

[2] Centro de Investigación en Computación - IPN
Av. Juan de Dios Batíz s/n casi esq. Miguel Othón de Mendizabal, Unidad Profesional "Adolfo Lopez Mateos" Col. Nueva Industrial Vallejo, Mexico D.F. C.P. 07738

Abstract. In [1] we showed that the measure of a rotatory machine can be made using an incremental encoder –an electronic digital device that produce several electric pulses on each revolution [3] – and then we proposed an algorithm to correct the intrinsic error produced in these measures. In this paper, we improve the algorithm in [1] and proposing a scheme that allows getting a better response, combining two approaches [2]: First we count the number of pulses produced in time, taking into account that the remainder time to evaluate the angular speed is dynamically computed and changed respect to value, reducing the sample time error.
This combined algorithm considered the programming in a float point microprocessor and implanted in a reprogramable digital device.

Keywords: Angular speed, rotatory machines, state machine

© A. Argüelles, J. L. Oropeza, O. Camacho, O. Espinosa (Eds.)
Computer Engineering.
Research in Computing Science 30, 2007, pp. 57-69

MEASURING ANGULAR SPEED

In many cases, incremental encoders are used to measure the speed of rotatory machines. The square pulses on the outputs of these devices, are taken into account to calculate the angular speed and direction. The basic limits considered in this paper didn't include the jitter and noise in the output of incremental encoders.

One way of measuring angular speed using incremental encoders, counting the number of pulses produced in a pediod of time:

$$Angular\ Speed\,(rpm) = \frac{60 * M}{N * k * T_b} \tag{1}$$

Where:

rpm: Revolution per minute.
N: It is the number of pulses per revolution produced. This depends on the incremental encoder used, with $N \in Z_+$.
M : It is the number of the pulses produced in the time kT_b by the incremental encoder. $M \geq 0$
k : It is a positive integer number. $k \in Z_+$
kT_b : It is the time used to count the pulses given by the incremental encoder.

The period P of each pulse depends naturally, to the angular speed into rotatory machine, and is given by

$$P = \frac{60}{Angular\ Speed * N} \tag{2}$$

And we supposing that P is an invariant and consequently stationary for any kT_b time, accomplishing, that

$$\min(kT_b) \geq \min(P), \forall k \in Z_+ \tag{3}$$

The kT_b value tends to be greater when the angular speed tends to zero.
In Figure 1, we showing the several values of P for some incremental encoders [2]

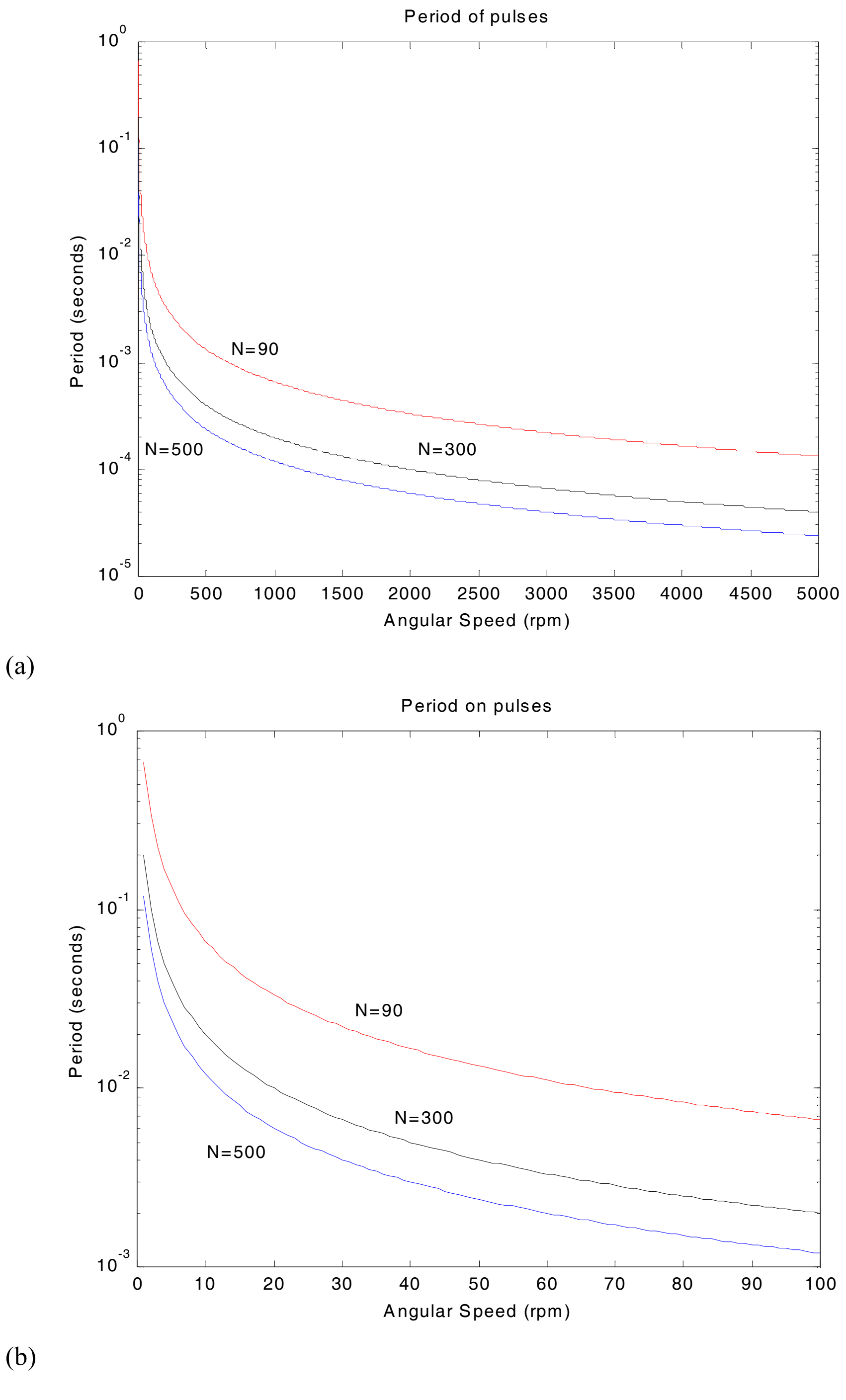


Figure 1. Period of pulses for some values of *N* (a) 0<rpm≤500 (b) 0<rpm≤100.

Among the (1) and (2) is observed that the exact number of pulses, which can be counted in time kT_b is given by (4):

$$M_{real} = \frac{k * T_b * N * rpm}{60}, \tag{4}$$

but unlike (1) now $M_{real} \in Z_+$.

This situation conduces to an error because the number of pulses measured with a digital system are the same for an interval of speeds (rpm$_1$ $_{to}$ rpm$_2$) and therefore if the speed measured is used as feedback in a control system the output of the last one can be wrong[4] [5]. Figure 2, shows and example of this.

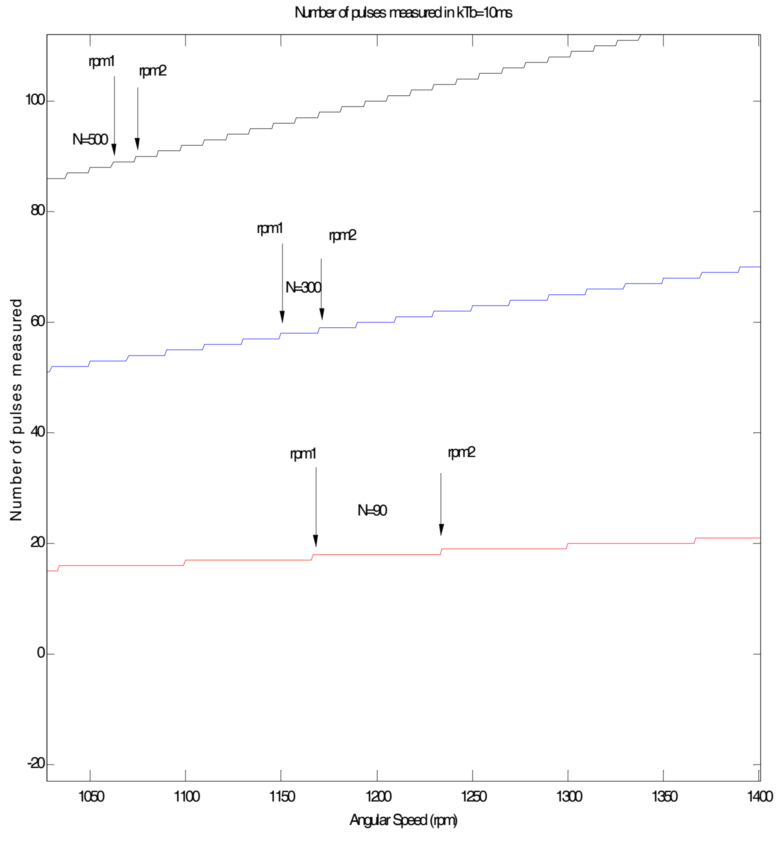


Figure 2. Number of pulses measured for *kT$_b$*= 10ms. Observe that the number of measured pulses is an integer.

In practice, once the measurement system is working, the only parameter that one can vary is kT_b, so we can use:

$$\frac{\partial rpm}{\partial(kT_b)} = -\frac{M*60}{N*(k*T_b)^2}, \tag{5}$$

Equation (5) means that measures tends to be more stable if kT_b raise, and variations are toward zero in a descend fashion.

If the maximum variation of measures requiered for an aplication is V (rev per minute units where $V>0$) then:

$$|kT_b| \geq \left|\sqrt{\frac{M*60}{N*V}}\right|, \tag{6}$$

Forcing $M=1$ in (6) the lowest time required to take a measure can be obtained with precision of $\pm V/2$.

kT_b in (6) can be calculated in practice using the algorithm proposed in [1] and showed again in Figure 3.

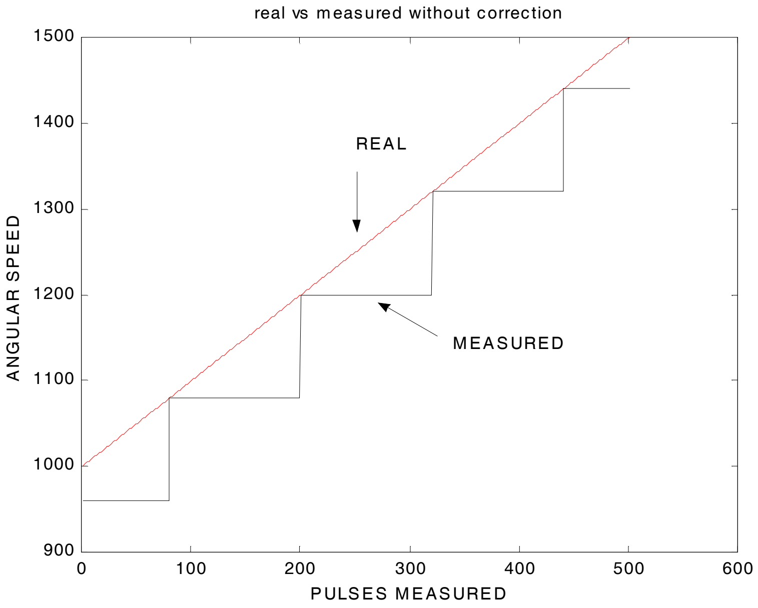
```
1. Begin with an arbitrary $kT_b$
2. Measure the number of generated pulses ($M$) in the
   $kT_b$ time.
3. Obtain the next ("optimized" value of $kT_b^+$
   according to $kT_b^+ = \frac{kT_b}{M} * K_1$, where $k_1>0$ is an fixed
   positive integer.
4. Take another measure of $M$ using this new value of
   $kT_b^+$
5. Obtain the speed in rpm units using (1).
```

Figure 3. Algorithm proposed in [1].

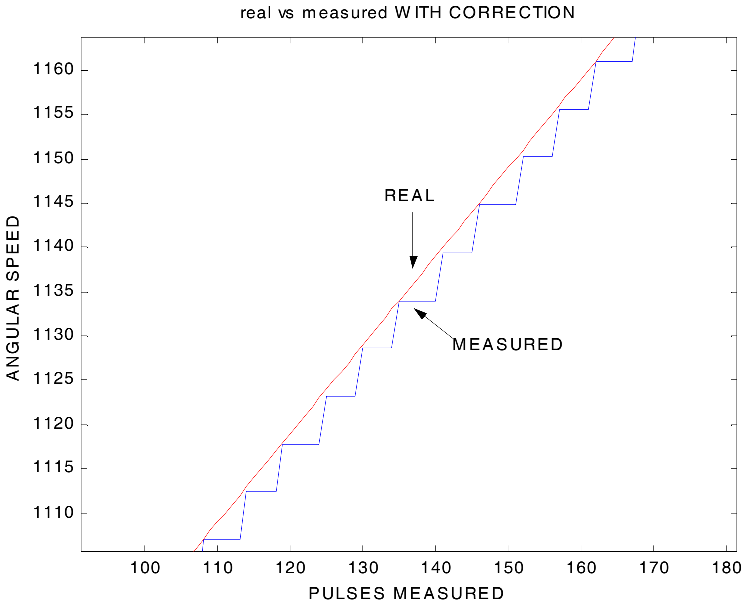
The method described in Figure 3, can be successfully used by beginning with a small kT_b and works better with bigger values of N. The minimum value of kT_b is:

$$kT_{b\min} = \frac{60}{(rpm)_{\max} * N}, \tag{7}$$

Implementing this algorithm an eight bits [1], RISC architecture microcontroller, we obtained good enough results. Simulations behavior about the algorithm [1] depicted in Figure 4 into the both sections.



(a)



(b)

Figure 4. Results of simulation of algorithm showed in Figure 3 (a) without using algorithm Figure 3 (a) Using algorithm Figure 3

IMPROVING THE ALGORITHM

When it is measuring the speed of an rotatory machine, is very important count the number of pulses produced by the incremental encoder in a period of time, another option is take a measure of the period of pulses, however both of them can leads to an error, even if the number of pulses missed is as small as one.

In this paper, we improve the algorithm showed in Figure 3. The basic idea combines the two strategies that frequently come apart: Count the number of pulses in a period of time, and measure the period of pulses. We use two subsystems, one of them for measure number of pulses and another one for measure the remained time, after kT_b finishes, in addition to this, we change dynamically the parameter k_1 in the system.
In Figure 5, simplifying the diagram designed to improve the algorithm, a brief pin description is showed in Table 1.

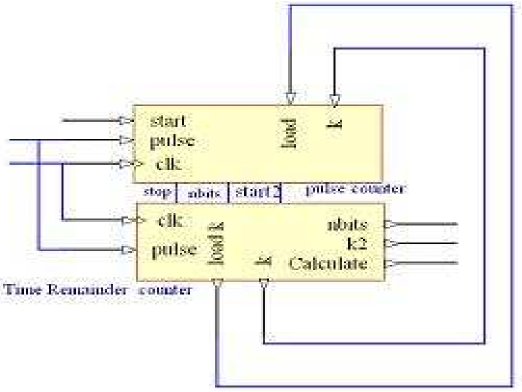


Figure 5. Basic diagram for the algorithm proposed.

PIN	DESCRIPTION
Start	Reset the entire system, begins execution of algorithm.
Clk	Clock signal for sincronizing all system.
Pulse	Pulses produced by incremental encoder.
stop	This singnal is generated by the *pulse counter* system.
n bits	The number of pulses measured in kT_b time.
nbits	Final measure
k2	Number of ticks after pulse counter systems finishes
calculate	Indicates the entire system has finished

Table 1. Pin description.

The *pulse counter* subsystem is basically a state machine that counts the pulses generated by an incremental encoder. T_b is the minimum "quantum" that corresponds with the resolution of system. A very simplified diagram is presented in Figure 6.

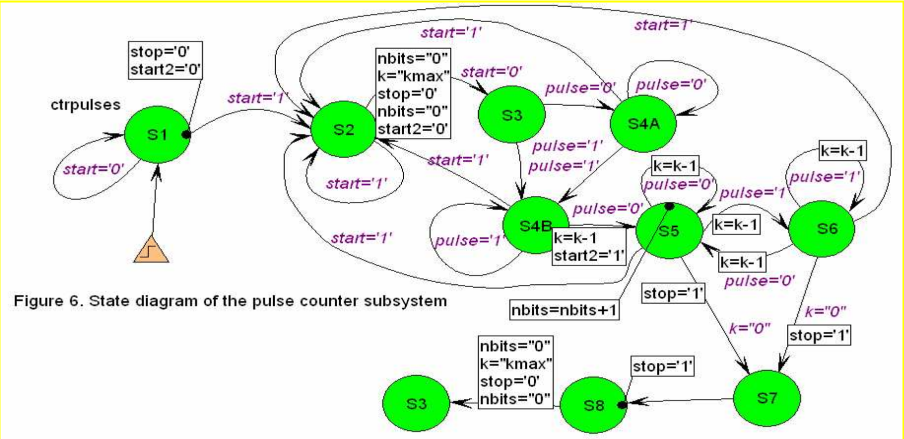


Figure 6. State diagram of the pulse counter subsystem

STATE **DESCRIPTION**

S1 Initial state. Subsystem waits for a high level in *start* signal.

A high level has arrived in *start* input. Wait in this state now for a low level in *start* signal.

S2 The state machine leaves this state after a falling edge in *start* signal. *nbits*, *k* and *stop* are all set to an apropiate value.

This state machine returns to state S2 again if high level presented in *start* signal.

S3, S4A and S4B These states allow waiting for the first falling edge of incremental encoder output.

Begins (continues) counting the number of pulses produced in time kTb.

S5 and s6 The state machine is changing alternative between S5 and S6 states, depending on falling or risig edge of incremental encoder pulses.

The kT_b time is since now (first time state machine gets into state 6) counting down.

S7 S3 and S4 states are leaved when the sample time kT_b has finished. Now, the number of pulses counted is passed to period counter subsystem, this one is notified by means of the stop signal.

S8 This state is used to permit that the state machine of second subsystem can read the output.

nbits, *k* and *stop* are set to an appropriate value. Return to S3.

Can be observed that this subsystem is a simplified version of algorithm presented in Figure 3. Later, we'll show how to adapt the kT_b value. The task of the second subsystem consists in measure the interval between the last pulses counted and the time remained before kT_b expires. This is also a state machine, whose diagram can be seen in Figure 7.

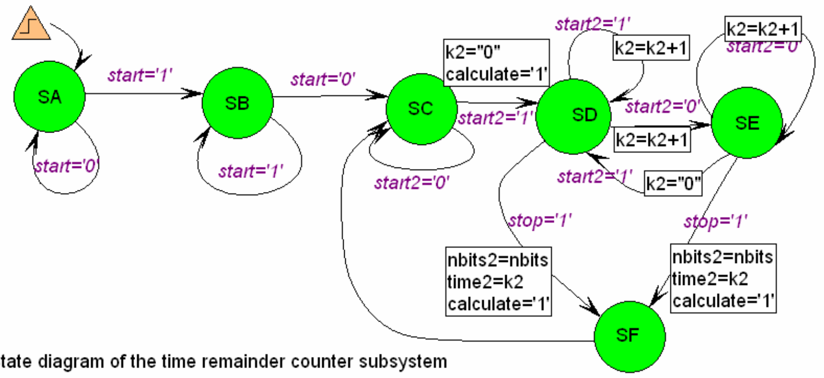


Figure 7. State diagram of the time remainder counter subsystem

STATE	DESCRIPTION
SA	Initial state. Subsystem waits for a high level in *start* signal.
SB and SC	These states allow to wait for the first pulse counted in the previous state machine. Begins (continues) counting the remainded time after a pulse has been counted.
SD and SE	The state machine is changing alternative between SD and SE states, depending on falling or risig edge of incremental encoder pulses. On each transition, k2 increments by one indicating that a tick has elapsed.
SF	Finally, when kT_b has finished stop incrementing k2. The values of k2, and nbits (number of pulses counted) are presented in outputs of this subsystem.

In order to compute the next value of k_1, we propose the use of an averager: so, when state machine arrives state SF, a recursive averager (not showed in simplified diagram of Figure 7) compute the average vaue of k_2. This is a fast way to compensate the next value of k_1 (kT_b^+) in the first state machine. Observe that for both subsystems, T_b is taken of `clk,` so we can obtain k_1 (average of k_2) mentioned in Figure 3.

SIMULATIONS

For all simulations, consider that the speed of a rotatory machine is constant at least while taking the measure. In practice acceleration must be also considered, [2][3]. The first simulation presented corresponds to running the algorithm with the following parameters: $k=1024$, $T_b=1x10^{-6}$ *sec* (state machines working at *1MHz*) and *M=500.* Figure 8 shows the number of pulses measured, observe that at this point, there is not any improvement with respect to the original algorithm, the number of pulses measured are always an integer, and this is a cause of error in calculating the speed.

The remained time is measured, and presented in Figure 9, observe a constant pattern that decays with respect to speed.

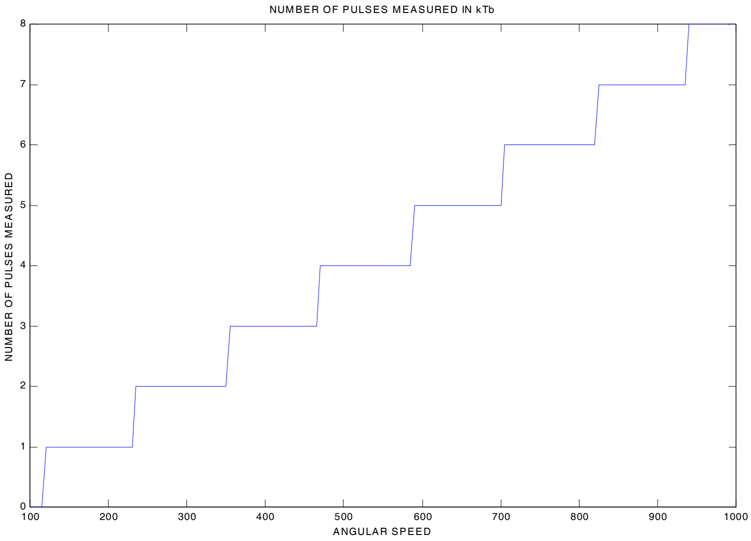


Figure 8. Number of pulses measured for T_b= 1us, k= 1024.

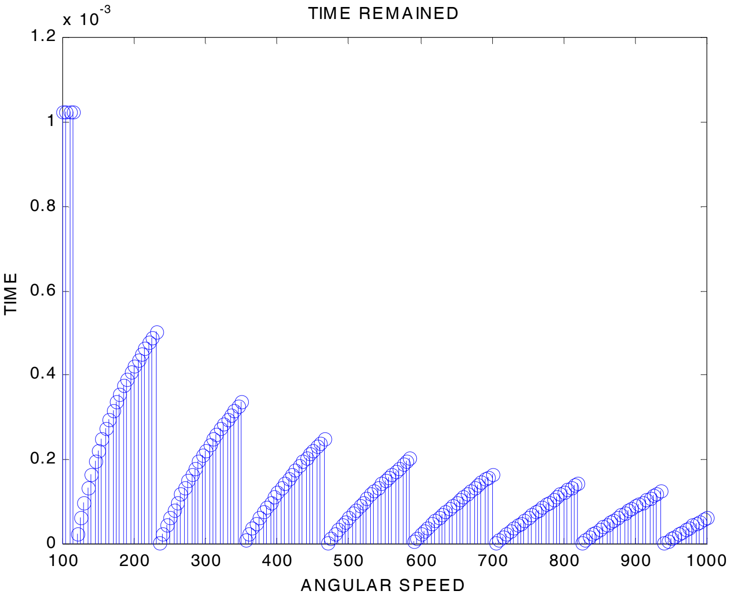


Figure 9. Remained time.

Finally, the algorithm computes the speed taking in account both the number of pulses and the remained time measured. The difference between real and measured speeds is near to one, but most important, the error can be a fraction._

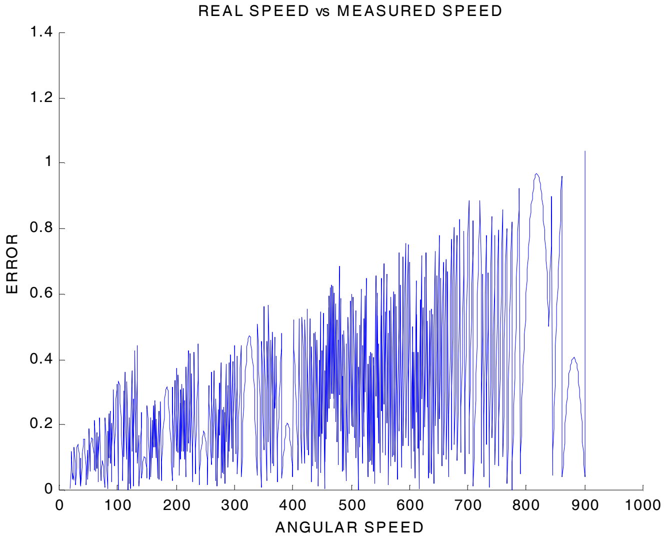
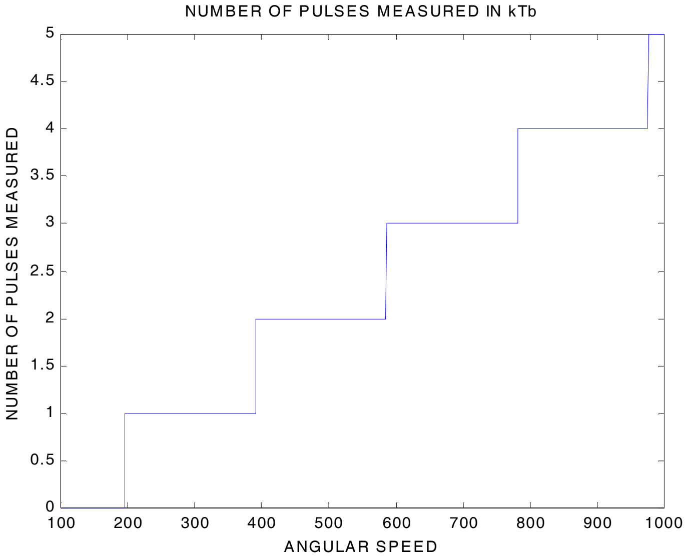
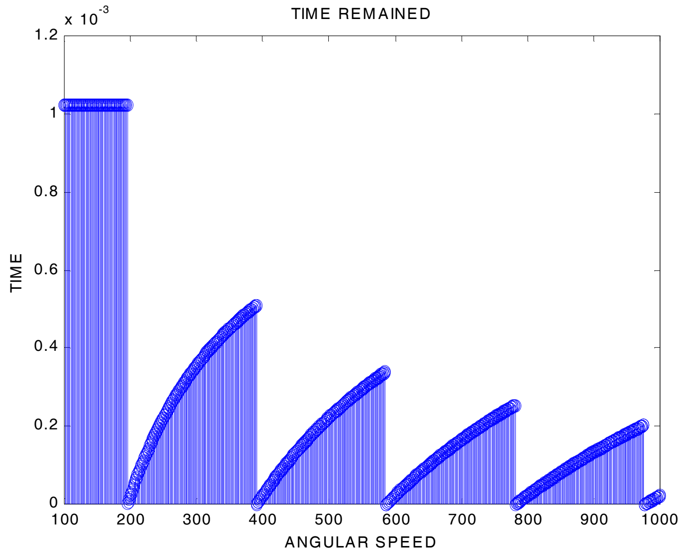


Figure 10. Error: difference between real and measured speed

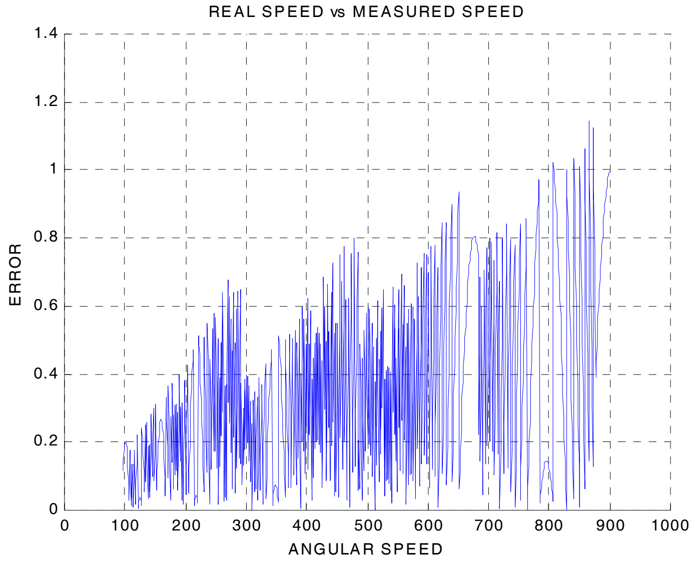
Figure 11 shows more results.



(a)



(b)



(c)

Figure 11. (a), (b) and (c) some other results.

The two main differences between this improvement and the original algorithm, is that the error is reduced and more over, is now a fraction, other important difference is that we can change parameters dynamically, using a recursive average.

CONCLUSIONS

In this paper, we have seen that when measuring angular speed with an incremental encoder, error presented is due to the number of pulses counted as integers. We propose a new improvement of an algorithm for measure of angular speed, in this new version the number of pulses measured and the remained time is taking into account to produce better results. In the simulations can be seen that the error is not necessary an integer, but can be a fraction.

The next step in this work is to take advantage of parallelism, we are working in create several instances of the improvement presented in this paper and introduce the outputs to a system capable to select the best measure.

Bibliography

[1] [1] "Reducción de error en la medición de velocidad de maquinas rotatorias", J. J. Medel-J., Guevara López Pedro, López Chau Asdrúbal, "CICINDI-2005. 6th International Conference on Control, Virtual Instrumentation, and Digital Systems

[2] [2] "An optimised algorithm for velocity estimation method for motor drives", Dapos,Arco, S.; Piegari, L.; Rizzo, R.

[3] 4th IEEE International Symposium on

[4] Diagnostics for Electric Machines, Power Electronics and Drives, 2003. SDEMPED 2003

[5] [2]http://www.gpi-encoders.com/, http://www.servotek.com/

[6] [3]"Sensores y Acondicionadores de Señal" 3ª edición. Ramón Payas Areny.

[7] Alfa Omega Marcombo. pag.455-456.

[8] [4] "Sistemas Modernos de Control, Teoría y Práctica". 2ª edición. Richard C. dorf. Addison Wesley Iberoamerica. pag. 508

[9] [5] "Sistemas de Control en Tiempo Discreto" 2a Edición. Katsuhiko Ogata. pag 8.

Neural Model for Diagnosis and Fault Detection in Modern Shrimp Ponds

José Juan Carbajal Hernández[1], Luís Pastor Sánchez Fernández[1]

Av. Juan de Dios Batíz s/n casi esq. Miguel Othón de Mendizabal, Unidad Profesional
"Adolfo Lopez Mateos" Edificio CIC, Col Nueva Industrial Vallejo, 07738, Mexico D.F
Email: carbajalito@hotmail.com.mx

Abstract. Nowadays, the methods based on artificial neural networks (ANN) have demonstrated to be useful in the treatment of environmental problems. Water quality is an important factor in shrimp farms. This study proposes a new method for evaluating the water quality in shrimp ponds based on an ANN, this method has been developed proving the importance and potentiality of the neuronal systems in this area. ANN's are used to find a relationship among the environmental variables that affect the shrimp habitat. The results show a good response obtaining for classification of the status of the water quality; excellent, good, regular and bad. This work gives an alternative tool that is used in the treatment of the water management.

Keywords: Water quality, neural networks, aquaculture, artificial intelligence.

1 Introduction

The shrimp farming is an important activity in aquaculture field that is practiced over the world. In Mexico the total area dedicated to the shrimp farming is approximated 52,648 ha; 51,059 ha is located in the Gulf of California. The farming system most used in the Gulf of California is the semi-intensive [1]. The shrimp ponds are supervised frequently because the environment must be controlled. In the marine water exist many environmental parameters that affect the habitat, such parameters are classified in tree categories: physical, chemical and biological. A good control of the environmental variables makes a good habitat (Table 1) [2].

The water management is an important factor in farming shrimp for having a good farming period. A bad control in water quality could generate a stress, low maturing or a decreasing in the immunological system. An organism weakened has more possibilities for getting any sickness for example Yellow-Head Virus, White Spot Syndrome Virus and Taura Syndrome Virus.

This work develops a neuronal model that allows diagnosing the status of water quality and detects faults in modern shrimp ponds; also the objective of the model is obtaining a good environment that allows an optimal maturing and surviving of the organisms [3],[4].

The shrimp farms are big places, which are composed of many ponds; each one has a different habitat. The measurement made to each variable is made with manual techniques in laboratories or using sensor devices, for example pH, temperature or conductivity sensors. The needed for techniques that allow the diagnosis of the status of water quality, the analysis of the environmental variables behavior and their integration in the assessment in such process is clearly recognized. Analyzing these aspects, some alternatives have been developed in the artificial intelligence and the control techniques areas. Some of the methodologies are based in neuronal systems, and these systems have been proved in real environmental problems [2].

Tabla 1. Physical-chemical parameters in a shrimp pond.

High Impact	Low Impact
Temperature	Hydrogen Sulfide
Oxygen dissolved	Non Ionized Hydrogen Sulfide
Turbidity	Nitrates
Salinity	Total inorganic Nitrogen
PH	Silicate
Non ionized ammonia	Phosphorus
	Chlorophyll A
	Total suspension solids
	Potential redox
	Alkalinity
	Dioxide of Carbon
	Total Ammonia

In the habitat the environmental variables must be in a valid range of values (Table 2), if they are out of this range, could generate a negative impact into the pond. There are some variables that have a bigger impact than others, if one or some of them are out of range will destabilize the water quality generating a bad status this is represented with two states "Good" and "Regular" depending of how many variables are not controlled, in other hand, if all the variables are in range, the pond is in optimal conditions and it is known as "Excellent", and the undesired status in when one of the variables or the combination of some of them could generate an status that will be lethal for the organisms that is known as "bad". The main purpose is to use these variables with an artificial neural network (ANN) for developing a system that obtains the status of the water quality [5], [6].

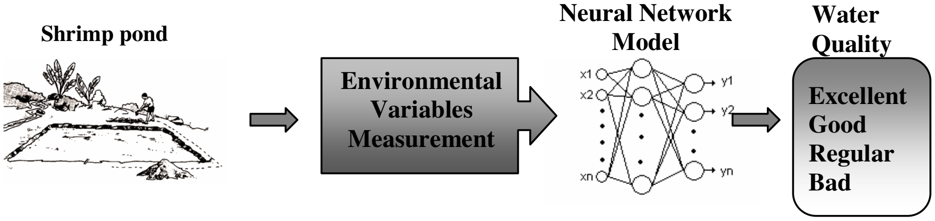


Fig. 1. The measurements of the environment will be analyzed by a model that describes the relationship among them, obtaining a status of water quality.

It is needed to indicate that water quality (WQ) in seawater is a function of the environmental variables; the next equation describes the relationship:

$$\text{Water Quality} = f\,(\text{Temp, DO, Salt, Turb, NIA, PH}, \ldots) \qquad (1)$$

Where Temp is temperature, DO is dissolved oxygen, Salt is salinity, Turb is turbidity and NIA is non ionized ammonia [7]. There are also some relationships between variables, for example the equation that represent the changes with pH in the Gulf of California (Hernandez and Zirino, 2003), is a function of total CO_2 (*Ct*), total alkalinity (*At*) and pressure (*P*):

$$\partial \mathrm{pH} = \left(\frac{\partial \mathrm{pH}}{\partial \mathrm{Ct}}\right)\partial \mathrm{Ct} + \left(\frac{\partial \mathrm{pH}}{\partial \mathrm{At}}\right)\partial \mathrm{At} + \left(\frac{\partial \mathrm{pH}}{\partial \mathrm{P}}\right)\partial \mathrm{P} \tag{2}$$

The environmental variables have relationships among them and trying to represent them is very complex to establish as the equation 2, however this complexity can be expressed with an ANN. The ANN is a powerful and efficient tool for recognizing and classifying patterns, that is why the ANN is used on intelligent systems. The ANN's are used in non-lineal mathematical relationships that are so hard to describe. In this study a neuronal system has been used for establish this relationship among them for evaluating the status of water quality.

Table 2 Ranges of environmental variables.

Environmental variables	Ranges of values
Temperature (ºC)	23º – 30º
Salinity (mg/l)	15 – 25
pH (mg/l)	7.6 – 8.6
Dissolved oxygen (mg/l)	6 – 10

2 Materials and Methods

2.1 Study area and water quality data

The measurements of the environmental variables were made in ponds located in Bahía de Kino in the Gulf of California, near from Sonora Estate, the water resources were obtained from seawater (Fig. 1). A total of 3000 measurements were made in a period of shrimp farm. The variables measured were pH, temperature, salinity, dissolved oxygen.

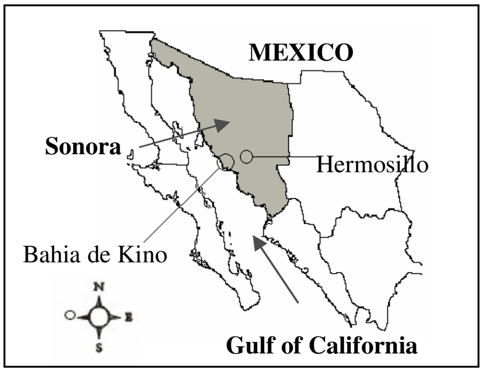


Fig. 2 Map of Sonora, Mexico indicating the two regions with sampling of shrimp farms (Bahia de kino and Hermosillo).

2.2 Artificial Neural Networks

The artificial neural networks were applied in this study for providing a non lineal relationship among the environmental variables (Input set) and the status classification for the water quality (Desire output). The environmental variables present no lineal relationships that are very hard to establish with a mathematical expression, in order to describe it, the ANN is a good technique to represent the behavior among them [8].

An ANN is constructed of singles processing elements called nodes o neurons, which are connected with some parameters called weights. The neurons are accommodated in a layer structure. The first layer is known as the input, and it is where the information of all environmental variables is introduced, the final layer is known as the output layer, this layer is where the neurons process all the data extracted giving a desired response. The layers used between the input and the output layer are denominated hidden layers (figure 4), and there is not a limit about how many hidden layers an ANN could contain [9], 10[].

Each neuron has a function, which is determined by the inputs that receives to other neurons (figure 3).

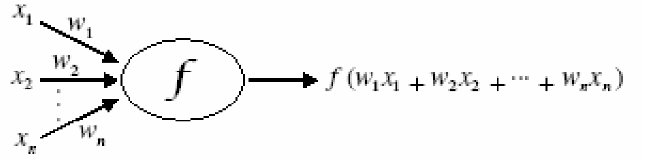


Fig.3 Representation of a neuron.

The behavior of a node can be expressed by the next equation:

$$s_i = \sum_{j=1}^{n} y_i w_{ij} + b_i \quad (1)$$

Where w_{ij} are the weights that interconnect the different neurons j of the previous layer with the actual neuron i, b_i is the bias of the neuron, y_i is the output of the neuron in the last layer and S_i is the new output. The result will be processed by a new function that is called the transfer function; the most common are the next:

$$y_j = \frac{1}{1+e^{-S_i}} \qquad \text{Exponencial} \quad (2)$$

$$y_j = \tanh(s) \qquad \text{Tangencial} \quad (3)$$

$$y_j = \begin{cases} 1 & y_j > 1 \\ y_j & -1 < y_j < -1 \\ -1 & y_j < -1 \end{cases} \qquad \text{Lineal} \quad (4)$$

The learning method (memorizing) of an ANN's is made using a data set extracted by a database. The data set contains the input values (patterns) and the output values (desired response). The ANN is able to establish a relationship between the input and the output values [6].

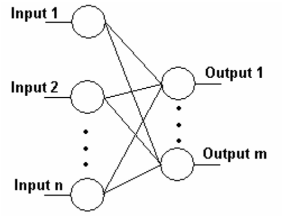


Fig. 4. ANN architecture.

The feature of the ANN's makes them an attractive case of study for resolving complex problems in engineering.

3. Results

Low concentrations of dissolved oxygen and high variations were observed in the most quantity of measurements made in the shrimp ponds. The conditions of the pond depend of the variations of the environmental variables for example the temperature and dissolved oxygen. The analysis made by the ANN showed that in the mayor part of the measurements the status of the water quality was a risk condition, the dissolved oxygen was one of the main factor of this result, due in the measurements obtained the values were lower than 4 mg/l, the high salinity concentrations was a second factor.

.

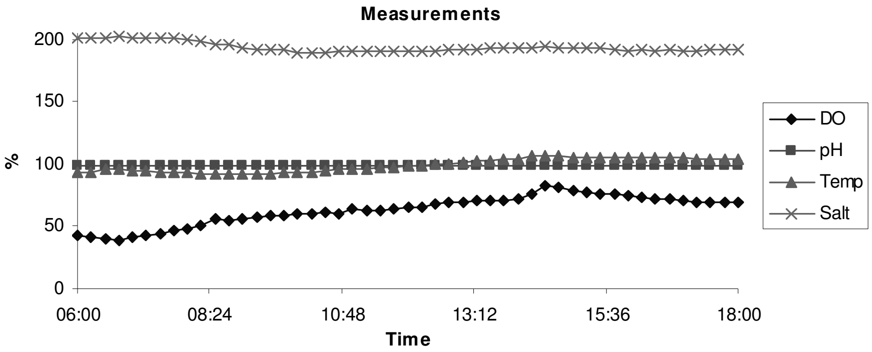


Fig. 2 Measurements obtained in a farming day.

3.1 Pre-processing

Some preprocessing steps make more efficient the RNA [3]. All the environmental variables were preprocessed before training the ANN, normalizing the data set in a range of [0, 1] is important because there are some variables that have big values and another that have low values, normalizing them will generate that all variables have the same range, to do this we used the next equation:

$$\text{var} = \frac{x - \min(x)}{\max(x) - \min(x)} \qquad (5)$$

Where X is the environmental variable and var is the normalized variable.

3.2 Patterns

When the shrimp farming is in process the weather is always in change and the environmental variables could disestablish the habitat [7], [8], [9], this problem is generated by the combinations of the bad controlled variables, such problems could be classified as follows:

1. Excellent: all the environmental variables are controlled.
2. Good: One variable is out of range; however it not represents any problem.

3. Regular: some variables are out of range and the combination of them could disestablish the habitat.
4. Bad: One or some variables are in range that could generate several or lethal problems in the habitat.

The Table shows the range of the controlled environmental variables. The combinations of the values of the environmental variables create patterns that define de status of the water quality. A complete data set that contains all the possibilities is huge, this problem was resolved creating an artificial dataset which cover different cases, and this set was composed of 7000 patterns.

The nature of this parameter and their relationship with the water quality make them an excellent set for their processing with an ANN.

3.3 ANN Model

The ANN developed in this work is backpropagation 4 – 10 – 10 – 10 – 4 layered network; the input layer receives all the data set information (Environmental variables values). The output layer gives the final status of the water quality (excellent, good, regular, bad). The three hidden layers is constructed by 10 neurons each one and its functions is to do a processing between the input layer with the output layer. The topology was determined by proving different kinds of ANN's, this topology was the best in time processing and response. We found that there were not significant differences with a bigger numbers of neurons in the hidden layer, and a lower number of neurons the ANN's had not a good response [10],[11],[12]. The activation functions used were exponential; using about 400 iterations for the training, the mse obtained was of $1x10^{-3}$.

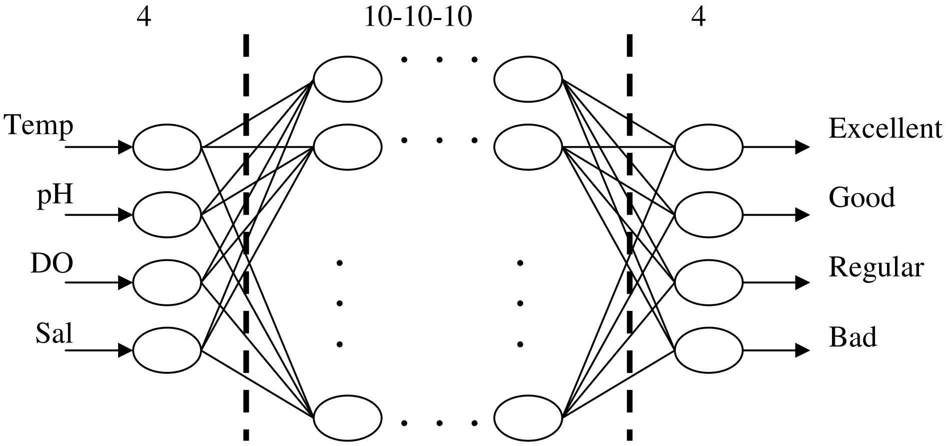


Fig. 4. Topology of the ANN. The 4 input layer is receives the environmental variables values, the four output layer is the response of the water quality in the shrimp habitat.

3.4 Water Quality Index

The measurement was tested with the neural network model; the results showed good performance of the ANN, obtaining a water quality status, however for practical reasons the output of the neurons was transformed for obtaining a index of water quality WQI, this process was done using a fuzzy logic system. The WQI is obtained in a range of [0,1], when the WQI is 0 means a bad water quality, in other hand if the WQI is 1, means a excellent water quality, and the values between 0 to 1 represents the good and regular status [8].

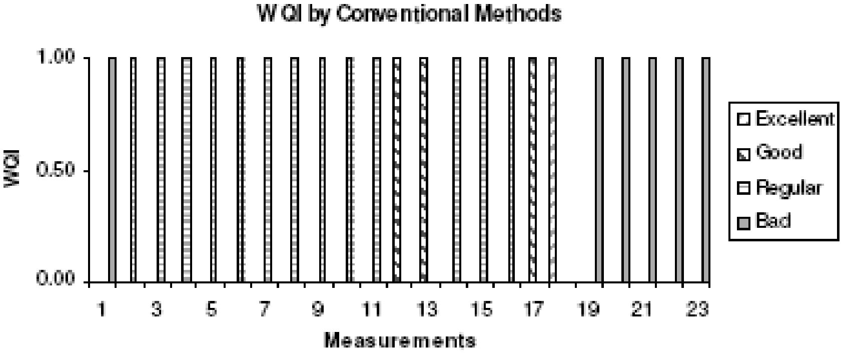


Fig. 3 The WQI is obtained by conventional methods.

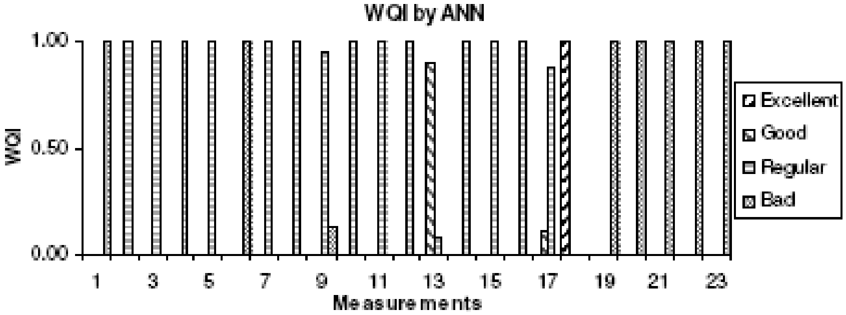


Fig. 4 The WQI is obtained by the ANN model.

4 Discussion

In this work, we present a neural model that gives a diagnosis for fault detection in management of water in shrimp; a mathematical equation that describes the behavior of the environmental variables in water management is very complex to develop. A research of the shrimp habitat was made for obtaining the environmental variables that have the highest impact in the WQI.

An ANN can model non linear systems in aquaculture systems; the topology of the network depends of how complex is the environment system. There are some questions about the regression methodologies because their purpose is to find the parameters that define the best linear approximation to the input and the output.

According with the results, the ANN gives a good diagnosis of the water management. We can conclude that the model proposed is an excellent tool for the diagnosis and detection of faults in modern shrimp ponds. The main advantage of our model over other evaluating methods is: speed on the assessment of water quality and its portability to another aquaculture system.

References

1. Casillas, R., Magallón, F., Portillo, G., Osuna, P.: Nutrient mass balances in semi-intensive shrimp ponds from Sonora, Mexico using two feeding strategies: Trays and mechanical dispersal. Aquaculture, Vol. 258. Elservier (2006) 289-298.
2. Páez O, F., “Camaronicultura y Medio Ambiente”. Instituto de Ciencias del mar y Limnología. UNAM. pp. 271-298, México, 2001.
3. Granvil, T.: Shrimp maturation and spawning, Texas A&M University, Sea Grant College Program. UJNR Technical Report No. 28, 2001.
4. Granvil, T.: Shrimp maturation and spawning. In Proceedings of the 28th Us-Japan Natural Resources Aquaculture Panel UJNR Technical Report No. 28, pp 121-134. TAMU-SG-01-814.2001
5. Angulo A., Angulo, U., “Estudio de calidad del agua y su relación con el crecimiento del camarón blanco (Litopenaeus vannamei), en la granja camaronera agua verde, S.A. de C.V. en Rosario, Sin”, Tesis, Universidad Autónoma de Sinaloa, 2003.
6. Martínez R., “Cultivo de Camarones Pendidos, *Principios y Practicas*”, Ed. AGT Editor S.A., 1994.
7. Hernández J., Zirino, A., Marione, S., Canino, R., Galindo, M.: PH-density relationship in seawater. Ciencis Marinas, Vol. 29(2003) 597-508.
8. Ocampo, W., Ferré, N., Domingo, J., Schuhmacher, M.: Assesing water quality in rivers with fuzzy inference systems: A case study. Environment International, Vol 32. Elservier (2006) 733-742.
9. Romero, C., Shan, J.: Development of an artificial neural network-based software for prediction of power plant canal water discharge temperature. Expert Systems with Application, Vol 29. Elservier (2005) 831-838.

10. Lek, S., Guiresse, M., Giraudel, J.: Predicting stream nitrogen concentration from watershed features using neural networks. Water Resources, Vol. 33. Pregamon (1999) 3469-3478.
11. Sharma, V., Negi, S., Rudra, R., Yang, S.: Neural network for predicting nitrate-nitrogen in drainage water. Agricultural Water Management, Vol 63. Elservier (2003) 169-183.
12. Principe, José, Luliano, Neil y Lefebvre W Curt, 2000, Neural and Adaptative systems: *Fundamentals through Simulations,* 1ª Edición, John Wiley & Sons.

FPGA implementation of Hebbian neural network for Engineering Educational

Marco A. Moreno-Armendariz[1], Osvaldo Espinosa Sosa[1] and Floriberto Ortiz Rodriguez[2]

[1] Centro de Investigación en Computación-IPN
CIC-IPN
AV. Juan de Dios Bátiz S/N, México D.F.,07738, México
marco_moreno@cic.ipn.mx
and
[2]Departamento de Control Automatico
CINVESTAV-IPN
A.P. 14-740, Av.IPN 2508, México D.F., 07360, México

Abstract. In this paper we are working in a new development of educational material for advanced courses in engineering. One of the hottest issues is the implementation of different computer intelligence algorithms in programmable logic technologies. We start this goal with the Hebbian neural network as an introduction of neural network courses. This article shows the design of neural networks using the Hardware Description Language VHDL and its implementation in Field Programmable Gate Arrays FPGAs. The code is totally open so that the user can verify the value of internal signals and make modifications to the structure of the design which can be very useful when a student is learning neural networks and digital design. The main advantages that are offered are that the design can be totally monitored and modified, is portable and it isn't necessary to view designs like a black box in where we do not know the internal structure of the system.

1 Introduction

Nowadays, programmable technologies have penetrated in many areas and places in where technology is present. One of the best cases is the Field Programmable Logic Array FPGA's , and the evolution of these devices has allowed to increase the density and capacity allowing integration of complete Systems on a Chip (SoC), in addition, these elements are offered in low cost and low power consumption devices. Designers have two options to obtain a design, the first one is the complete design of the circuit and second one is to purchase existing designs called CORE's that allows designers to use it as if they were a black box allowing to diminish dedicated effort. A disadvantage of the second option is that it implies the cost of acquisition of these CORE's. Although there are free ones, these usually are tied to a family of FPGAt's and this creates dependency. Another disadvantage is that they are not susceptible to be modified nor verified

© A. Argüelles, J. L. Oropeza, O. Camacho, O. Espinosa (Eds.)
Computer Engineering.
Research in Computing Science 30, 2007, pp. 81-88

its internal state. From an educational point of view, this is unacceptable for our requirements. On the other hand, design totally one circuit has the advantage to offer independence of companies, can be portable to any device of any company and with the characteristic to allow to students to use, monitor and even modify the characteristics of designs as much as it is required[1].

The designed circuit corresponds to a hebbian neural network. In the field of neural networks, the non supervised algorithm could be the denominated hebbian learning method that consists of increasing the value of weights that join two neurons if they activate simultaneously, and to diminish the value if they activate on a differential manner. A hebbian neural network can be arranged in a single layer or several: the inputs propagate to the internal layer, and when coming out, and after the propagation, the weights change in the indicated form. The hebbian learning is equivalent to an analysis of main components of inputs.

2 Hebbian neural network

In 1949, Donald Hebb gives the first learning law for an artificial neural network; the main idea is trying to represent how the brain learns at the cellular level [2]. In the brain, a neuron receives many inputs from a large number of other neurons through synaptic connections. Hebb´s law states that if a neuron A is repeatedly activated by another neuron B, the neuron A will become more sensitive whenever both of them fire simultaneously. So, this learning can be viewed as strengthening a synapse according to the correlation between the activation levels of the neurons it connects. The neuron output signal $y_{net}(k)$ for the network in figure 1 can be expressed as [3],

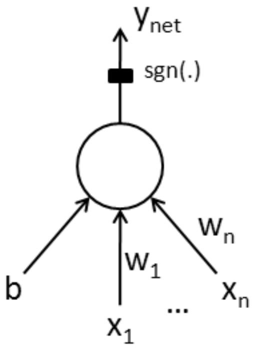


Fig. 1. Hebbian neural network model.

$$y_{net}(k) = sgn\left(\sum_{i=1}^{n} w_i(k)\, x_i(k) + b\right) \quad i = 1, \cdots, n$$

where $sgn(.)$ is the signum function. Given a set of training data $(x_i(k), y_i(k))$,the learning law used to update the weight values $w_i\,(k)$ and the bias $b(k)$ are,

$$\begin{aligned} w\,(k+1) &= w\,(k) + x^T(k)y\,(k) \\ b(k+1) &= b(k) + y(k) \end{aligned}$$

Thus, if the correlation between the input and the output signals, which can be represented by the product $x^T(k)y\,(k)$ is positive, it enhances the strength of the synapse weight value. Otherwise, if it is negative, the weight value should be decreased [4].

This net is a linear associator network, since associate one or more pairs of vectors $(x(k), y(k))$ so that given $x(k)$ as the input, the network will produce $y_{net}\,(k)$. Furthermost, when a vector close to $x(k)$ is presented, the network will produce a vector close to $y_i(k)$. The representation is (consider $b(k) = 0$),

$$y_{net}\,(k) = x(k)w(k)$$

Suppose the network has learned to associate n pairs of vectors. That is,

$$w(k) = \sum_{i=1}^{n} x_i^T(k)y_i\,(k)$$

Futher, assume that the vectors $x_1(k), x_2(k), \cdots, x_n(k)$ are orthogonal and of unit length (i.e, they are orthonormal): If $j = i$, $x_j(k)x_i(k) = 1$;else it is 0. Then $x_j(k)$ can always be transformed into $y_j(k)$ by the network without error because

$$x_j(k)w(k) = \sum_{i=1}^{n} \left(x_j(k)x_i^T(k)\right) y_i(k) = y_j(k)$$

In a d-dimensional input space, the maximum possible number of orthogonal vector is d. Thus, the number of vector pairs that we can associate exactly is limited to d. If x_i´s are not orthogonal, the we get an error when attempting to retrive $y_j(k)$ using $x_j(k)$. This error ϵ can be estimated by

$$\epsilon = \sum_{i \neq j} \left(x_j(k)x_i^T(k)\right) y_i(k)$$

In some cases, the weights $w(k)$ can be chosen so that the error ϵ is small.

3 Real time design

In order to make this work, it was used the software provided by Altera denominated Quartus II[5], this software allows to capture descriptions of circuits using the VHDL Hardware Description Language, after that, it makes a syntax revision of code. Once the description in VHDL is the correct one, a synthesis is made to transfer it to an equivalent design of gates and logic elements. It is

Input		Output
x_1	x_1	y
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

Table 1. Training set for the learning phase.

important to mention that after the synthesis it is possible to use a simulator to verify the functionality of the circuit, that is, to verify that logically the design behaves in a correct form when the corresponding inputs are applied to it and the generated outputs corresponds to results expected by designers. When simulation is correct, it comes to make a process called ”place and route” that has the objective to locate and connect the different components of the design inside the selected FPGA (Altera DE2 [6] with cyclone EP2C35F672C6 in our case). With the obtained data, it is possible to be know the total amount of resources used by the design, as well as the internal delays that are very important when it is desired to make a timing simulation (essential when it is desired that the circuit continues working when it operates to high frequencies of clock). These tools allow to experienced users as well as beginners to make the process before mentioned.

3.1 Set the experiment

From 2, we take $n = 2$ and we select the case of learning the AND-gate as shown in Table 1.

Apply the learning laws in (1) to this training set and using as an initial conditions $W_0 = (0, 0)$, $b_0 = 1$, we obtain the following values:

$$W = (2, 2) \;\; b = -1$$

Now we describe the main parts of the VHDL code that was used to implement the hebbian neural network.

Entity For educational purposes we design an entity that allows to the student to manipulate different signals:

1. Clock signal: The student generate this signal to check the transitory values of the weights and bias of the neural network.
2. Reset signal: The student set the initial values for the weights and bias.
3. Aprender signal: The student can set the neural network in learning and testing mode.

As an output ports we manage four seven segment displays to show values of the parameters of the network during learning or the testing steps.

```
library IEEE; use IEEE.STD_LOGIC_1164.ALL; use
IEEE.STD_LOGIC_ARITH.ALL; use IEEE.STD_LOGIC_SIGNED.ALL; use
work.pack_red_heb_and.ALL;


entity red_heb_and is
    Port (
              clk       : in std_logic;
              reset     : in std_logic;
              aprender  : in std_logic;
              disp_0_w1 : out std_logic_vector(6 downto 0);
              disp_1_w1 : out std_logic_vector(6 downto 0);
              disp_0_w2 : out std_logic_vector(6 downto 0);
              disp_1_w2 : out std_logic_vector(6 downto 0);
              disp_0_b  : out std_logic_vector(6 downto 0);
              disp_1_b  : out std_logic_vector(6 downto 0);
              disp_0_y  : out std_logic_vector(6 downto 0);
              disp_1_y  : out std_logic_vector(6 downto 0)
         );
end red_heb_and;
```

Architecture This fragment of code shows the implementation of hebbian neural network in two ways: the learning and the testing modes.

```
architecture una_neurona of red_heb_and is
begin
process(clk, reset)
begin
    if reset='1' then
        w1 <= "0010";
        w2 <= "0011";
        b  <= "0001";
        y  <= "0000";
        elsif (clk'event and clk='1') then
            if aprender='1' then
                w1 <= w1 + rom_x1 * rom_y;
                w2 <= w2 + rom_x2 * rom_y;
                b  <= b  + rom_y;
            else
                I <= rom_x1 * w1 + rom_x2 * w2 + b;
                    if I < 0 then
                        y <= "1111";
                    else
                        y <= "0001";
                    end if;
            end if;
    end if;
end process;
end una_neurona;
```

Since this kind of neural network uses bimodal format (+1,-1) we use the two's complement representation for the output of the seven-segment displays.

3.2 Simulations

Now we present the simulation results of the VHDL code in figure 2. Here both phases are reviewed, first the learning signal (aprender) is activated, so we can see how the transitory values of weights w_1, w_2 and bias b change in,

```
disp_0_w1, disp_1_w1, disp_0_w2, disp_1_w2, disp_0_b, disp_1_b
```

Then, the testing mode starts where these values are fixed and we send other input values to check the response of the neural network y_{net} in,

```
disp_0_y, disp_1_y
```

Due to the response is correct, we successfully finish the simulation and we proceed to download this program to the FPGA.

For this implementation we use 53 logic elements, 59 pins and 4 embedded multipliers, that is < 1 percent of the total resources of the Altera Cyclone FPGA.

4 Conclusion

From an educational point of view, the design and simulation of neuronal networks as well as its implementation in programmable logic devices constitute an important element in formation of students related to design of digital circuits, electronics as well as digital control and related areas. Although commercial designs exists from several companies, its use generates technological dependency, therefore to provide a description with Hardware Description Languages such as VHDL in this case allows to have an open technology which does not have dependency disadvantages and even allows the access to all internal signals and full modification of the structure. Students of graduate and undergraduate courses can take advantage of this contribution to improve their abilities. After the simulations and the implementations, the design proved to be very efficient for demonstrations and practices in laboratory.

Acknowledgments

Dr. M. A. Moreno-Armendariz thanks to the Centro de Investigación en Computación of the IPN (CIC-IPN) and the Secretaría de Investigación y Posgrado of the IPN, under Research Grant no. 20070612 and CONACyT. Also, the authors thanks to Altera [6] for the donation of the Altera DE2 kits and academic licenses of Quartus II software.

References

1. Romero-Troncoso, R. de J., Ordaz-Moreno A., Vite-Frias J.A., Garcia-Perez A., 8-bit CISC Microprocessor Core for Teaching Applications in the Digital Systems Laboratory, Reconfig 06 (IEEE), (2006)
2. Fu L., Neural Networks in Computer Intelligence, McGraw-Hill, (1994) pp. 55-58
3. Pardo F., Boluda J.A., VHDL Lenguaje para síntesis y modelado de circuitos, Alfaomega, 2da. edición,(2004) pp. 273-309
4. Sheu B.J., Choi J., Neural information processing and VLSI, Kluwer Academic Publishers,(1995) pp. 21
5. (2007) Altera: Quartus II. [Online]. Available: http://www.altera.com/education/demonstrations/online/design-software/onl-design-software-demos.html
6. (2007) Altera: University Program. [Online]. Available: http://www.altera.com/education/univ/unv-index.html

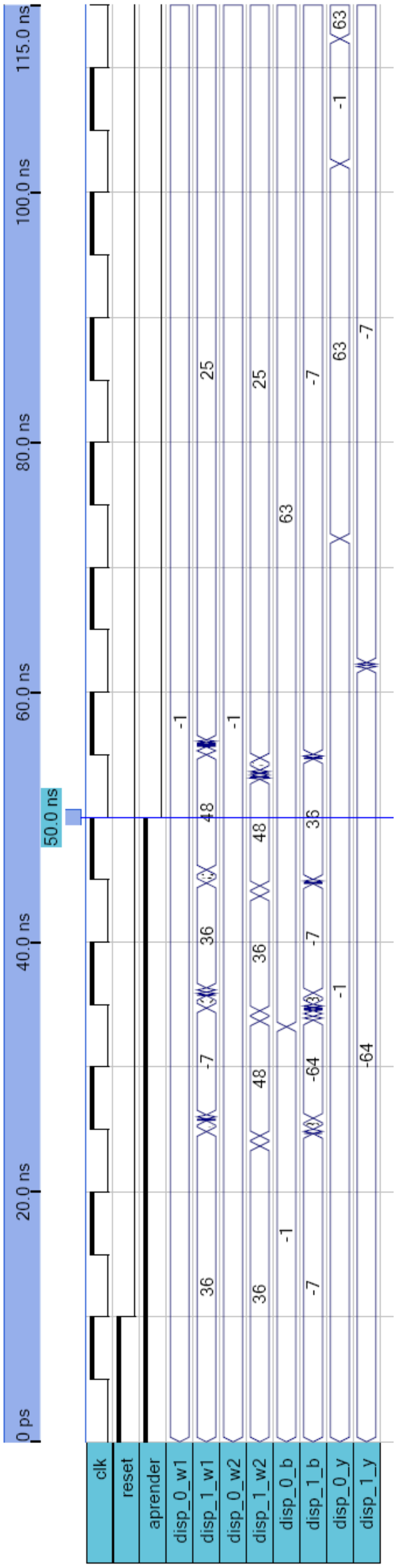


Fig. 2. Simulation waveform of the hebbian neural network.

Engine Simulator for ECMs Diagnosis

José Ignacio Huertas[1] and Natalia Navarrete[1]

[1] Instituto Tecnológico de Estudios Superiores de Monterrey, Automotive Engineering
Research Center, Eduardo Monroy Cárdenas # 2000,
50110 Toluca, México
jhuertas@itesm.mx, natalia.navarrete@invitados.itesm.mx

Abstract. Currently, the number of companies that provide diagnosis, repair and maintenance services to the electronic control modules – ECMs of the vehicles is very limited. Even though the demand of the service is still unsatisfied, the possibility of expansion of this companies is limited by the need of an engine simulator to semi-automate the ECMs diagnosis process. To fulfill this requirement, the present paper describes the design, implementation and testing of an electronic device that simulates the electrical signals generated by the sensors and transducers commonly installed on engines. The device was programmed to simulate different models of commercial engines and to perform automatically the standard procedures followed to identify the most common failures of the ECMs.

This device incorporates systems to guarantee the integrity and safety of the information gathered during the diagnosis process and the physical integrity of the ECMs being diagnosed.

Keywords: Engine simulator, ECMs, engine sensors, diagnosis.

1 Introduction

Currently 100% of the commercial vehicles have electronic control modules – ECMs to control and optimize the vehicle operation. These ECMs receive the information generated by the sensors and transductors installed around the vehicle. The variables monitored in a vehicle are essentially the same regardless of the manufacturer. Most of them are related to the engine operation.

ECMs use the information coming from sensors to [1]:

- Display to the driver the operational conditions of the vehicle such as speed, fuel level, oil levels, etc.
- Regulate the vehicle operation by controlling its different actuators, as for example: fuel injectors, fuel pump, etc.
- Diagnose and display the vehicle functional problems

© A. Argüelles, J. L. Oropeza, O. Camacho, O. Espinosa (Eds.)
Computer Engineering.
Research in Computing Science 30, 2007, pp. 89 - 95

Although the ECMs were designed to stand the adverse conditions of operation to which a vehicle is typically exposed to, the cases of faulty ECMs have become more frequent.

Vehicles manufacturers recommend replacing faulty ECMs by new ones. Nevertheless, this solution is not reasonable for the case of heavy duty vehicles where in addition to the replacement cost, the high opportunity cost of keeping a vehicle in standby during the delivery and calibration time needed by a new ECM must be taken into account. Some companies have been created in response to this business opportunity.

The diagnosis process is carry out connecting the ECM to an engine testing cell [2] and a laptop with a specialized software. Then, a standard procedure is followed where the engine is operated under different working regimes.

This procedure requires to have available an engine of the same specifications of the one for which the ECM was originally designed and programmed. Consequently, these companies have been forced to count with a representative sample of the engine models most used in the region where they are located. In practice, these companies must acquire an engine whenever a new model is launch to the market.

Finally, after the failure is identified, staff with specialized knowledge repair the ECM and perform the physical tests required to guarantee the quality of the service. These know how is the main barrier to the incoming of new competitors in the market.

The high cost of the required infrastructure and the need of preserving its “know how” have limited the expansion of these companies. As an alternative, the companies have proposed to semi-automate the diagnosis process so that it could be performed in several subsidiaries and to concentrate the repair process in the company headquarters.

Considering that for the diagnosis purposes the engine testing cell is simply a source of electrical signals, here it is proposed to design a device to simulate those electrical signals that come from the engine to the ECM. To address this need this paper describes the design and implementation of an engine simulator to diagnose ECMs.

2 Engine Simulator

The engine simulator is a device that generates in its respective operational ranges and signal nature, the electrical signals that the sensors and transducers installed on the engine produce at different work regimes [3].

The design of the simulator should take into account the following parameters:

2.1 Engine Simulator Design Parameters

Besides to generate the signals that simulate the engine at different working regimes, the engines simulator must fulfill the following characteristics [4]:

- Real time operation.
- Display the value or state of each one of the variables that controls and monitors the performance of the engine. It must report the physical value of the measured variable instead of the representative electrical signal generated by the sensor. Table 1 lists the main variables to be considered.
- Include protective systems to avoid electrical overloads to the ECM.
- Include a self-diagnosis system to verify the correct operation of the simulator
- Allow its reconfiguration to simulate different engine models.
- Generate the typical values that are commonly reported by the sensors installed in the engine when it operates at low, medium and high load under steady state conditions. This involves, for example, the simulation of the engine temperature that typically is reported by this sensor when the engine operates at low, medium and high load.
- Allow the manual variation of the most important variables. This allows users to simulate arbitrary work conditions of the engine and to induce, on purpose, errors in the sensors and transducers.
- Include a multipurpose data registration system. It could be used for example, to save results of a diagnosis process made with the simulator. It should contain the operator identification, date, ECM model, type of tests performed, ECM error codes, etc.
- Include a security system to restrict the use by non-authorized users.
- Operate under adverse environmental conditions of temperature, humidity, vibration and shocks.

Table 1. Set of variables that usually area monitored in an engine.

No	VARIABLES	No	VARIABLES
1	Engine speed	11	Brake switch
2	Accelerator angle	12	Test signal for diagnosis
3	Engine position	13	Cruise control signal Set/Resume
4	Intake pressure	14	Cruise control signal On/Off
5	Barometric pressure	15	Engine break - On/Off
6	Coolant temperature	16	Ralenti - Increase/Decrement
7	Intake manifold temperature	17	Ignition
8	Oil temperature	18	Coolant level - High/Low
9	Oil pressure	19	Injector 1/2/3/4/5/6
10	Clutch switch		

2.2 Simulator Design

The engine simulator has the following internal modules:

- Power
- Communication
- Control
- Data storage
- User interface
- Signal conditioning and electrical protections

Figure 1 shows the different subsystems that are included in the simulator. Next, a description of each module is presented.

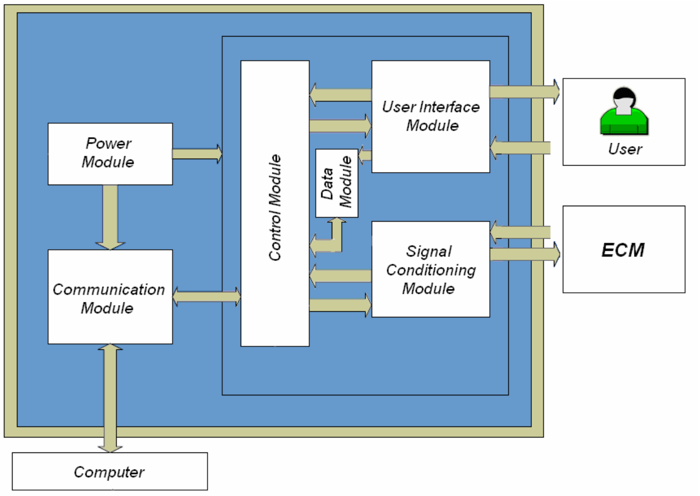


Fig. 1. General configuration of the engine simulator.

Power module. It provides the energy that under normal conditions the battery of the vehicle provides to the ECM. Additionally, this module is the power supply of each one of the electronic devices included within the simulator. For its operation, the power module requires a 120V AC connection. The module has a capacity of 200Watts distributed on voltage outputs of ±12V and ±5V. [5].

Communication module. It is designed to control the serial communication with any computer under RS232 protocol. It also includes a software developed in Java [6] to download the information stored in the simulator through a user interface.

Control module. It is the main module of the simulator. It includes a commercial microprocessor. It sends instructions and information to the other modules using a data bus under the I2C protocol.

This module contains the information related to the calibration curves of sensors and transducers and the routines for the self-diagnosis of the simulator, ECM errors report and data storage.

Data module. The main components of this module are a real time clock and a serial memory EEPROM of 2kbit which are controlled by the control module. Its main function is to acquire and store the information gathered during the diagnosis process. The stored information is used to evaluate the performance of the simulator. In the near future the historic information stored in the simulators will also be used to design new test procedures to identify ECMs problems.

User interface module. This module includes 3 commercially available microcontrollers [[7]].

As input elements this module has a matrix keyboard. This element allows the user to enter passwords, values or to choose the standard test to be made. Additionally, there are switches and knobs to allow users, in an independent way, to control the ignition module and the value of each variable.

As output elements this module has indicators leds, 7-segments displays and a liquid crystal display - LCD. These devices allow the operator to know the state or value of the different variables that are being simulated.

The user interface module make possible the interaction with the user to change the configuration of the engine simulator, select between manual o automatic operation, carry out standard test, perform a self-diagnosis test, modify the calibration curves of sensor and transducers, display the information stored in memory, include new users and create and change passwords. Different security passwords are handled to protect the different levels of information and operation of the engine simulator.

Signal conditioning and electrical protections module: This module generates the electrical signals that simulate each of the sensor and transducer of the engine. In most of the cases, the sensors generate analogous signals [[5]]. To simulate these signals, 8-bit digital potentiometers controlled through I2C data networks were used. The speed sensor and the crankshaft position sensor generate signals that have the form of a square wave. For these cases, the PWM modules of a microcontroller of the same family than the mentioned before were used to simulate them. The variables that are controlled through switches do not require any additional type of control within the engine simulator.

3 Implementation and Operational Tests

Electrical circuits were designed to perform the operations described before. In all the cases commercial components with the latest technology were used. Additionally, it was designed and manufactured a chassis to keep the electrical circuits and to place the user interface components described previously. Figure 2 shows the first prototype of engine simulator manufactured.

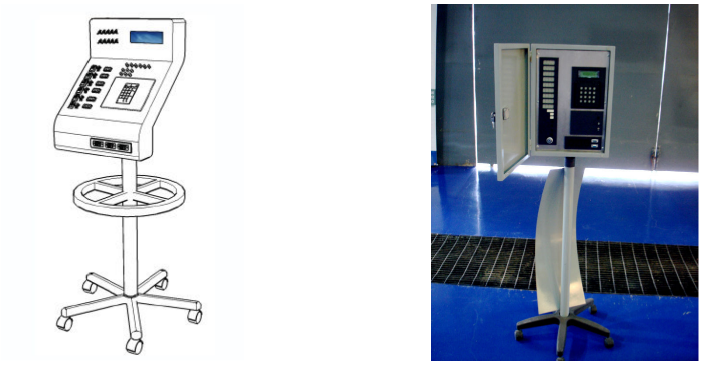


Fig. 2. Engine simulator manufactured to diagnose ECMs.

Currently, a set of operational tests has been performed on the engine simulator prototype to evaluate its performance. In addition, a set of physical endurance tests has been performed to verify its resistance to adverse conditions of vibrations, temperature and humidity.

As future work it has been considered to use the information stored in the data module to strengthen the current standard tests and to design new standard tests to diagnose the most frequent ECMs problems.

4 Conclusions

An engine simulator was designed and manufactured to semi-automate the diagnosis process of ECMs problems. This device integrates the state of the art technology to process, condition, transmit and store information. The device simulates the nature and intensity of the electrical signals generated by the sensors and transducers commonly installed on commercial engines. It can simulate different models of commercial engines. The device was programmed to carry out in an automatic way a set of standard tests to identify the most common ECMs failures. It incorporates systems to guarantee the integrity and safety of the information gathered during the diagnosis process and the physical integrity of the ECMs being diagnosed.

Currently, a set of operational tests has been performed on the engine simulator prototype to evaluate its performance. In addition, a set of physical endurance tests has been performed to verify its resistance to adverse conditions of vibrations, temperature and humidity.

As future work it has been considered to use the information stored in the data module to strengthen the current standard tests and to design new standard tests to diagnose the most frequent ECMs problems.

Acknowledgments. Financial founding to develop this project was provide by Automovil y Tecnología S. de R.L de C.V. The authors thank engineer Roberto Barraza García by his contributions to the development of this project.

References

1 DuPuy, R.K.: Automotive Electrical and Electronic Systems. Chek-Chart, USA. (2000)

2 Plint, M. y Martyr, A. Engine Testing: Theory and practice. Butterworth – Heinemann, 1997

3 Muñoz, M. y Payri, F. Motores de Combustión Interna Alternativos. E.T.S.I.I Fundación General-UPM. Madrid: España (1989).

4 Cummins S. de R.L. de C.V., http://www.cummins.com

5 Horowitz, P., Hill, W.: The Art of Electronics. Cambridge University Press, United Kingdom (1980).

6 Deitel, P., Deitel, H.: Java How to Program. Prentice Hall, United States of America (2006)

7 Freescale Semiconductor, http://www.freescale.com

Performance comparison on secure processors using a temporized page encryption technique

Osvaldo Espinosa Sosa, Luis Villa Vargas and Oscar Camacho Nieto

Center for Computing Research CIC-IPN, Av. Juan de Dios Bátiz, esq. con Miguel Othón de
Mendizábal, México, D.F., 07738. México
espinosa@cic.ipn.mx , lvilla@cic.ipn.mx, oscarc@cic.ipn.mx
Phone: +(55)57296000 ext. 56519

Abstract. Secure processors have been proposed as the solution for problems in computer security such as hardware attacks, viruses and intruders as well as piracy of software. Researchers have proposed several techniques based on encryption of memory contents where the processor is the only capable entity of decrypt that information before use it. It is evident that encryption and decryption processes increase security levels on computer systems but there is a real penalty on processor performance due to the higher memory access latency. This paper shows the effect on performance of secure processors using temporized techniques for memory page encryption. The main objective is to show performance loses when secure processors are compared with baseline architectures (insecure processors). Performance loses are well justified with the major security level offered with the inclusion of encryption and decryption system. Two cases are evaluated: the first one using Direct Encryption mode and the second one using Counter Mode Encryption.

1 Introduction.

Nowadays, new attacks to computer systems constantly appears and they are produced by malicious software that take advantage of operating systems vulnerabilities and hardware weaknesses in computing systems. Duplication in an illegal way of software (piracy) [1] is also a very important problem; causing millionaire loses to software industry. To reduce these problems, several techniques have appeared at microprocessor level [3]. In these techniques the microprocessor is the only entity authorized to access to information, any other hardware component is considered vulnerable to the attacks due to the fact that anybody can be monitoring the information flowing through the buses [4]. Programs are then stored in memory in an encrypted form and only can be decrypted on-chip, taking into account that the encryption engine is usually placed between level two of cache and main memory.

This paper shows the effect in the performance of a superscalar processor due to the inclusion of an encryption and decryption system including the capacity to encrypt

© A. Argüelles, J. L. Oropeza, O. Camacho, O. Espinosa (Eds.)
Computer Engineering.
Research in Computing Science 30, 2007, pp. 97 - 105

pages of main memory at regular intervals of time, changing on every encrypted page the key to be utilized in order to raise the security level. It is clear that this encryption system will add more latency to main memory access and this will affect the overall processor performance. It is important to find a well balanced configuration between memory page size and the period of time used by page encryption in a manner that performance will not be affected severely and it can offer an adequate security level. This work contains a methodology used to evaluate the proposal (temporized page encryption) which uses an execution-driven simulator to obtain detailed statistics of realized experiments, results obtained are included also and finally we have conclusions and bibliographic references.

2 Encryption of memory in secure processors.

On previous work it has been proposed to encrypt data contained in main memory to offer a good security level against attacks [2]. The encryption and decryption system is usually inserted between level two of cache memory and main memory due to it is the place where processor performance is degraded in a minor quantity and because of the fact that levels one and two of cache memory normally reside on chip, it is used also as a bus interface with an insecure external world. When the processor performs a read operation to main memory, the data obtained must be decrypted to be used by the processor; likewise, when a write operation to main memory is performed the data must be encrypted before.

There are two approaches to do this: the first one is called *Direct Encryption Mode* where the encryption/decryption engine is placed serially between main memory and the second cache level. This mode encrypts and decrypts data moving between level two of cache and main memory. This encryption mode has the characteristic of exhibit the whole latency of encryption system and then an access to main memory increments its normal latency adding the encryption engine latency resulting in a higher total latency. The second approach is called *Counter Mode.* Unlike Direct Encryption system it does not need to wait until data arrives from memory (it does not work serially), instead of that, the system encrypts already well-known information at the moment of the access to memory such as memory address and/or the value of a counter, producing a new data called *data pad* which can be used for encryption or decryption. The data pad is calculated in parallel with memory access and the encryption/decryption latency is hidden with the memory access latency. When data read from cache (or ready to be written) and data generated for encryption engine (data pad) are available then both are XOR'ed to produce a new data decrypted to be read (or encrypted to be written). Latencies showed by Direct Encryption and Counter Mode Encryption are depicted on figure 1.

We can notice the minor total latency of Counter Encryption Mode. For encryption/decryption is usually utilized the AES algorithm (Advanced Encryption Standard) which is an algorithm of fixed parameters. The AES requires as input a data block of 16 bytes. In order to encrypt a cache memory line of 64 bytes, four AES blocks are required as shown in figure 2.

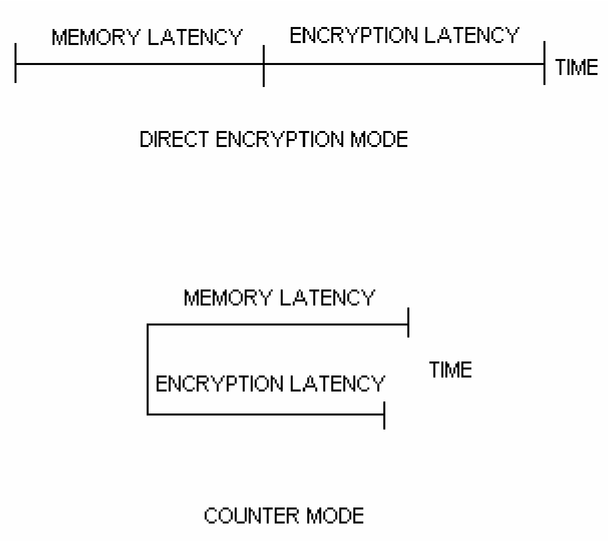


Fig. 1 Two approaches for encryption/decryption

The encryption or decryption process can use always the same key, which represents vulnerability due to the possibility that the algorithm can be broken by an expert intruder.

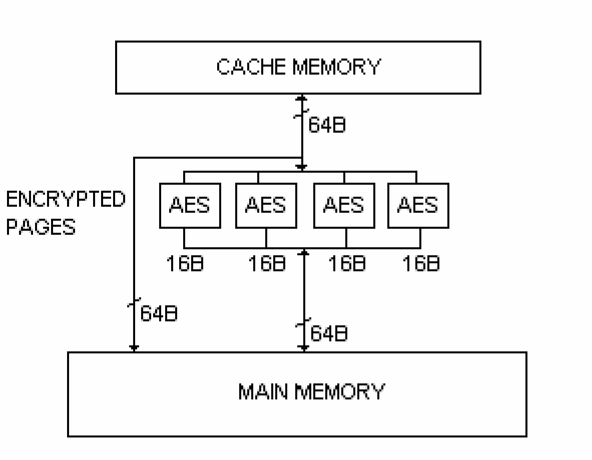


Fig. 2 Encryption and decryption circuit detail.

Other option is to change the key after certain number of encryptions and decryptions, however when the key is replaced the main memory must be re-encrypted causing that system could be stopped a large interval of time (in the order of seconds) in a system working at frequencies in the order of Ghz. To improve security in a computer system without important performance degradation we propose a system where periodically keys are replaced using a pull of keys, even more, we can use different keys for each memory page, as we are going to explain in the next section.

3 Proposed architecture.

Our encryption system is shown in figure 3. The main memory is divided in pages of fixed size (e.g. 4KB). There is a *timer* that activates the mechanism for key replacing periodically at regular intervals of time. Keys are generated in a random form. Every time this mechanism is activated, a memory page is decrypted using the old key and re-encrypted using the new key. Afterwards, the encrypted information is sent back to main memory.

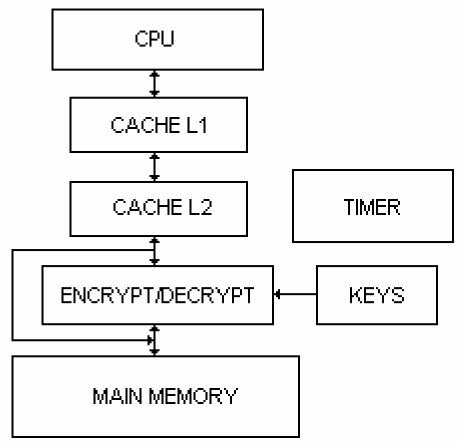


Fig. 3 Proposed architecture.

At the end of the period, a new page of main memory will be selected for re-encryption following a Round Robin scheme. The new encryption process will use a new key (and a new one for every page) to offer a higher level of security in comparison with proposals which use only one key for whole memory.

Using this new page encryption model the security level increases without important performance degradation. In order to know what memory pages are active on memory, the processor has a special group of page registers working together with operating system which is the responsible of resource management (memory in this case). The processor includes a set of special registers to store the old key for decryption (it comes from main memory) and other one to store the new key that will be used for page re-encryption before are sent back to main memory. It is important to notice the following fact: when a page is re-encrypted, information pass through the decryption circuit and data is sent back directly to main memory without affecting the cache contents.

4 Methodology.

We have evaluated our models using the simplescalar 3.0 simulator tool set which performs a very detailed simulation of a superescalar processor with out of order execution [5]. The simulator was configured as an Alpha 21264 because this architecture has been considered the best superscalar processor at its time of appearance. This processor contains two level 1 cache memories (Instructions and data) of 64 KB 2-way associative with 64 byte blocks. The second level of cache is unified with size of 1 MB being 8-way associative with 64 byte block. The SPEC CPU 2000 was used as a benchmark set, which is composed of twelve applications fixed point and fourteen floating point programs. In this work the performance is monitored whenever we change the page size or the period of time for page re-encryptions. Five hundred millions instructions were simulated for each program skipping the first 1 X 10^8 instructions with the aim of eliminate initialization effects on statistics. Results are shown in terms of IPC average for the 26 SPEC CPU programs.

5 Evaluation.

As we can see on figure 4, our reference is the bar labelled as *baseline*, which corresponds to an insecure processor and corresponds to the maximum performance attainable. We compare the *baseline* with Direct Encryption Mode (XOM) using pages of 4 KB, encrypting every 1,000,000 cycles and 100,000 cycles (XOM+1e6 and XOM+1e5 respectively). We can notice that performance is diminished significantly with the inclusion of encryption engine. Direct Encryption reduces performance to 90,84% in average respect to the baseline but including page encryption every 1,000,000 cycles performance degradation is minimal (descends to 90.23%) . If page encryption is now realized every 100,000 cycles the performance degradation is more important (84.73%). It is clear that increasing security (reducing re-encryption's period) performance will be reduced in a major form. Figure 5 shows how the page size impacts on performance. We consider pages of 4 KB, 8 KB and 16 KB (labelled XOM+4K, XOM+8K and XOM+16K respectively) and there is a comparison with baseline and Direct Encryption Mode (XOM). It is evident that increasing page size, performance decreases. In this case, page re-encryptions take place every 100,000 cycles. With pages of 4 KB the performance is 90.23% respect to the baseline and 85.62% for pages of 8 KB. The performance is even worse with 16 KB pages, where it is reduced until 69.84%. Figure 6 shows results with a period of 1,000,000 cycles between page re-encryptions. We can see that with a higher period the latency of encryption system is better hidden. As it is depicted, page size could be bigger and the system is less sensitive to the latency inserted by the encryption engine, having performances of 90.65% on average respect to the baseline with 8 KB pages and 88.04% with 16 KB pages.

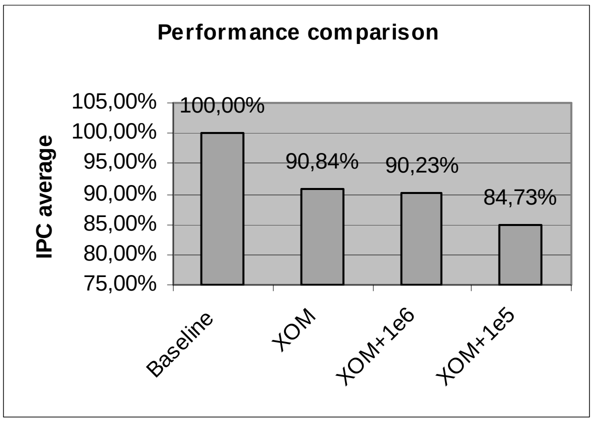


Fig. 4 Direct Encryption Mode with 4 KB pages and different periods.

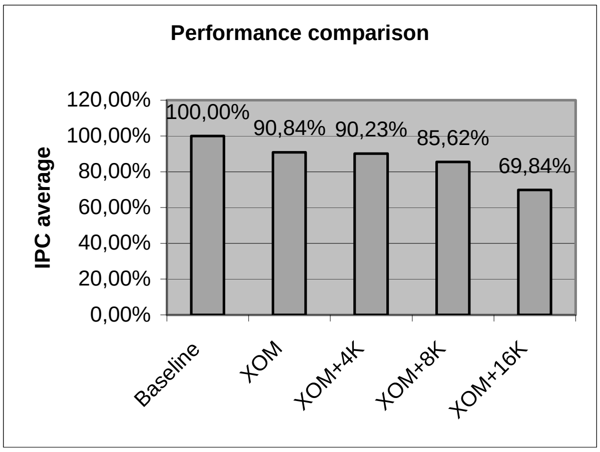


Fig. 5 Direct Encryption Mode with page encryption every 100,000 cycles.

On the other hand, figure 7 shows results for the Counter Mode Encryption system, the insecure processor is labelled baseline and represents the maximum performance attainable by the system. The Counter Mode only encryption system corresponds to the bar CM, and when we include page encryption every 1,000,000 and 100,000 cycles (labelled CM+1e6 and CM+1e5 respectively). Is evident that performance loses are lower than Direct Encryption Mode, in fact performance reductions for CM+1e5 is less than 5%. Figure 8 depicts the case of page re-encryption every 100,000 cycles, showing that we can select page sizes of 8 KB having the performance loss to approximately 10%, better than Direct Encryption case.

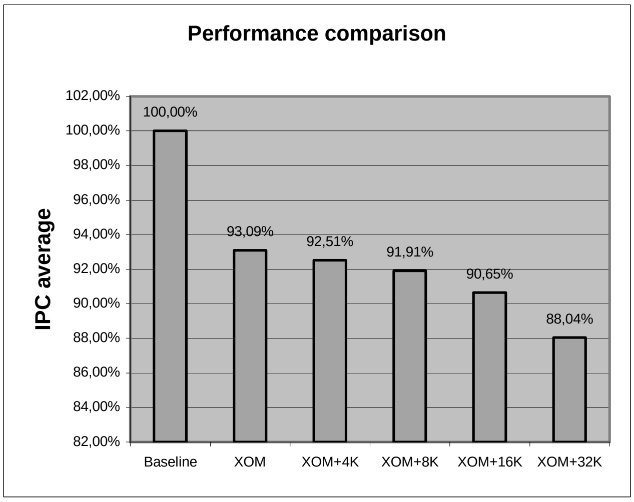


Fig. 6 Direct Encryption Mode with page encryption every 1,000,000 cycles.

Figure 9 corresponds to the case of page encryption every 1,000,000 cycles. Reducing the period of time between re-encryptions the system is less sensitive to an encryption process. We can notice that page size of 32 KB represents a performance loss of less than 5%.

Comparison of performance loses give us an idea that Counter mode is better in terms of performance instead of Direct Encryption which have better levels of security (by the fact that Counter Mode exhibits part of information through the buses for example).

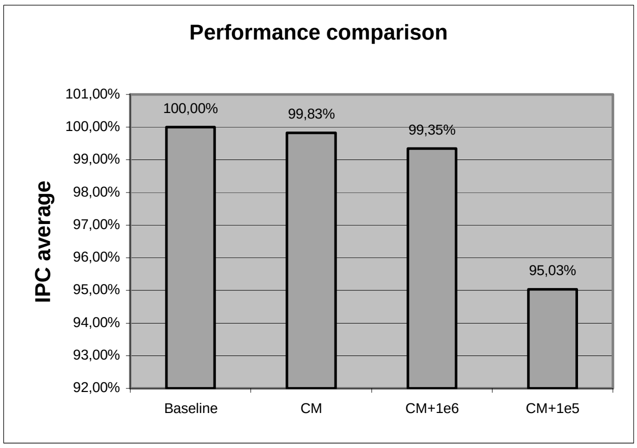


Fig. 7 Counter Mode Encryption with 4 KB pages and different periods.

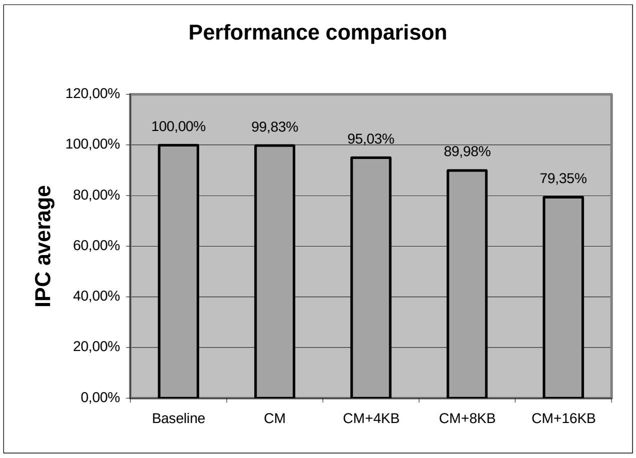


Fig. 8 Counter Mode Encryption with page encryption every 100,000 cycles.

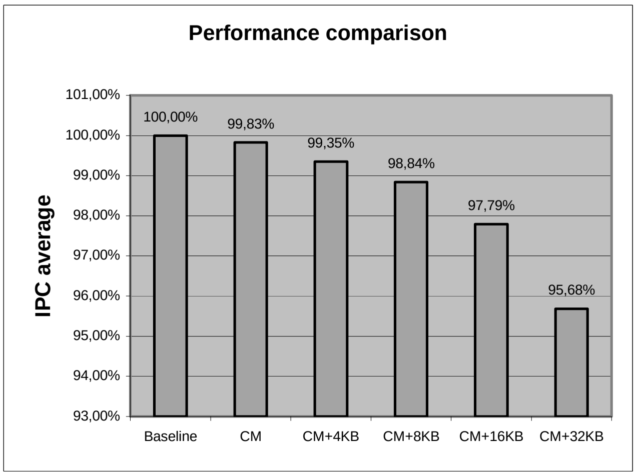


Fig. 9 Counter Mode Encryption with page encryption every 1,000,000 cycles.

6 Conclusions.

As it is studied in this work, to include an encryption system reduces processor performance in an important manner, if additionally we insert the temporized page encryption technique we are adding additional loss. These performance loses are well justified in terms of increased security. We can eliminate 10% of performance in order to gain a higher level of security. Evidently, choosing an adequate period for page re-encryption and the page size could increase even more the security of the system. Direct Encryption is better in terms of security but Counter mode is better for system performance.

7 References.

1. Yang,Zhang,Gao. Fast secure processor for inhibiting software piracy and tampering. Proceedings of the 36th International Symposium on Microarchitecture MICRO 36-2003.

2. Ruby B. Lee, Peter C. S. Kwan. Architecture for protecting critical secrets in microprocessors. Proceedings of the 32nd Annual International Symposium on Computer Architecture 2005.

3. T.Kgil, L.Falk and T. Mudge. ChipLock: support for secure microarchitecures. Workshop on architectural support for security and anti-virus, 2004.
4. Chenyu Yan, Brian Rogers et. Al. Improving cost, performance and security of memory encryption and authentication. International Symposium on computer architecture ISCA 2006.
5. Burger,Dough. The simplescalar toolset, version 3.0 *Computer Architecture News*, **25** (3), pp. 13-25, June, 1997.

Digital Fuzzy Inference Engine Simulator

Antonio Hernández Zavala[1], Oscar Camacho Nieto[1], Ildar Batyrshin[2], Osvaldo Espinosa Sosa[1]

[1]Instituto Politécnico Nacional, [2]Instituto Mexicano del Petróleo
antonioh_z@hotmail.com, oscarc@cic.ipn.mx, batyr@imp.mx, espinosa@cic.ipn.mx

Abstract. Fuzzy control has growing to many applications in the course of time since it was developed at early 60's. Recent works as they increase in complexity require every time more calculations; this is why usage of specific fuzzy hardware is becoming more suitable to satisfy processing time demands. It has been noticed that conventional fuzzy processors and simulators use operations such as multiplications and divisions over an input space given by the number of resolution bits used, this is reflected in the total time consumed to give a crisp result out from an input. To reduce time consumed we avoid usage of these equations by replacing them with adders and shifting. As representation of real functions in binary terms is really not continuous, membership functions are not represented at all elements, they are discretized into m quantization levels called $\alpha - levels$ which are on dependence of the number of resolution bits used for membership universe space and is independent of bits used for input space. There is no simulator that let us visualize how inference procedure is performed in terms of integer numbers as they are used by the computer architecture, on this work a "Digital Fuzzy Inference Engine Simulator (DiFES)", which is realized to satisfy fuzzy hardware design demands according to mentioned representation, is presented in order to visualize inference procedure at a hardware equations level to make easier this design process.

Keywords: Fuzzy Logic, Fuzzy Sets, Defuzzification, Discrete Numbers, Quantization, Inference Simulation.

1 Introduction

Since the first application of fuzzy logic to control a steam engine realized by Mamdani at 70's [1], several applications have grown up, among this we can mention Sugeno's work on an automatic parking car [2], high performance applications on water treatment [3][4], nuclear reactors [5][6], robotics[7], pattern recognition [8] and physics experiments [9], among others.

Fuzzy logic has been widely accepted because it can be used to efficiently translate human knowledge into control rules for different applications. It has been shown that a fuzzy inference controller is robust and gave better results than conventional controllers. There are researchers that have shown all advantages that fuzzy logic controllers offer, considering them as universal aproximators [10][11][12][13].

© A. Argüelles, J. L. Oropeza, O. Camacho, O. Espinosa (Eds.)
Computer Engineering.
Research in Computing Science 30, 2007, pp. 107 - 118

While applications are extended to more complex processes that require faster processing speed, they have turned to use specific inference hardware. First digital fuzzy processor was realized by Watanabe and Togai at 1985. Since then, many interesting architectures have been presented [14], [15], [16], [17] y [18]. Each of them has improved the processor features, obtaining every time less complicated hardware that realizes inferences faster, consuming less processing time and decreasing number of hardware resources.

With FPGA boom, because of its binary reprogrammable capabilities, some researchers have used this dispositive to implement fuzzy hardware [19][20][21][22][23], on this works digital circuitry is adapted according to the mathematics of fuzzy logic. The tendency is to avoid as possible time consuming operations as multiplications and divisions, these two operands need many instruction cycles to give a result.

Actually there is no simulator system where all steps required to implement digital algorithms of fuzzy logic are visualized, on this paper we give an approach to this, first, we give an introduction to fuzzy control where general controller is shown to explain its blocks digital realization on consequent chapters. A simulator on C++Builder programming language, was realized with equations shown in content, to finally give some results and conclusions.

2 Fuzzy Control

The parts which compose a fuzzy controller are shown on Fig. 1; outside the dotted line is the crisp type environment of the controller. Fuzzifier and defuzzifier are translators between crisp and fuzzy numbers at input and output respectively; rule base contains all the rules that describe system behaviour from where inference decides which rules use to make a decision. Each block is commented on following parts.

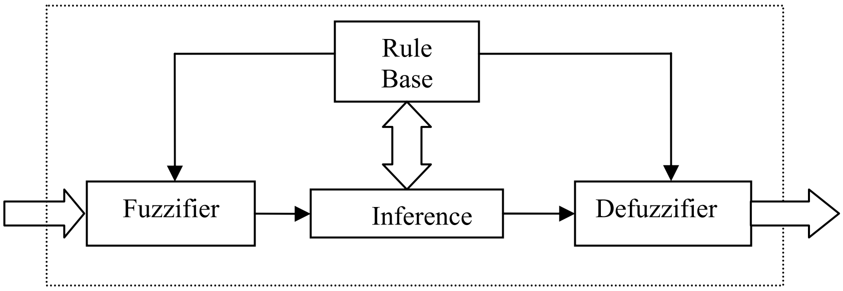


Fig. 1. Fuzzy controller blocks.

2.1 Fuzzifier

First block from left on Fig. 1 corresponds to the fuzzifier stage where the real input value is mapped into the corresponding universe of discourse, on binary numbers this is called discretization or quantization of input universe into *m* number of $\alpha - levels$ [24] and a membership value is assigned to each of *n* points in *x*.

As mentioned before, the operator fuzzifier can take different forms, on this document $\alpha - levels$ are used to assign different membership values to a membership function in the following way:

Let *M* be a trapezoidal membership function, with $x_i \in X; i = 0,1,\ldots,n, X \in [0, 2^n - 1]$; the number of points in *x* axis, and *m* discretization levels that give the resolution on *y* axis. Let's assume that we have the base of *M* with height equal to 0 or α_0, in the range from *x=0* and *x=n*; calling these two points initial ($x_0^{\alpha_0}$) and final ($x_f^{\alpha_0}$) values respectively, this segment will be called the base and its length is given by the following equation (1):

$$Lenght_{\alpha_0} = x_f^{\alpha_0} - x_0^{\alpha_0} \tag{1}$$

As we are using integer numbers, α_0 will have height equal to 0, then its area is equal to 0, this is an interval defined by the length of the segment and is used just to know if input *x* corresponds to this function as on (2).

$$IF\ x \in \left[x_0^{\alpha_0}, x_f^{\alpha_0}\right] THEN\ x \in M \tag{2}$$

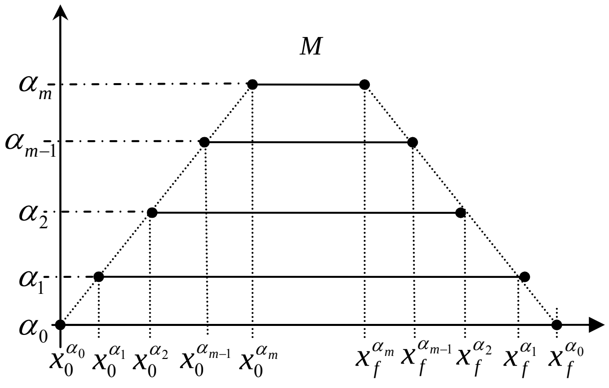

M
α_m
α_{m-1}
α_2
α_1
α_0
$x_0^{\alpha_0}$ $x_0^{\alpha_1}$ $x_0^{\alpha_2}$ $x_0^{\alpha_{m-1}}$ $x_0^{\alpha_m}$
$x_f^{\alpha_m}$ $x_f^{\alpha_{m-1}}$ $x_f^{\alpha_2}$ $x_f^{\alpha_1}$ $x_f^{\alpha_0}$

Fig. 2. Discretization of a trapezoidal membership function into $\alpha - levels$.

Every $\alpha - level$ for a membership function will be then defined by the following four parameters: *M* is the name of the function and is used to group $\alpha - levels$, $x_0^{\alpha_m}$ is the initial point and $x_f^{\alpha_m}$ is the final point of the interval; α_m is the height or membership value for all the elements in $\left[x_0^{\alpha_m}, x_f^{\alpha_m}\right]$. Fig. 2 shows a discretization of a membership function with m $\alpha - levels$ and $n = x_f^{\alpha_0}$ points on the space of X. The whole M membership function is then defined by the set of all its $\alpha - levels$ which describe it totally.

To evaluate a point on M we check first (2) and if true we can test on which $\alpha - levels$ of M this point matches and keep record of which was the higher membership value, and the name of the function. With this information for each function activated we can perform inference procedure, which will be shown on the following section.

2.2 Mamdani Inference

Inference procedure used here is Mamdani form corresponds to middle blocks on Fig.1, exemplified on Fig. 3 with a two input-one output fuzzy system, with two and three membership functions respectively; $x \in X, y \in Y, z \in Z$ to form truth table, there is an output function activated for each pair of inputs. When two points *(x, y)* are evaluated as input, they can be operated with any *t*-norm or *t*-conorm to realize conjunction or disjunction, this gives as result the height on α and the membership function to be evaluated.

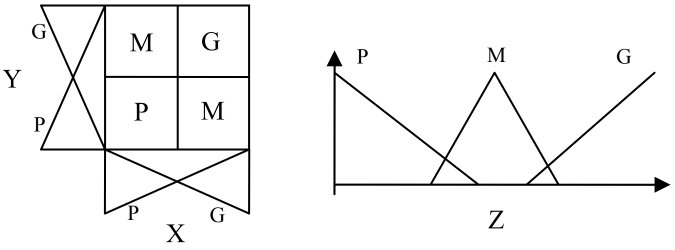


Fig. 3. Mamdani Inference Example

Let us consider as example: an input value is in *M* membership function when (2) is true, min operator is used between *x* and *y* inputs to set the height of the rule to be evaluated according to expression (3), *z* is a one-dimensional weight vector used to obtain the shape of output which is completely defined by its $\alpha - levels$.

$$z_i = \min\left(\alpha_{x_i}, \alpha_{y_i}\right); \qquad 0 \le i \le Rules_{TOT} \tag{3}$$

Next step is to create homogeneous intervals, this is realized using *min* and *max* operators on each level for each activated function obtaining the length for output function, this is repeated for each $\alpha_i < \alpha_{\max}$ until reaches maximum membership value $\alpha_{\max}$, the initial and final points are obtained with (4). The resultant set of $\alpha - levels$ is aggregated to obtain the area, this is realized with (5). Up to now we have described the shape of resultant function, which is next, is to defuzzify and obtain a crisp output in the following section.

$$\alpha_i^z \begin{cases} z_0^{\alpha_i^z} = \min\left(z_0^{\alpha_i^{PP}}, z_0^{\alpha_i^{PG}}, z_0^{\alpha_i^{GP}}, z_0^{\alpha_i^{GG}}\right) \\ z_f^{\alpha_i^z} = \max\left(z_0^{\alpha_i^{PP}}, z_0^{\alpha_i^{PG}}, z_0^{\alpha_i^{GP}}, z_0^{\alpha_i^{GG}}\right) \end{cases} \tag{4}$$

$$A_Z = \sum_{i=0}^{\alpha_{\max}} \left(z_f^{\alpha_i^z} - z_0^{\alpha_i^z}\right)\alpha \tag{5}$$

2.3 Defuzzification

There exists many methods to defuzzify, right block on Fig. 1, all of them satisfy different precision needs and speed requirements, most commonly used on literature for digital implementations are center of gravity COG [25], center of average [26], among others. A good research on this is found at [27], where new methods are introduced and compared with previous ones. Another form of fast defuzzifier is on [28].

We have distinguished that all of this methods need at least $k-1$ iterations according to the input spaces given by 2^n where n is the number of bits used. We will use Center of Slice Area Average defuzzifier, on which we need $\alpha_{\max}$ iterations and sum midpoints of every α_i, then divide them between the maximum α reached, this is expressed by (6). As example on Fig. 4, there is a case where $m=4$ here, the 4 centers are shown for every α_i, and output is marked as COSAA.

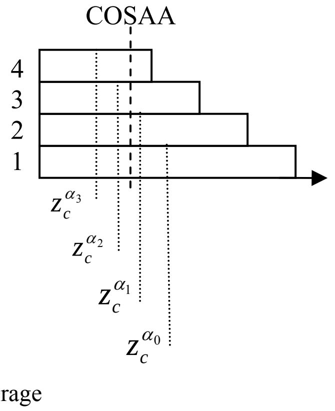


Fig. 4. Center of Slice Area Average

$$COSAA=\frac{\sum_{i=0}^{\alpha_{\max}}\left(\frac{\left(x_f^{\alpha_i}-x_0^{\alpha_i}\right)}{2}+x_0^{\alpha_i}\right)}{\alpha_{\max}} \tag{6}$$

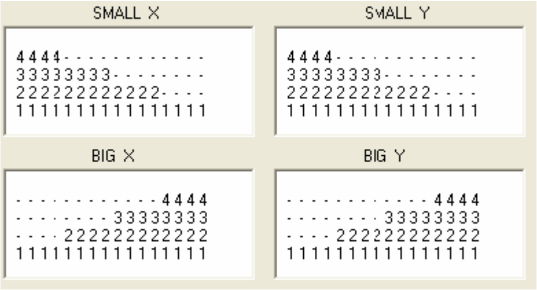
3. Fuzzy Inference Simulator (DiFES)

Methodology presented here was simulated on a C++Builder program on which every inference stage was implemented according to equations (1-6). Here two inputs with two membership functions each and one output with three membership functions as on Fig. 3 is discretized at 2 bits for membership i.e. four $\alpha - levels$, and 4 bits for *x* axis, i.e. 16 values on the input universe.

DiFES Simulator is divided on four sections: Input Functions, Output Functions, Inference Evaluation and Defuzzification is the last section, following sections detail them.

3.1 Input Membership Functions

First section corresponds to the input functions for both inputs *x, y* shown on Fig. 5 where correspondent functions are for *x* input: SMALLx and BIGx, for *y* input are SMALLy and BIGy, they are shown separately with the objective of visualize them in a better way. Numbers that compose each function are used to identify its membership value, which is at least 1 and at most 4. These functions are identical for each input and are defined by equation (7) for SMALL and (8) for BIG, where can be noticed that equations correspond to trapezoids defined by intervals and are overlapped each other. As can be seen and as we are using two bits to represent membership value a continuous function can not be represented because of discretization.



```
SMALL X                            SMALL Y
4444------------                   4444------------
33333333--------                   33333333--------
222222222222----                   222222222222----
1111111111111111                   1111111111111111

BIG X                              BIG Y
------------4444                   ------------4444
--------33333333                   --------33333333
----222222222222                   ----222222222222
1111111111111111                   1111111111111111
```

Fig. 5. Input Membership Functions for *x* and *y*.

$$\mu_{SMALL} = \begin{cases} \alpha_{MAX}, & [0,3] \\ -\dfrac{\alpha_{MAX}}{12}x + 4, & [3,15] \end{cases} \tag{7}$$

$$\mu_{BIG} = \begin{cases} \dfrac{\alpha_{MAX}}{12}x, & [0,12] \\ \alpha_{MAX}, & [12,15] \end{cases} \tag{8}$$

Following code is realized to create the set of $\alpha - levels$ that define input membership functions (7) (8) for the variable *x*.

```
For (int a=0; a<=3;a++)  //**********SMALLx construction
        {
         MF[a].function=1;
         MF[a].start=0;
         MF[a].end= 15 - 4*a;
         MF[a].area = (MF[a].end - MF[a].start) +1;
         MF[a].alfalevel=a+1;
        }
For (int a=4; a<8;a++)  //*********BIGx Construction
        {
         MF[a].function=2;
         MF[a].start=4*(a-4);
         MF[a].end=15;
         MF[a].area = ((MF[a].end - MF[a].start)+1);
         MF[a].alfalevel=a-3;
        }
```

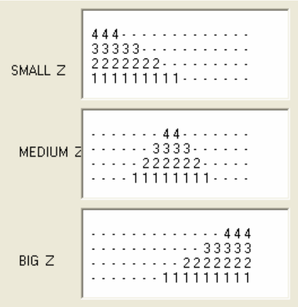
3.2 Output Membership Function

Second part corresponds to output membership functions for output variable *z*, they are SMALLz, MEDIUMz and BIGz; these functions are defined by the equations (9-11), Fig. 6 shows simulator output membership functions, according to the equations mentioned. Discretization is like on previous section at 2 bits for membership and 4 bits for input space.

$$\mu_{SMALLz} = \begin{cases} \alpha_{MAX}, & [0,2] \\ -\dfrac{\alpha_{MAX}}{12}x + 4, & [2,8] \end{cases} \tag{9}$$

$$\mu_{MEDIUMz} = \begin{cases} \dfrac{\alpha_{MAX}}{3}x - 4, & [4,7] \\ \alpha_{MAX}, & [7,8] \\ -\dfrac{\alpha_{MAX}}{3}x + 11, & [4,7] \end{cases} \tag{10}$$

$$\mu_{BIGz} = \begin{cases} \dfrac{\alpha_{MAX}}{6}x - 4, & [7,13] \\ \alpha_{MAX}, & [13,15] \end{cases} \tag{11}$$



```
444-------------
33333-----------
2222222---------
111111111-------
```

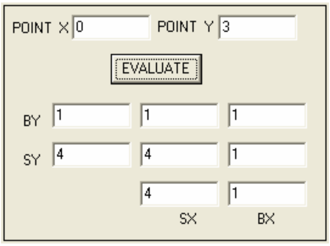
```
-------44-------
------3333------
-----222222-----
----11111111----
```

```
-------------444
-----------33333
---------2222222
-------111111111
```

Fig. 6. Output Membership Functions

3.3 Inference Evaluation

Third section of simulator is where crisp inputs are introduced for *x* and *y* over the input universe (4 bit), on top of Fig. 7, these values are evaluated on input membership functions for each when pushing EVALUATE button.



POINT X 0 POINT Y 3

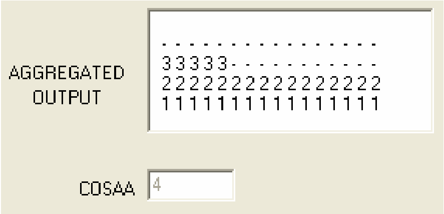
BY	1	1	1
SY	4	4	1
		4	1
		SX	BX

Fig. 7. Inference Evaluation

Down evaluation button, there are two columns labelled SX and BX, they correspond to membership functions SMALLx and BIGx respectively; the rows are the same for *y* input, SY, BY correspond to SMALLy and BIGy respectively. Each of mentioned labels has a text box where membership value activated for that respective input is shown, these values are used to generate rule weight matrix with the intersection points using *min* operator as on (3).

3.4 Defuzzification

After stages mentioned, last to do is to obtain a resultant membership function, by means of aggregation operator (5), and to get a crisp output from it using a defuzzifier (6). Where figure is the resultant aggregated function, and COSAA is the crisp type output value obtained from center of slice area average method. On top of Fig. 8, there is aggregated output membership function which is used to obtain defuzzified output COSAA as a crisp value over the output universe.



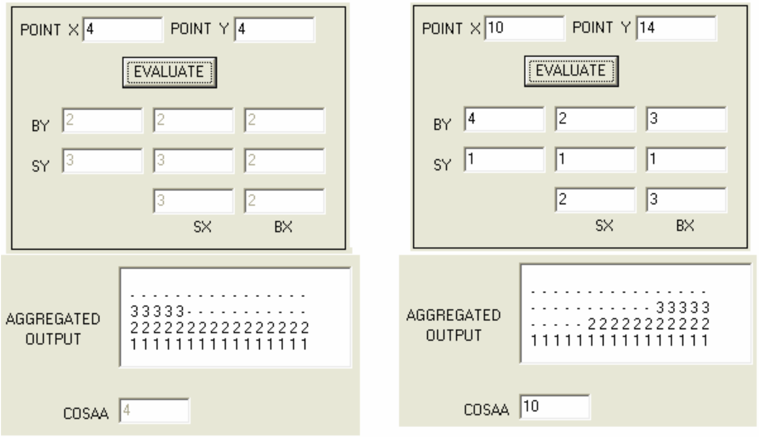
AGGREGATED OUTPUT

```
- - - - - - - - - - - - - - - -
3 3 3 3 3- - - - - - - - - - -
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

COSAA 4

Fig. 8. Aggregation and defuzzification of output membership function

4 Results



POINT X 4 POINT Y 4

BY 2 2 2

SY 3 3 2

SX 3 BX 2

AGGREGATED OUTPUT

```
- - - - - - - - - - - - - - - -
3 3 3 3 3- - - - - - - - - - -
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

COSAA 4

(a)

POINT X 10 POINT Y 14

BY 4 2 3

SY 1 1 1

SX 2 BX 3

AGGREGATED OUTPUT

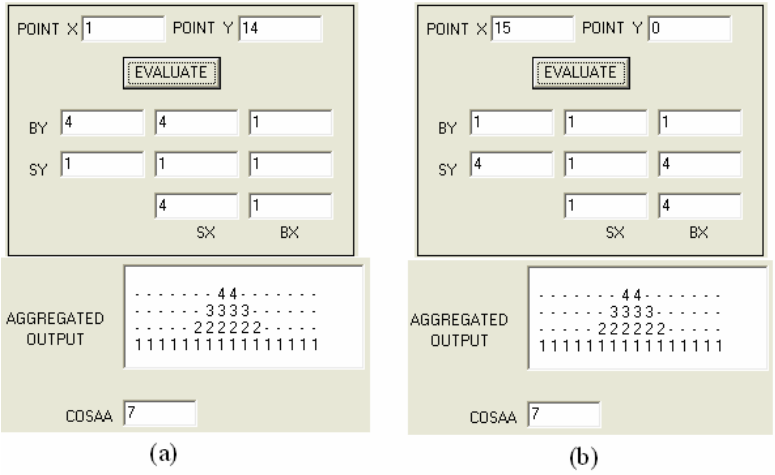
```
- - - - - - - - - - - - - - - -
- - - - - - - - - - - 3 3 3 3 3
- - - - - 2 2 2 2 2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

COSAA 10

(b)

Fig. 9. Simulation Results for Two Different Cases, a) x=4, y=4. b) x=10, y=14

Simulator DiFES presented here was tested for different pairs of inputs *x*, *y*, where for each case a defuzzified crisp output was obtained using input and output membership functions as mentioned on previous sections. As can be seen on Fig. 9 a and b and on Fig. 10 a and b, here are four different cases, on top there are inputs for both points x and y, when evaluating this points, we obtain a crisp output at the bottom part of each figure, as result of COSAA defuzzifier. It can be easily seen that aggregated output membership function, changes its form according to the different cases presented.



(a)

POINT X 1 POINT Y 14

BY 4 4 1

SY 1 1 1

SX 4 BX 1

AGGREGATED OUTPUT

```
-------44-------
------3333------
-----222222-----
1111111111111111
```

COSAA 7

(b)

POINT X 15 POINT Y 0

BY 1 1 1

SY 4 1 4

SX 1 BX 4

AGGREGATED OUTPUT

```
-------44-------
------3333------
-----222222-----
1111111111111111
```

COSAA 7

Fig. 10. Simulation Results for another Two Different Cases, a) x=1, y=14. b) x=15, y=0

5 Conclusions

In this work, there appeared the tool of simulation DiFES, which was used to demonstrate and to conclude that there is a form to represent fuzzy logic inference engine by means of simple operations such as additions and shifting, which gives as result a reduction on time consumed during complex calculation operands on conventional implementations [25][26][27][28], which use at less *n-1* iterations, our propose iteration number is less or equal than the *m* number of $\alpha - levels$ that is proportional to the number of bits used to discretize *y* axis into a truth space with height equal to 2^n-*1*.

The $\alpha - level$ method used to represent membership functions can be adapted to almost any shape, because it is represented as square slices with α height and a variable interval as its length and its resolution its on dependence on how many bits are used to discretize membership values.

Defuzzification method used calculates the center of area of every slice that forms aggregated output membership function, in terms of $\alpha - levels$ and calculate an average from them, requiring at most *m-1* iterations.

There justifies itself the need of the design of tools of simulation that allow us to experiment with new proposals of hybrid systems fuzzy-hardware.

6 References

1. Mamdani E. H., and Assilian S.: A case study on the application of fuzzy set Theory to Automatic Control. Proceedings of IFAC Stochastic Control Symposium. Budapest. (1974).
2. Sugeno M. and Murakami K.: An experimental study of fuzzy parking control using a model car. In: M. Sugeno, (eds.): Industrial Applications of Fuzzy Control. North-Holland, Amsterdam. (1985). 125-138.
3. Itoh 0., Gotoh K., Nakayama T., and Takamizawa S.: Application of fuzzy control to activated sludge process. Proceedings of 2ndIFSA Congress. Tokyo, Japan. (1987). 282-285.
4. Yagishita 0., Itoh 0., and Sugeno M.: Application of fuzzy reasoning to the water purification process. In M. Sugeno (eds.): Industrial Applications of Fuzzy Control, Amsterdam: North Holland. (1985). 19-40.
5. Kinoshita M., Fukuzaki T., Satoh T., and Miyake M.: An automatic operation method for control rods in BWR plants. In Specialists' Meeting on In-Core Instrumentation and ReactorCore Assessment. Cadarache, France. (1988).
6. Bernard J. A.: Use of rule-based system for process control. IEEE Control Systems Magazine Vol. 8, no. 5. (1988). 3-13.
7. Ruspini E. H.: Reactive Fuzzy Control of Autonomous Robots. On Hirota, Sugeno (eds.): Industrial Applications of Fuzzy technollogy in the world. World Scientific. (1995)
8. Ostrowski D. J. and Cheung P. Y. K.: A fuzzy Logic Approach to Handwriting Recognition. On M. j. Patyra, D. M. Mlynek (eds): Fuzzy Logic, Implementation and applications. Chapter 10. (1996)
9. Falchieri D., Gabrielli A., Gandolfi E., Masetti M.: Design of Very High Speed CMOS Fuzzy Processors for Applications in High Energy Physics Experiments. Seventh International Conference on Microelectronics for Neural, Fuzzy, and Bio-Inspired Systems. Granada, Spain. (1999). 284-291
10. Kosko B.: Fuzzy systems as universal aproximators. IEEE international conference on Fuzzy systems. Vol. 43 No. 11. (1992). 1329-1333.
11. Wang L. X.: Fuzzy systems are universal aproximators. IEEE Transactions on Systems Man and Cybernetics. (1992). 1163-1170.
12. Buckley: Sugeno type controllers are universal controllers. Fuzzy sets and systems. Vol. 53 No. 3. (1993). 299-303.
13. CastroJ. L.: Fuzzy logic controllers are universal aproximators. IEEE Transactions on Systems, Man and Cybernetics. (1995)
14. Togai M.: Expert system on a chip: an engine for real-time approximate reasoning. Proceedings of the ACM SIGART international symposium on Methodologies for intelligent systems. (1986). 147-154
15. Watanabe H.: RISC approach to design of fuzzy processor architecture. In Proceedings of 1st International Conference on Fuzzy Systems. (1992). 431–441
16. Cardarilli G.C., Re M., Lojacono R.: VLSI Implementation of a Real Time Fuzzy Processor. Journal of Intelligent and Fuzzy Systems. No. 3, Vol. 6. (1998). 389-401
17. Gabrielli A., Gandolfi E.: A Fast Digital Fuzzy Processor. IEEE Micro, Vol. 19, Issue 1, Jan.-Feb. (1999). 68-79.

18.Ascia G., Catania V., and Russo M.: VLSI Hardware Architecture for Complex Fuzzy Systems. IEEE Transactions on Fuzzy Systems. Vol.7, No.5. (1999). 553-570.

19. Surmann H., Ansgar Ungering and Karl Goser. Optimized Fuzzy Controller Architecture for Field Programmable Gate Arrays. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo. (1993). 124-133.

20. V. Salapura and V. Hamann. Implementing fuzzy control systems using VHDL and Statecharts. Proc. of the European Design Automation Conference EURO-DAC '96 with EURO-VHDL '96. Geneva, Switzerland. September. IEEE Computer Society Press. (1996). 53-58.

21. S. Sánchez-Solano, R. Senhadji, A. Cabrera, I. Baturone, C. J. Jiménez, A. Barriga. Prototyping of Fuzzy Logic–Based Controllers Using Standard FPGA Development Boards. Proc. 13th IEEE International Workshop on Rapid System Prototyping. Darmstadt. (2002). 25-32.

22. F.J. Garrigos-Guerrero y R. Ruiz Merino. Implementacion de sistemas fuzzy complejos sobre FPGAs. Computación reconfigurable y FPGAs. (2003). 351-358.

23. R. Raychev, A. Mtibaa, M. Abid. VHDL Modelling of a Fuzzy Co-processor Architecture. International Conference on Computer Systems and Technologies – CompSysTech (2005).

24. K. Uehara & M. Fujise, Fuzzy inference Based on Families of a-level Sets, IEEE Transactions on Fuzzy Systems, vol. 1, No. 2. May (1993). 111-124.

25. G.C. Cardarilli, M. Re, R. Lojacono, M. Salmeri, A new Architecture for High-Speed COG based Defuzzification. International Workshop on Tool Environments and Development Methods for Intelligent Systems, TOOLMET' 97, pp., April 17-18, (1997). 165-172.

26. Gaona, A.; Olea, D.; Melgarejo, M. Distributed arithmetic in the design of high speed hardware fuzzy inference systems;, 2003. NAFIPS 2003. 22nd International Conference of the North American Fuzzy Information Processing Society. 24-26 July (2003). 116 – 120.

27. Banaiyan, A.; Fakhraie, S.M.; Mahdiani, H.R.; Cost-Performance Co-Analysis in VLSI Implementation of Existing and New Defuzzification Methods. Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce. Volume 1. (2005). 828 – 833.

28. Hakaru Tamukoh, Keiichi Horio and Takeshi Yamakawa A bit-shifting-based fuzzy inference for self-organizing relationship (SOR) network IEICE Electronics Express, Vol. 4, No. 2. (2007). 60-65.

The Actual Back-End Tendency in Speech Synthesizer

Sergio Suárez Guerra[1]; Ismael Díaz Rangel[1]

[1] Center for Computing Research, National Polytechnic Institute,
Juan de Dios Batiz esq Miguel Othon de Mendizabal s/n, P.O. 07038, Mexico
ssuarez@cic.ipn.mx, idra06@sagitario.cic.ipn.mx

Abstract. At the moment the models of greater use used for the implementation of the synthesis of voice in their stage back-end are: the articulator model, that still is in an immature stage, but is outlined in the future like the model with greater potential; the model by formants, that given its limitation an artificial voice but with great flexibility and intelligibility produces, and the model by concatenation, that enjoys a great popularity nowadays, nevertheless, require of greater computational cost and it practically lacks synthesis flexibility. It is possible to mention that great challenges of the voice synthesis are in their stage front-end, since the determination of how to pronounce the numbers, abbreviations, acronyms, names, etc. they are possible to be turned problems of difficult solution; as well as the correct analysis of prosody. It writes or it sticks the text here to translate.

Keywords: Synthesis, back-end, Front-end, Formants, Glottal Excitation.

1 Introduction

The speech is the main form of communication between the people. A synthesizer of voice is a device able to create of artificial way articulated voice [1]. A type of synthesizer of voice is the call Text Speech System (TTS). The TTS has the capacity to read any text aloud, or introduced by a user, or generated by a system of OCR (Optical Character to Recognizer), or even pertaining when coming out of a system of consultation with data with the results of a request on the part of a user.

The fundamental difference with other systems talking (like it could be a tape reproducer), is that our interest is centered in the capacity to reproduce new phrases or texts automatically, which eliminates of the process the idea that a recording of such mediates. Even so, it can that we need to refine plus our initial definition: systems that, for example, simply concatenate prerecorded words or phrases (typically calls systems of vocal answer), are only applicable when the vocabulary of the application very is limited, of the order of few hundreds of words. In the context of systems TTS,

© A. Argüelles, J. L. Oropeza, O. Camacho, O. Espinosa (Eds.)
Computer Engineering.
Research in Computing Science 30, 2007, pp. 119 - 128

it is almost impossible to raise the recording of all the words of the language, so that he is more reasonable to define them as automatic production systems of speech, through a process of transcription of graphemes to phonemes. It is capacity to read must be distinguished to the language and geographic location of the user; since the diction is different not only between languages, but that even in he himself language (it does not sound identical the Spanish in México that the Spanish in Cuba). In this sense, a good margin in México exists to work in voice synthesis.

A system TTS consists basically of two modules (figure 1), the module of text analysis to which we will call "front-end" and the module of speech synthesis to which we will call "back-end". In the analysis module, the following operations are made: normalization of text and abbreviations, syntactic analysis, semantic analysis, syllabication, accentuation and converter grapheme-allophone (the most elementary unit of the sound). Once analyzed the text, it is had a set of textual parameters that are necessary in the synthesis process; for example, Mr. by Mister is translated and he is indicated that the second syllable must go marked, eliminate the letters that are not pronounced and the possible places of prosodic changes are marked.

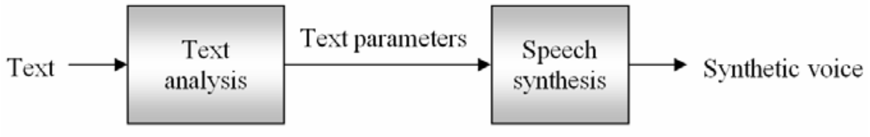


Fig. 1 Simplify diagram of TTS system

In the implementation of a system of voice synthesis first that we must consider it is the model for the stage back-end. At the moment the synthesis is dominated by three systems, that depending on the requirements of the application we must show preference for one or another one. In order to sustain this idea, in the following points we are going to describe briefly to each one of them.

2 Articulator Model

The articulator synthesizers provide synthetic voice of high quality, but its disadvantage is that the parameters are very difficult to obtain and to control them automatically.

The voice wave is the answer to the system of filters of vocal tract for one or more sources of sound. This affirmation, expressed in terminology of acoustics and electrical engineering, implies that the voice waves have unique specifications in source terms and 2 filtrate characteristics [2].

The articulator synthesis determines the characteristics of filter of vocal tract by means of the description of the geometry of vocal tract (like the size of the oral cavity, the trachea and the position of the language, among other variables) and places the sonorous sources within this geometry. These factors are related to each other to produce a voice that is resembled in the greater measurement of the possible thing the human voice. The articulator synthesis applies harmonic signals to the sonorous signal and establishes an analogy between parameters related to the articulator organs, its movements and characteristics.

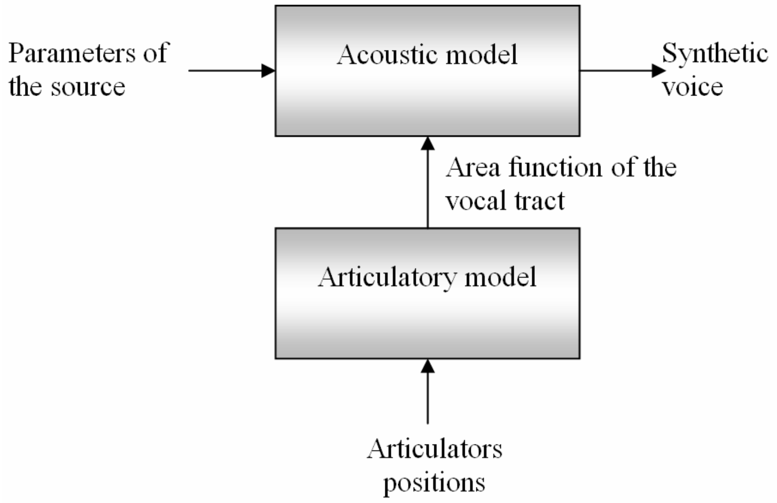


Fig. 2 Basic structure in the articulator synthesis

Depending on the synthesizer, the geometry of vocal tract can be described in one, two or three dimensions. The one-dimensional model represents vocal tract by means of its function of area directly. The area function describes as the area of the representative one of the tubes of the vocal zone varies between the glottis and the opening of the mouth. Assuming a one-dimensional propagation in tract, the function of the area contains all the information to specify the characteristics of the filter. Therefore, with respect to the acoustic simulation, the models bi-dimensional and three-dimensional of the vocal zone also are transformed finally into a one-dimensional function of the area. The advantage of the multidimensional models is that the form and position of the articulator's can be specified of a very direct way.

The artificial articulator's of these models generally are controlled by means of a small set of articulator parameters. The variation of these parameters in the time allows that the function of area of vocal tract changes during a pronunciation. An acoustic model is used to calculate the wave of voice from the sequence of the area functions and its corresponding sources of sound.

In summary, an articulator synthesizer needs the following thing at least:

- A geometric description of vocal tract implemented in a set of articulator's parameters.

- A mechanism of parameters of control during each word.

- A model for the acoustic simulation, including the generation of the sound sources.

The entrance for the acoustic simulation is generally a constant segment of the function of area, corresponding to a zone of vocal tract integrated by several cylindrical sections of a tube according to the illustrated thing in figure 3. The figure shows how the vocal zone is excited by means of a glottal function of the speed of the volume (acoustic source) and radiates a sound wave of the pressure in the nasal orifices and the opening of the mouth.

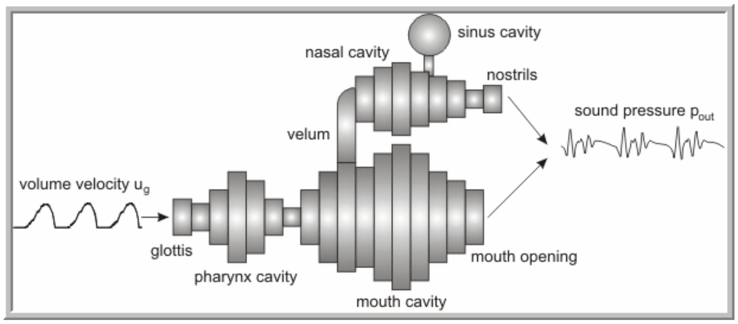


Fig. 3 Three-dimensional model of the vocal tract [4]

The main problem of the articulator models is the enormous amount of internal parameters of control that need and make difficult to the coordination and derivation of the parameters of control available to the entrance of the synthesizer; and on the other hand, the great amount of information that is needed to obtain analyzing (in a three-dimensional space) the position and the movement of the articulator organs of a person that speaks normally, thing very difficult to measure in these conditions.

The idea that bases to this synthesizer makes us think that it is adapted to totally reach the objectives of the back-end; nevertheless, not yet it has been managed to model correctly to each one of the formants, thus we thought that although at the moment it is not the system that gives better results, the constant improvement of his characteristics will take it to be the dominant system.

3 Concatenation Model

The principle of the concatenation is to produce voice by means of the connection of pre-recorded sounds, with which a sound of intelligible and natural voice is obtained. Nevertheless, the synthesizers by concatenation are limited by only talking one, which must record all the units of speech that the system is going to use. The size of these units is based on the naturalness that is tried to reach, this is, to obtain a greater naturalness is preferable to use great units. The type of unit to concatenate is a critical parameter to obtain a good quality of the synthesized voice: it is necessary to arrive at a balance between the inter-segmental quality possible (to greater length of the segments, less points of concatenation and therefore greater quality) and the amount from memory necessary to store the prerecorded units. The recorded pieces do not have to be words by two fundamental reasons. In the first place, the pronunciation of a phrase is very different from the one from a sequence of recited words separately, since in a phrase the words last one more shorter than when they are isolated and the rate, intonation and accentuation, that depend on semantic and syntactic factors, they are totally unnatural when recorded words are concatenated separately. A second problem is the innumerable existing words in a language, if we consider for example the own names, as well as the formation of words by means of suffixes, area codes and conjugations. The syllable is an interesting unit very linguistically, but there are a great number of them. Another proven unit is the phoneme, whose number is smaller of 30, but the turn out to concatenate phonemes is not satisfactory due to co-articulators effects between adjacent phonemes that produce changes of the acoustic manifestations of a phoneme depending on the context. The co-articulators effects tend to diminish themselves in center acoustic of a phoneme, which took to propose difonema, the voice piece that in the middle of goes from half of a phoneme the following phoneme, like the most satisfactory unit for the concatenation. In Spanish 900 can be considered about. In addition it can be necessary to introduce allophones different to make the distinction between the marked and atonics vowels or the inclusion of triphone, that are an extension to groupings of three phonemes when the co joint effects are so great that the segmentation in difonemas is not possible.

Although this system is the one that more computational resources it requires has become more and more popular, partly because its implementation is not so complicated (but very laborious), and to that the systems of computer every time are quicker and accessible, idem the memories. Nevertheless the parameters of the voice as the fundamental tone cannot be modified; for that reason, if we took like reference to [1] to define voice synthesis, we noticed that this system does not make true synthesis of voice.

Let us think that this system is a good option for a system that require good intelligibility, naturalness, a limited vocabulary and that is sufficient one or two speakers.

4 Formants Model

These synthesizers are based on the acoustic theory of voice production, which in its simpler form, says that it is possible to see the voice as the result of the excitation of a linear filter with one or more sonorous sources.

A simplified approach in the mechanism of the speech production in the acoustic dominion was proposed at the end of the 50 decade and was called “to source-filter model” [3]. In this model, the production system of voice is divided in two:

- Source of excitation.

- Resonance tract.

These two parts assume one “interaction” and one connection not to linear. The formants are the resonances of vocal tract. A synthesizer by formants reproduces the structure of formants of vocal tract.

In the 60 decade, appeared the first done discreet synthesizers with formants. The resonant ones were implemented in configurations of series (in cascade) or parallel. Flanagan (1957) concluded that the form series is better to reproduce sonorous sounds, and the no nasal ones; whereas the structure in parallel is better for the nasal sounds, and the no sonorous.

In 1980 appears the combined parallel/series system of Klatt. This configuration, made to improve the capacity to reproduce nasal and no sonorous sounds of the system. This type of synthesizers has an ample diffusion but the quality of the synthesized voice is smaller [Montero, 2002]. Although with the model parallel/series, using an appropriate specification of the synthesis variables and a correct configuration, it is possible to obtain a synthesized voice of high quality.

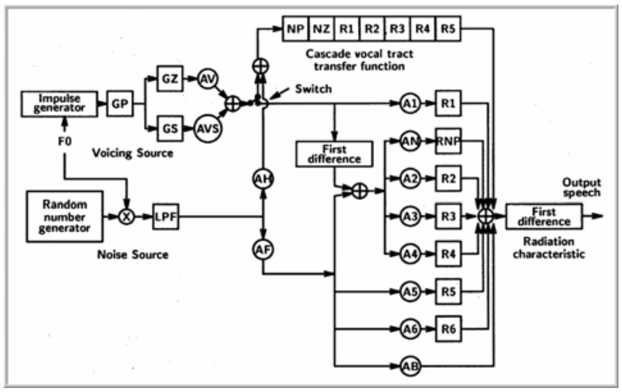


Fig. 4 Parallel/series model formants synthesizer [5]

The factor most important to obtain a synthesis of voice of high quality is the extraction of the parameters of synthesis, applying adapted procedures of analysis to a voice signal. Most of these procedures they use an acoustic signal of voice like source to determine the formants [Alku, 1992; Childers and Lee, 1991; Klatt and Klatt, 1990; Markel and Gray, 1974; McCandless, 1974; Trim off lower branches of, 1971]. Another factor important to obtain a synthesis of high quality, is the design of the excitation source [Childers, 1995; Childers and Ahn, 1995; Childers and Hu, 1994; Childers and Lee, 1991; Childers and Wu, 1990].

Synthesis parameters extraction

The parameters of our interest are bandwidth and the frequency of the formants, which are the picks in the surrounding one of the spectral of the voice signal which they represent the frequencies of resonance of vocal tract [6]. The formants frequencies can vary from a person to another one because all we have a constitution of vocal tract only; but in general form they are within a well-known rank.

Several methods for the calculation of formants of a voice signal exist; the methods that were used to extracts the formants from the spectrum of the lineal predictions coefficients (LPCs) of a voice sample. The steps to obtain the LPCs are:

1. Segment the useful signal, without silences at the beginning and end, in units of 25 or 20 ms (I segments).

2. To apply a window of Hamming to each segment of signal.

3. To calculate the amount of **p** values of the coefficients of autocorrelation for each segment.

4. To calculate the $\mathbf{a_p}$ coefficients LPC for each segment.

5. To obtain the LPCs average.

Once obtained the LPCs, graphical its spectrum in frequency, and took the frequency from the picks of the surrounding one like the formants, also of this graph we can calculate the bandwidth of each formant. In figure 5 it is showed to the frequency response of the LPCs of a recording without controlled conditions of the vowel ´a´, monaural and with a frequency of sampling of 16000 Hertz. In figure 6 we showed an approach of the first pick of the spectral shown in figure 5, and of which we can accurately determine the frequency of the first formant.

In figures 5 and 6 the vertical axis corresponds to the amplitude, and horizontal axis corresponds to the frequency in Hertz.

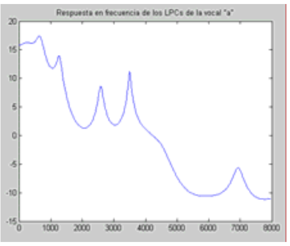


Fig. 5 Frequency spectrum response of LPC Vowel ‘a’

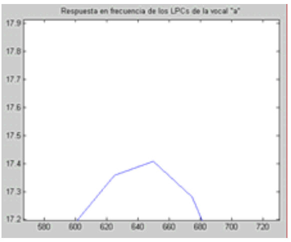


Fig. 6 First format pick of the frequency spectrum frequency spectrum of vowel ‘a’

This way it is been able to determine bandwidth formants and for each one of the units of voice of sonorous type that our system requires. The following table presents the data collected for the five vowels of the Spanish; in her we both showed to first respective formants and their bandwidths:

Table 1. Vowels formats

	F1 / W1 [Hz]	**F2 / W2 [Hz]**	**F3 / W3 [Hz]**	**F4 / W4 [Hz]**	**F5 / W5[Hz]**
A	650 / 100	1200 / 150	2600 / 250	3500 / 300	4500 / 200
E	400 / 200	2000 / 150	2700 / 350	3650 / 150	3850 / 200
I	250 / 50	2200 / 150	3000 / 300	3500 / 200	4000 / 200
O	400 / 100	800 / 200	2000 / 200	3200 / 200	4000 / 200
U	400 / 100	800 / 200	2500 / 200	3500 / 200	4000 / 200

In the tests it is observed that the quality of the synthesis improves when increasing the amount of formants; but this is only truth with the first formants, after the first three formants the improvement is every smaller time; for that reason one says that five formants are an suitable amount to approach us the Maxima quality that this type of synthesizer can offer.

Excitation source

The other point of relevance is the source of excitation for the system of synthesis by formants, was proven with different models: Delta, Rosenberg and Liljencrants-Fant (LF). Being this last one from that better perceivable result we obtained. Model LF tries to equal the waveform produced in vocal tract (figure 7), and changing some of his parameters it is possible to obtain different pitch characteristics in the synthesized signal.

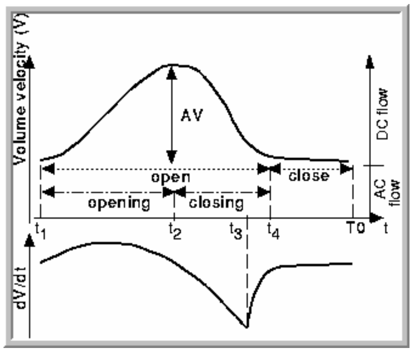


Fig. 7 Glottal Speech volumetric wave and its derivate [7]

In figure 7 we can observe the following parameters:

- t1 initiates the opening of the vocal cords and begins to flow the air.
- t2 moment of maxima glottal opening (AV) and maximum air flow.
- t3 beginning of the closing of the glottal cavity and maximum change of the glottal flow (harsh- introduces components of greater frequency).
- t4 the glottal cavity is completely closed and ideally there is air flow no.
- To duration of a complete period.

The equations that model LF proposes to model the derived one from the waveform is:

$$g(t) = E_0 e^{\alpha t} \sin(\omega_g t) \qquad t_1 \le t \le t_3$$

$$g(t) = \frac{Ee}{\varepsilon t_\alpha}[e^{-\varepsilon(t-t_3)} - e^{(T_0-t_3)}] \quad t_3 \le t \le t_4 \le T_0$$

Where:

- E_e is the amplitude of the excitation.
- tα constant of the exponential curve that determines form of the curve between T3 and t4.
- ω_g establishes the duration of the phase of opening.
- E_0 is an amplitude factor.
- ε is coefficient that it increases of exponential way to the sinusoid.

In order to obtain the waveform that will be used in the synthesizer only it is necessary to integrate the equation of the LF model.

In figure 8 we can observe the vowel ‘a’ real and in figure 9 we observed the vowel ‘a’ synthesized. The graphs show amplitude against sample. The source of used excitation was provided by model LF.

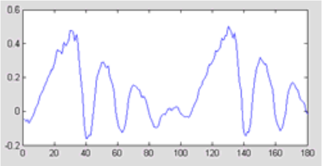


Fig. 8 Vocal ‘a’ real

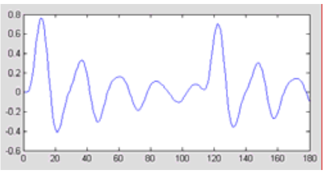


Fig. 9 Vocal ‘a’ synthetic

5 Conclusions and future works

In intelligibility approach the voice synthesis already reach an acceptable level; nevertheless, the naturalness implies so many dynamic changes that at the moment it is continued investigating the most suitable method to reach an acceptable level; in that sense, the articulator synthesis seems to be the key, but lack work to do much. The synthesis would concatenate, although it is not synthesis in a strict sense, at the moment widely is used, due to the superiority in naturalness that obtains, but its lack of dynamism and discharge demand of resources prevents its universality. On the other hand, the model by formants, although produces an artificial voice, doing a configuration adapted of its modules, a correct calculation of the formants and providing a source to him of excitation next to which generates vocal tract, we can obtain satisfactory results, with the advantage to be able to change parameters in the speech synthesized.

This word is support by National Polytechnic Institute (IPN) of Mexico in de project 20070331.

References

1. Donald G. Childers. “Speech Processing and Synthesis Toolboxes”, Jonh Wiley & Sons, Inc. (2000)

2. Donald G. Childers. “Speech Processing and Synthesis Toolboxes”, Jonh Wiley & Sons, Inc. (2000)

3. Gunnar Fant. “Acoustic Theory of Speech Production”, Mouton, The Hague. (1960)

4. http://wwwicg.informatik.uni-rostock.de/~piet/speak_main.html

5. J. Acoust. “Software for a cascade/parallel formant synthesizer”, Soc. Amer., vol. 67, pp. 971–995, 1980.

6. Alfredo Mantilla. Tesis doctoral "Análisis, reconocimiento y síntesis de voz esofágica". Escuela Superior de Ingeniería Mecánica y Eléctrica, Culhuacan DF México. (2007)

7. http://www.ims.uni-stuttgart.de/phonetik/EGG/page13.htm

IIR Adaptive Wavelet Network with Reinforcement Learning

Iván S. Razo-Zapata, Luis E. Ramos-Velasco, and Julio Waissman-Vilanova

Research Center in Information Technologies and Systems,
Autonomous University of the State of Hidalgo,
Carr. Pachuca-Tulancingo, Km. 4.5, C.P. 42084, Pachuca,
Hidalgo, Mexico. {ri096373,julio,lramos}@uaeh.reduaeh.mx
http://www.uaeh.edu.mx/investigacion/sistemas/

Abstract. This paper presents a novel approach of reinforcement learning for continuous systems. The scheme is based in wavelet networks to approximating the continuous space of states. The structure of the wavelet network is dynamically generated accord to the explored regions and trained with a modified Q-Learning algorithm. The wavelet network include a IIR filter in order to make smooth controllers. This novel approach is called adaptive wavelet reinforcement learning control (AWRLC). Simulations of applying the proposed method to underactuated systems are performed to demonstrate the properties of the adaptive wavelet network controller.

1 Introduction

Reinforcement learning (RL) is learning to perform sequential decision tasks without explicit instructions, only optimizing a criterion about how the task is perform. So, the learner doesn't know which actions to take, but instead must discover which actions yield the most reward by trying them. This method, is goal-directed, and seems better adapted to the solution of a kind of control problems [1, 2], which ones about searching a final goal, and the problem is to find a policy that reach this goal [3].

The basic RL algorithms use a look-up table scheme in order to represent the value function $Q(s, a)$. Unfortunately this representation is limited when working with continuous spaces like physical systems. Several approaches can be applied to deal with this problems, like function approximation techniques. Neural networks offers an interesting perspective due to their ability to approximate nonlinear functions [4].

In recent years, wavelets have attracted much attention in many scientific and engineering research areas. Wavelets possess two features that make them especially valuable for data analysis: they reveal local properties of the data and they allow multiscale analysis. The local property is useful for applications that requires online response to changes, such a controlling process. Wavelets and neural networks have been combined [5, 6], to form a class of networks, so called

© A. Argüelles, J. L. Oropeza, O. Camacho, O. Espinosa (Eds.)
Computer Engineering.
Research in Computing Science 30, 2007, pp. 129- 138

wavelet networks, which are capable of handling moderately high-dimensional problems [4].

Inspired by the theory of multi-resolution analysis of wavelet transform and suitable adaptive control laws, an adaptive wavelet network is proposed for approximating action-value functions [7]. In this paper, we propose an adaptive wavelet reinforcement learning control (AWRLC) whose design is based on the promising function approximation capability of wavelet networks. The goal of the paper is to propose a control scheme based on RL algorithms and an AWRLC to control underactuated systems. We proposed a IIR filter in order to avoid bang-bang controllers. In this work the the *Pendubot* was used like example to evaluate the advantages and disadvantages of AWRLC methods for control of underactuated systems.

The work is organized as follows. Section 2 presents the reinforcement learning approach. In Section 3 is summariezed the background about wavelets networks, while Section 4 shows the control scheme which is implemented in the system. Section 5 gives the results obtained by numerical simulation. Finally, in Section 6 conclusions from results and future work are presented.

2 Reinforcement Learning

Q-Learning is a reinforcement learning method where the learner builds incrementally a Q-function which attempts to estimate the discounted future rewards for taking actions given states. the system is assumed as a Markov Decision Process (MDP) [3]. So, in a common control task maximize the total return R_t expressed in (1) is the main objective.

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \tag{1}$$

Where R_t is the total return at state s_t and r_t is the reward value (numerical) when the system reach the state s_t.In this way, the output of the Q-function for state s_t and action a_t is denoted by $Q(s_t, a_t)$. When action a_t has been chosen and applied, the system is moved to a new state, s_{t+1}, and a reinforcement signal, r_{t+1}, is received, $Q(s_t, a_t)$ is updated by [3]:

$$Q(s_{t+1}, a_{t+1}) \longleftarrow Q(s_t, a_t) + \alpha\delta \tag{2}$$

where

$$\delta = r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)$$

$0 \leq \alpha \leq 1$ is the *learning rate*, and $0 \leq \gamma \leq 1$ is called the *discount*, this parameter is used to decrease r_{t+1} in the total return (1).

3 Wavelet Networks

Wavelets are a class of functions which have some interesting and special properties. These properties are localization in scale and time, compact support, multiresolution analysis among others. The original objective of the theory of wavelets is to construct orthogonal bases of $L_2(\mathbb{R})$. These bases are constituted by translations and dilations of the same function ψ called "mother wavelet" [8].

The structure of a wavelet network is a type of building block similar to a RBF network [6]. This building block allows the approximation of unknown functions by the concept of the multi-resolution approximation. The building block is formed by shifting and dilating the basis function ψ, (the modified version is its "daughter wavelet") and a "father wavelet" ϕ. Most commonly, wavelet bases are derived using shift-invariance and dyadic dilation. In this way we use the dyadic series expansion

$$\psi_{jk}(x) = 2^{j/2}\psi(2^j x - k), \quad j,k \in \mathbb{Z} \tag{3}$$

which is integral power of 2 for frequency partitioning. The daughter wavelet (3) is obtained from a mother wavelet function ψ by a binary dilation (i.e. dilation by 2^j) and a dyadic translation (of $k/2^j$).

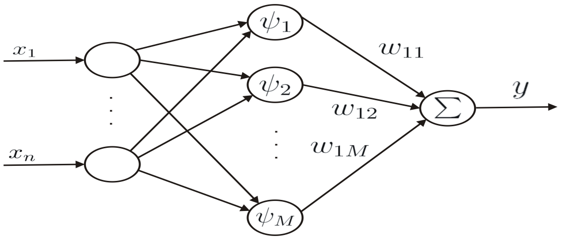


Fig. 1: Structure of wavelet network.

In this way combination of wavelet and neural networks can handle problems of large dimensions well and can make constructing network easily. The basic structure of a wavelet network is illustrated in Fig 1. The operation of each layer is summarized as follows [9]

- Using I_i^j and O_i^j to denote the input and output of the ith node in the jth layer, in first layer inputs are introduce into the network

$$O_i^1 = I_i^1 = x_i, \quad i = 1, 2, \ldots, n$$

- Second layer consists of wavelet which one corresponding to pairs of (j,k) in (3), and the inputs and outputs of the wavelet nodes in this layer can be described as

$$\begin{cases} I_i^2 = \left[O_1^1, \ldots, O_n^1\right]^T \\ O_i^2 = 2^{j/2}\psi(2^j I_i^2 - k) \end{cases} \quad i = 1, 2, \ldots, m$$

– Finally the input-output reltion in third layer is expressed with

$$y = O^3 = \sum_{i=1}^{m} w_{ij} O_i^2$$

4 Control Scheme

The control scheme is based in a wavelet network ($\mathbb{W}$) trained with reinforcement learning. Accord with Fig. 1 the structure of the wavelet network is composed by three layers. First layer is the input layer, third layer is the output layer, and the second layer is the hidden layer with wavelet functions as activation functions. In order to deal with continuous actions our scheme presents an extra layer. This layer is an Infinite Impulse Response (IIR) synopsis network, which computes a continuos action accord to the outputs in the third layer (see Fig. 2a).

The main motivation for to implementing a wavelet network in this scheme is because artificial neural networks allow to approximate unknown functions, and with RL algorithms the idea is to approximate a Q-function. The wavelet network structure could be represent like a MISO system, with n input variables and only one output variable, like in Fig. 2a.

4.1 Building the Wavelet Network

The process of building the architecture of the wavelet network is performed on-line with the exploration through new states, due in part to the absence of data generated in past operations (training data). This growing of wavelet basis provide support to new states in order to approximate the Q-function with better accuracy.

New wavelet basis are generated with series expansion techniques, in this case was applied (3) with n-dimensions, with a tensor product [10–12].

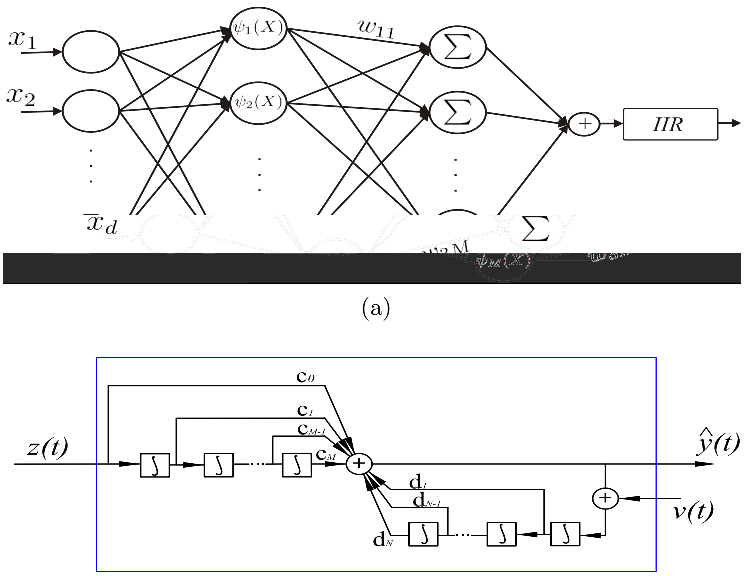
$$\psi(X) = \psi(x_1, x_2, \ldots, x_n) = \prod_{j=1}^{n} \psi(x_j) \tag{4}$$

The mother wavelet implemented in this scheme is Mexican Hat defined as follows

$$\psi(x) = \frac{2}{\sqrt{3\sqrt{\pi}}}(1 - x^2)e^{\frac{-x^2}{2}} \tag{5}$$

The translation parameter in each one of the dimensions is determined by $k = x_i * 2^j$ where j is the scale of the wavelet basis and x_i is an element without support in the structure of $\mathbb{W}$, when the learning process begins, the first neuron with support in $\mathbb{W}$ is created over the initial coordinates $x_1 = 0, x_2 = 0, ..., x_n = 0$.

In this way is possible initialize the learning with $\mathbb{W}$ empty, and $\mathbb{W}$ grows accord to the explored regions. The growing is controlled by the election of a threshold ξ which determines the minimum value in order to consider if a given state has support in the structure of the network. The diagram shown in Fig. 3 presents the algorithm for the construction of the wavelet network $\mathbb{W}$.



(a)

(b)

Fig. 2: *IIR Adaptive Wavelet Networks Structure: (a) Like a MISO system with n inputs and one output, in the second layer activation functions are wavelets; (b) IIR Model.*

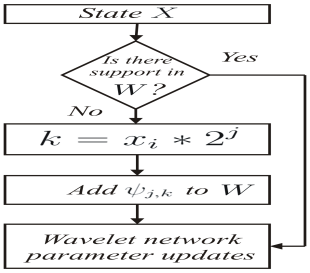


Fig. 3: Building the wavelet network.

4.2 Training the Wavelet Network

The training process in $\mathbb{W}$ consists in update the weights of layer 3. This training is performed on-line according to the interaction in the environment.

The updating rule is a combination between reinforcement learning and the gradient descent method. The reinforcement learning method implemented is *Q-Learning* due to its capacity of learning over the best action independently of the action selected. The updating rule applied in $\mathbb{W}$ is given by:

$$\begin{aligned} w_{i,j}(t+1) &= w_{i,j}(t) + \triangle w, \quad j = 1, \ldots, m \\ \triangle w &= \alpha \left[r_{t+1} + \gamma \max_{y_M} y_j(t+1) - y_j(t) \right] |\psi(X)| \end{aligned} \tag{6}$$

where α is the learning rate, r_{t+1} is the reward at the time $t+1$, and γ is the discount with the same function that in (2).

The structure shown in Fig. 2a could be see like a MIMO system until layer 3 [10]. At the time t there is a vector $Y = [y_1, y_2, \ldots, y_m]^T$ with m outputs. So $y_i(t)$ is the output value of the neuron i in layer 3, where $i \in 1, 2, \ldots, m$. The value of i represents the selected action in the exploration process, treated in the next section.

In the training applied to neuronal networks, target values are required. And, in this scheme the target values is the maximum output given by $\mathbb{W}$ at the time $t{+}1$ in neurons of layer 3. In this way, the approximation of $Q(s, a)$ is achieved on-line without training data. In each iteration several neurons could be activated, but only the weights of neurons with value of activation bigger than the threshold $0 < \xi < 1$ are modified.

4.3 Action Selection

During the learning process, the network $\mathbb{W}$ allows to make elections between the set of actions $A(s)$. This selection can be applied with some kind of exploration like *Softmax* or $\epsilon-$ *greedy* [3]. The selection of actions is performed with the output values in layer 3 of $\mathbb{W}$, $(y_1, y_2, \ldots, y_m)$. Each output represents the approximated value for the function $Q(s, a)$. And in a *greedy* exploration, the neuron with maximum value always is taken as control action.

4.4 Operation Process

After the learning process ($i.e$ $\mathbb{W}$ is built and trained), the last layer in $\mathbb{W}$ computes continuos actions, which ones will be applied to the continuous system. This layer computes actions accord to the discrete actions learned for neurons in layer 3. In the last layer only one neuron is presented (Fig. 2a), and consist in a IIR filter.

5 Application

The control scheme AWRLC presented in Section 4 was applied to one underactuated system (*Pendubot*) in order to control it in one equilibrium point. For the Pendubot the equilibrium position is the called UP-UP configuration (the first and the second link in vertical position like in Fig 4). The set of parameters applied in learning process with $\mathbb{W}$ is summarized in Table 1.

Table 1: Operation parameters

Parameter	Description	Value
n	Number of inputs.	4, $X = [x_1, x_2, x_3, x_4]^T$
m	Number of neurons (outputs) in second layer.	3, $Y = [y_1, y_2, y_3]^T$
$\theta_1, \dot{\theta_1}, \theta_2, \dot{\theta_2}$	Initial conditions	$X = [\pi, 0, \pi, 0]^T$
$A(s)$	Set of actions.	$-1, 0, +1$
α	Learning rate.	0.2
γ	Discount.	0.9
ϵ	Exploration $\epsilon-greedy$.	0.1
$\psi(x)$	Mother Wavelet.	Mexican Hat
j	Scale of wavelets.	3
ξ	Threshold.	0.2

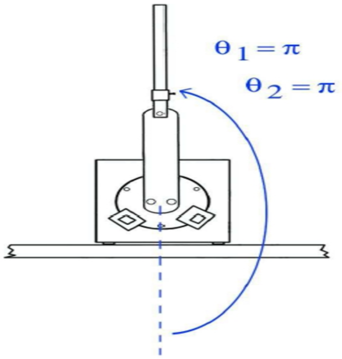


Fig. 4: Pendubot configuration UP-UP.

The control task in the system, is realized by $\mathbb{W}$ with a learning process based in RL. And the rewards for learning were assigned in continuous form, according to $\theta_1, \dot{\theta_1}, \theta_2$ and $\dot{\theta_2}$. The desired value for θ_1 and θ_2 is around π (UP-UP Configuration). So rewards were given accord to the angular position of both links, this assignation was realized as follows:

- -3 When $\theta_1 \simeq \pi$, $\theta_2 \neq \pi$, $\dot{\theta_1} > 0$ and $\dot{\theta_2} > 0$
- -2 When $\theta_1 \simeq \pi$, $\theta_2 \simeq \pi$, $\dot{\theta_1} > 0$ and $\dot{\theta_2} > 0$
- -1 When $\theta_1 \simeq \pi$, $\theta_2 \simeq \pi$, $\dot{\theta_1} \simeq 0$ and $\dot{\theta_2} > 0$
- $+1$ When $\theta_1 \simeq \pi$, $\theta_2 \simeq \pi$, $\dot{\theta_1} \simeq 0$ and $\dot{\theta_2} \simeq 0$

The simulation was applied with a time step of 0.01s during 2000 episodes, with $j = 3$. A pruning process was applied in order to reduce the number of neurons. This pruning process is due to the growing of number of neurons in second layer related with the exploration property of RL algorithms, and some neurons doesn't have influence in the control task. So, 10 operation process were performed and the most important neurons were identified. Originally the network $\mathbb{W}$ has 2463 neurons, but with the pruning process the number was reduced to 131.

The reduced structure of AWRLC was applied to the system allowing to control the system in the desirable position, although to disturbances introduced to the torque of the first link.

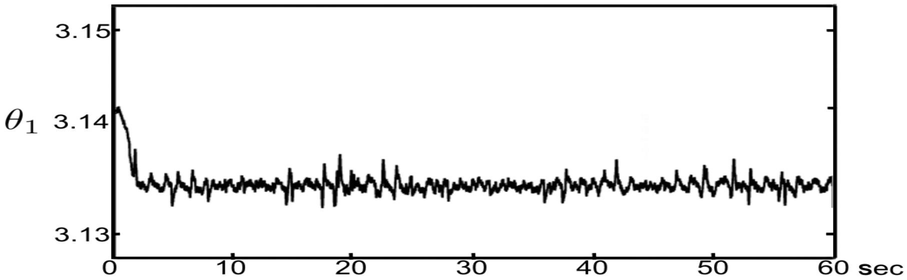


Fig. 5: θ_1, mass and large of both links like in training.

Fig. 6.a shows the behavior of the first link in 60 seconds of operation. Random disturbances were applied every 0.5 seconds of $+2$ and -2. In this plot the value of θ_1 is around to the desirable value in spite of the disturbances. In the same way Fig. 6.b presents the behavior of the second link during 60 seconds. It is possible to observe that the position is stable around an ideal value.

The main difference between the scheme presented in [10] and the actual AWRLC is the application of continuous actions. In [10] actions applied to the underactuated system are purely discrete actions. Fig. 7 presents the continuous actions applied to the system and disturbances introduced in 3 seconds of operation.

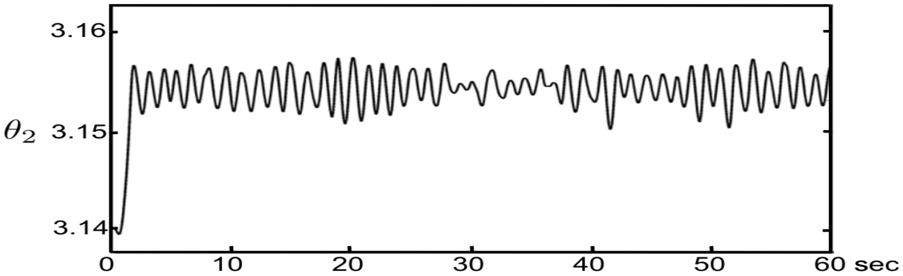


Fig. 6: θ_2, mass and large of both links like in training.

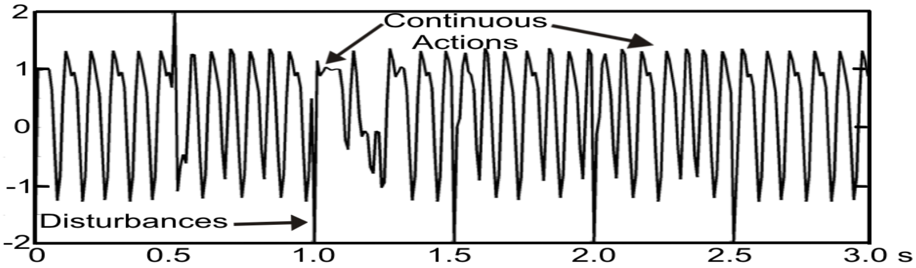


Fig. 7: Continuous actions and disturbances.

6 Conclusions and Future Work

6.1 Conclusions

In this work is presented the AWRLC scheme based in Reinforcement Learning algorithms, Wavelet Networks structure and IIR filter. AWRLC allow to deal with continuous spaces of states and actions. This represents an improvement with respect to the work presented in [10]. A simulation results of a *Pendubot* system was presented as illustrative example. The control of this system is specially difficult since is an underactuated mechanisms (two degrees of freedom and only one input).

Control scheme is based on learning methods, modifying one of the most popular RL algorithms: Q-Learning with adaptive wavelets networks. This approach uses a wavelet networks to approximate a Q-function, where the function gives the optimal control policy. Finally a IIR filter in order to avoid bang-bang controllers, which is applied to the underactuated system.

The simulation results show that this controller provides a good performance when keeping the systems in the unstable vertical positions. Results indicate that the AWRLC is a potentially attractive alternative for underactuated systems.

The algorithm for build wavelets networks in Fig. 3 represents an advantage working with physical systems with unknown limits of operation, because this method of generating neurons create support in regions accord to the explorations. Multi-resolution is other attractive property handled by this kind of wavelet networks. The scales of resolution allow to approximate with good accuracy unknown functions, and in this case coarse approximations doesn't produces optimal control policies.

6.2 Future Work

The optimality over the policy built by AWRLC is one of the topics in further discussion. Besides, convergence and stability studies are in development actually.

References

1. I. S. Razo-Zapata, L.E. Ramos-Velasco and J. Waissman-Vilanova, Reinforcement Learning of Underactuated Systems, *in International Symposium on Robotics and Automation (ISRA 2006)*, San Miguel Regla, Hidalgo, México, 2006, pp. 420-424.
2. K. S. Fu, Learning Control Systems-Review and Outlook, *IEEE Transactions on Automatic Control*, vol. AC-15, 1970, pp. 210–221.
3. R. S. Sutton, A. G. Barto, *Reinforcement Learning An Introduction*, The MIT Press, 1998.
4. M. T. Hagan, H. B. Demuth and M. Beale, *Neural Network Design*, PWS Publishing Company, 1996.
5. Q. Zhang, and A. Beneveniste, Wavelet Networks, *IEEE Trans. Neural Networks*, vol. 3, 1992, pp. 889–898.
6. Q. Zhang, Using Wavelet Networks In Nonparametric Estimation, *IEEE Trans. Neural Networks*, vol. 8, 1997, pp. 227–236.
7. J. Xu and Y. Tan, *Nonlinear Adaptive Wavelet Control Using Constructive Wavelet Networks*, Proceedings of the American Control Conference, Arlington, VA, 2001.
8. S. Mallat, A Theory For Multiresolution Signal Decomposition: The Wavelet Representation, *IEEE Transactions Pattern Recognition and Machine Intelligence*, vol. 11, 1989, pp. 674–693.
9. W. Sun and Y. Wang and J. Mao, Using Wavelet Network for Identifying the Model of Robot Manipulator, *World Congress on Intelligent Control and Automation*, 2002, pp. 1634–1638.
10. I. S. Razo-Zapata, J. Waissman-Vilanova and L.E. Ramos-Velasco, Reinforcement Learning in Continuous Systems: Wavelet Networks Approach, Analysis and Desing of Intelligent Systems using Soft Computing Techniques, Series: Advances in Soft Computing , Vol. 41, Melin, P.; Castillo, O.; Ramrez, E.G.; Kacprzyk, J.; Pedrycz, W. (Eds.), 2007, XXI, 855 p. ISBN: 978-3-540-72431-5, Imprint: Springer-Verlang Brlin Heidelberg, 2007.
11. R. Wai and J. Chang, Intelligent Control of Induction Servo Motor Drive Via Wavelet Neural Network, *Electric Power Systems Research*, 2002, pp. 67–76.
12. J. Zhao, B. Chen and J. Shen, Multidimensional Non-Orthogonal Wavelet-Sigmoid Basis Function Neural Network for Dynamic Process Fault Diagnosis, *Computers and Chemical Engineering*, 1998, pp. 83-92.

Start Times Matricial Estimation with Stochastic Behavior for Periodic Real Time Tasks into Concurrent Systems

J. J. Medel [3,2], P. Guevara López [1,2] , F. Ríos Suriano [2,3]

[1] DGCH, CICATA- I. P. N. [2] CICATA-I. P. N., [3] UPIITA, – I. P. N.,

Tel.57296000 Ext. 50427, Ext. 64344

(pguevara, jjmedel) @ipn.mx, frios99@yahoo.com

Abstract. A multivariable stochastic dynamic model describing start times respect to real-time tasks with bounded properties: Stationary conditions, first order, with jitter and external perturbations bounded with a normal distribution without correlation that closely represent periodical behavior of real time tasks. To bear closer the task model in a concurrent system, internal dynamics description, and the parameter matrix in regressive model with oscillations used as multivariable estimator. Results of an example performed on a real-time platform presented, considering periodic and concurrent tasks, an instrumental variable algorithm is a basic tool in this kind of systems because it is a good estimator with convergence probability sense and its relatively easy implementation.

Keywords: Start time, estimation, real-time, periodic tasks, instrumental variable.

© A. Argüelles, J. L. Oropeza, O. Camacho, O. Espinosa (Eds.)
Computer Engineering.
Research in Computing Science 30, 2007, pp. 139- 145

1 Introduction

Periodic tasks are usually found in several applications like airplanes and control process where uniform monitoring is required. Modeling them is not simple because each of those needs an adequate representation. A set of periodic concurrent tasks represented as multivariable state model where in explicit way the model is function of the internal dynamic and previous states of the system's output information.

These matrix and perturbations are going to give the start times characteristics through system evolution. A parameter matrix estimator needed in order to adjust the model for reconstruction, tracing and prediction in real time.

2 Start Times Model in PRTT (Periodic Real Time Tasks)

As seen in [1], [4], [5], [6], [10], [14] and [15]**,** PRTT model is represented by a stochastic, stationary, first order and first grade type difference equation; considering that external processor perturbations without correlation and obeying a normal distribution function.

Proposition 1. (Absolute Arrival Time for RTT). Into instance set, is the absolute arrival time vector L_k with index *k* described as:

$$L_k = L_{k-1} + \Pi_k . \tag{1}$$

Proposition 2. (Inter-arrival Time for RTT). Into instance set, is the Inter-arrival time vector Π_k with index *k* described as:

$$\Pi_k = A_k(\Pi_{k-1} - W_{k-1}) + U_k + W_k \tag{2}$$

Where: *A_k is the system parameter matrix with unknown dynamics but its bounded in agreement with [7], [11], [12] y [14]; Π_k is the Inter-arrival times vector of instances with index k; W_k is the external processor perturbations vector, represented through random variables with Gaussian distribution; U_k is the Inter-arrival time vector reference.*

Proposition 3. (Start Times for RTT). The starting times vector S_k of a feasible task with index *k* has a recursive description as:

$$S_k = S_{k-1} + \Pi_k + V_k - V_{k-1} . \tag{3}$$

Comment 1. If an inter-start times vector modeled as Π'_k require adding the *jitter* vector knowing it as V_k, justifying it as internal perturbation to the state equation such that:

$$\Pi'_k = X'_k + W_k \,, \tag{4}$$

$$X'_k = A_k X'_{k-1} + U_k + V_k \,, \tag{5}$$

Considering to (5) into (4), the inter-start times vector model is:

$$\Pi'_k = A_k(\Pi'_{k-1} - W_{k-1}) + U_k + W_k + V_k \,. \tag{6}$$

Proposition 4. (Periodic Tasks in Real Time). A PRTT set of J_i all its instances have inter-arrival times vectors Π_k round off to a periodic vector $T_{k.}$.

Para A_k $\{a_{i,j,k}\} \subset [0,1]$ $a_{i,j,k}$ constante $\forall$ $i,j,k \in Z^+$

Para U_k $\{u_{i,k} = T_i - a_{i,k} T_j\}$ $u_{i,k}$ constant $\forall$ $ilk \in Z^+$

3 Real Time Parameter estimator (RTPE)

In order to trace the parameter matrix evolution into concurrent PRTT, the Real Time Estimator (RTPE) in agreement to [3], [7],[10], [14], [15], and [16] has a basic description:

Definition 1. (Real Time Parameter Estimator RTPE). All RTPE is a digital filter with the following basic conditions:

a. To Extract and to emit observable information (input and outputs observable signals, where $\{u(k)_i \in \mathbf{U}(k)\}$ and $\{y(k)_j \in \mathbf{Y}(k)\}$, respectively , with $i, j, k \in \mathbf{Z}^+$), in the sense considered by [2], [3], [7], [8], [11], [12] and [13],

b. To give correct answers respect to the process considered in some established criteria illustrated and conceptually described in [2],[11], [12] and [13],

c. Expressing the model filter in recursive sense (see [3], [7], [10], [13], [17]),

d. Convergence value bounded, respect to perturbation variance.

e. Matrix operation bounding the process dynamics restrictions.

Proposition 5. (Convergence in all Multivariables RTPE). All RTPE as a parameter estimator has an error functional bounded in probability sense (to see: [2],[7], [9], [16] and [17]), such that :

$$m^* = \arg\min \inf_{k \geq m} P\{|\,\hat{a}_k - a\,| \leq \Delta\} = 1 \,. \tag{7}$$

Where Δ is the error limit, defined by noise variance; m is the convergence interval and m^* is the intervals set where the RTPE has a convergence rate in probability sense.

4 Estimation of Periodic Real Time Task Start Times with stochastic jitter using the Instrumental Variable technique.

As an example, we considered a model of concurrent PRTT set, described in (6): The relative arrival time model (*Πk)* used as the observable signal, with the stochastic properties:

$$E\{W_{k+1}\Pi_k^T\}=[0],\quad E\{W_k\Pi_k^T\}=\Theta_{w_k}^2,\quad E\{V_k\Pi_k^T\}=\Theta_{v_k}^2,$$
$$E\{W_k(W_k)^T\}=\Theta_{w_k}^2,\ E\{V_k(V_k)^T\}=\Theta_{v_k}^2,\ E\{V_k(W_k)^T\}=[0]. \tag{8}$$

The estimator in agreement to [9] and obeying the properties expressed in Definition 1., has the form:

$$\hat{A}_k:=(\hat{A}_{k-1}B_{k-1}+\Pi_k Z_k^T)B_k^{-1}. \tag{9}$$

The estimation error in agreement to [2], [3], [7] and [9] is:

$$\Delta_k=\left|\hat{A}_k-A\right|. \tag{10}$$

Recursively, and in agreement to [9] the estimation error is:

$$\Delta_k:=((-A\Theta_{w_k}^2+\Theta_{v_k}^2)(I-A^2))(\Theta_{w_k}^2(2A^2+I+A^2)+\Theta_{v_k}^2(I+2A))^{-1}, \tag{11}$$

As the functional error in accordance with [9] and [15], expressed by the second probability moment in vector sense:

$$J_k=E(\Delta_k(\Delta_k)^T). \tag{12}$$

And considering its recursive description:

$$J_k=\tfrac{1}{k}(\Delta_k\Delta_k^T+(k-1)J_{k-1}). \tag{13}$$

The following data considered as hypothetical results evolution using the model expressed in (6):

$$A=\begin{bmatrix}0.5 & 0.2\\ 0.3 & 0.4\end{bmatrix},\ \Theta_{v_k}^2=\begin{bmatrix}0.95 & 0.93\\ 0.97 & 0.94\end{bmatrix},\ \Theta_{w_k}^2=\begin{bmatrix}0.95 & 0.93\\ 0.97 & 0.94\end{bmatrix}. \tag{14}$$

For the experimental implementation of the RTPE the following considered:

a. The maximal system deadline $\mathbf{D}_{k,\,max}$, is equal to period $\mathbf{T}_k$.
b. The Start time ($\mathbf{S}_k$- $\mathbf{L}_k$) was obtained such that $\mathbf{S}_k = \mathbf{L}_k+0.0015$ ms.
c. The sampling period T_k (Real Time temporizator impulse) for task activation is 10 ms.
d. Minimal deadline $\mathbf{D}_{k\,min}$= 1 ms.
e. Convergence deadline of the PRTT set is: *d*= 4 s.

For RTPE experiment (Fig. 1) obtained us the following results:

m_{max}= 354 intervals, t_{c_max}= 3,54 s. The convergence time is t_c =3.54 s.

a. m= [354, 213, 241, 318] intervals,
b. t_c= [3.54, 2.13, 2.41, 3.18] s,
c. d= 4 s. (400 intervals).

Finally, we considered the estimation algorithm expressed in (9). The Fig.1 and Fig. 2 are just examples of a RTPE behavior used as a parameter estimator trough the instrumental variable technique.

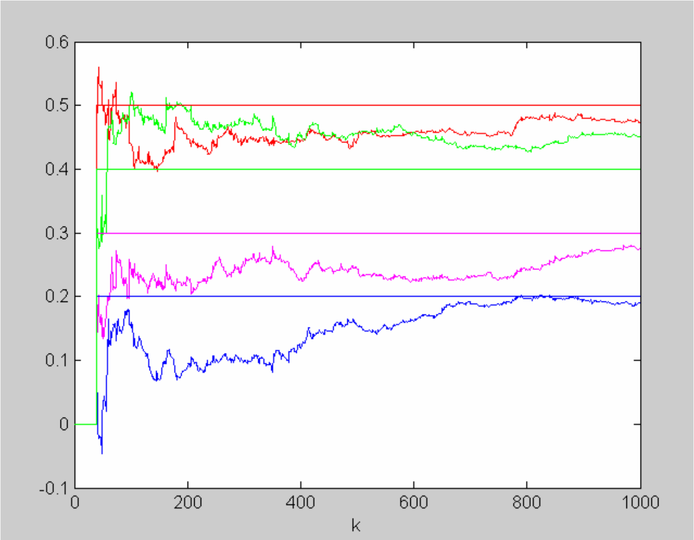


Fig.1 *Matrix parameter estimation "A" using the RTPE.*

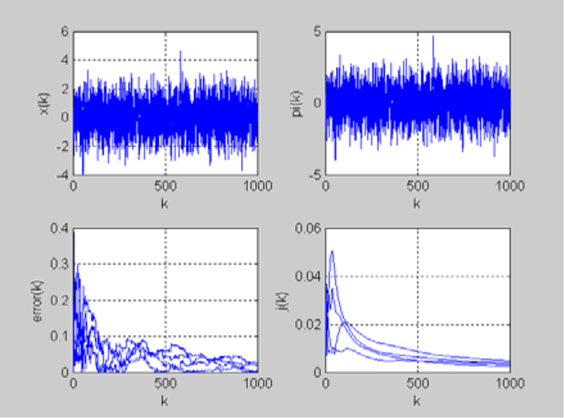


Fig. 2 *Internal state X_k, observable signal Π_k, estimation error and functional error J_k , for RTPE model.*

5 Conclusions

The internal dynamic system for Periodic Real Time Tasks into a concurrent system is into the real-time system, the start time model. Tasks set, depended on its arrival and start times in accordance with its matrix parameters, its entrance and internal computing equipment and external perturbations, generating the proposition set. The model showed capable of characterize several Real Time Task behaviors and obeyed characteristics mentioned by several authors round off the real-time systems. By the other hand, the real-time internal description considers that the task characteristics, synchronicity, sampling periods and convergence time, form part of its properties. For start time estimation, the convergence times where bounded by an Δ estimation error, described by second probability moment respect to internal and external perturbations and convergence periods acceptable in probability sense. The instrumental variable technique considered in its estimator had a good enough response in time sense obeying to Real Time System conditions, approaching to real parameters rapidly and accomplishing its deadlines.

Bibliography

[1] Baras J. (1999), "*Symbolic and numeric Real-time signal processing*", Technical Report University of Maryland, USA. pp 226

[2] Caines P. (1986). "*Linear Stochastic Systems*". Ed. Wiley, Canada.

[3] Chui C., Chen G. (1999), "*Kalman Filtering with Real-time Applications*". Ed. Springer, USA.

[4] Cruz-Pérez D. (2004). "Modelo Dinámico para Tareas en Tiempo Real". Tesis de Maestría en Ciencias de la Computación, 7 de agosto de 2004, CIC-IPN, México D.F.

[5] Ecker K. H. (2000). *"Overview on Distributed Real-time Systems. Requirements, Analysis, Specification, Operating Systems, Case Studies"*. Institute of Informatik, Technical University of Clausthal, Germany.

[6] Guevara P., Medel J. J., Cruz D. (2004). "*Modelo Dinámico para una Tarea en Tiempo Real*". Revista Computación y Sistemas, ISSN 1405-5546, Vol. VIII No. 1, México, Septiembre de 2004.

[7] Haykin S. (1991). *Adaptive filter theory.* Prentice Hall information and system sciences series.

[8] Kotel'nikov V. A. (1933). *"On the transmission capacity of "ether" and wire in electrocommunications "*. Izd, Red. Upr. RKKA (Moscow URSS) (Material for the first all-union conference on questions of communications), vol. 44, 1933.

[9] Medel J. J. (2002). "*Análisis de dos métodos de estimación para sistemas lineales estacionarios e invariantes en el tiempo con perturbaciones correlacionadas con el estado observable del tipo: Una entrada una salida*" Computación y sistemas volumen 6 número 1, México.

[10] Medel J., Guevara P., Flores A. "*RTMDF: Real-Time Multivariable Digital Filter*". International IEEE Workshop Signal Processing 2003, Poznan Polonia, October2003.

[11] Nyquist, H. (1928). *Certain Topics in Telegraph Transmission Theory*. USA. AIEE Transactions.

[12] Shannon C. E. (1948), *"A mathematical theory of communication"*. Bell Syst. Tech. J. vol. 27, pp. 379-423, 623-656, July-Oct.

[13] Whittaker E. T. (1915), *"On the functions which are represented by the expansion of interpolation theory"*. In Proc. Roy. Soc. Edimburgh, vol. 35, pp. 181-194.

[14] Guevara P., Medel J.J., Cruz D. (2004). Modelo Dinámico para Tiempos de Arribo de una Tarea en Tiempo Real. Revista Computación y Sistemas, ISSN 1405-5546, Vol. VIII No. 1, pags. 190-209, México, Septiembre de 2004.

[15] J.J. Medel, P. Guevara, A. Poznyak. (2004). "Real-time Multivariable Digital Filter using Matrix Forgetting Factor and Instrumental Variable". Automatic Control and Computer Sciences Vol. 38, No. 1 pages 40-53, ISSN 0132-4160, February. 2004, (ISI), Latvia.

[16] J.J. Medel, P. Guevara, A. Flores. (2004). "Caracterización de Filtros Digitales en Tiempo Real para Computadoras Digitales". Revista Computación y Sistemas, ISSN 1405-5546, Vol. VII No. 3, México.

[17] Guevara-López P. (2004) "Resumen de Tesis Doctoral". Revista Computación y Sistemas, ISSN 1405-5546, Aceptado para publicación, Septiembre de 2004.

Author Index

Índice de autores

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Noviembre de 2007.
Printing 500 / Edición 500 ejemplares.