# Advances in Computing Science and Applications

# Research in Computing Science

## Series Editorial Board



***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 147, No. 12,** diciembre 2018. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No.: 04-2005-121611550100-102, 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:** *Grigori Sidorov, RFC SIGR651028L69*

**Research in Computing Science** is published by the Center for Computing Research of IPN. **Volume 147**, **No. 12**, December 2018. Printing 500. The authors are responsible for the contents of their articles.  Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

# Advances in Computing Science and Applications

**Juan C. Chimal**
**Christian E. Maldonado (eds.)**


Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2018

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



Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico

ISSN 1870-4069

# Editorial

As much as the information age revolution has deeply transformed the world and our day to day life, the rate of change is far from slowing down. Quite contrary, every day new tasks are automated or improved upon with the addition of computer models, algorithms and hardware.

The current number of *Research in Computing Science* (RCS) is witness to this trend as it covers a broad array of applied and theoretical advances in different areas of Computer Science. Novel and seasoned authors provide excellent insights and descriptions of methods to apply technological solutions or improve upon previous existing ones.

From vast and comprehensive compilations of satellite imagery to novel ideas on the implementation of a CUDA algorithm to solve Poisson's equation –passing through several innovative uses of existing hardware on medical applications– the richness of the present volume lies on its variety.

This issue is the product of the submission and thorough examination of 70 articles with 30 accepted papers, thus, an acceptance rate of 42.85%. The amount of works with innovative contributions was both inspiring and challenging on its selection.

We want to express our gratitude to the numerous people responsible for bringing this number into fruition. To the authors for their confidence in this platform to present the products of their research and inventive to the scientific and academic community, as well as to all the reviewers and editors, whom without their dedication and hard work this endeavor would not have been possible. Our special thanks go to the *Centro de Investigación en Computación del Instituto Politécnico Nacional* of Mexico (CIC-IPN) for their ample support and nurturing environment.

Finally, it should not go unnoticed that the submission, review and selection of the articles was enabled by the excellent and free of charge platform EasyChair, www.easychair.org.

Juan C. Chimal
Christian E. Maldonado
*Guest Editors*
December, 2018

ISSN 1870-4069

# Table of Contents

Page

ISSN 1870-4069

# A Method for Malware Analysis by Virtual Machine Introspection Technique

Luis Enrique Héctor Almaraz García, Raúl Acosta Bermejo

Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico City, Mexico
lalmarazg1400@alumno.ipn.mx, racostab@cic.ipn.mx

**Abstract.** Malicious code has become one of the biggest threats in the field of computer security. Traditional malware monitoring tools are installed in the physical host, they trust in the integrity of the host, however, they are vulnerable to being infected by malware and delivering erroneous results about monitoring. In this paper, a method based on Virtual Machine Introspection technique is proposed to obtain the memory image of a Virtual Machine, from outside, with the help of the VirtualBox API, also analyze its internal content such as running processes, threads, network connections, and open files with the use of the Volatility Framework to interpret the low-level bytes into high-level information and finally, report this information in a monitoring register. This approach has been tested with the execution of 3 samples of malware inside a 32-bit Microsoft Windows XP SP3 Virtual Machine and the results obtained support the main hypothesis that if the Virtual Machine Introspection technique is applied to a Virtual Machine then it is possible to obtain the activities of a process and according to its behavior, identify malware.

**Keywords:** virtual machine introspection, malware, process monitoring, memory forensics, dynamic analysis.

## 1 Introduction

Computer forensics is the science that identifies, collects, preserves, analyses and presents the data that has been processed and stored on electronic media, in a legal process [1]. Virtual Machines provides efficient use of resources, ease of management, low operation and maintenance costs, flexible systems and even efficient power consumption [2, 3]. Virtual Machine forensic has focused on collecting forensic data to uncover the malicious content in the memory [4, 5], in this case, malware.

Malware or malicious code refers to a program that is inserted into another program and that compromises the confidentiality, integrity or availability of data, applications or the operating system itself [6]. Malicious code has become one of the biggest threats in the field of computer security, the number of malware has grown in recent years [7], [8] and it is reported that every 4.6 seconds a new malware specimen emerged in 2017 [9].

Traditional malware monitoring tools are installed in the physical host, they trust in the integrity of the host, however, they are vulnerable to being infected by malware and delivering erroneous results about monitoring [10]. The approach proposed in this paper is based on Virtual Machine Introspection (VMI) [11] technique to obtain the memory image of a Virtual Machine (VM), from outside, in this case, with the help of the VirtualBox API [12], analyze its running processes, threads, network connections, and open files with the use of the Volatility Framework [13] and finally, report this information in a monitoring register.

The rest of the paper is organized as follows. Section 2 introduces a background information about the Virtual Machine Introspection technique. Section 3 presents related work to Virtual Machine Introspection in search of possible malicious processes. The design of the proposed method is described in section 4. Section 5 specifies the implementation of the method. The experimental results are provided in section 6 and finally, the conclusions and future work are discussed in section 7.

# 2 Background

## 2.1 Virtual Machine Monitor

Virtual Machine Monitor (VMM) or hypervisor is a software that enables communication between Virtual Machines and real host [4]. It provides the virtual environment by means of a Virtual Machine where other programs can be executed just as they do in a real environment [14]. There are two types of VMMs [2]. Type I VMM, Fig. 1, is one that runs directly on the hardware, some examples of hypervisors of this kind are: VMware ESX/ESXi, Citrix Xen Server and Oracle VM, they are used in data centers and in server environment. The type II VMM, Fig. 2, is installed on the operating system of the real host as another user program, examples of this kind of hypervisor are: VMWare Workstation/Fusion/Player, VirtualBox, Parallels and Microsoft Virtual PC.

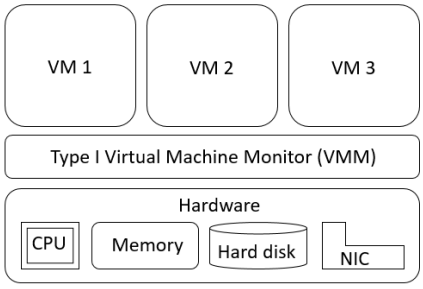



**Fig. 1.** Type I Virtual Machine Monitor.

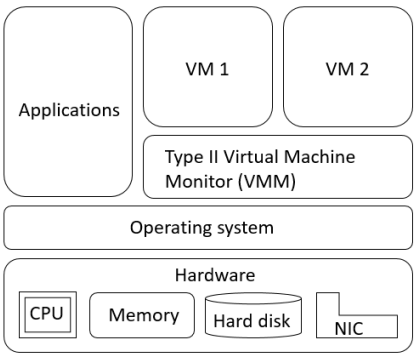



**Fig. 2.** Type II Virtual Machine Monitor.

### 2.2 VirtualBox Hypervisor

The approach proposed is based in one of the type II VMM that is called VirtualBox, Fig. 3. It provides a main API that is implemented using the Component Object Model (COM), an interprocess mechanism for software components originally introduced by Microsoft for Microsoft Windows. In this way, in a host with a Microsoft operating system, VirtualBox uses COM and in the rest of operating systems such as Linux and macOS, VirtualBox uses the Cross Platform Component Object Model (XPCOM), a free software implementation of COM originally created by the Mozilla project for their browsers [12]. The VirtualBox front-ends use COM/XPCOM to call the Main API and each VM is working with a VirtualBox client, which helps the VM interact with the VBoxSVC process, the service corresponding to the VMM [14].

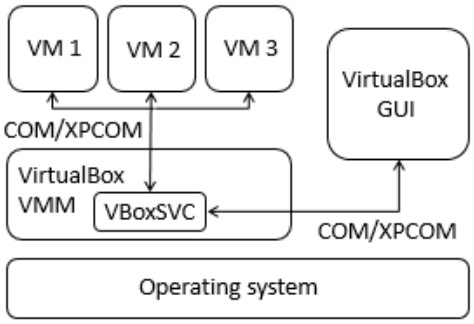



**Fig. 3.** Architecture of VirtualBox.

### 2.3 Virtual Machine Introspection

Virtual Machine Introspection is a technique to analyze the memory of a given VM to detect its internal activities from outside over the Virtual Machine Monitor layer [10], [11]. Such activities are related with memory, disk, CPU registers, network connections and available kernel symbols of the VM [15, 16]. This technique has been used for intrusion detection, malware analysis and memory forensics [17]. The method described in this paper performs Virtual Machine Introspection through of the VirtualBox API.

### 2.4 Volatility Framework

The Volatility Framework is a collection of tools developed in Python, for the extraction of digital artifacts from volatile memory (RAM). The framework is not a memory acquisition tool [13]. It supports investigations for different guest operating systems like macOS, Linux and Windows and extracts the operating system data structure from memory pages of introspected Virtual Machine identifying processes in execution and details of hidden processes that allow detecting malware in the virtual environment [15].

## 3 Related Work

The proposal of VMI has been introduced since 2013 [18]. In recent years, researchers have adopted this technique to detect malware. Authors in [19] have designed and implemented a process detection system called VmRecoverySystem, with KVM as a hypervisor that consists of four modules. Semantic reconstruction module reconstructs the processes and kernel data structures found in memory. Detection module checks the existence of critical tasks and their system calls in the Virtual Machine. The policy module contains the user's configuration to delete or start a process and likewise, to restart or restore a Virtual Machine.

The recovery module executes the previously actions. They wrote a simple rootkit which tampered the system calls table to test the operation of their system. This idea is also used in [5], with the same KVM hypervisor, but with a scheme of 3 modules, the Virtual Machine process search module searches all the running process identifiers associated with the Virtual Machine. Then, using the Virtual Machine memory dump module to dump the memory of each process obtained previously to finally, the memory forensics analysis module reads the Virtual Machine memory dump file, of each process, to obtain the user, pid, cpu usage and memory usage. Their purpose is to ensure the integrity and efficiency of the information of the processes in the different files.

In [20], a VM Introspection method is introduced to monitor the presence of malware in the volatile memory of the Virtual Machine through the analysis of its processes, files, registers and network activities. They perform introspection to the Xen hypervisor with the help of LibVMI library and analyze system behavior using memory forensics integrating the Volatility Framework. They tested their method running 20 advanced and 8 script-based malware samples and their results were verified with three sandboxes: sandbox CIA, Cuckoo Sandbox and CaptureBAT, they got similar detections in terms of processes, files, registers and network activities. Another approach for Xen hypervisor based Virtual Machine is presented in [15], it has 4 components, the State Information Extraction uses the LibVMI library to reconstruct the memory from raw data to readable information.

The Anomaly based Detection Engine knows which processes access certain system calls, as well as the order of them. The Malicious Port Detection Engine holds a database of well-known backdoor ports. The Notification generates an alert to the user to indicate the presence of possible malware in the Virtual Machine. Average Coder and Jynx2 rootkits were used in their experiment to detect process abnormal behavior through system calls and ports found.

Researchers in [4] propose a mechanism to monitor processes in a VMware Virtual Machine with Windows XP. A snapshot of the Virtual Machine is made. Next, the Virtual Machine is suspended and cryptographic MD5 sums of the next files are taken: vmem, vmsn, and vmss, they keep the information about the current state, snapshots and saved state of the Virtual Machine respectively. They tested their methodology over the Virtual Machine and got the list of processes from the previously files and from the memory dump of the host operating system.

# 4 Design of the Method

This section details the design and operation of the proposed method to analyze malware in a Virtual Machine through Virtual Machine Introspection technique, see Fig. 4. It consists of the following five steps:

1. Access to the asset: This is achieved through the interprocess mechanism called COM/XPCOM that implements the VirtualBox API.
2. Collection: It generates a memory dump of the Virtual Machine volatile memory.
3. Analysis: It translates the low-level bytes into high-level information with the help of the Volatility tool, through the profile of the virtual machine and extracts objects from the operating system.
4. Logging: It generates the log of the malware analysis.
5. Containment: The COM/XPCOM interprocess mechanism sends a killing command to finish the malware execution from outside the Virtual Machine.

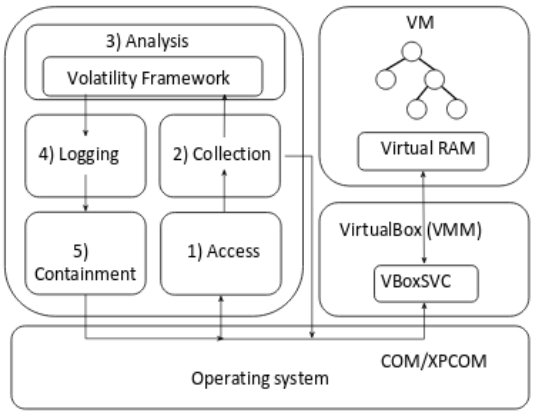



**Fig. 4.** Architecture of the proposed approach.

# 5 Implementation of the Method

## 5.1 Access to the Asset

The first step consists in a programmed module that identifies the Virtual Machine to be monitored from a set of Virtual Machines available from the VirtualBox hypervisor. This is done through the COM/XPCOM interprocess mechanism that calls the VirtualBox API, which in turns, interacts with the VBoxSVC process that manages the current Virtual Machine. This module is responsible for determining the current state of the Virtual Machine, if it is off, then it is turned on and a session is created to interact with the Virtual Machine. If the Virtual Machine state is on from the beginning, then only the session is created.

### 5.2 Collection

This step generates a memory dump of the Virtual Machine volatile data with the help of a programmed module. First, the state of the Virtual Machine is paused, then the module interacts with the hypervisor through the VirtualBox API to request a memory dump. By means of the VirtualBox debugger it is possible to acquire memory data in ELF format by default, next the module extracts the first section of the ELF file that is called LOAD where the memory dump data is placed, this is possible by searching this section and its correct offset in the file, also the size of the data. Once the above is done, a raw file is created, this new file contains the memory image that can be analyzed for the next module, also the state of the Virtual Machine is resumed.

### 5.3 Analysis

The goal of this step is to solve the problem of the semantic gap by using the Volatility Framework. It takes the raw file obtained in the previous step and interprets the low-level bytes into high-level information by obtaining the process list from memory dump, number of threads, network connections and open files of them. Also, this step determines and choose the correct profile associated with the operating system of the Virtual Machine to be monitored.

### 5.4 Logging

It generates a log in plain text that contains the name of the processes, their identifiers, identifiers of the parent processes, the number of threads, network connections and open files associated with them within the Virtual Machine through the monitoring time, this step is automated by means of a programmed module. Each log is created inside a directory that receives the name of the Virtual Machine monitored and each log is named with the start date and time of the monitoring.

### 5.5 Containment

The last step requires minimal human effort to determine that the information delivered by the generated logs, in each monitoring, report the presence of unknown processes by the security administrator, which can be malicious, once this has been detected, then he, through the implementation of this module, executes a command to kill the execution of the process inside the Virtual Machine. The COM/XPCOM interprocess mechanism is the responsible for sending the command form outside to the monitored Virtual Machine and guarantee its execution.

## 6 Experimental Results

In order to test the operation and functioning of the approach, three experiments were performed, each of them consisted in the execution of a different sample of malware, Table 1, inside the monitored Virtual Machine. The experiments were performed with physical machine of Intel ® core (TM) i7 CPU @ 2.60 GHzx4, 64-bit Kali GNU/Linux

Rolling as operating system with 8 GB in memory and a Virtual Machine with 32-bit Microsoft Windows XP SP3 as guest operating system with 1 GB in memory.

**Table 1.** Main characteristics of malware samples.

| Experiment | Malware | MD5 | Antivirus detection from VirusTotal |
|---|---|---|---|
| 1 | Trojan | 7583a73f73638d23298ddb4900def643 | 56/64 |
| 2 | Trojan | 8915452ee0b8e754ee7b047a849a01a2 | 58/68 |
| 3 | Trojan | c334b788e3da78c413364ef1e163b8ff | 43/68 |

Using the method described in section 4, the approach started with the access module that selected the Virtual Machine to be monitored. The acquire module got the raw dump file to be analyzed by the next module. The analyze module detected the correct profile to be used with the Volatility Framework on the Virtual Machine, it was the "WinXPSP3x86" profile. It extracted relevant data of each sample of malware such as their identifiers, the number of their threads, network connections and open files.

The reporting module generated each monitoring log, the results obtained by this module, Table 2, allowed knowing the behavior of each sample of malware during its execution within the Virtual Machine.

In the first experiment the introduced malware executed 7 threads, 4 network connections and 24 open files. The four network connections correspond to IP addresses located in Canada, Italy, Pakistan and Germany, they are considered as part of blacklist according to VirusTotal. Among the total files opened by this malware, the suspicious files correspond to those whose functions are related to the Microsoft Windows

**Table 2.** Results reported by the reporting module.

| Experiment | Malware | #Threads | Network connections | | # Open files | Suspicious files |
|---|---|---|---|---|---|---|
| | | | Port | Address | | |
| 1 | Trojan | 7 | 8143 | 199.7.136.88 | 24 | ws2_32.dll, ws2help.dll, mswsock.dll, dnsapi.dll, 996E.exe. |
| | | | 1743 | 151.80.142.33 | | |
| | | | 243 | 202.69.40.173 | | |
| | | | 7447 | 78.47.66.169 | | |
| 2 | Trojan | 5 | 1029 | 0.0.0.0 | 21 | wsock32.dll, ws2_32.dll, crypt32.dll, autoexec.bat. |
| 3 | Trojan | 1 | 80 | 211.104.175.45 | 12 | ws2_32.dll, mswsock.dll, wshtcpip.dll, rpcrt4.dll, dnsapi.dll. |

Sockets API, as well as to the file that contains these functions to establish connections to the Internet, a file was also found that is responsible for the resolution of domain names to their corresponding IP addresses. The file that is called 996E.exe, was opened by the malware, which is known to be used by Windows operating system to assure some specific programs to run properly, however the malicious code uses this file to circulate its own infectious files through its created threads and make the user believe that it is a benign software.

In the second experiment 5 threads, 1 network connection and 21 open files were registered, the only connection made by the malware corresponds to the IP address 0.0.0.0, the malware that uses this logical address do so in order to establish a connection remote from the attacking host, regardless of whether the victim host has multiple network interfaces, in this way, perform other actions such as establishing an ftp or ssh connection, related to it, the suspicious files opened by the malware allow to establish network connections and one of them contains encryption functions typical of the Microsoft Windows API. The file that is called autoexec.bat is altered by the malware to establish its self-execution with each start of the operating system.

The last experiment consisted in the execution of the third sample of malware, which was associated with the creation of 1 thread, 1 network connection and 12 open files, the IP that corresponds to its unique connection is located in South Korea. Its open files are related to network application activities and the port used to connect to the IP is port 80, typical of the HTTP protocol, it also uses the Remote Procedure Call API for communication with the Internet and the module that develops the resolution of domain names, translates the following URL registered in the log: http://download.everytoolbar.co.kr/setup/everytoolbar2_setup.exe to the aforementioned remote IP address, which is considered part of the blacklist stored in VirusTotal.

With the last module, of the last step, each malware sample execution, in each experiment, was killed from outside by means of this programmed module and as a consequence, each of the processes associated with them finalized their execution inside the Virtual Machine, this was achieved with the implemented “Kill” command concatenated with the process identifier which was reported in the log monitoring.

## 7 Conclusions and Future Work

In this paper, a method for malware analysis based on the Virtual Machine Introspection technique was proposed with the design of a 5 steps method: access, collection, analysis, logging and containment. By analyzing the memory image of the Virtual Machine, the behavior of three samples of malware in the Virtual Machine such as their identifiers, identifiers of their parent processes, their number of threads, also their network connections and open files of them were obtained. This information was reported in a log monitoring and it allowed to the security administrator the ability to kill a process, from outside the Virtual Machine, which he could consider as malicious. In this way, the experiments results verify the hypothesis that is possible to identify malware by means of Virtual Machine Introspection technique and solving the semantic

gap problem by using the Volatility Framework to interpret the low-level bytes into high-level information

There are some aspects that should be improved as: Apply the method on the Linux guest operating system, find a better way to locate important structures in the volatile memory dump to replace the Volatility framework with one proposed and with the steps to analyze malware from memory dump, extend the method to identify rootkits in the Linux guest operating system.

**Acknowledgment.** The authors are thankful to the Instituto Politécnico Nacional (IPN) and Consejo Nacional de Ciencia y Tecnología (CONACyT) for their support in the development and achievement of this research.

## References

1. López, C., Guadrón, R.: Computer forensics. In: 2016 IEEE 36th Central American and Panama, pp. 1–6. IEEE, San Jose, Costa Rica (2016)
2. Riaz, H., Ashraf, M.: Analysis of VMware virtual machine in forensics and anti-forensics paradigm. In: 2018 6th International Symposium on Digital Forensic and Security (ISDFS), pp. 1–6, IEEE, Antalya, Turkey (2018)
3. Thongthua, A., Ngamsuriyaroj, S.: Assessment of Hypervisor Vulnerabilities. In: 2016 International Conference on Cloud Computing Research and Innovations (ICCCRI), pp. 71–77, IEEE, Singapore, Singapore (2016).
4. Huseinović, A., Ribić, S.: Virtual Machine Memory Forensics. In: 2013 21st Telecommunications Forum Telfor (TELFOR), pp. 940–942. IEEE, Belgrade, Serbia (2013)
5. Guangqi, L., Lianhai, W., Shuhui, Z., Shujiang, X., Lei, Z.: Memory Dump and Forensic Analysis Based on Virtual Machine. In: 2014 IEEE International Conference on Mechatronics and Automation, pp. 1773–1777, IEEE, Tianjin, China (2014)
6. Souppaya, M., Scarfone, K.: Guide to Malware Incident Prevention and Handling for Desktops and Laptops. NIST Special Publication 800-83 Rev 1 (2013)
7. Liu, J., Wang, Y., Wang, Y.: The Similarity Analysis of Malicious Software. In: 2016 IEEE First International Conference on Data Science in Cyberspace (DSC), pp. 161–168, IEEE, Changsha, China (2016)
8. Zhang, D., Zhang, Z., Jiang, B., Tse, T.: The Impact of Lightweight Disassembler on Malware Detection: An Empirical Study. In: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), pp. 620–629, IEEE, Tokyo, Japan (2018)
9. Kan, Z., Wang, H., Xu, G., Guo, Y., Chen, X.: Towards Light-Weight Deep Learning Based Malware Detection. In: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), pp. 600–609, IEEE, Tokyo, Japan (2018)
10. Li, N., Li, B., Li, J., Wo, T., Huai, J.: vMON: An Efficient Out-of-VM Process Monitor for Virtual Machines. In: 2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, pp. 1366–1373, IEEE, Zhangjiajie, China (2013)
11. Nemati, H., Dagenais, M.: VM processes state detection by hypervisor tracing. In: 2018 Annual IEEE International Systems Conference (SysCon), pp. 1–8. IEEE, Vancouver, BC, Canada (2018)
12. Oracle VM VirtualBox.: Programming Guide and Reference. Oracle Corporation (2018)
13. Volatility Foundation.

14. Jiang, S., Cai, H.: Monitoring VirtualBox Performance. In: Department of Computer Science and Engineering, University of Notre Dame, pp. 1–6, University of Notre Dame, Notre Dame, USA (2012)
15. Ajay, M., Jaidhar, C.: Virtual Machine Introspection based Spurious Process Detection in Virtualized Cloud Computing Environment. In: 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), pp. 309–315, IEEE, Noida, India (2015)
16. Hebbal, Y., Laniepce, S., Menaud, J.: K-binID: Kernel binary code identification for Virtual Machine Introspection. In: 2017 IEEE Conference on Dependable and Secure Computing, pp. 107–114, IEEE, Taipei, Taiwan (2017)
17. Qiang, W., Xu, G., Dai, W., Zou, D., Jin, H.: CloudVMI: A Cloud-Oriented Writable Virtual Machine Introspection. In: IEEE Access, IEEE, vol. 5, pp. 21962–21976, National Natural Science Foundation, China (2017)
18. Garfinkel, T., Rosenblum, M.: A virtual machine introspection based architecture for intrusion detection. In: Network and Distributed Systems Security Symposium, pp. 191–206, NDSS, San Diego, California, USA (2003)
19. Hua, Q., Zhang, Y.: Detecting Malware and Rootkit via Memory Forensics. In: 2015 International Conference on Computer Science and Mechanical Automation (CSMA), pp. 92–96, IEEE, Hangzhou, China (2015)
20. Wei, C., We, J., Chieh, S., Kuo, S.: Memory forensics using virtual machine introspection for Malware analysis. In: 2017 IEEE Conference on Dependable and Secure Computing, pp. 518–519, IEEE, Taipei, Taiwan (2017)

ISSN 1870-4069

# Reconocimiento de rostros mediante estructuras faciales antropométricas

Yair Velasco-Ramírez, Edgardo M. Felipe-Riverón, Ricardo Barrón-Fernández

Instituto Politécnico Nacional, México
rbarron@cic.ipn.mx

**Resumen.** Las estructuras faciales antropométricas son estructuras que se obtienen mediante puntos somatométricos (los cuales son empleados para la extracción de características craneofaciales). En este trabajo se propone una estructura facial antropométrica que contribuya al desarrollo de un método que realice reconocimiento de rostros de manera automática. Este método está basado principalmente en cinco etapas: extracción de coordenadas de los puntos somatométricos, creación de la estructura antropométrica, cálculo de distancias entre puntos, detección de ángulos de los puntos y cálculo de proporciones respecto a las distancias. Las distancias y los ángulos se obtienen con base en la relación de los puntos, relación que es establecida con la estructura antropométrica propuesta. A diferencia de otros trabajos relacionados con el reconocimiento de rostros, el método propuesto utiliza solamente nueve puntos, lo que es, hasta ahora, uno de los menores números de puntos utilizados.

**Palabras clave:** reconocimiento, rostros, puntos somatométricos, estructura antropométrica.

## Recognition of Faces using Anthropometric Facial Structures

**Abstract.** Anthropometric facial structures are structures obtained through somatometric points (which are used for the extraction of craniofacial features). In this work we propose an anthropometric facial structure that contributes to the development of a method that performs face recognition automatically. This method is based mainly on five stages: extraction of coordinates of the somatometric points, creation of the anthropometric structure, calculation of distances between points, detection of angles of the points and calculation of proportions with respect to distances. Distances and angles are obtained based on the relationship of the points, a relationship that is established with the proposed anthropometric structure. Unlike other works related to the recognition of faces, the proposed method uses only nine points, which is, until now, one of the lowest number of points used.

**Keywords:** recognition, faces, somatometric points, anthropometric structure.

## 1. Introducción

Todo sistema de reconocimiento de rostros requiere de una previa adquisición de características, mismas que sean capaces de brindar información sobre el rostro de una persona. A partir de dichas características es posible generar una alternativa para representar el rostro, la cual, para este trabajo, es una estructura antropométrica.

Esta estructura necesita cubrir al menos dos aspectos importantes: que sea capaz de representar un rostro y que dicha estructura sea única para cada persona.

Debido a que hay rasgos faciales que tienden a sufrir cambios con el paso del tiempo o pueden ser modificados por procedimientos menores (como maquillaje, barba, etc.), la estructura antropométrica que se propone está basada en nueve puntos somatométricos, mismos que fueron específicamente seleccionados debido a que son poco alterables inclusive bajo la influencia de algunas expresiones faciales.

Utilizando estos nueve puntos, se está realizando un trabajo de reconocimiento de rostros con un número pequeño de puntos, lo cual resulta una ventaja al momento de no contar con otros puntos visibles o, simplemente, que éstos no puedan ser obtenidos.

El propósito sustancial de utilizar este número de puntos es reducir redundancias a la hora de realizar la clasificación y, de igual manera, evitar procesar mucha información para el reconocimiento de un rostro.

El principal factor a considerar en el método propuesto para el reconocimiento de rostros es el conjunto de características o conjunto de patrones necesarios para realizar comparaciones entre rostros. Por esta razón, es necesario que para cada una de las imágenes la zona facial se muestre completa, es decir, que la persona esté de frente, esto es con la finalidad de evitar que alguno de los nueve puntos se pierda. De igual manera, las imágenes requieren tener un cierto nivel de iluminación, ya que si son demasiado obscuras podría no ubicarse un punto de manera correcta o, en el peor de los casos, omitirse por completo.

## 2. Estado del arte

Existen diferentes técnicas de reconocimiento facial; todas ellas extraen ciertas características del rostro para la etapa de clasificación. En el trabajo desarrollado por Dubey et al. [3] mencionan que el éxito de la metodología en el reconocimiento facial depende en gran medida de la selección de las características utilizadas por el sistema de reconocimiento. Proponen un método de selección de características a través de la Somatología (ciencia derivada de la Antropología) y sugieren tomar dos imágenes del sujeto para la extracción de 37 puntos: una frontal, de la que se obtienen 25 puntos, y otra de perfil, de la cual se extraen 12 puntos y con base en esos puntos, calculan 12 distancias euclidianas entre los puntos de la imagen frontal y 8 distancias más entre los puntos de la imagen de perfil.

Gupta et al. [5] desarrollaron en el 2010 un algoritmo de reconocimiento facial en tres dimensiones que emplea distancias geodésicas y euclidianas en 3D para 10 puntos faciales somatométricos, llamado "Anthroface". Aislaron 70 proporciones antropométricas (propuestas por Farkas [4]) asociadas con la región facial, de las cuales identificaron que 23 tienen valores altos de desviación estándar para poblaciones de adultos. Se etiquetaron de manera manual los 25 puntos faciales asociados a las 23 proporciones identificadas. Con base en la experimentación, Gupta concluyó que algunos de los 25 puntos utilizados eran redundantes, por lo que se determinó un subconjunto de 10 puntos; al comparar los resultados del reconocimiento de los 25 puntos propuestos inicialmente y estos 10 puntos, los resultados fueron estadísticamente equivalentes, de modo que al final se utilizaron 10 de los 25 puntos propuestos.

La tabla 1 contiene trabajos sobre reconocimiento de rostros, mostrando los resultados obtenidos para cada uno, mismos que servirán como punto de comparación para el método propuesto en este trabajo.

**Tabla 1.** Resultados de trabajos afines.

| Artículo | Resultados (accuracy) |
|---|---|
| Face detection and recognition with SURF for human-robot interaction [2] | 96 % |
| Face detection using information fusion [1] | 93 % |
| Face Recognition Using Local Binary Decisions [6] | 91-96 % |
| Fast face recognition based on fractal theory [8] | 93.3 % |
| Applying artificial neural networks for face recognition [7] | 94.7-96.7 % |

## 3. Desarrollo

### 3.1. Puntos somatométricos seleccionados

El conjunto de puntos sobre el cual se basa la construcción de la estructura antropométrica (mediante observación y con base en el estudio realizado por Gupta) es considerado como el conjunto que aporta más información geométrica del rostro y que mejor lo caracteriza debido a la mínima variación que presentan los puntos ante expresiones faciales y ligeras rotaciones en los distintos planos: axial, coronal y sagital.

Los puntos somatométricos utilizados son: de la región de los ojos, los dos exocanthiones (ex, ex') y los dos endocanthiones (en, en'), así como el nasion (n); de la región de la nariz únicamente el subnasal (sn); y de la región de la boca el stomion (sto) y los dos goniones (go, go') (figura 1).

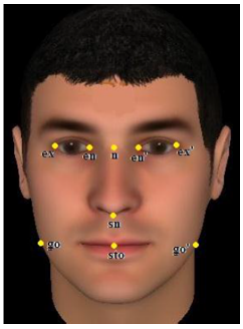



**Fig. 1.** Puntos somatométricos seleccionados.

### 3.2. Obtención de las coordenadas de los puntos y estructura propuesta

El trabajo realizado toma como principio que los puntos ya están ubicados en el rostro, por lo que para la obtención de las coordenadas, se procesa una imagen en escala de grises con los puntos somatométricos resaltados en rojo (figura 2a). Las coordenadas de los pixeles de los puntos se van almacenando en orden de aparición de derecha a izquierda (tomando como referencia la perspectiva del rostro en la imagen) de tal forma que faciliten la construcción de la estructura antropométrica (figura 2b).

El propósito principal es, entonces, establecer patrones a partir de la estructura que permitan establecer un margen de similitud o diferencia bajo comparaciones entre dichos patrones para poder realizar el reconocimiento de un rostro mediante un algoritmo de clasificación.

La estructura antropométrica está conformada por los puntos (nodos) y las distancias entre estos (aristas). Tomando el supuesto que los rostros que se utilizarán para los experimentos son simétricos, se plantea una relación entre nodos donde, si se divide la estructura exactamente por la mitad de tal manera que solo tendrán relación aquellos puntos que se encuentren del lado derecho con los puntos del lado derecho y los que se encuentren del lado izquierdo tendrán relación solamente con los del lado izquierdo.

**Corrección en el plano sagital** Las invariantes de rotación en los planos representan un problema a la hora de extraer las mediciones de la estructura, ya que para poder obtener resultados satisfactorios es necesario que el rostro esté de frente, por lo que es preciso corregirlas antes de su cálculo.

Para la corrección de rotación del plano sagital, se toman en cuenta las coordenadas de los dos endocanthiones. Se obtiene la pendiente de la línea generada entre el endocanthion derecho y el endocanthion izquierdo. El ángulo de rotación del rostro es determinado por el arco tangente de la pendiente (a, b)

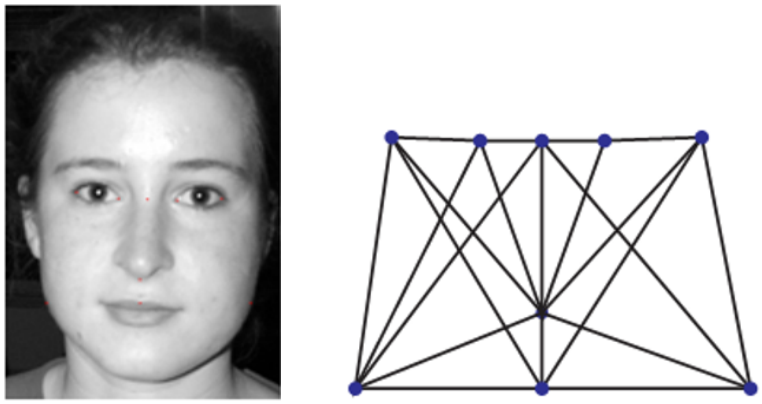


**Fig. 2.** a) Puntos ubicados en el rostro, b) Estructura antropométrica utilizada.

(ecuación 1):

$$\theta = tan^{-1}\left(\frac{b_2 - b_1}{a_2 - a_1}\right). \tag{1}$$

En un plano de coordenadas x, y la rotación de una imagen se obtiene mediante el producto de una matriz de rotación por el vector correspondiente a las coordenadas [x, y] de cada uno de los pixeles de la imagen. Se debe tener en cuenta que para una imagen el eje x corresponde a las filas y el eje y a las columnas, por tanto, las coordenadas de los pixeles de la imagen rotada responden a la ecuación 2:

$$\begin{aligned} x' &= y \cdot \sin\theta + x \cdot \cos\theta, \\ y' &= y \cdot \cos\theta - x \cdot \sin\theta. \end{aligned} \tag{2}$$

**Corrección en el plano coronal** La rotación en el plano coronal implica un problema debido a que produce una asimetría en la estructura antropométrica. Esta asimetría tiene dos implicantes; la primera, que la coordenada y del nasion es diferente a la coordenada y del subnasal y el stomion. Puesto que, al tener el rostro de frente, la distancia entre el gonion derecho y el stomion debe ser semejante a la distancia entre el gonion izquierdo y el stomion. La segunda implicante que presenta la asimetría de la estructura antropométrica es la diferencia de distancia entre estos puntos en el eje x. De modo que, si el rostro está rotado hacia la derecha, la distancia entre el gonion izquierdo y el stomion es mayor que la distancia entre el gonion derecho y el stomion (figura 3).

La diferencia entre las distancias del gonion izquierdo–stomion y gonion derecho–stomion se establece con una diferencia máxima de pixeles. Si dicha tolerancia es superada existe una rotación del rostro en el plano coronal.

A diferencia de la corrección del ángulo de rotación sobre el plano sagital, la corrección sobre el plano coronal no se realiza sobre la imagen, esta corrección

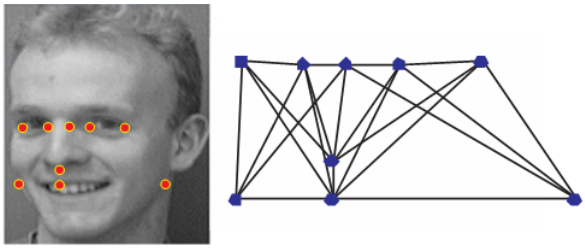


**Fig. 3.** El plano coronal.

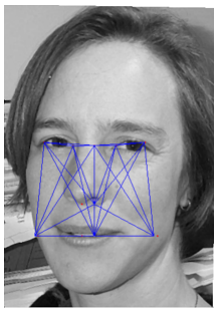


**Fig. 4.** Corrección de la rotación en el plano coronal.

se realiza únicamente modificando las coordenadas x de todos los puntos de la estructura antropométrica, con excepción del nasion, ya que éste es tomado como referencia. Para poder establecer las nuevas coordenadas de los puntos se calcula la distancia promedio entre puntos pares (gonion izquierdo – gonion derecho, etc.)

Posteriormente, tomando como referencia la coordenada x del nasion, se reubican los puntos a la distancia calculada y se actualizan las coordenadas. En el caso del stomion y subnasal, quienes no son considerados como puntos pares, sus nuevas coordenadas son establecidas por la coordenada x del nasion. En la figura 4 se puede ver la reubicación de los puntos sobre la imagen, así como los puntos originales.

**Obtención de los patrones** La estructura antropométrica propuesta permite obtener diversas características que pueden ser patrones para el algoritmo de clasificación, sin embargo, no todas son útiles ya que algunas poseen menos información que caracterizan al rostro inconsistentemente, es decir, puede haber ciertas características que no se presten para realizar comparaciones entre las mismas ya que no se pueden normalizar o sus valores varían demasiado entre diversas imágenes del rostro de una misma persona. Por este motivo se seleccionó un conjunto de tres características que, con base en los experimentos realizados,

permiten establecer un rango de error entre mediciones y, al mismo tiempo, establecer un margen de diferencia entre rostros de personas distintas. Las características seleccionadas son:

- ***Distancias entre puntos.*** Con base en la relación de puntos en la estructura antropométrica propuesta, se obtiene la distancia euclidiana de cada arista una de las aristas.
- ***Ángulo de la pendiente.*** El ángulo de la pendiente de cada una de las aristas se obtiene con un criterio distinto a las distancias. Se toma como origen al subnasal y se calculan las pendientes de aquellas aristas que sean adyacentes a este punto. Son un total de 9 mediciones más dos que corresponden a la inclinación de los ojos.
- ***Proporciones de distancias.***Para resolver la invariante del tamaño del rostro se calculan proporciones de las distancias obtenidas. Una proporción equivale a dividir la distancia de cada arista entre el alto del rostro, de esta manera se asegura que todas las distancias de un mismo rostro varían en una mínima cantidad y así poder establecer una métrica más precisa de comparación entre personas.

Es importante resaltar que, a pesar de que las proporciones se calculan a partir de las distancias, las dos características de la estructura aportan información importante y se valoran de manera distinta en el clasificador.

**Clasificación** La base de datos cuenta con 470 imágenes de rostros que corresponden a 27 personas diferentes. A fin de comprobar que los patrones formados representan de manera precisa a un rostro, se utilizaron tres clasificadores:

- *KNN,*
- *Clasificador bayesiano,*
- *MLP.*

A fin de comprobar que los patrones formados representan de manera precisa a un rostro, se utilizaron tres clasificadores, dos de ellos probabilísticos ya que, al tener diversas clases con diferente número de elementos entre ellas, es posible establecer una función de densidad o directamente la probabilidad de que un elemento $x$ pertenezca a la clase $C_j$.

## 4. Resultados

### 4.1. Corrección de invariantes y cálculo de mediciones

La rotación de los rostros se soluciona de manera satisfactoria. En la figura 5 se muestra la corrección de la rotación del plano sagital para 4 rostros diferentes.

De igual manera, la corrección del plano coronal es satisfactoria. En la figura 6 se muestra la reubicación de los puntos para dos personas distintas y la estructura antropométrica sobre el rostro, esto con fin de apreciar la reestructuración de la misma.

**Fig. 5.** Corrección de la rotación en el plano sagital.

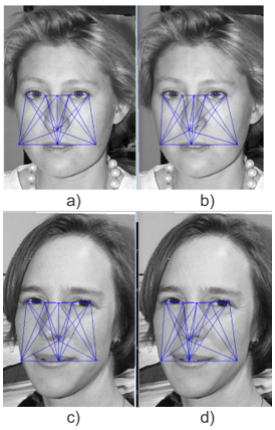



**Fig. 6.** Corrección del plano coronal, a) y c) corresponden a las imágenes originales, b) y d) corresponden a los puntos y la estructura corregida.

La corrección de la escala se probó en dos escenarios. El primero comparando las proporciones de la imagen de un rostro contra la misma imagen, solo que a esta última se le aplicó una reducción de tamaño del 30 %.

El segundo escenario fue comparar la imagen del rostro con otra imagen de la misma persona. Al final se obtienen mediciones bastante cercanas entre ellas (figura 7).

Los ángulos de las pendientes de los puntos se calculan planteando un sistema de coordenadas donde el subnasal es el origen (figura 8). Posteriormente, todas aquellas aristas que sean adyacentes a ese punto son procesadas. Aunados a estos puntos, están dos ángulos más, los ángulos que corresponden a las pendientes de los ojos. Estos últimos ángulos son procesados de manera independiente ya que no son adyacentes al subnasal, sin embargo, proporcionan información relevante del rostro.

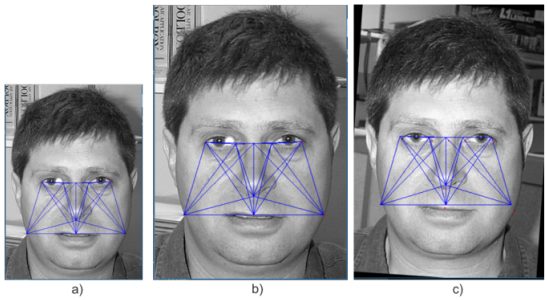



**Fig. 7.** Imágenes del rostro a comparar, a) reducción de la imagen, b) al 30 %, c) diferente imagen del mismo rostro.

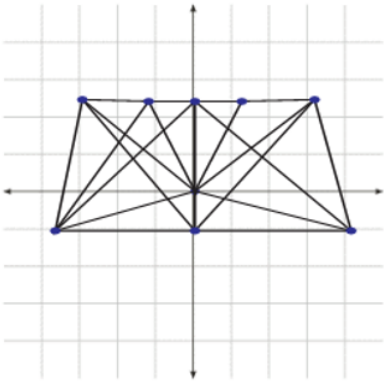


**Fig. 8.** Sistema de coordenadas.

### 4.2. Clasificación

Para las pruebas en la etapa de clasificación se utilizaron un total de 22 personas con una promedio de 20 imágenes para cada una, esto es debido a que algunas personas poseen un número bajo de imágenes para poder clasificarlas o, la iluminación de las mismas no permiten una adecuada localización de los puntos. Por tanto, se utilizó un total de 446 imágenes, donde se aplicó el 70 % para entrenamiento y el 30 % para prueba. Respecto al clasificador del vecino más cercano, se utilizó una configuración donde se establece un k=5. Las mediciones de recall, presicion y accuracy se muestran en la tabla 2.

## 5. Conclusiones

Se observan resultados por debajo del promedio de los trabajos afines, por lo que es hay dos posibles alternativas. Replantear las características propuestas

**Tabla 2.** Resultados de la clasificación de los rostros con los dos clasificadores.

| | KNN | Bayesiano | MLP |
|---|---|---|---|
| Presicion | 64.5 | 68.88 | 68.33 |
| Recall | 61.32 | 70 | 73.33 |
| Accuracy | 62 | 71.66 | 75.5 |

o, en su defecto, incrementarlas añadiendo áreas significativas o alguna otra característica que pueda brindar información del rostro. La segunda, probar con otro tipo de clasificador en donde se puedan ponderar los pesos o la importancia que se le da a cada característica.

A pesar de que los resultados son bajos, los resultados que arroja el clasificador bayesiano y la MLP son más altos que los del clasificador KNN. Es probable que, con base en estas observaciones, la solución más factible pudiera ser el proponer un clasificador más robusto.

Por otro lado, los resultados que se obtienen con el clasificador KNN indican que las mediciones entre rostros de personas diferentes sean muy similares. Hay que establecer un margen de diferencia más preciso para poder tener mayor distinción entre rostros.

**Agradecimientos.** Proyectos SIP 20181698, 20181895.

## Referencias

1. Aarabi, P., Lam, J.C.L., Keshavarz, A.: Face detection using information fusion. In: Information Fusion, 10th International Conference on. pp. 1–8 (2007)
2. An, S., Ma, X., Song, R., Li, Y.: Face detection and recognition with surf for human-robot interaction. In: Automation and Logistics. ICAL'09. IEEE International Conference on. pp. 1946–1951 (2009)
3. Dubey, S., Sharma, T.: A face recognition system through somatology. International Journal on Computer Science and Engineering 3(1), 155–160 (2011)
4. Farkas, L., Munro, I.: Anthropometric facial proportions in medicine. Thomas Books, first edn. (1987)
5. Gupta, S., Markey, M., Bovik, A.: Anthropometric 3d face recognition. International Journal of Computer Vision 90(3), 331–349 (2010)
6. James, A.P., Dimitrijev, S.: Face recognition using local binary decisions. IEEE Signal Processing Letters 15, 821–824 (2008)
7. Le, T.H.: Applying artificial neural networks for face recognition. Advances in Artificial Neural Systems 15 (2011)
8. Tang, Z., Wu, X., Fu, B., Chen, W., Feng, H.: Fast face recognition based on fractal theory. Applied Mathematics and Computation 321, 721–730 (2018)

ISSN 1870-4069

# Sistema de clasificación de deformaciones pédicas por procesamiento digital de imágenes y lógica difusa en LabVIEW

Héctor García Estrada[1], Omar Alejandro Linares Escobar[1], Angelo Pastrana Manzanero[1], Luis Carlos Martínez Ruiz[1], María Guadalupe Ramírez Sotelo[2], Agustín Ignacio Cabrera Llanos[1]

[1] Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Biotecnología, Departamento de Bioprocesos, Ciudad de México, México
[2] Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Biotecnología, Departamento de Bioingeniería, Ciudad de México, México
aicllbuda@yahoo.com

**Resumen.** En el presente trabajo se muestra el desarrollo de un sistema de clasificación de malformaciones pédicas usando procesamiento digital de imágenes en LabVIEW. El sistema funciona en tres etapas: captura de las imágenes del pie, procesamiento de las imágenes y clasificación. Para la toma de las imágenes se utilizaron webcams colocadas en la base de un andador de hierro con superficie de acrílico, de manera que se obtiene la planta del pie. Posteriormente, se umbraliza la imagen adquirida usando el rango de intensidad en la zona donde se ubica la huella del pie de manera que se obtiene la superficie que hace contacto con el acrílico al momento de la pisada. A partir de la huella se obtienen las medidas de la longitud del arco y del ancho del pie, desde los cuales se calcula el porcentaje de relación entre estas mediciones. Finalmente, este porcentaje se ingresa a un sistema de lógica difusa, pudiendo clasificar las deformaciones en pie plano, normal, cavo, cavo fuerte y cavo extremo. Este sistema disminuye el tiempo de respuesta para clasificación de la deformación pédica por parte del especialista y permite el desarrollo de un sistema de apoyo al diagnóstico de la malformación.

**Palabras clave:** LabVIEW, procesamiento digital de imágenes, lógica difusa, diagnóstico por imagen, malformación pédica.

# Classification System of Pedic Deformations Using Digital Image Processing and Fuzzy Logic in LabVIEW

**Abstract.** The present Project shows the development of a classification system for foot malformation using digital image processing in LabVIEW. The system Works in three stages: acquisition of the foot image, image processing and classification. For the image capture were used webcams at the base of an iron walker with acrylic surface, so that the sole of the foot is obtained. Later, the acquired image become binary by using thresholding by way of obtain the sole surface that makes contact with the acrylic surface at the moment of the footstep.

From the footprint is obtain the measure of arc long and foot width from which the percentage of relationship between these measurements is calculated. Finally, this percentage enters to a fuzzy logic system being able to classify the deformations in flat foot, normal, cavus foot, strong cavus foot and high cavus foot. This system decreases the response time for classify the foot malformations given by the specialist and allows the development of a support system for malformation diagnostic.

**Keywords:** LabVIEW, digital image processing, fuzzy logic, diagnostic imaging, pedic malformation.

# 1. Introducción

## 1.1. Malformaciones pédicas

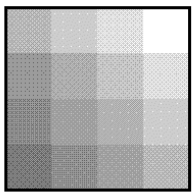
Una malformación es una anomalía morfológica, generalmente de origen congénito. En el caso de las malformaciones pédicas, estas se dan en la planta del pie, provocando que la huella se deforme alterando su biomecánica [1].

Las malformaciones se pueden clasificar por la forma en la que se aprecia la planta del pie, siendo las más comunes pie plano y pie cavo (Fig. 1). El pie plano se caracteriza por la ausencia de arco y una huella uniforme; el cavo presenta un arco muy pronunciado en la huella. La técnica más común para el diagnóstico de malformaciones pédicas es el fotopodograma, el cual consiste en obtener una imagen de la huella del pie para realizar mediciones desde las cuales se puede obtener una valoración por parte del especialista [2].

**Fig. 1.** Ejemplos de fotopodogramas de diferentes malformaciones.

## 1.2. Procesamiento digital de imágenes

El procesamiento digital de imágenes es una disciplina y un conjunto de técnicas para trabajar, analizar y obtener información de imágenes a través de medios electrónicos usando los valores numéricos de las matrices que componen las imágenes digitales, donde cada pixel corresponde a una posición de la matriz y cada valor numérico a una intensidad (Fig. 2) [3].



| | | | |
|---|---|---|---|
| 124 | 135 | 141 | 255 |
| 110 | 120 | 128 | 141 |
| 108 | 115 | 120 | 135 |
| 98 | 110 | 119 | 128 |

**Fig. 2.** Comparación de imagen digital con su imagen correspondiente.

Las técnicas del procesamiento digital de imágenes son muy variadas y permiten analizar diferentes tipos de información en las imágenes pasando de la determinación de un color a la detección de objetos [4].

Para el presente trabajo se utilizó el umbralado, en esta técnica se toma en cuenta un intervalo de tonalidad, los pixeles que tengan su valor dentro del intervalo se les asigna el valor más alto de luminosidad mientras que a los que se encuentren fuera se les asigna 0; quedando únicamente las formas que se encuentren en los intervalos.

### 1.3. Lógica difusa

La lógica *fuzzy* o lógica difusa es una técnica que permite simular el proceso del pensamiento humano a partir de incertidumbres o juicios de valor, por ejemplo "el sol es brillante" o "la noche es oscura". Esto lo logra utilizando como referencia sistemas basados en el conocimiento.

A diferencia de la lógica Booleana que considera la pertenencia absoluta o la no pertenencia de la variable a analizar con un conjunto, la lógica difusa toma en cuenta valores de pertenencia a dicho conjunto; por ejemplo: una persona que mide 1.70 en lógica booleana únicamente puede ser clasificada como alta o no alta, en cambio con lógica difusa la misma persona tiene un porcentaje de pertenencia a alta. Adicionalmente, la lógica difusa permite hacer relación de variables lingüísticas como se observó en el caso anterior con la palabra alto.

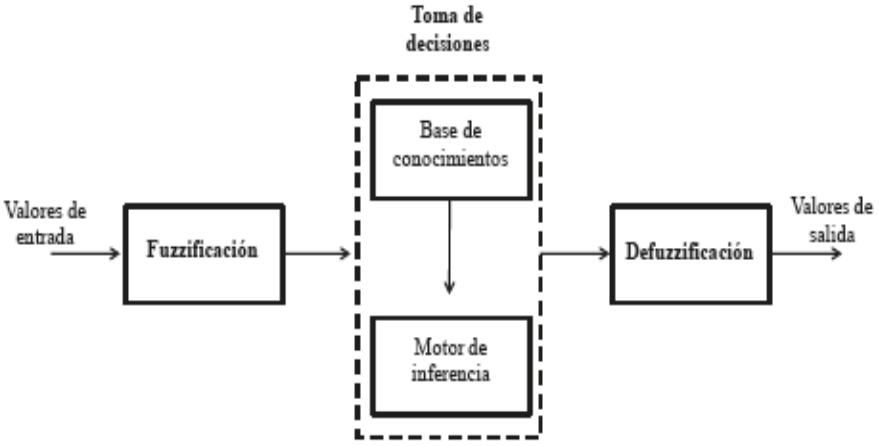



**Fig. 3.** Diagrama de bloques de un sistema de lógica difusa.

Para lograr los valores de pertenencia, se utilizan variables de membresía, las cuales relacionan los valores que puede tener la variable con la pertenencia al conjunto. A partir de la función de membresía la variable pasa por un proceso de *fuzzificación* por el cual se obtiene el valor de pertenencia dentro de las funciones a las que llegue a pertenecer. Posteriormente se evalúa la entrada a partir de un conjunto de reglas que define el usuario, el resultado de esta evaluación pasa por un proceso de *defuzzificación* y se obtiene la salida del sistema (Fig. 3) [5].

## 2. Metodología

Este trabajo se desarrolló en tres etapas: captura de las imágenes, procesamiento de las imágenes y clasificación de la malformación. A continuación, se desarrollan cada una de ellas.

### 2.1. Captura de la imagen

Para capturar las imágenes se utilizaron cámaras web Logitech C170, con una resolución de 5 megapixeles; estas cámaras se colocaron en un andador construido con ángulo de hierro y plataforma de acrílico, posicionadas de manera que adquirieran la imagen de la planta de los pies del paciente que suba al dispositivo (Fig. 4).

### 2.2.

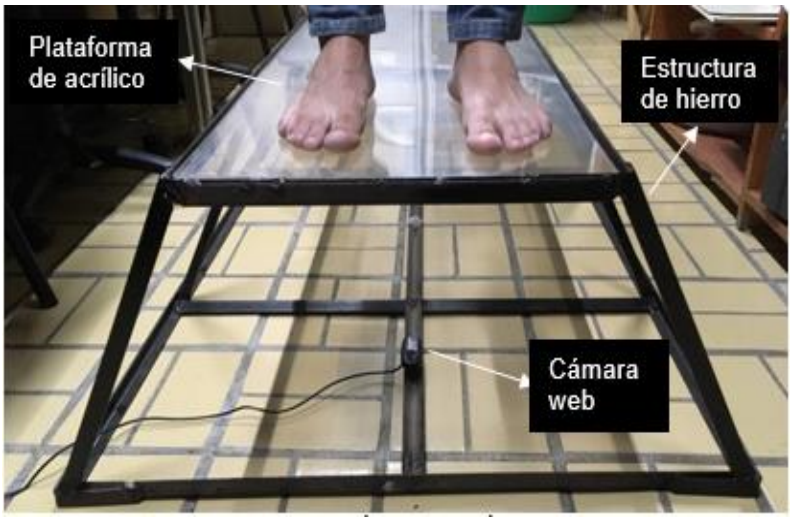



**Fig. 4.** Dispositivo para capturar fotografías de la planta del pie.

El control de las cámaras se realizó por medio de LabVIEW, configurando la cámara para capturar la fotografía en blanco y negro con una resolución de 8 bits (Fig. 5).

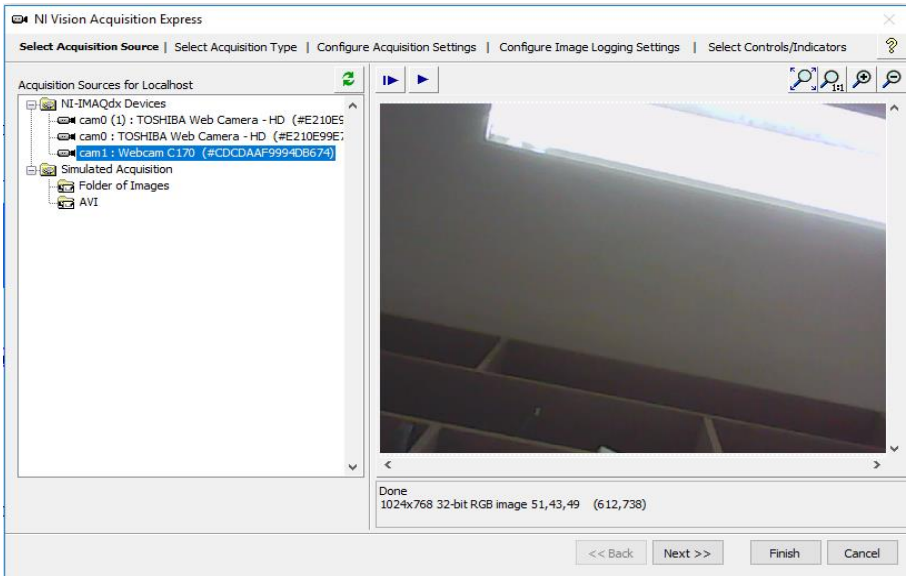



**Fig. 5.** Configuración de las webcams.

## 2.3. Procesamiento de la imagen

Después de obtener la imagen se le aplica un filtro de suavizado gaussiano para eliminar cualquier ruido generado en la captura de la imagen, este filtro es de comportamiento lineal, utilizando para éste una máscara [0, 1, 0; 1, 4, 1;0, 1, 0], adicionalmente se obtiene el histograma de la imagen para poder visualizar la variación de tonalidades en ella.

A partir del histograma se define el intervalo de intensidades de interés y se aplica umbralado por *threshold* permitiendo mantener la eliminación de los valores por arriba y abajo del umbral delimitando la planta del pie del resto de la imagen. La selección de la ventana de umbral es manual, a partir del histograma de la sección a umbralizar, se toma una media y se da una tolerancia

El proceso de umbralado se aplica siguiendo la ecuación 1, donde *g(x,y)* es la función de intensidad de la imagen umbralada, *f(x,y)* la función de intensidad de la imagen original y el intervalo *[a,b]* como los valores de intensidad en los que se delimita la información de interés:

$$g(x,y) = \begin{cases} 255 & a \leq f(x,y) \leq b, \\ 0 & c.o.c. \end{cases} \quad (1)$$

## 2.4. Clasificación de la malformación

Para empezar con la clasificación de la malformación se deben obtener medidas del pie, esto se logró mediante la toma de líneas rectas limitando el ángulo de detección en 80 a 110 grados evitando el contorno lateral del pie. Se genera una línea para el empeine

y otra para el arco. Una vez que se obtienen las líneas se calcula su longitud mediante la ecuación de distancia entre dos puntos y se utiliza la ecuación 2, donde *Le* es la longitud del empeine y *La* la del arco para encontrar la relación entre las distancias antes mencionadas. Debido a que esta es un porcentaje, la distancia se obtiene en pixeles sin necesidad de convertir a otro sistema de medición:

$$R = \frac{Le-La}{Le} x\ 100. \quad (2)$$

La relación obtenida es ingresada a un sistema de lógica difusa donde se clasifica el estado de malformación del pie. Para el diseño de las funciones de membresía se utilizó información de una guía de diagnóstico por podometría [2]. Se diseñaron siete funciones de entrada y cinco de salida a partir de esta información. Las funciones de salida se definieron como: plano, normal, cavo, cavo fuerte y cavo extremo (Fig. 6).

El proceso de *defuzzificación* se llevó a cabo mediante la técnica del centroide o centro del área. Esta técnica considera el área de las funciones de membresía con las que se empata la variable a analizar por las reglas del sistema difuso y las considera una sola. A partir de esto calcula el centro del área total obtenida, este centroide será la salida del sistema.

Se usaron siete reglas, siendo la respuesta del sistema la siguiente (Fig. 7) teniendo un valor cuantitativo de la malformación del pie.

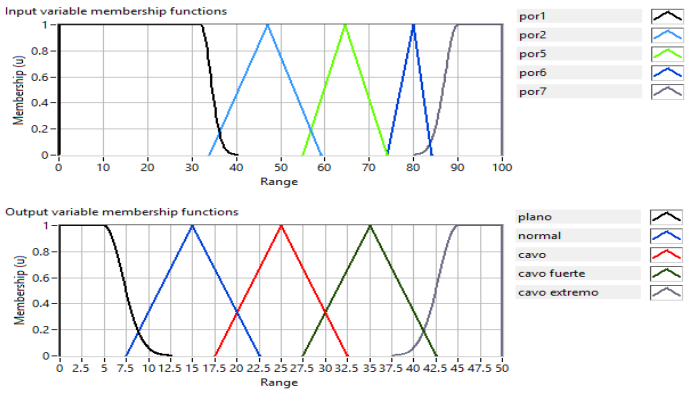



**Fig. 6.** Funciones de membresía del sistema de clasificación.

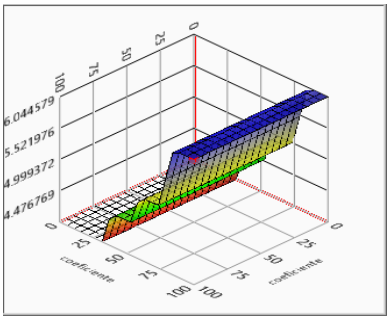



**Fig. 7.** Respuesta del sistema de lógica difusa.

## 3. Resultados

Se desarrolló un instrumento virtual para la clasificación de malformación pédicas en LabVIEW (Fig. 8) [6]. En éste se despliegan la imagen obtenida por la cámara, la imagen umbralizada, el histograma, la relación entre las medidas, la salida del fuzzy y la clasificación de la deformación.

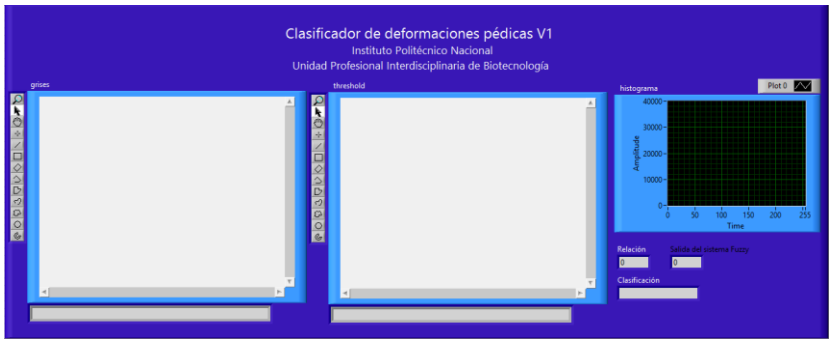



**Fig. 8.** Panel frontal del instrumento virtual.

Se obtiene la imagen mediante la cámara y se convierte de color a escala de grises (Fig. 9).

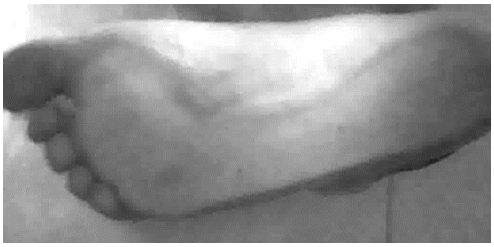


**Fig. 9.** Fotografía obtenida con el dispositivo de captura.

La imagen umbralizada se muestra a un costado, en esta se procura mantener la huella del pie (Fig. 10).

**Fig. 10.** Obtención de la huella mediante el umbralado.

Finalmente se muestra en el panel frontal las líneas generadas para realizar la clasificación, el porcentaje de relación, el resultado del sistema Fuzzy y el tipo de malformación de la malformación obtenida (Fig. 11).


**Fig. 11.** Interfaz en funcionamiento.

Se realizaron pruebas con 10 sujetos diferentes con diagnóstico previo; cuatro de ellos sin malformación, dos con pie cavo y el restante con pie plano. Se muestran los resultados en la tabla 1 [7].

**Tabla 1.** Resultados de pruebas realizadas con el clasificador.

| Diagnóstico previo | Resultado del Programa | % de relación | Salida del Fuzzy |
|---|---|---|---|
| Pie normal | Pie normal | 56.41 | 19.04 |
| Pie normal | Pie normal | 52.76 | 14.99 |
| Pie normal | Pie normal | 50.76 | 15.01 |
| Pie normal | Pie normal | 52.86 | 14.98 |
| Pie cavo | Pie cavo | 66.51 | 25.04 |
| Pie cavo | Pie cavo | 72.69 | 25.12 |
| Pie cavo | Pie cavo fuerte | 82.07 | 35.32 |
| Ausencia de arco | Pie cavo extremo | 100 | 46.04 |
| Pie plano | Pie plano | 10.53 | 3.95 |
| Pie plano | Pie plano | 10.53 | 3.95 |

## 4. Conclusiones

Se logró un sistema de clasificación de malformaciones pédicas, el cual selecciona entre cinco categorías ampliando el rango de clasificación con respecto a sistemas tradicionales. La resolución de las cámaras utilizadas fue suficiente para este sistema. El umbralado de la imagen ayuda al análisis al eliminar la información no deseada permitiendo el trazo de las líneas para la obtención de las medidas fundamentales. La determinación de las medidas fundamentales por medio de funciones lineales cumplió con las expectativas de la clasificación. El resultado de la clasificación se encuentra dentro de las variables de membresía del sistema Fuzzy para realizar la clasificación. La implementación del sistema de clasificación en un soporte permite que se expanda

el sistema para realizar análisis de la marcha. Todo esto, facilita el proceso de diagnóstico por parte del especialista.

## Referencias

1. Silverman, M.: Ortopedia y Traumatología. Médica Panamericana. Argentina (2010)
2. Barrera, R., Siles, J. A., Concepción, L.: Aplicación didáctica para la valoración de un fotopodograma en las clases de educación física. Revista digital efdeportes No. 141, Argentina (2010)
3. Ready, S., Kwon K.: Practical guide to machine vision software. Wiley VCH. Singapur (2015)
4. Grimson, W. E. L., Huttenlocher, D. P.: Object recognition by computer: the role of geometric constraints (2015)
5. Ponce, P.: Inteligencia Artificial con Aplicaciones a la Ingeniería. Alfaomega, México (2010)
6. Lajara, J.R., Pelegri, J.: LabVIEW entorno gráfico de programación. Marcombo, España (2011)
7. Sanchez-Velarde, E., Sotelo de Avila, A. A., Cabrera-Llanos, I. A.: Clasificador fuzzy-triage en sala de urgencias, aplicando máquina de estados en LabView. En: Memorias del VI Congreso Nacional de Tecnología Aplicada a Ciencias de la Salud, México: INAOE. (2015)

ISSN 1870-4069

# Differential Neural Network Online for Identification of an Electrocardiographic Signal

Héctor García Estrada[1], Karen Jazmín Mendoza Bautista[1], Angelo Pastrana Manzanero[1], Omar Alejandro Linares Escobar[1], María Guadalupe Ramírez Sotelo[2], Agustín Ignacio Cabrera Llanos[1]

[1] Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Biotecnología, Departamento de Bioprocesos, Mexico City, Mexico

[2] Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Biotecnología, Departamento de Bioingeniería, Mexico City, Mexico

aicllbuda@yahoo.com

**Abstract**. This work shows the development of a Differential Neural Network (DNN) applied to an online processing to signal from a 12-lead electrocardiograph. This system was divided in three stages: Acquisition of the ECG signal, design and tracking of the DNN. Firstly, a 12-lead electrocardiograph with a hybrid processing was constructed. Acquiring 8 leads with the USB DAQ-6009 and calculating the other four by programming. Then, based on the theory of Differential Neural Networks, the algorithm is developed, determining the values of the network by trial and error tests. To finally visualize the tracking of the network through our virtual instrument. The use of this technique is very common in the analysis of complex nonlinear dynamical systems in engineering and medicine areas, showed an excellent performance in the description of electrocardiographic biopotentials developed in this work.

**Keywords:** dynamic neural networks (DNN), electrocardiogram, LabVIEW, USB DAQ-6009, tracking of a biopotential.

## 1 Introduction

On the basis of Static Neural Networks (SNN) capability to approximate any nonlinear continuous function, a natural extension is to approximate the input-output behavior of nonlinear systems by Differential Neural Networks (DNN): their information process is described by diff erential equations for continuous time or by diff erence equations for discrete time. The existing results about this extension require quite restrictive conditions such as: an open loop stability or a time belonging to a close set.

This type of networks was developed based on the Lyapunov stability approach for the development of the laws of learning [1], these have had successful applications in fields such as biotechnology when estimating variables in a fermentation process [2], using observation schemes and using a technique that has been called virtual sensor [3]; in the estimation of drug doses for cancer, by generating a control signal in the dosage in such a way that the growth of cancer cells is not large [4].

Tracking is a phase of great importance and interest for the analysis and control of the process or systems, since we can, among other things, identify the parameters of the

structure of the system to be studied and thus be able to design a model that behaves in a very similar to the system in question.

### 1.1 Electrocardiograph

An electrocardiograph is an electrical device that captures the electrical potentials of the heart by recording the voltage it generates and transmits through the body through a system of electrodes, cables and a recording console. These electrodes placed in specific parts of the body, are responsible for detecting cardiac depolarizations.

The heart cannot be seen from a single place, since it is a three-dimensional organ, so it must be from different places to be able to assess the electrical activity. Depending on the place where the explorer or derivative electrode is placed, they will be the electrical characteristics that we will appreciate.

The standard electrocardiogram consists of 12 leads, which can be divided into:

- Monopolar (aVR, aVL, aVF, V1 to V6)
- Bipolar (DI, DII and DIII)

The monopolar derivations aVR, aVL and aVF, arose in 1942, when Goldberger observed that the signals of Wilson could be increased (since a 50% increase in the value of the detected signal is observed) if the average of the other points involved was taken. On the other hand, leads V1 to V6 complete the information needed to study the heart on almost all sides that provide information on how cardiac depolarization occurs according to what was previous mentioned.

The bipolar derivations, (also called standard) constitute a closed circuit (Kirchhoff's Law) and they comply with a law called Einthoven, which says that DII = DI + DIII, and is used to verify the correct placement of the cables [5].

### 1.2 Differential Neural Networks (DNN)

In the eighties J. Hopfield proposed the principle of operation of a stable recurrent network, which consists of a group of neurons where the output of each neuron serves as feedback at the entrances, except for its own entrance. This network arises from the theory of geometric control based on differential geometry [6].

The training process is an iterative process in which the input signals are applied, and the output is calculated; the process is repeated until the output signal is constant, which is when the network is said to be a stable network. In case that the output is not a constant output and is a variable output, you have an unstable network.

The continuous time Hopfield Neural Network or, on our terminology, the Differential Neural Networks, can be described by an electric circuit, which is based on a RC network connecting nonlinear amplifiers.

### 1.3 Data Acquisition Device

The device used to acquire data online is the NI USB DAQ-6009 card, which has a performance and sampling rate suitable for a certain number of applications that include

control and automation of systems. Due to the use of this device, it was used to implement the software of graphic programming known as LabVIEW both company National Instrument. Same software has a group of exclusive components for this device, in which the user can acquire or generate various signals, which can be analog or digital [7].

## 2 Metodology

### 2.1 Signal Acquisition

For the acquisition of the signal it was necessary, firstly, the adequacy of the same, so it was designed as a 12-lead signal electrocardiograph. For this, an amplification was performed with the necessary gain to be acquired by the card, for which the AD620 instrumentation amplifier was used due to its characteristics of the low noise, low input bias current, high CMRR, and low power, this make it well suited for medical applications, such as ECG and noninvasive blood pressure monitors.

Implemented circuit shown in Figure 1. In it you can see the connections of the right arm, left arm and left leg in three of the nine op amp's placed following Einthoven's relationship, as well as also the precordial and corresponding arrangement for increased referrals.

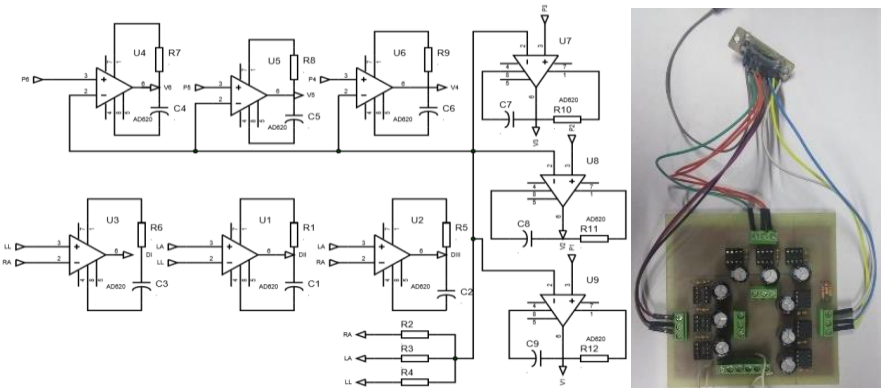



**Fig. 1.** Schematic and real circuit PCB of the 12-lead electrocardiograph.

The signal is acquired through a series of suction electrodes placed following the AHA system, these electrodes are chosen based on the use of op amps with high input impedance characteristics, it is possible to reduce the effect of polarization, thus avoiding a electric flow which modifies the potential of half cell electrodes, causing distortions in the recorded signal [8]. These signals are transmitted from the patient to the circuit by means of a patient cable to ECG, presented in Figure 2. It is worth mentioning that all power is carried out by a series of connected batteries which gave us the voltage and the electric flow needed to feed the op amps, making the portable system free of any electrical hazard for the patient [9].

**Fig. 2.** Patient cable to ECG register.

On the other hand, this test was performed at rest, which was considered to belong to the same frequency range between 0.5 to 150 Hz, so each of the eight lines were subjected to 120 Hz analog filtering through an RC array in the gain resistance of the instrumentation amplifier.

This arrangement modified the gain equation present in the datasheet of the AD620. Causing that the amplifier also acted as a band pass filter, when the frequency goes to zero the gain will be 1, and when it goes to infinity the gain will be approximately 2. Both gains are despicable when compared with the design gain, this equation is shown in (1):

$$G = 1 + \frac{49.4\ k\Omega Cs}{R_G Cs + 1}. \tag{1}$$

Eight circuits like this were implemented, to acquire the monopolar branches V1 to V6 and bipolar DI and DII. Then the signals are digitized by means of the USB DAQ-6009 card using its eight analog channels. To facilitate the monitoring part of the neural network, high frequency noise was eliminated by configuring the inputs with a sampling frequency of 250 Hz as shown in Figure 3.

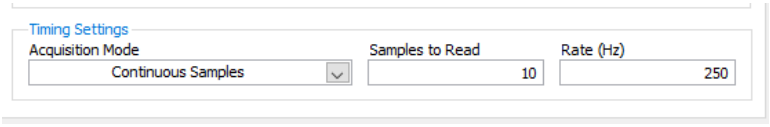



**Fig. 3.** DAQ-6009 configuration.

Finally, the signals are subjected to a digital filtering process using infinite response to the impulse filters (IIR) to the fourth order impulse, which are: high-pass at 0.5 Hz, low-pass at 120 Hz and a reject band at 60 Hz. To obtain the 12 derivations from 8 of them, we proceeded to calculate the rest. The derivation DIII was obtained following the relationship between the bipolar derivations, applying the law of Kirchoff voltages obtaining equation (2):

$$DIII = DII - DI. \tag{2}$$

And for the increased derivations the analysis of the connections for each one of the derivations was carried out, arriving at equations (3-5):

$$aVR = \frac{-(DI+DII)}{2}, \tag{3}$$

$$aVL = \frac{(DI - DIII)}{2}, \tag{4}$$

$$aVF = \frac{(DII + DIII)}{2}. \tag{5}$$

### 2.2 Dynamic Neural Network Design

Unfortunately, when carrying out the identification, it was difficult to know if it had the external measurements that can be obtained from the system, components due to external or parametric disturbances were presented, this makes the identification process difficult, therefore, the use of the DNN helped us overcome these difficulties. Below is a description of the process of creating the tracking system based on a DNN.

In the tracking of a system, it is assumed that both the input and output values can be measured, so they are a vital part of network training, following the ideas proposed in [10], the mathematical description of the network is given by a differential equation (6):

$$\dot{\hat{x}}_t = A\hat{x}_t + W_{1,t}\sigma(\hat{x}_t) + W_{2,t}\phi(\hat{x}_t)u_t. \tag{6}$$

where $\hat{x}_t \in \Re^n$ is the state of the neural network, $u_t \in \Re^q$ the input from the model, $W_{1,t} \in \Re^{n \times k}$ the weight matrix of the feedback state layer, $W_{2,t} \in \Re^{n \times r}$ the weight matrix of the input layer, $A \in \Re^{n \times n}$ a stable matrix selected by trial and error tests, the activation functions $\sigma(\hat{x}_t)$ and $\phi(\hat{x}_t)$ are composed of sigmoid functions in each of these equations (7-8):

$$\sigma(\hat{x}_t) = a_1\left(1+e^{-a_2\hat{x}_t}\right)^{-1} - a_3, \tag{7}$$

$$\phi(\hat{x}_t) = b_1\left(1+e^{-b_2\hat{x}_t}\right)^{-1} - b_3. \tag{8}$$

Defining $\Delta t$ as the error between the states of the original system and the states generated by the DNN considered. The DNN weights are adjusted by a set of differential matrix equations, which are equations (9-10):

$$\frac{dW_{1,t}}{dt} = -K_1 P \Delta_t \sigma^T(\hat{x}_t), \tag{9}$$

$$\frac{dW_{2,t}}{dt} = -K_2 P \Delta_t \phi^T(\hat{x}_t) u_t{}^T. \tag{10}$$

Where $K_1$ and $K_2$ are constants by means of which the adjustment of the weights is carried out and must be selected by the trial and error method. $P$ is a positive definite matrix that provides the solution of an algebraic Riccati equation (11) described by:

$$A^T P + PA + PRP + Q = 0. \tag{11}$$

where the following facts are required to be met:

- There is a positive definite matrix $Q_0$ such that the Riccati equation which has a positive solution $P = P^T > 0$

– It is also required that the *R* and *Q* matrices are described by equations (12-13):

$$R = \Lambda_f^{-1} + W_{l,t}^{*} \Lambda_{\sigma}^{-1} (W_{1,t}^{*})^{T} + W_{2,t}^{*} \Lambda_{\phi}^{-1} (W_{2,t}^{*})^{T} + K_1 \Lambda_1^{-1} K_1^{T} + K_2 \Lambda_2^{-1} K_2^{T} + \Lambda_{\xi 1}^{-1}, \quad (12)$$

$$Q = D_{\sigma} + v_0 D_{\phi} + \Lambda_1 + Q_0. \quad (13)$$

### 2.3 Tracking of the Network

For the operation of the neural network, the algorithm described in equations 6 to 10 was introduced within a control loop by means of the LabVIEW module called *"Control Design and Simulation System"*. As part of the design of the virtual instrument, a selector was use within the algorithm, allowing the user to choose one of the 12 derivations obtained, which enters in the DNN algorithm, thus visualizing the chosen derivation, the tracking of the network, and the error of the same.

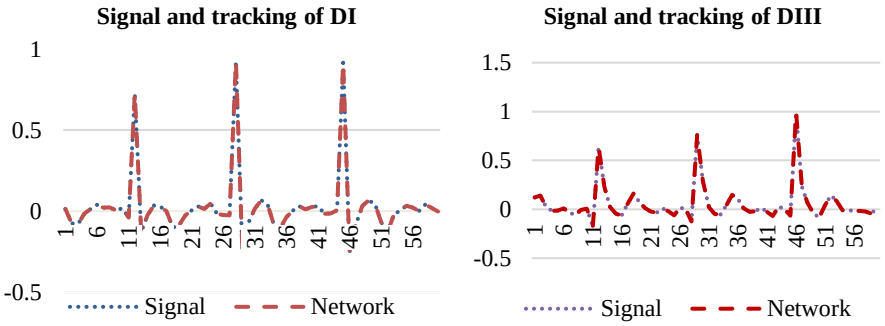
## 3 Results

The evaluation of the monitoring of the network can be carried out by placing the prototype of electrocardiograph on the patient, the inclusion of the acquisition card to the computer next to the above-mentioned electrocardiograph and the deployment of the virtual instrument. The patient can view one of the twelve calculated derivations and you will see the corresponding monitoring network as well as the dynamics of the error over the training of the same time.

For example, the DI and DIII connections were followed in an apparently healthy 22-year-old patient, observing the results through the front panel of the program developed in LabVIEW, as shown in Figure 3.


**Fig. 3.** Front panel of the program in operation.

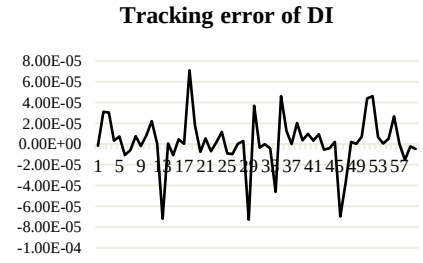
The signal and the monitoring of the network are shown in the Graphs 1 and 2, where the ECG signal is represented by the dotted line and the network behavior by the hyphen line. As we can see, a total splicing is observed between both, taking as an example two selected derivation.



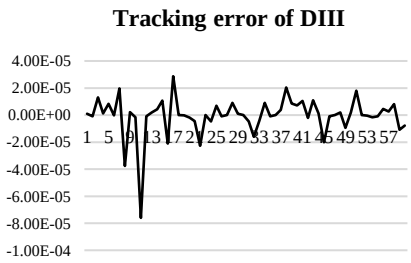

**Graphs 1, 2.** The signal (*dotted line*) and the monitoring of the network (*hyphen line*) in DI and DII.

As we can see in the Graphs 1 and 2, it is impossible to detect the difference between the real values of the signal and the values generated by the network. In both you can see how the network manages to reproduce very well the smooth and abrupt variations in the ECG signal.

Now in the Graphs 3 and 4, the calculation of the error between the values of the ECG signal of the values generated by the DNN is shown. This indicates a good performance of the network because it can reproduce the signal of the derivations shown with errors of approximately ± 0.0001. From this we can say that the use of DNN is an excellent alternative for tracking of electrocardiographic signal.




**Graph 3.** Error of the network in the tracking of DI.




**Graph 4.** Error of the network in the tracking of DIII.

## 4 Conclusions

In the present paper, a tracking system that uses differential neural networks on line applied to biological signal particularly the ECG obtained from a healthy patient has shown a good performance. In these signals the error obtained by the difference

between the DNN generate signal and the ECG measure signal is approached asymptotically to zero. The tracking process is shown, and the error signal value less to $10^{-4}$ units was obtained. The methodology based on the DNN, showed that even with abrupt changes, such as those experienced between each peak of the signal, it was possible to obtain almost immediately and precisely such variations, although these were of very short duration. It is considered as future work to expand the functionality of the network to convert it into a prediction system, considering the corresponding changes in the network.

## References

1. Poznyak, A.S., Sánchez, E.N., Yu, W.: Differential Neural Networks for Robust Nonlinear Control. World Scientific, pp. 215–251 (2000)
2. Cabrera, A.I., Poznyak, A.S., Poznyak, T., Aranda, J.S.: Identification of a Fedbatch Fermentation Process: Computational and Laboratory Experiments. In: Bioprocess and Biosystems Engineering. Springer. vol. 24, No. 5, New York: Clarendon, pp. 319–327 (2002)
3. Cabrera, A.I., Aranda, J.S.: Estimating the Trehalose Cytoplasmatic Content during a baker's yeast. In: 10th International Symposyum on Computer Application on Biotechnology. Cancún, México, IFAC (2007)
4. Aguilar, N.C., Chairez, I.: Neuro Tracking Control for Immunotherapy Cancer Treatment. In: IJCNN '06 International Joint Conference on Neural Networks. IEEE, Vancouver, BC, 5316–5323 (2006)
5. Kligfield, P., Gettes, L.S., Bailey, J.J., Childers, R., Deal, B.J., Hancock, E.W., Mirvis, D.M.: Recommendations for the Standardization and Interpretation of the Electrocardiogram. Journal of the American College of Cardiology, pp. 1109–1127 (2007)
6. Ponce, P.: Inteligencia Artificial con Aplicaciones a la Ingeniería. Alfaomega, pp. 238–242 (2010)
7. National Instruments. Manuals. Retrieved from National Instruments: www.ni.com/pdf/manuals/371303n.pdf (2005)
8. García, M.T., Jiménez, A., Ortiz, M.R., Peña, M.A.: Potenciales bioeléctricos: origen y registro. Universidad Autónoma Metropolitana, Unidad Iztapalapa, pp. 65–79 (1997)
9. Webster, J.G., Neuman, M.R., Olson, W.H.: Medical Instrumentation: Application and Design. 4th Edition. John Wiley & Sons, Inc. pp. 638–674 (2010)
10. Cabrera, A.I., Ramírez, M.G., Galvez, G.: Independent Neuro-Fuzzy Control System. Elsevier, pp. 237–242 (2005)

ISSN 1870-4069

# Vehicle Recognition and Classification Model by Digital Accelerometer

Marco Antonio Jasso-Juárez[1], Ignacio Hernández-Bautista[2], Juan José Carbajal-Hernández[3], Juan Francisco Mosiño[1], Raúl Santiago-Montero[1]

[1] Instituto Tecnológico de León, División de Estudios de Posgrado e Investigación, León, Guanajuato, Mexico

[2] Cátedra CONACYT - Instituto Tecnológico de León, División de Estudios de Posgrado e Investigación, León, Guanajuato, Mexico

[3] Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico City, Mexico

{marco.a.jasso.j, jfmosino}@gmail.com; ihernandezb@conacyt.mx; jcarbajalh@cic.ipn.mx; raul.santiago@itleon.edu.mx

**Abstract.** Currently, the pollution by urban vibration is a growing problem in big cities. This work proposes an analysis system of vibrations generate by automobiles over urban avenues. By digital processing of signals from measurements on the ground, a computational method that allows identifying the kind of automobiles is presented. Experimental results show evidence of the good functioning of the system, providing a tool that allows studies to go deeper into the problems generated by vibrations on communication routes, as well as its effect on nearby buildings.

**Keywords:** accelerometer, vibrations, pattern recognition, vehicle classification, artificial neural network.

## 1 Introduction

Statistical studies are used as a likely tool for the evaluation and generation of strategies vehicular flow analysis. The main objective of these studies is counting frequency and determining the kind of vehicle that passed on a track [1]. Although, counting vehicles seems an easy work, actually it is an irritating and complicating task if the vehicular flow is large. For this reason, actual technologies started to use inductive loop sensors; these being the most commonly used since the1960´s [1].

For several years, different types of sensors have been used for vehicle detection [2, 3, 4] and different techniques based on artificial intelligent were implemented for vehicle classification [1, 5, 6]. This permits knowing with details for each class about the data of traffic sensors, which can be classified as intrusive and non-intrusive, where the latest have been installed over the road surface, and which have been more popular and recurrent due to their easy maintenance [4].

In recent years, applications of vibrations in the engineering field have motivated research on machine design, foundations, structures, engines, turbines and control systems [7]. According to this, accelerometers are the most common sensors used for measuring mechanical vibrations. Those vibration can be propagated from vehicles to ground surface [5, 6, 8].

Vehicles vibrations are different between classes and their characteristics can be modeled by stochastic studies, where some are associated with a specific vehicle [9]. Recent systems have incorporated other kinds of sensors [6] or used anticipated seismic studies as compensation [5]; both involved vehicle classification. However, vibrations analysis for vehicles analysis has not been properly explored. This study is a way of understanding the pollution made by released energy automotive vibrations, and leads to understanding what kind of vibrations have a bigger affectation on surfaces subjected to these forces.

This work proposes a computational model for assessment of vehicle vibrations through the use of digital accelerometers. A preprocessing step is performed in order to create a normalized database. A feedforward artificial neuronal network (ANN) is used for recognizing different types of vehicles according to measured signal vibrations. Our results reveal significant differences between small and heavy vehicles.

## 2 About Vibrations and Vehicle Classification

### 2.1 Vehicle Classification

Human activities involve vibration in different ways; e.g., we can hear or see because our eardrums vibrate or light waves vibrate. Breathing is associated lungs vibration and walk involve an oscillating movement (periodic) of legs and hands [7]. However, surface vibrations must be carefully monitored, because when they are generated by vehicular traffic, they go through the buildings where the highest amplitudes will considerably affect their structure; but the oscillations are insufficient for cause damages. Also, vibrations can affect people and even lower levels can damage the sensitivity of laboratory equipment or affect the manufacturing of micro electric circuits [9].

In works proposed by [5] and [6], their authors developed different methods for vehicle recognitions using vibrations. The first fits data vibrations by seismic studies of analyzed road. The second used an accelerometer and another type of sensor to compensate vehicle classification. Finally, both used ANN as classifier methods for better results.

It is important to remark, that diverse ways to classify a vehicle without using vibrations was determined; e. g., to determine the kind of a vehicle in a dynamic way, sensors were implemented for obtaining the main characteristics and depending on the approach, different kinds of data is collected [10]. Interruption of current using inductive sensors can be used, where disturbances are measured by the vehicle passage, supported by ANN for a more accurate classification [1]. Another case, is the coalition process of Nordic countries through the project NorSIKT [2], where they used several sensors to make a vehicle classification, obtaining as a result five different classes.

## 2.2 Artificial Neural Networks (ANN)

An Artificial Neural Network is a model that is useful for recognizing patterns when a specific target is associated. The basic model [11] that represent an ANN is shown in Fig. 1. The ANN inputs can be represented by the following expression:

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}, \tag{1}$$

where $j = 1,\ldots,M$ and $M$ (shown in Fig. 1) is a linear combination of the input variable $x_1,\ldots, x_D$, and the superscript "(1)" indicates that the corresponding parameters is in the first 'layer' of the network. We shall refer to the parameter $w_{ji}^{(1)}$ as *weights*, parameter $w_{j0}^{(1)}$ as *biases* and the quantities $a_j$ are known as *activations*. Each is then transformed using a differentiable, nonlinear activation function $h(\cdot)$ to give:

$$z_i = h(a_j). \tag{2}$$

This represents the *hidden units* and where the nonlinear function $h(\cdot)$ is generally chosen to be a sigmoidal function. All these values are combined to give an *output unit activation:*

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}, \tag{3}$$

where $k = 1,\ldots,K$ and $K$ is the total number of outputs. This transformation corresponds to the second layer of the network, and again $w_{k0}^{(2)}$ are the bias parameter. Finally, the output unit activations are transformed using an appropriate activation function to give a set of network outputs $y_k$ as follows:

$$y_k = \sigma(a_k). \tag{4}$$

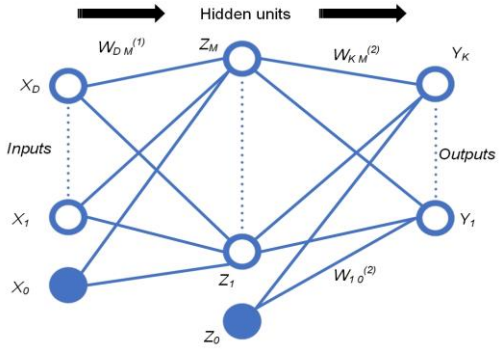



**Fig. 1.** The scheme that represents a neural network is shown. The hidden, and output variables are represented by nodes, and the weight parameters are represented by links between the nodes. Bias parameters are denoted by links coming from additional input and hidden variables such as $x_0$ and $z_0$. Arrows denote the direction of information flow through the network during forward propagation.

# 3 Methodology and Development

## 3.1 Data Acquisition

One accelerometer was used for measuring vibrations that belong to different vehicles. The digital accelerometer MMA8451q has the capacity of detecting those vibrations in three fundamental axes *x*, *y*, *z*; however, only *z* axis measures vibrations generated by automotive vehicles in a surface level [8]. This axis will be considered for data extraction. Using 8 and 14 bits resolution, it allows an output data range between 1.56 Hz and 800 Hz. An $I^2C$ communication protocol was used with a dynamically scale of ±2g, ±4g y ±8g [12]. Arduino UNO board [13], is used as a data acquisition device because it has an $I^2C$ bus [14] and allows a quick data transfer. The system basically is confirmed by three devices which can be consulted in Fig. 2.

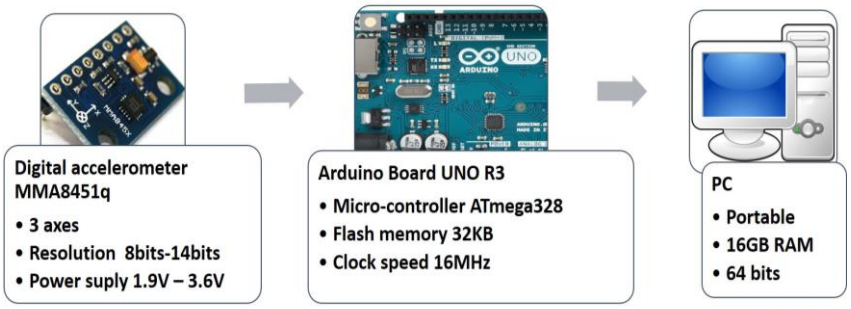



**Fig. 2.** Computational system proposed, where data information is measured, transmitted and processed.

## 3.2 Experimentation

An experimental step was designed through a test which is repeated several times. It used two vehicles of different classes: GMC Safari truck model 1993 with a weight of 2 Ton and Nissan Sentra car model 2017, with a weight of 1.2 Ton. Average speed for experiments is between 20 km/h and 30 km/h. Fig. 3 shows the distance which the test will be registered, avoiding loss data information.

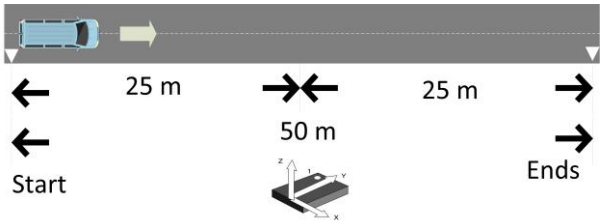



**Fig. 3.** The scheme of how to conduct a test is shown. The total distance for beginning and finishing the measurement of vibration is 50 meters, where accelerometer position is half of that distance.

### 3.3 Signal Processing

The number of tests to build the database in this experimental step was 10 for each vehicle. To validate if the test was measured correctly, signals were checked to have the shape shown in Fig. 4, where the section surrounded by an oval is the vibration generated for the vehicle passage, near to the accelerometer. The sections surrounded by a rectangle are the common behavior of accelerometer, so that, this information was ruled out and keeping only the information into the rectangle section as shown in Fig. 4.

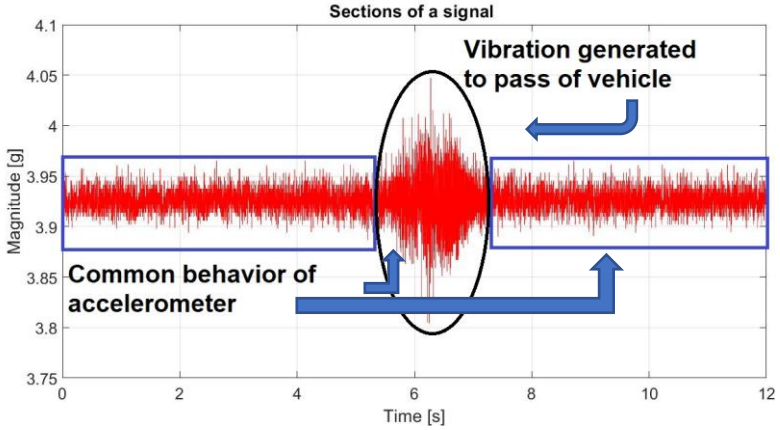



**Fig 4.** Event register. These data series have two parts, the common behavior of accelerometer and the vibration generated to pass vehicle, in this case, is a truck.

A method to separate between each class of vehicles was developed by measuring the maximum peaks of each test as shown in Fig. 5. In this way, the information is feasible to integrate the database.

For signal processing and defining a main vector, a digital pass-band FIR filter was implemented for reducing signals that generate noise, focusing on the fundamental frequency of the vehicle studied. The filter equation is described in (5), which employs a band of frequency between 100 Hz and 250 Hz. In this range, main frequencies of each kind of vehicles can be found. Then, a Hamming window was used for removing signal noise (7). Hamming is employed because after tested different windows such as rectangular, Bartlett, Von Hann, Hamming and Blackman, in different orders between 3 to 50, the proposal in a 30 order allowed the minimum attenuation without vanished the maximum peaks of signal, which are the main characteristic of the signal. This filter allowed passing characteristic frequencies of both class of vehicles according to:

$$H_d(\Omega) = \sum_{n=0}^{N-1} h_d[n] e^{-jn\Omega}, \tag{5}$$

where $H_d(\Omega)$ represents the FIR filter, $N$ is the amount of data, $h_d[n]$ is the product of the pass-band filter $h[n]$ with the Hamming window $w[n]$; $f_a$ and $f_b$ are the cut-off frequencies of the filter, $\omega$ equals the bandwidth center, and $\Omega$ is the sample frequency. The following expressions define the Hamming window filter:

$$h[n] = 2f_a \frac{\sin(n\omega_a)}{n\omega_a} - 2f_b \frac{\sin(n\omega_b)}{n\omega_b}, \tag{6}$$

$$w(n) = 0.54 - 0.46\cos\left(2\pi \frac{n}{N}\right), \quad 0 \le n < N, \tag{7}$$

$$h_d[n] = w[n]h[n]. \tag{8}$$

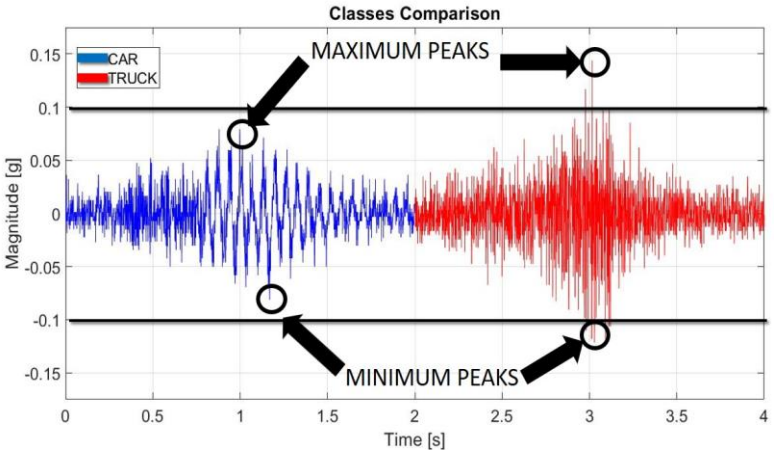



**Fig 5.** Vibrations generated by a car (blue) versus truck (red), where are pointing the maximum peaks (Top circle) and the minimum (lower circle).

After filtering the signals, the Discrete Cosine Transform (DCT) was used in order to get the average power spectrum [15], as:

$$s(i) = \frac{a(i)}{N}\left[\sum_{q=0}^{N-1} s(q)\cos\left(\pi \frac{i(2q+1)}{2N}\right)\right]^2, \quad \text{for } 0 \le i \le N-1 \tag{9}$$

$$a(i) = \begin{cases} 1, & 1 \le i \le N-1 \\ \frac{1}{\sqrt{2}}, & i = 0 \end{cases}, \tag{10}$$

where $N$ is the vector length.

The resulting vector $X$ is obtained by interpolating $S(i)$ in order to have a final length of 1000 elements [15] as follows:

$$X = F_{1000}^{-1}\{F_N\{S(i)\}\}, \tag{11}$$

where $F_M\{\cdot\}$ means the direct Fourier transform of length $N$, $F_{1000}^{-1}\{\cdot\}$ is the inverse Fourier transform of length 1000.

### 3.4 Database

Each vehicle sample is represented by a 1000 length vector. In this work, two different classes have been studied: cars and trucks. About 10 samples for each vehicle were registered, having a database with a total of 20 vectors. The exact speed is not specified in each sample, but measurements were performed within 20 to 30 km/h range. This guarantee obtained representative harmonics for studying a vehicle.

### 3.5 Feed-forward Neural Network

The purpose of database is evaluated by ANN. For this task, the network developed has 1000 inputs that represent the length of each vector. In this case, a dimensionality reduction method was not available because each element is unique. After tested different combinations of nodes (from 3 to 50) and hidden layers (from 2 to 10), it was found that maximum number configuration obtained the best results of classification (50 nodes and 10 hidden layers). As ANN output, 2 nodes are useful to determine two different classes. For the ANN training, Matlab software was used as development tool. The scaled conjugate gradient method for updating weights and bias values was used. Training automatically stops when generalization stops improving, as indicated by an increase in the cross-entropy error of validation samples; nevertheless, 1000 epochs was defined as rule stop. Finally, to evaluate the performance of the ANN, the internal procedure of the software divides the proposed database in training patterns (90%), validation (5%), and test (5%). Fig. 6 shows the scheme of the ANN developed. Black circles represent nodes, input and hidden layers; green circles represent the bias and finally, dashes represent the connection of the last hidden layer with outputs nodes.

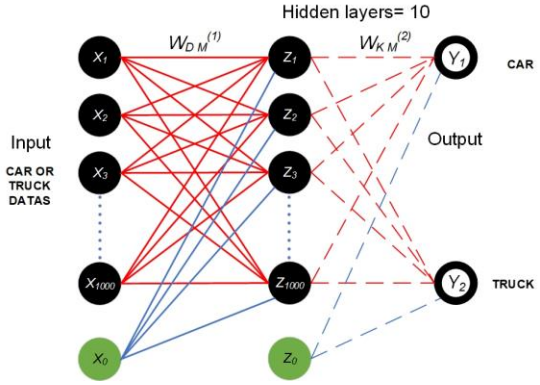



**Fig. 6.** Scheme of ANN used in the classification problem. The nodes represent inputs and hidden layers, and links represent the weights parameter. This scheme only has 2 outputs because each represents a class of vehicle analyzing in database.

## 4 Results and Discussions

Database acquisition is not an easy task; vehicle vibrations depend of driver behavior, where measurements should made for vehicles passing near the sensor and with

constant velocity. For this work, a [20 – 30] km/h speed range was chosen as a first attempt for studying this kind of vibrations.

The detection of maximum and minimum peaks allows identifying the amplitude of the highest oscillation and it can be used to define a linear separability between each class of different vehicles. Signal separability is defined by differentiating the maximum and minimum peaks of a certain vehicle class against those of the opposite class, allowing the construction of a threshold. The maximum peak is enough to determine the separability in the signal. Fig. 7 shows the vehicle signal, where 500 points was enough for showing the shape in the graphic.

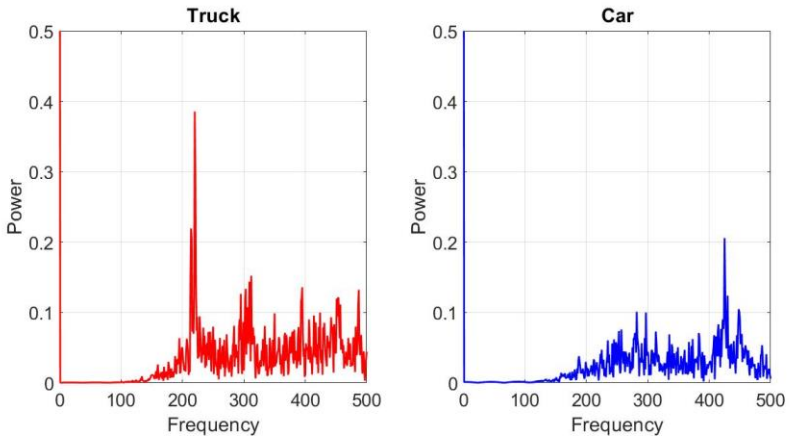



**Fig. 7.** Comparison between signals measured from a car against truck, which have been processed. Axis *y* represents power and *x* b frequency in Hz. Although signal length has 1000 points, 500 are enough for observing the signal behavior.

Table 1 shows the results obtained in the classification training, validation, test, and evaluation of the proposed database. Two different tests were evaluated; each has a different distribution in order to create different scenarios. Each test has two columns, where the left column represents the sample distribution size of the database and the right column, is the classification results expressed as the accuracy percentage. In order to have a better accuracy, 10 tests per distribution with random sample selection were performed. In this case, good classification results were obtained for each vehicle class.

**Table 1**. Classification results from experimentation between a car and a truck. Two different distributions were performed with their corresponding results. Database (Db), Classification (C).

| Tests | Distribution 1 | | Distribution 2 | |
|---|---|---|---|---|
| | % Db used | % C results | % Db used | % C results |
| Training | 80% | 83.75% | 70% | 90.72% |
| Validation | 10% | 90% | 20% | 70% |
| Testing | 10% | 65% | 10% | 77% |
| Complete Db | 100% | 82.05 | 100% | 85.5% |

As mentioned previously, current literature depends of additional information such as other sensors [6] and surface studies [5], and studies are focused on studying other vibrations characteristics. However, this works provides a model that is suitable to be improved in order to identify more vehicles or increase the effectiveness on data classification.

## 5 Conclusions

The proposed vehicle recognition system turned out to be effective to measure the vibrations and thus proving its implementation. The effective classification of recently tests using ANN allows identifying two vehicle classes and also determining the class to which they belong. A future work will be to increase the database size to build a more effective classification tool.

## References

1. Oliveira, H.A., Barbosa, F.R., Almeida, O.M., Braga, A.P.S.: A vehicle classification based on inductive loop detectors using artificial neural networks. In: 2010 9th IEEE/IAS International Conference on Industry Applications, pp. 1–6, IEEE (2010)
2. Vaa, T., Melén, P., Andersson, D., Nielsen, B.B.: NorSIKT–Nordic System for Intelligent Classification of Traffic. Procedia - Soc. Behav. Sci. 48, pp. 1702–1712 (2012)
3. Klein, L.A., Mills, M.K., Gibson, D.R.P.: Traffic Detector Handbook. U.S. Dep. Transp. Fed. Highw. Adm. II, 462 (2006)
4. Elena Mimbela Project Manager, L.Y., Klein, L.A., Kent, P.: Summary Of Vehicle Detection And Surveillance Technologies Used In Intelligent Transportation Systems. Submitted To: Federal Highway Administration (FHWA) Intelligent Transportation Systems Joint Program Office (2000)
5. Lan, J., Lan, T., Nahavandi, S.: A Novel Application of a Microaccelerometer for Target Classification. IEEE Sens. J. 4, pp. 519–524 (2004)
6. Kleyko, D., Hostettler, R., Birk, W., Osipov, E.: Comparison of Machine Learning Techniques for Vehicle Classification Using Road Side Sensors. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp. 572–577, IEEE (2015)
7. Rao, S.S.: Vibraciones mecánicas. Pearson Educación (2012)
8. Hostettler, R., Birk, W., Lundberg Nordenvaad, M.: Feasibility of road vibrations-based vehicle property sensing. IET Intell. Transp. Syst. 4, 356 (2010)
9. Hunt, H.E.M.: Stochastic modelling of traffic-induced ground vibration. J. Sound Vib. 144, pp. 53–70 (1991)
10. Gupte, S., Masoud, O., Martin, R.F.K., Papanikolopoulos, N.P.: Detection and classification of vehicles. IEEE Trans. Intell. Transp. Syst. 3, pp. 37–47 (2002)
11. Bishop, C.M.: Pattern Recognition and Machine Learning (2013)
12. Semiconductors, N.: MMA8451Q 3-Axis, 14-bit/8-bit Digital Accelerometer, Data sheet
13. Arduino Uno Rev3, https://store.arduino.cc/usa/arduino-uno-rev3
14. AN10216-01 I Integrated Circuits 2 C Manual. Jean-Marc Irazabal – I Steve Blozis – I (2003)
15. Hernández Bautista, I., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Camacho-Nieto, O., Pogrebnyak, O.: Adjustment of Wavelet Filters for Image Compression Using Artificial Intelligence. Polibits, 53, pp. 23–30 (2016)

ISSN 1870-4069

# Asistente de velocidad vehicular como agente de control en entornos urbanos

Rodrigo Velázquez

Instituto Politécnico Nacional, Centro de Investigación en Computación, Ciudad de México, México

**Resumen.** En la actualidad es común encontrar vehículos cada vez más seguros y confortables, esto se debe al desarrollo tecnológico de los últimos años y a diversas iniciativas y proyectos a nivel mundial, entre las cuales podríamos citar el proyecto AutoNOMOS Labs a cargo del Dr. Raúl Rojas de la Universidad Libre de Berlín. Como ejemplo de estos avances se pueden mencionar algunos sistemas como el de antibloqueo de frenos (ABS), control crucero (CC) y, sistema asistente de velocidad inteligente (ISA). En este trabajo, primero se presentan diferentes implementaciones de asistentes de velocidad inteligente, que permiten asistir al conductor sobre la velocidad que lleva con el vehículo y, en determinadas ocasiones tomar el control de la aceleración y frenado. En una segunda etapa del trabajo se propone el modelo de un sistema asistente de velocidad, basado en coordenadas de mapas digitales y la ubicación actual del vehículo. Cuando la información de latitud y la longitud del GPS instalado en el vehículo se aproxime a las coordenadas que están almacenadas en la base de datos mariaDB (ingresadas a través de una aplicación desarrollada con JavaScript y Google Maps), verificará el límite de velocidad almacenado y la velocidad del vehículo para emitir una alerta al conductor.

**Palabras clave:** asistente de velocidad, mapas digitales, vehículos.

## Vehicle Speed Assistant as a Control Agent in Urban Environments

**Abstract.** Currently, it is common to find vehicles that are increasingly safe and comfortable, this is due to the technological development of recent years and various initiatives and projects worldwide, among which we could mention the AutoNOMOS Labs project, led by Dr. Raúl Rojas at the Free University of Berlin. As an example of these advances we can mention some systems such as the anti-lock braking system (ABS), cruise control (CC) and intelligent speed assist system (ISA). In this work, we first present different implementations of intelligent speed assistants, which allow the driver to be assisted on the speed that he carries with the vehicle and, in certain occasions, to take control of the acceleration and braking. In a second stage of the work the model of a speed assistant system is proposed, based on coordinates of digital maps and the current

location of the vehicle. When the latitude and longitude information of the GPS installed in the vehicle is close to the coordinates that are stored in the mariaDB database (entered through an application developed with JavaScript and Google Maps), it will verify the speed limit stored and the speed of the vehicle to issue an alert to the driver.

**Keywords:** speed assistant, digital maps, vehicles.

## 1. Introducción

El presente trabajo busca plantear el modelo de un sistema asistente de velocidad vehicular que sea capaz de identificar coordenadas lanzadas por un dispositivo GPS en el vehículo y compararlas con las coordenadas almacenadas en una base de datos de marcas de velocidad de zonas urbanas, y alertar al conductor si excede en alguna de ellas a medida que vaya avanzando en su recorrido, lo que puede contribuir de gran manera al desarrollo de esta línea de investigación y a mejorar de forma notable las posibilidades de obtener en un futuro no muy lejano un automóvil que pueda ser completamente asistido por un computador.

Parte de las normativas que mencionamos están constituidas por las señales de tránsito que regulan el comportamiento vial de los automóviles, específicamente las señales de máxima velocidad, para nuestro caso, las mismas resguardan la seguridad de los conductores y peatones para determinadas zonas.

### 1.1. Objetivos

El objetivo de este trabajo será proponer el modelo que incluye software y hardware, de un sistema asistente de velocidad de un vehículo, según su ubicación en tiempo real, para posteriormente comunicar por medio de sus coordenadas, su ubicación con respecto a unas marcas de límite de velocidad y así poder trabajar la información procesada pudiendo utilizarla para alertar al conductor sobre exceso de velocidad.

En detalle se tienen los siguientes objetivos: Proponer el diseño del sistema asistente de velocidad, proponer pruebas y experimentos con prototipos robóticos que permitan identificar coordenadas en movimiento y preparar un plan de pruebas para diferentes escenarios que permitan perfeccionar el diseño.

### 1.2. Motivación

Los mapas se utilizan para dos propósitos principales: localización y navegación. La localización ayuda a situarnos comparando los puntos de referencia que se ven a nuestro alrededor con los puntos de referencia que figuran en el mapa y la navegación es la que nos guía de un lugar a otro en una ruta desconocida.

En este trabajo, el concepto de mapa abarcará solamente el propósito de localización, ya que tanto el usuario que ingresa las marcas como el sistema

necesitarán establecer solamente las coordenadas de donde estará situada la máxima de velocidad y donde está el vehículo actualmente.

Este diseño surgió de la necesidad de contar con algún tipo de asistente, que permita al conductor alertar sobre los límites de velocidad, para cada zona en determinadas rutas, haciendo también posible alertar a los conductores acerca de los controles de velocidad bajo radar. Esto podría ser de gran utilidad para resguardar la seguridad tanto de peatones como del conductor y la posibilidad de evitar infracciones por exceso de velocidad.

### 1.3. Estructura del trabajo

Luego de los planteamientos presentados en función de los objetivos y la motivación de este trabajo, el contenido del mismo se organiza en el estado del arte y en la propuesta del diseño. Dentro de cada uno de ellos, y en sus secciones, se mencionan las estrategias de asistentes más relevantes y los trabajos de investigación previos y que han sido tomados como punto de inicio.

En la sección 2 se describe los conceptos que son necesarios conocer para que se pueda comprender los objetivos del presente trabajo y los últimos desarrollos existentes en el campo de la tecnología.

En la sección 3 se presenta el modelo para la implementación de un sistema asistente de velocidad para diferentes tipos de vehículos en zonas urbanas. Además, se presenta un listado de requerimientos necesarios para una posible implementación del mismo, como así también de una serie de ventajas y desventajas del diseño.

Finalmente en la sección 4 las conclusiones y resultados del presente trabajo, así como algunas líneas de investigación que podrían generarse a raíz de las contribuciones presentadas aquí.

## 2. Estado del arte

Enfocados en el Sistema Avanzado de Asistencia a la Conducción, o más conocido como Sistemas ADAS, presentamos los conceptos técnicos referentes a nuestros objetivos y a los proyectos de investigación; **AutoNOMOS**, de la Universidad Libre de Berlín, y el Programa **Autopía**, de la Universidad Politécnica de Madrid. Además, las nuevas aportaciones hechas para el control longitudinal de vehículos se presentan también en esta sección, las mismas son planteadas como diferentes tipos de estrategias aplicadas para conseguir dicho control.

### 2.1. Control longitudinal

Se trata acerca del comportamiento del vehículo, ante las diferentes entradas y perturbaciones que pudieran llegar a surgir con respecto al avance del mismo, en este caso el modelo será un modelo longitudinal, despreciando así las fuerzas

y movimientos laterales del vehículo debido a que apenas afectan para el cálculo de la velocidad y movimiento longitudinal [6].

Los Sistemas de navegación inercial (INS), permiten determinar la posición del vehículo a partir de los valores de la aceleración lineal y la velocidad angular medidos por el sistema. Un INS es un sistema formado por sensores que miden aceleraciones y variaciones angulares. Integrando esta información y la obtenida de los odómetros, podemos calcular la posición actual del vehículo, a partir de su posición inicial [6].

### 2.2. Sistema de aceleración electrónica (TAC)

**Throttle Actuador Control (TAC).** La aceleración electrónica suprime el acoplamiento mecánico entre la mariposa y el acelerador en vehículos a gasolina, siendo sustituido por una conexión eléctrica por medio de una unidad de control, que a la vez comanda la inyección y el encendido del motor para un mejor arranque en frío. La entrada de aire se controla mediante una señal eléctrica, a medida que cambia la posición del pedal [1].

Con la instalación de sistemas de aceleración electrónica se consigue la reducción de los gases de escape, una aceleración controlada y precisa sobre todo en velocidades invariables [1].

**Sensor de Posición de la Mariposa del Acelerador (TPS)** La señal indica en qué posición está la mariposa del acelerador a la PCM, ganando el incremento de la potencia del motor. Contiene un potenciómetro en el eje de la mariposa que trabaja como una resistencia que varía según el movimiento del eje, se alimenta con 5 voltios, tiene una señal variable de voltaje hacia la ECU y masa [1].

**Sensor de oxígeno** Verifica que la dosificación aire/combustible sea exacta para el convertidor catalítico por medio de la señal eléctrica del sensor de oxígeno a la ECU, y ajustará la mezcla idónea a inyectar en la cantidad de aire que entra al colector de admisión. Este sensor consta de cuatro cables: 2 al calefactor del sensor, una señal y una masa [1].

**Sensor de posición del pedal del acelerador (APP)** Este sensor puede venir equipado en el propio pedal o a su vez que un cable de gobierno se envíe al mismo, de este modo el conductor ejecuta la acción sobre un resorte que moverá los potenciómetros del APP para saber las exigencias a las que se está realizando la aceleración. Los sensores APP pueden constar de dos o tres potenciómetros, si es un sensor de dos potenciómetros tienen dos señales diferentes por lo que el voltaje de un potenciómetro se amplía y la otra decrece al mover el pedal del acelerador.[1]

La ECU examina cómo se desenvuelven los potenciómetros verificando que las tensiones se encuentren dentro de los rangos precisados [1].

### 2.3. Arduino Uno

Es una plataforma que está constituida de un microcontrolador con reguladores de tensión, un puerto USB acoplado a un módulo adaptador USB-SERIE que permite programar el microcontrolador y ejecutar pruebas de comunicación con el propio chip [1].

Esta placa también dispone de 6 pines de entrada analógicos que transportan las señales a un convertidor analógico/digital de 10 bits. Tiene un lenguaje de programación Processing (integrado de códigos abiertos) [1].

### 2.4. Sistemas inteligentes de transporte (ITS)

Los Sistemas Inteligentes de Transporte (ITS) pueden ser definidos como la integración de tecnologías de comunicación y electrónica con el fin de mitigar los problemas de transporte terrestre. Una aplicación podría consistir en la implementación de límites de velocidad variable en la vía para la adaptación de la conducción al estado del tráfico, a las características de la vía o a las condiciones meteorológicas en cada momento (**Gestión de la velocidad en las vías**). El objetivo de los sistemas de transporte inteligente es mejorar la movilidad, seguridad y eficiencia del transporte, mejorando la funcionalidad de los vehículos y las vías usando las tecnologías de la información [4].

### 2.5. Proyectos de investigación

**AutoNOMOS**, de la Universidad Libre de Berlín, inicia sus trabajos en el año 1998 construyendo robots autónomos, logrando en varias competencias internacionales obtener premios destacados, y el Programa **Autopía**, de la Universidad Politécnica de Madrid, el cuál fue iniciado en el año 1997, y desde entonces ha centrado su trabajo en la aplicación de técnicas de control, desarrolladas primero para robots móviles, para luego aplicarlas a vehículos autónomos reales.

**Programa Autopía** El programa Autopía tiene dos objetivos esenciales. El primero, implementar una conducción automática de vehículos comerciales sobre carreteras reales. Aunque este objetivo se puede considerar utópico en este momento, es un punto de partida importante para explorar el futuro. El segundo objetivo es el desarrollo de un sistema de guiado automático formado por componentes modulares que pueda incorporarse de manera sencilla en la industria del automóvil [7].

Las principales entradas sensoriales al sistema de guiado son una cámara de visión y un GPS de alta precisión. El sistema incluye una interfaz de comunicación y un sistema computacional de conducción de bajo nivel en el que reside el conocimiento y la experiencia humana, que se han modelado mediante lógica borrosa [8].

**AutoNOMOS** En la Universidad libre de Berlín, el Dr. Raúl Rojas ha estado construyendo robots autónomos desde 1998. Después de muchas participaciones exitosas en el RoboCup, al ganar el Campeonato Mundial dos veces y el Campeonato Europeo en cinco ocasiones, se decidió extender, como se menciona en [3]. En 2006, se tomó la decisión de participar en Grand Urban Challenge.

Se compró una caravan de Dodge retroadaptada, ya modificada, para que una persona con discapacidad pueda operarla utilizando solo un joystick y un touch pad. Con solo un pequeño presupuesto y 6 meses de tiempo de desarrollo, el auto llamado "Spirit of Berlin"(o más corto, SpoB) llegó a las semifinales de una competencia celebrada por la Agencia de Investigación Avanzada de Defensa (DARPA). Hasta el momento, el proyecto ha funcionado en tres automóviles diferentes [3].

### 2.6. Estrategias

Con el paso del tiempo y los avances tecnológicos, se fueron presentando diferentes mecanismos para el control de velocidad de los vehículos y a continuación se mencionan las siguientes estrategias de implementación.

**Acelerador inteligente** Con la intención de evitar aceleraciones no intencionadas, algunas marcas utilizan el acelerador inteligente, una solución bastante simple que hace que las órdenes del conductor sobre el pedal del freno siempre prevalezcan sobre las del acelerador. Esto significa, que si se nos queda accionado por error el acelerador, bastaría un pisotón al freno para poder parar el vehículo, ya que si ambos pedales están presionados a la vez la electrónica forzaría al motor a obedecer las órdenes del freno, pudiendo así desacelerar y parar el vehículo [4].

**Aplicación android asistente Velociraptor** es una nueva aplicación para Android que tiene una función muy especifica pero funciona muy bien. Se trata de un velocímetro que nos indica el límite de velocidad. Al ser una app flotante nos aparecerá en Google Maps. Se basa en nuestra ubicación y en los datos de Here Maps [2].

**Asistente de velocidad inteligente (ISA)** Dentro del proyecto iSafer, en el año 2017, el servicio de transporte de Londres probó un sistema ISA obligatorio en una pequeña cantidad de autobuses. El sistema demostró ser eficaz y resultó ser especialmente útil para evitar el exceso de velocidad en zonas de 20 mph. La prueba halló que, a pesar de que el sistema utilizado era obligatorio, hubo algunas ocasiones en que los autobuses excedieron el límite de velocidad en las secciones de descenso de la carretera. El sistema probado no aplicó automáticamente los frenos, solo evitó la aceleración por encima del límite de velocidad publicado [5].

## 3. Sistema asistente de velocidad

En principio se utiliza la librería Javascript de la API de Google Maps. La aplicación es básicamente HTML, CSS y JavaScript trabajando juntos. Los mapas son solo imágenes que se cargan en el fondo a través de peticiones ejecutadas por la tecnología de AJAX, y se insertan en un bloque  en la página HTML. Mientras navegas en el mapa, el API envía información acerca de las nuevas coordenadas y los niveles de "zoom" del mapa a través de AJAX y esto retorna las imágenes [9].

### 3.1. Modelo del sistema

Cuando la información de latitud y longitud del GPS se aproxime a los datos que están almacenados en la base de datos, verificará el límite de velocidad almacenado y la velocidad del vehículo. Si se excede el límite de velocidad especificado, el sistema emitirá una alerta al conductor. Teniendo en cuenta el proceso de desarrollo en el cual se encuentra actualmente este trabajo, queremos exponer en la Figura 2 citearticle, el comportamiento dinámico para diferentes valores de $T_m$, el cual sería el que más se ajusta a lo que se pretende con este trabajo. En la Figura 1 se ve el diagrama de bloques.

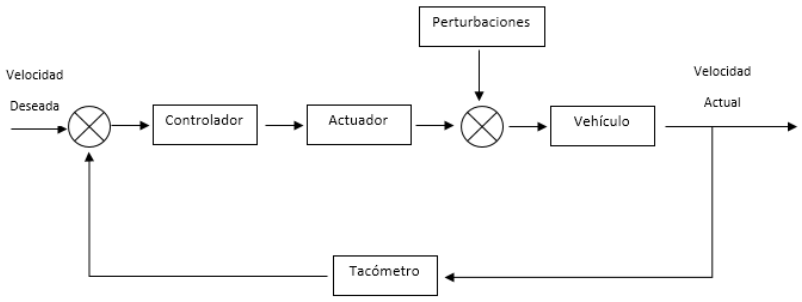



**Fig. 1.** Sistema de control de velocidad en lazo cerrado.

**Requerimientos** En cuanto a requerimientos técnicos, el sistema está diseñado para todo tipo de vehículo que cuente con inyección electrónica y con un dispositivo del tipo GPS que permita obtener y evaluar los datos de latitud y longitud.

**Ventajas y desventajas**

**Ventajas:** Control de velocidad en zonas de control por radar, disminución de accidentes, menos emisión de contaminantes al controlar la aceleración.

**Desventajas:** No reduce velocidad en pendientes, limita su aplicación las especificaciones técnicas de vehículos.

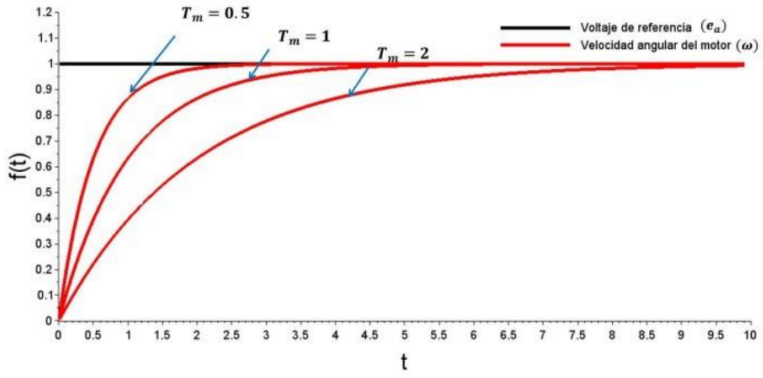



**Fig. 2.** Respuesta dinámica del modelo del motor de DC para diferentes constantes de tiempo.

### 3.2. Aplicación para ingreso de coordenadas

En esta sección pasaremos a ver las herramientas utilizadas para el desarrollo de la aplicación que permite el ingreso de las coordenadas de las marcas de velocidad. La cual hará posible también que se pueda consultar y comparar con la ubicación actual del vehículo. Como se ve en la Figura 3, la pantalla de ingreso de coordenadas en un teléfono móvil.

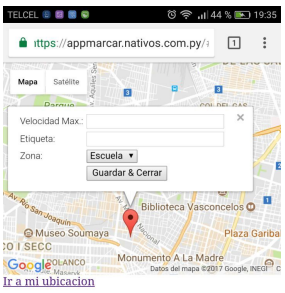



**Fig. 3.** Esta es la imagen de la aplicación.

**Aplicación** Las coordenadas están expresadas y almacenadas usando números decimales separados por coma. La latitud siempre precede la longitud. La latitud es positiva si va después del punto mostrado en el mapa y negativo si va antes. En los mapas físicos, las coordenadas están expresadas en grados, así que la posición de La Facultad Politécnica sería:

$$25°20.7' \ \mathbf{S} \ 57°31.22' \ \mathbf{O}.$$

La forma de convertir estos datos a decimales sería:

- $25°20.7' = (25 + (20 / 60) + (7 / 3600)) =$ -25.335,
- $57°31.22' =$ -$(57 + (31 / 60) + (22 / 3600)) =$ -57.522.

Como se ve en la Figura 4, así se encuentran almacenadas las coordenadas en la base de datos.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ☐ | Editar | Copiar | Borrar | 71 | 40 | Cervepar | -25.336876 | -57.528992 | otros |
| ☐ | Editar | Copiar | Borrar | 69 | 20 | Universidad Nacional de Asunción | -25.358959 | -57.518314 | Universidad |
| ☐ | Editar | Copiar | Borrar | 70 | 20 | Faculta Politecnica | -25.336327 | -57.521797 | Universidad |

**Fig. 4.** Ejemplo de cómo se almacenan los datos.

**Base de datos** Para el almacenamiento de las coordenadas de las marcas de velocidad en las diferentes zonas urbanas se utiliza el SGBD mariaDB. La aplicación mariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License). Es desarrollado por Michael (Monty) Widenius (fundador de MySQL), la fundación mariaDB y la comunidad de desarrolladores de software libre. Introduce dos motores de almacenamiento nuevos, uno llamado Aria -que reemplaza con ventajas a MyISAM- y otro llamado XtraDB -en sustitución de InnoDB. [10]

## 4. Conclusiones

En este trabajo se propuso el modelo que incluye software y hardware, de un sistema asistente de velocidad de un vehículo, según su ubicación en tiempo real, para posteriormente comunicar por medio de sus coordenadas su ubicación con respecto a unas marcas de límite de velocidad y así poder trabajar la información procesada pudiendo utilizarla para alertar al conductor sobre exceso de velocidad. Para su aplicación, el sistema necesitará una conexión a Internet o una base de datos en un dispositivo de almacenamiento dentro del vehículo conectado con GPS.

## Referencias

1. Buenaño-Moyano, L., Mena-Jiménez, R., Venegas-Núñez, J.S., Razo-Sifuentes, A.: Repotenciación de un banco de pruebas de inyección electrónica J20A a través de la adaptación de un sistema de aceleración electrónica TAC, para la implementación en el laboratorio de inyección electrónica de la escuela de ingeniería automotriz (2015)
2. Collado, C.: Ya puedes ver el límite de velocidad de la carretera en google maps gracias a velociraptor (2016)

3. Czerwionka, P.: A Three Dimensional Map Format for Autonomous Vehicles. Ph.D. thesis, Universidad Libre de Berlín, Instituto de Ciencias de Computación, Alemania
4. Espinoza-Ventura, R.: Sistemas inteligentes de transportes-ITS
5. European Transport Safety Council: iSAFER—ETSC (2017)
6. López-Montes, D.: Sistema de control longitudinal para vehículo eléctrico urbano. Universidad de Cantabria, España (2014)
7. Milanés, V., Naranjo, J., González, C., Alonso, J., García, R., de Pedro, T.: Sistema de posicionamiento para vehículos autónomos. Revista Iberoamericana de Automática e Informática Industrial 5(4), 36–41 (2008)
8. Pérez-Rastelli, J.M.: Agentes de control de vehículos autónomos en entornos urbanos y autovías. Ph.D. thesis, Facultad de Físicas – Universidad Complutense de Madrid, España
9. Rodríguez-Colón, A.: Google Maps API V3 introducción y primeros pasos (2016)
10. Wikipedia: MariaDB (2018)

ISSN 1870-4069

# Asistente computarizado para la determinación de la regla del fuera de lugar en el fútbol *soccer*

Jesús Jaime Moreno Escobar, Joshua Romero-Tapia,
Oswaldo Morales Matamoros

Instituto Politécnico Nacional, ESIME-Zacatenco,
Ciudad de México, México

jemoreno@esimez.mx

**Resumen.** En el presente documento se presenta una metodología con el objetivo de crear un asistente para la determinación del fuera de juego utilizando visión por computadora a partir de procesos de segmentación y detección de objetos esto a partir de imágenes capturadas mediante el uso de una cámara web la cual se encontrara dentro de un espacio de iluminación controlado. Obteniendo así una imagen del terreno de juego en la cual se determinaran los elementos que estén en él, así como las distancias que ocupan dentro de la imagen para con ello obtener la existencia del fuera de juego. Para que esta metodología sea posible es necesario conocer los elementos teóricos necesarios para determinar el fuera de juego en el fútbol *soccer*, así como los principios de visión por computadora necesarios para realizar la captura de imágenes, la segmentación y la detección de objetos así como las herramientas necesarias para su aplicación.

**Palabras clave:** visión por computadora, histograma de gradiente orientado, segmentación, detección de objetos, captura de imágenes.

## Computerized Assistant Referee for Determining the Offside Rule in Soccer

**Abstract.** In this document we present a methodology in order to create an assistant for determining the offside using computer vision from processes of segmentation and object detection this from captured using a webcam pictures which was within a space of controlled lighting. Thus, obtaining an image of the field in which the elements that are in it, and the distances they occupy within the image to thereby obtain the existence of offside is determined. For this approach possible is necessary to know the theoretical elements necessary to determine the offside-rule in *soccer*, as well as the principles of computer vision necessary to perform image capture, segmentation and object detection as well as the tools necessary for its implementation.

**Keywords:** computer vision, histogram of oriented gradient, segmentation, object detection, image capture.

## 1. Introducción

El fútbol es uno de los deportes más populares alrededor del mundo. A lo largo de su historia siempre ha existido polémica alrededor de este deporte. Sobre todo, a la falta de exactitud de los jueces o árbitros, quienes determinan las faltas o infracciones que se dan a lo largo de un partido.

La evolución de este deporte ha provocado que el futbolista sea considerado como una inversión en comparación con épocas pasadas donde se consideraba el aspecto deportivo para el desarrollo de este deporte. Esto ha provocado un incremento en la velocidad de juego, trayendo como consecuencia la determinación de sanciones erróneas por parte de los jueces o árbitros. Sin embargo, las transmisiones televisivas son capaces de demostrar cómo estas determinaciones erróneas perjudican a alguno de los equipos que se encuentran en el terreno de juego[2].

Al presentarse con mayor frecuencia este tipo de determinaciones, en la actualidad existe un gran cuestionamiento sobre la honestidad del deporte, con lo que los organismos reguladores del fútbol buscan con mayor rapidez opciones que sean capaces de minimizar el error mediante el uso de la tecnología. Así, que mientras en deportes como el fútbol americano o el tenis existen apoyos tecnológicos capaces de ayudar a los jueces o árbitros en aquellas jugadas difíciles de analizar, en el fútbol es muy escaso el apoyo de la tecnología al existir sólo un sistema, el cual es capaz de determinar si el balón ha entrado o no en la portería rival.

Uno de estos cuestionamientos se da en la marcación del fuera de juego, la cual es decisiva en los partidos de fútbol por lo que se ha convertido en una necesidad primordial el crear un dispositivo que sea capaz de asistir en la verificación de la también conocida regla 11 del fútbol, como la menciona el reglamento de juego de la FIFA, véase la Figura 1.


**Fig. 1.** Ejemplo de marcación de fuera de juego.

Establecida por la Universidad de Cambridge, en Inglaterra, en lo que se considera el primer reglamento de fútbol en el año de 1863 y modificada por la Federación Internacional de Fútbol Asociación (FIFA), en 1990. La regla del fuera de juego establece que un jugador se encontrará en posición antirreglamen-

taria si se encuentra más cerca de la línea de meta contraria que el balón y el penúltimo adversario. Esto se toma en cuenta a partir del momento en que el balón es tocado o jugado por uno de sus compañeros, véase la Figura 2.

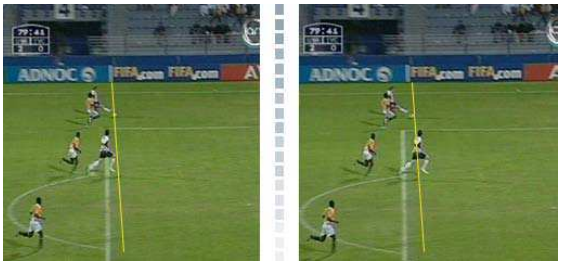


**Fig. 2.** Ejemplo de posición de fuera de juego [2].

Considerando que hasta el momento no existen opciones que mediante el uso de la tecnología apoye al árbitro en la determinación de esta regla, este trabajo tiene como objetivo desarrollar un asistente computacional para la determinación de la regla del fuera de juego en el fútbol *soccer*. Este trabajo se apoya en la visión por computadora mediante el uso de software capaz de realizar procesos de segmentación, detección de objetos y procesamiento digital de imágenes[3].

El asistente presentado en este trabajo representa una opción para apoyar a los árbitros en la toma de decisiones; esto a través de una mayor precisión en el análisis de esta regla mediante la detección de formas y sus posiciones dentro de la imagen. Pero resulta difícil que el sistema por sí mismo señale o diferencie entre la existencia o no de intención por parte del atacante en tener participación en la jugada, por lo cual es necesario remarcar que este dispositivo es únicamente un asistente y no un sustituto del árbitro[9].

A su vez, es importante señalar que el sistema descrito en este documento no determina en tiempo real el fuera de juego, sino que realiza la captura en un instante determinado para comenzar el análisis a partir de la imagen seleccionada y a petición de alguna autoridad, como se hace en otros deportes como el tenis[5], tenis de mesa[8], béisbol o fútbol americano. El único sistema que detecta en tiempo real, es la marcación del gol en el Hockey sobre hielo, ya que la velocidad del disco o *puck* alcanza los 160 km/h[4,6].

## 2. Entorno del proyecto

### 2.1. Regla del Fuera de Juego

Para la realización del algoritmo presentado en este artículo es necesario conocer la regla del fuera de juego la cual es resultado de la reforma de 1960 en el reglamento de la FIFA donde se contempla en el punto 11 la regla del fuera de juego. En esta regla se indica que existe una posición de fuera de juego si un jugador se encuentra más cerca de la línea de meta (portería) contraria que el

balón y el penúltimo adversario. Esto se toma en cuenta a partir del momento en que el balón es tocado o jugado por uno de sus compañeros con el fin de realizar un ataque[2]. Un ejemplo de esto se muestra en la Figura 2.

A su vez no existe posición de fuera de juego si el jugador está en su propia mitad de campo o a la misma altura que el penúltimo o los dos últimos adversarios (Figura 3). A su vez no hay infracción si un jugador recibe el balón en un saque de banda, un saque de meta o como resultado de un tiro de esquina[7].

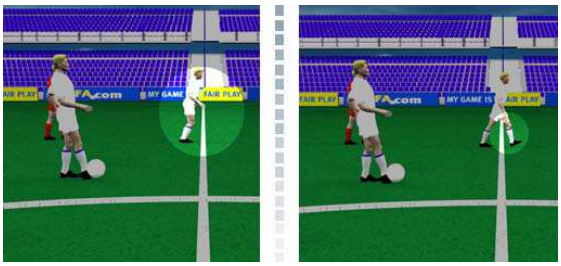


**Fig. 3.** Ejemplo de posición reglamentaria en media cancha[2].

Dentro de la regla del fuera de lugar existe el término de juego activo el cual se presenta cuando un jugador se encuentre interfiriendo a un adversario o cuando haya ganado ventaja de la posición en la que se encuentra[10]. Por parte de la IFAB se toman algunas consideraciones para sancionar el fuera de juego; la primera de ellas plantea que se debe tener en consideración cualquier parte de la cabeza, cuerpo o pies del atacante en relación al penúltimo adversario, el balón o la línea media. Mientras que, para la interpretación de esta decisión, los brazos no se consideraran parte del cuerpo[1]. Dentro de la participación en el juego activo se toma en consideración que un jugador no está cometiendo una infracción simplemente por estar en posición de fuera de juego. Mientras que, existe infracción cuando existe posición de fuera de juego con la participación activa en el juego. Como consecuencia de la infracción por fuera de juego, el árbitro otorga un tiro libre indirecto que es lanzado desde el lugar donde estaba el jugador infractor en el momento en que el balón fue jugado o tocado por uno de sus compañeros. Dentro de esta regla se recomienda a los árbitros asistentes, quienes son los encargados de marcar esta regla es necesario que el árbitro se encuentre concentrado y centrar su atención, esto resulta clave en la determinación y sanción de la regla, también es importante considerar la colocación del árbitro para reducir el error en la marcación[2].

## 3. Algoritmo del asistente para la determinación del fuera de juego

El algoritmo del asistente para la determinación del fuera de juego parte de una imagen capturada mediante el uso de un sensor, Figura 4. Este algoritmo

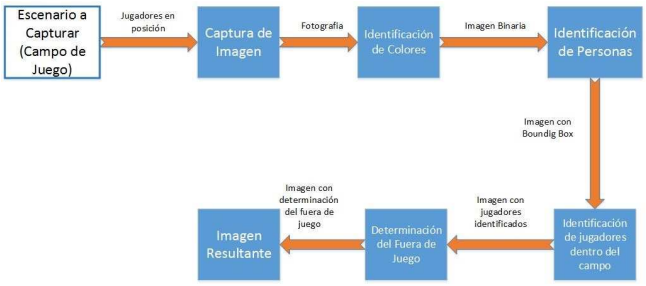



**Fig. 4.** Algoritmo del asistente para la determinación del fuera de juego.

busca que a partir de una imagen capturada se identifique el color del uniforme de ambos equipos para posteriormente identificar a cada jugador que se encuentre en el campo de juego. Así, el algoritmo ubica al ofensor más adelantado, así como al último defensor. Una vez hecho esto se compara la posición del ofensor contra el defensor y se determina si existe o no fuera de lugar.

### 3.1. Captura de imagen

El primer paso del algoritmo se realiza a partir de contar con un sensor compatible con el software MatLab, así como de contar con los elementos que se analizan dentro de un espacio de iluminación controlada, una vez que se cuenta con esto es posible realizar la captura de la imagen mediante la creación de un objeto de sistema el cual contiene las características del sensor conectado al dispositivo de cómputo que contenga el software MatLab R2014 o posterior.

### 3.2. Identificación de colores

Una vez que se tiene la imagen capturada, el siguiente paso es identificar el color principal de los uniformes de ambos equipos para con esto obtener una imagen que sólo muestre las áreas de interés, esto se realiza mediante un proceso de segmentación del color, el cual sigue los siguientes pasos:

1. Se toma un pixel dentro de la imagen $I$ con el color a identificar.
2. Se toma un valor de tolerancia el cual determina el intervalo de error $\varepsilon$.
3. Los resultados son almacenados dentro de un arreglo para los canales $I_R$, $I_G$ y $I_B$ con lo cual se calculan los valores de referencia.
4. Se realiza una comparación canal por canal con base en los valores de referencia y a la tolerancia señalada.
5. Mediante operaciones lógicas se comparan los resultados obtenidos.
6. Se realiza una segementación canal por canal para crear una imagen resultante $\widehat{I}$.
7. Se obtiene la imagen binaria $\overline{I}$ mediante la multiplicación de la imagen capturada por la imagen segmentada, es decir, $\overline{I} = I \times \widehat{I}$.

Este análisis se realiza para los jugadores de ambos equipos, por lo cual se obtienen dos imágenes resultantes, $\overline{I}$ y su complemento $1 - \overline{I}$, que representan las regiones con mayor cantidad de los colores indicados por el usuario al inicio de esta etapa, tomando en consideración la tolerancia registrada $\varepsilon$.

### 3.3. Detección de personas

Una vez obtenidas las imágenes binarias con los colores del uniforme de ambos equipos se utiliza esa información para conocer la posición de los jugadores dentro del terreno de juego, para esto se toma en cuenta la información proporcionada por las imágenes binarias antes generadas, $\overline{I}$ y $1 - \overline{I}$. Así se obtiene una nueva imagen, la cual a partir de la imagen capturada original se muestra a cada uno de los elementos que se encuentren en el terreno de juego. Este proceso se realiza de la siguiente manera:

1. Se generan los objetos de sistema necesarios para habilitar las funciones de detección de personas.
2. Se establecen los parámetros necesarios para realizar la detección de personas.
3. Se realiza la detección de personas en cada una de las imágenes binarias obtenidas en la identificación de colores, $\overline{I}$ y $1 - \overline{I}$.
4. Se crea la imagen resultante $I_s$ en la cual ya se encuentran señalados los elementos considerados persona dentro del terreno de juego.

Una vez terminado este proceso la imagen obtenida muestran a todos los jugadores de ambos equipos.

### 3.4. Determinación del fuera de juego

Una vez identificados todos los elementos necesarios para la detección del fuera de juego, el siguiente paso es realizar la determinación de la regla del fuera de juego, esto se realiza de la siguiente manera:

1. Una vez que se tienen todos los elementos detectados se identifica la ubicación en el eje x de cada uno de los jugadores utilizando los datos obtenidos en la detección de personas. Esto es posible mediante el uso de la información proporcionada por la etapa de detección de personas, donde a través de conocer la posición de los elementos detectados se ubican los dos más adelantados para ambos equipos los cuales son comparados entre sí.
2. Con la posición de ambos jugadores conocida se compara si la posición del atacante es menor a la posición del defensor, si esto ocurre *no existe fuera de juego* en caso contrario se determina *el fuera de juego*. Es por ello que se consideran los dos valores en el eje de las abscisas de los jugadores más adelantados mediante el uso de una sustracción $Pos = X_{J_1} - X_{J_2}$ de la siguiente manera:
   - Primeramente, si el valor de la sustracción $Pos > 0$ no existe fuera de juego.

- En caso contrario si el valor de la sustracción $Pos \leq 0$ existe fuera de juego.

3. Una vez que comparadas las posiciones de ambos jugadores y se determina el fuera de juego se crea una nueva imagen resultante $I_{offside}$ la cual indica la existencia o no del fuera de juego. Un ejemplo del resultado $I_{offside}$ de aplicar este algoritmo propuesto se observa en las Figuras 5 y 6.


**Fig. 5.** Resultado del algoritmo propuesto o $I_{offside}$, se detecta que *hay Posición Adelantada*.


**Fig. 6.** Resultado del algoritmo propuesto o $I_{offside}$, se detecta que *no hay Fuera de Lugar*.

## 4. Resultados experimentales

Una vez realizadas cada una de las etapas del algoritmo lo siguiente es conocer si el algoritmo propuesto es capaz de determinar el fuera de juego,

así como las condiciones necesarias para determinarlo. Por lo cual, mediante una serie de capturas utilizando figuras Playmobil® a las cuales se les pegan imágenes de jugadores de fútbol con distintos colores de uniforme, siendo el equipo azul el equipo ofensivo y el equipo amarillo el defensivo, con lo cual se ha considerado que el descriptor de HOG utilizado en la fase de detección de personas se encuentra más familiarizado con aquellas figuras que asemejen la estructura del ser humano.

Otro punto importante que se considera son aquellos casos donde al acercar los jugadores de ambos equipos de forma que no sean perceptibles totalmente por el sensor o incluso en casos donde el sensor no es capaz de detectarlos lo cual produce que sólo aquellos jugadores detectados por la cámara sean considerados para el análisis del asistente para la determinación del fuera de juego, a estas situaciones se les llama *casos críticos*.

El rango del sensor utilizada para el experimento es de 90° con respecto al terreno de juego, mientras que la altura está determinada por las características físicas del sensor. Así, se presentan dos casos para evaluar el funcionamiento del algoritmo los cuales permiten diferenciar la existencia o no del fuera de juego los cuales son:

1. Consiste en tener al equipo ofensor en posición reglamentaria
2. Un atacante fue desplazado por delante de la línea defensiva con lo cual se espera que el algoritmo determine la posición del fuera de juego.

Para este experimento se determinaron los siguientes parámetros dentro del código del algoritmo:

1. Rango de radio del balón: 71 a 78 pixeles
2. Polaridad de Balón: Clara
3. Sensibilidad de detección del balón: 0.9822
4. Umbral de detección: 0.26
5. Factor de escala para personas: 1.7222
6. Tamaño mínimo: 102 a 49 pixeles

Así, se propone la modificación de la altura del sensor para una mejor captura, así como establecer un ambiente controlado donde el brillo y la sombra percibida en el terreno de juego disminuyan. La altura del sensor se modifica a 18 cm sobre el terreno de juego y con respecto al algoritmo se realizaron los siguientes cambios:

1. El rango de radio cambia de entre 50 y 55 pixeles.
2. Se mantiene la polaridad del balón en *clara*.
3. Sensibilidad de detección del balón: 0.98225
4. Umbral de detección a un valor de 0.2655
5. Para este caso, se descarta el factor de escala para personas:1.7222
6. De igual manera, se descarta el tamaño mínimo: 102 a 49
7. Mismo modelo de detección
8. Umbral de clasificación de personas: 1.65

Una vez que se realizaron estas modificaciones se capturan en las mismas posiciones que se utilizaron en el caso anterior. En este experimento se demuestra que es posible determinar el fuera de juego mediante el uso del algoritmo propuesto, tomando en consideración que los pixeles seleccionados proporcionen la información suficiente para la detección de colores y personas, mientras que la modificación del Ángulo facilita la detección del balón, lo cual permite pasar al algoritmo de fase de determinación del fuera de juego.

Un problema encntrado al aplicar el algoritmo en este segundo modelo es que en las capturas que no se detecta el fuera de juego se observa que la función *peopleDetector* determina el orden de las personas considerando la distancia focal del sensor. Esto provoca que al solicitar al elemento que se encuentre más adelantado se detecta al elemento más cercano al sensor.

Al modificar la posición del sensor y tomar en cuenta la función *peopleDetector* que toma en cuenta la distancia focal al propio sensor, ahora es necesario tomar en consideración la posición de los elementos en el eje de las abscisas, con lo que es posible obtener los resultados almacenados en el arreglo *bboxes* de la función *step* de la biblioteca *peopleDetector* donde la primera columna contiene los valores de x en el eje de las abscisas.

Una vez que se obtienen los valores en el arreglo se hacn las siguientes modificaciones al algoritmo:

1. Se extraen los valores de x del arreglo *bboxes*, los valores se almacenan en un nuevo arreglo.
2. Se busca el valor máximo dentro del arreglo, que es el valor más grande dentro del eje x el cual representa al jugador más alejado del extremo izquierdo de la imagen.
3. Una vez que se encuentro el valor más grande se busca su posición dentro del arreglo *bboxes*.
4. Se buscó el valor asociado a la fila de valor x para obtener las coordenadas en el eje de las ordenadas.
5. Se realizan los pasos del 2 al 4 para el arreglo *bboxes2* y se obtienen los puntos $x_2$ y $y_2$
6. Conocidos los puntos $x_2$ y $y_2$ y al aplicar la ecuación de la recta se obtiene la recta entre los dos puntos.
7. Se obtuvo el punto medio de la recta mediante la ecuación $x_m = \frac{x_2+x_1}{2}$
8. Se compara el punto medio de la imagen y el punto medio de la recta, lo cual se aplica bajo las siguientes condiciones:
   *a*) Si el valor del punto medio de la resta es mayor que el obtenido en la imagen se realiza lo siguiente:
      1) Se realiza una sustracción entre el valor de ambos puntos.
      2) El resultado de esta operación es el número de columnas dentro de la imagen que se eliminan.
      3) Se crea un arreglo que contiene la última columna de la imagen
      4) Se realiza la concatenación de imágenes, así en primer lugar se inserta la imagen sustraída de la original y posteriormente el arreglo repetido la cantidad de columnas que se eliminaron en la imagen original.

   *b*) Si el valor del punto de la recta es menor que el obtenido en la imagen se realiza los siguiente:
      1) El resultado de esta operación es el número de columnas máximo que tiene la imagen
      2) Se crea un arreglo que contiene la primera columna de la imagen.
      3) Se realiza la concatenación de imágenes primeramente se inserta en el arreglo repetido de la cantidad de columnas que se eliminan en la imagen original y posteriormente la imagen sustraída de la original.
   *c*) En caso de que ambos puntos se encuentren en la misma posición no se modifican en la imagen.
9. Para determinar el fuera de juego se realiza una sustracción entre los puntos $x_1$ y $x_2$ obtenidos del arreglo *bboxes* por lo que se considera lo siguiente:
   *a*) Si el valor de la sustracción es mayor que cero no existe fuera de lugar.
   *b*) Si el valor de la sustracción es menor o igual que cero existe fuera de lugar.

Al considerar que el porcentaje obtenido en el experimento anterior no es lo suficientemente alto para considerar que el objetivo del asistente para la determinación del fuera de juego se cumple. Ahora, se considera que la forma de las figuras Playmobil no representan con exactitud a un ser humano, lo cual provoca que la función *peopleDetector* encuentre a todos los jugadores en el terreno de juego.

Para solucionar esto se utilizan fotos de jugadores de futbol con los mismos colores de uniforme que las figuras playmobil sobre estas y con ello se realizan pruebas en diferentes posiciones del terreno de juego tanto en posición reglamentaria como en fuera de juego.

A su vez se consideran algunos casos críticos los cuales se presentan al acerca los jugadores de ambos equipos de forma que no son perceptibles totalmente por el sensor o incluso en casos donde el sensor no es capaz de detectarlos, lo cual produce que sólo aquellos jugadores detectados por el sensor sean considerados para el análisis del asistente para la determinación del fuera de juego.

Una vez que el asistente demostró su funcionamiento, resulta necesario conocer su tiempo de respuesta, para lo cual se realizaron 30 tomas, 15 en posición reglamentaria con 5 casos críticos y 15 en posición de fuera de juego con 5 casos críticos. Obteniendo los resultados mostrados en las Tablas 1 y 2.

De los resultados obtenidos se concluye que de las 15 en posición reglamentaria con sus 5 casos críticos, 12 resultaron aciertos con 4 casos críticos positivos, lo que esto representa un 80 % de efectividad con un tiempo promedio de 72.98 ms, mientras que para las 15 posiciones de fuera de juego con sus 5 casos críticos, 11 resultaron positivos con 2 casos críticos positivos lo que representa un 73.3 % de efectividad con un tiempo promedio de 143.53 ms.

De los resultados obtenidos se concluye que de las 30 capturas en posición reglamentaria, 27 de ellas resultaron correctas, lo que representa un porcentaje de efectividad del 90 %. Mientras que de las 30 capturas en posición de fuera de juego 23 resultaron correctas lo que representa un 73.67 % de efectividad. Por lo tanto, contabilizando las 60 pruebas totales resultando correctas 52 capturas

**Tabla 1.** Resultados obtenidos en el Experimento 4 en Posición Reglamentaria.

| Muestra | Caso Crítico | Detección | Tiempo (ms) |
|---|---|---|---|
| 1 | No | Sí | 166.136 |
| 2 | No | Sí | 23.69 |
| 3 | No | Sí | 19.59 |
| 4 | No | No | 17.829 |
| 5 | No | Sí | 16.86 |
| 6 | No | No | 18.16 |
| 7 | No | Si | 17.82 |
| 8 | No | Si | 19.059 |
| 9 | No | Si | 21.32 |
| 10 | No | Si | 19.13 |
| 11 | Sí | Si | 75.72 |
| 12 | Sí | Si | 297.12 |
| 13 | Sí | No | 304.1 |
| 14 | Sí | Sí | 48.83 |
| 15 | Sí | Sí | 29.39 |

**Tabla 2.** Resultados obtenidos en el Experimento 4 en posición de fuera de juego.

| Muestra | Caso Crítico | Detección | Tiempo (ms) |
|---|---|---|---|
| 1 | Sí | No | 27.17 |
| 2 | Sí | Sí | 24.37 |
| 3 | Sí | No | 31.18 |
| 4 | Sí | No | 454.19 |
| 5 | Sí | Si | 60.52 |
| 6 | No | Si | 29.21 |
| 7 | No | Si | 22.73 |
| 8 | No | Si | 23.35 |
| 9 | No | No | 20.85 |
| 10 | No | Si | 42.17 |
| 11 | No | Si | 282.33 |
| 12 | No | Si | 120.48 |
| 13 | No | SÍ | 524.3 |
| 14 | No | Sí | 450.77 |
| 15 | No | Sí | 39.33 |

se obtiene un porcentaje del 86.6 % de efectividad total del asistente para la determinación del fuera de juego.

## 5. Conclusiones

Finalmente, como resultado de los experimentos realizados se denuestra que es posible desarrollar un asistente para la detección del fuera de juego por medio de Visión por computadora utilizando procesos de segmentación y detección de objetos utilizando la herramienta MatLab.

A su vez se concluye que para que el objetivo de la determinación del fuera de juego tenga el menor error posible es necesario contar con las condiciones físicas ideales como son una iluminación uniforme dentro del terreno de juego, además de considerar que los elementos que se encuentren dentro de este se asemejan lo más posible a la forma del ser humano, esto con el fin de evitar falsos positivos y errores que no permitan la detección del fuera de juego. También se demuestra que es posible desarrollar un asistente para la determinación del fuera

de juego en un tiempo aceptable siempre y cuando las condiciones del equipo de procesamiento lo permitan, ya que de no ser así esto retrasa la respuesta, lo cual no resuleve la problemática en el análisis de esta regla en el campo de juego.

**Agradecimientos.** Este trabajo es desarrollado con recursos e instalaciones del Instituto Politécnico Nacional, México, por medio del Proyecto SIP 20180514 y la Comisión de Operación y Fomento de Actividades Académicas (COFAA). Cabe resaltar que este trabajo es parte de la tesis de Nivel Licenciatura del Becario BEIFI Joshua Romero Tapia, dirigida por el Dr. Jesús Jaime Moreno Escobar. También, se le agradece por un lado a la Ing. Isabel Meraz por el apoyo soporte lógistico y técnico; y por otro a los revisores que aportaron sus valiosos conocimientos para mejora del presente artículo.

## Referencias

1. Aribowo, A., Gunawan, G., Tjahyadi, H.: Adaptive edge detection and histogram color segmentation for centralized vision of soccer robot. In: 2016 International Conference on Informatics and Computing (ICIC). pp. 49–54 (Oct 2016)
2. FIFA: Reglas de Juego 2015/2016. FIFA, FIFA-Strasse 20 Apdo. postal 8044 Zúrich Suiza (2015)
3. Gholami, A., Bigham, B.S.: A learned soccer goalkeeper petri net model. In: 2017 Artificial Intelligence and Robotics (IRANOPEN). pp. 102–108 (April 2017)
4. Hardegger, M., Ledergerber, B., Mutter, S., Vogt, C., Seiter, J., Calatroni, A., Tröster, G.: Sensor technology for ice hockey and skating. In: 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN). pp. 1–6 (June 2015)
5. Leong, L.H., Zulkifley, M.A., Hussain, A.B.: Computer vision approach to automatic linesman. In: 2014 IEEE 10th International Colloquium on Signal Processing and its Applications. pp. 212–215 (March 2014)
6. Pileggi, H., Stolper, C.D., Boyle, J.M., Stasko, J.T.: Snapshot: Visualization to propel ice hockey analytics. IEEE Transactions on Visualization and Computer Graphics 18(12), 2819–2828 (Dec 2012)
7. Song, X., Zhou, Z., Guo, H., Zhao, X., Zhang, H.: Adaptive retinex algorithm based on genetic algorithm and human visual system. In: 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC). vol. 01, pp. 183–186 (Aug 2016)
8. Wong, P.K.C.: Developing an intelligent assistant for table tennis umpires. In: First Asia International Conference on Modelling Simulation (AMS'07). pp. 340–345 (March 2007)
9. Yao, Q., Kubota, A., Kawakita, K., Nonaka, K., Sankoh, H., Naito, S.: Fast camera self-calibration for synthesizing free viewpoint soccer video. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1612–1616 (March 2017)
10. Yu, X., Leong, H.W., Xu, C., Tian, Q.: Trajectory-based ball detection and tracking in broadcast soccer video. IEEE Transactions on Multimedia 8, 1164–1178 (2006)

ISSN 1870-4069

# Optimización por enjambre de partículas en el diseño de un árbol de transmisión

Derlis Hernández-Lara[1,3], Emmanuel Alejandro Merchán-Cruz[1], Ricardo Gustavo Rodríguez-Cañizo[1], Edgar Alfredo Portilla-Flores[2], Álvaro Marcos Santiago-Miguel[1]

[1] Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, México
[2] Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, México
[3] Tecnológico de Estudios Superiores de Ecatepec, México
{derlis392, marcoversal}@hotmail.com, {eamerchan, rgrodriguez, aportilla}@ipn.mx

**Resumen.** En este trabajo se presenta la utilización de metaheurísticas para apoyar al diseño de sistemas mecánicos, como una propuesta de metodología de diseño robusta. Se empleó el algoritmo de optimización por enjambre de partículas, PSO (del inglés, *Particle Swarm Optimization*), para el diseño óptimo de un árbol de transmisión hecho de materiales compuestos, con el fin de obtener los parámetros de diseño adecuados para soportar las cargas específicas a las que estará sometido este elemento. El problema en este tipo de diseños, es encontrar el espesor adecuado de la pieza, es decir el número de láminas y sus respectivas orientaciones que la conformarán, para lo cual existen demasiadas posibles combinaciones al trabajar con laminados de materiales compuestos, la metodología propuesta busca la mejor solución posible de acuerdo con los parámetros del problema. La principal aportación de este trabajo es el uso de metaheurísticas para resolver problemas de diseño complejos y difíciles de resolver por métodos tradicionales. Los resultados experimentales conllevan a concluir que este tipo de implementaciones son de gran utilidad para resolver problemas de optimización numérica en el diseño con materiales compuestos para diversas aplicaciones en ingeniería.

**Palabras clave:** optimización por enjambre de partículas (PSO), diseño con materiales compuestos, árbol de transmisión.

## Particle Swarm Optimization Approach in the Drive Shaft Design

**Abstract.** This paper presents the use of metaheuristics to support the design of mechanical systems, as a proposal of robust design methodology. The particle

swarm optimization algorithm (PSO) approach in the drive shaft made of composite materials, in order to obtain the optimal design parameters to support the loads specific to which this element will be subject. The problem in this type of design, is to find the adequate thickness of the piece, that is to say, the number of sheets and their respective orientation that will form it, for which there are too many possible combinations when working with laminates of composite materials, the proposed methodology look for the best possible solution according to the parameters of the problem. The main contribution of this work is the use of metaheuristics to solve design problems and complexes of resolution of traditional methods. The experimental results will lead to the conclusion that this type of implementations is very useful to solve problems of numerical optimization in the design with composite materials for diverse engineering applications.

**Keywords:** particle swarm optimization (PSO), design with composite materials, drive shaft.

## 1. Introducción

Uno de los principales objetivos al implementar metaheurísticas en problemas de optimización, es el de resolver situaciones complejas y buscar soluciones factibles dentro de un intervalo definido por las cotas de diseño. Para este estudio se realiza la búsqueda del número de laminados y mejor secuencia de apilamiento en el diseño de un árbol de transmisión para automóviles hecho de materiales compuestos, se usa el algoritmo de optimización por enjambre de partículas para obtener los parámetros de diseño adecuados para las cargas y esfuerzos requeridos en su trabajo específico.

En [1] mencionan que en las últimas décadas las industrias como la aeronáutica, automotriz y la de prótesis prefieren utilizar materiales compuestos en lugar de materiales tradicionales, debido a su excelente relación resistencia/peso y alta rigidez especifica. Otra ventaja de usar materiales compuestos es que la estructura se puede diseñar seleccionando la fibra y las orientaciones adecuadas para cumplir con los requerimientos específicos. Un problema clásico de diseño con materiales compuestos tiene gran cantidad de variables discretas, como el número de láminas, orientación de las mismas, espesor y tipo de material. La flexibilidad en seleccionar estas variables para cumplir con los requisitos introduce complejidad en el problema de diseño. Además, en la mayoría de los problemas de diseño, se conocen ciertas especificaciones a priori, como, espesor del laminado, opciones para orientaciones de los mismos y tipo de material. Por lo tanto, el diseño de una estructura mediante materiales compuestos se reduce para buscar orientaciones discretas de capas apropiadas y parámetros geométricos en un rango dado para lograr la resistencia y rigidez especificada.

## 2. Antecedentes y trabajos previos

En [2] utilizaron metaheurísticas para el diseño, optimización y selección de vigas mediante la metodología de diseño paramétrico, variando así las secciones transversales

de las vigas de estudio, con el fin de optimizarlas para soportar cargas especificas sin fracturarse. Sus resultados experimentales concluyen que este tipo de implementaciones son de gran utilidad para resolver problemas de diseño en diversas aplicaciones.

En [3] formularon una metodología de diseño óptimo multiobjetivo de vigas de pared delgada PRF (Plásticas Reforzadas con Fibras). El problema de optimización se plantea de manera de considerar restricciones estructurales y geométricas preestablecidas en el diseño, proponen minimizar el peso de la estructura conjuntamente con los desplazamientos máximos producidos, para lo cual usan la heurística SA (*Simulated Annealing*) empleando dos esquemas de búsqueda diferentes.

En [1] utilizaron optimización por enjambre de partículas y algoritmos genéticos (AG) para el diseño de una viga hueca, con un enfoque multiobjetivo, el diseño de la viga fue hecho con materiales compuestos y aplicación para la industria aeronáutica, los resultados mostraron que PSO obtuvo mejores resultados que el AG para este tipo de diseños.

En [4] se diseñó un árbol de transmisión para vehículos automotores hecho de materiales compuestos, la metodología de diseño implicó encontrar la combinación adecuada del material para que la pieza no falle en su funcionamiento, en este trabajo se utilizó un algoritmo propuesto por el autor, los resultados obtenidos fueron satisfactorios, concluyendo que la utilización de heurísticas en el diseño con materiales compuestos es de gran utilidad.

En [5] usaron algoritmos genéticos para diseñar un árbol de transmisión, obteniendo el espesor y la secuencia de apilamiento óptima de materiales compuestos para la pieza, para este caso diseñaron con fibra de vidrio y fibra de carbono e hicieron la comparación con respecto al diseño con acero.

Respecto a los antecedentes mencionados, las desventajas son que los resultados obtenidos son muy ideales y difíciles de llevar a la práctica, sin embargo, esta investigación pretende contribuir a subsanar esta situación, mediante la implementación de restricciones que permitan obtener resultados factibles de materializar y que conlleven a utilizar esta metodología para diversos diseños con materiales compuestos en ingeniería.

## 3. Metodología

De acuerdo con la literatura, las heurísticas pueden adaptarse al diseño con materiales compuestos, ya que son métodos de optimización global y pueden utilizarse para problemas no lineales o de variables discretas. En ingeniería siempre hay que analizar detalladamente el fenómeno en cuestión, para poder determinar todas las variables involucradas al respecto, en este caso interesa el estudio y comprensión del diseño de un árbol de transmisión hecho de fibra de carbono.

Se inicia con una propuesta de análisis del sistema de tal forma que se diseñe adecuadamente, para este caso se analiza el sistema como se observa la figura 1.

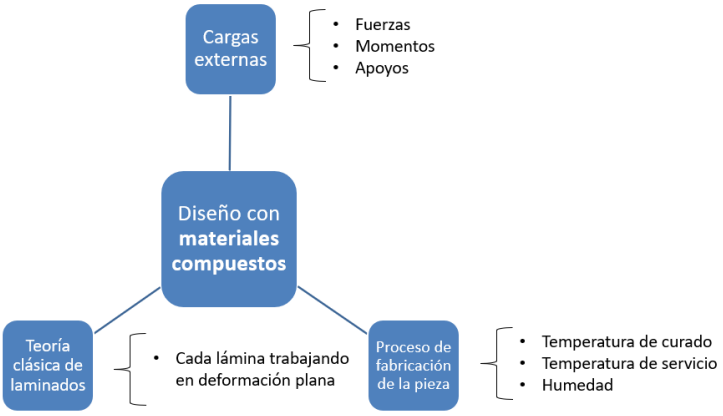



**Fig. 1.** Elementos a considerar en el diseño con materiales compuestos.

### 3.1. Teoría clásica de laminados

Para diseñar tomando en cuenta los esfuerzos aplicados al árbol de transmisión, y las reacciones internas que se producen en el material del cual está conformado se tiene que usar la teoría clásica de laminados. Para crear una relación lineal entre tensión-deformación para un material anisótropo se parte de la teoría de elasticidad como se muestra en la ecuación 1, mediante la ley de Hooke generalizada [4].

$$\{\sigma_{ij}\} = [Q_{ij}]\{\varepsilon_{ij}\}, \qquad \text{Siendo } i, j=1, 2, 3, 4, 5, 6. \tag{1}$$

Donde $[Q_{ij}]$ recibe el nombre de la matriz de rigidez. Para un material genérico, esta matriz tiene 36 componentes para definir completamente el material como se muestra en la ecuación 2.

$$[Q_{ij}] = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & Q_{14} & Q_{15} & Q_{16} \\ Q_{21} & Q_{22} & Q_{23} & Q_{24} & Q_{25} & Q_{26} \\ Q_{31} & Q_{32} & Q_{33} & Q_{34} & Q_{35} & Q_{36} \\ Q_{41} & Q_{42} & Q_{43} & Q_{44} & Q_{45} & Q_{46} \\ Q_{51} & Q_{52} & Q_{53} & Q_{45} & Q_{55} & Q_{56} \\ Q_{61} & Q_{62} & Q_{63} & Q_{46} & Q_{65} & Q_{66} \end{bmatrix}. \tag{2}$$

Para este tipo de diseños los laminados de fibra de carbono son delgados y no se aplican cargas fuera del plano, se consideran como problemas de deformación plana, y se definen como se muestra en las ecuaciones 3 y 4.

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = [Q] \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix}, \tag{3}$$

$$\begin{Bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{Bmatrix}=\begin{bmatrix}Q_{11} & Q_{12} & 0\\ Q_{21} & Q_{22} & 0\\ 0 & 0 & Q_{66}\end{bmatrix}\begin{Bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}\end{Bmatrix}. \quad (4)$$

La matriz de rigidez para cada lámina queda definida por las ecuaciones 5 y 6.

$$[Q_{ij}]=\begin{bmatrix}Q_{11} & Q_{12} & 0\\ Q_{21} & Q_{22} & 0\\ 0 & 0 & Q_{66}\end{bmatrix}, \quad (5)$$

$$[Q_{ij}]=\begin{bmatrix}\frac{E_{11}}{1-v_{12}v_{21}} & \frac{v_{12}E_{22}}{1-v_{12}v_{21}} & 0\\ \frac{v_{12}E_{22}}{1-v_{12}v_{21}} & \frac{E_{22}}{1-v_{12}v_{21}} & 0\\ 0 & 0 & Q_{66}\end{bmatrix}, \quad (6)$$

donde: $E_{ij}$=módulo de elasticidad en el eje correspondiente.

$v_{12}$=Coeficiente de Poisson en XY.

La matriz $[Q_{ij}]$ está expresada en ejes locales. Para convertir esta matriz en ejes globales se necesita operar con la matriz de transformación como se muestra en las ecuaciones 7 y 8.

$$\{\sigma'_{ij}\}=[\overline{Q_{ij}}]\{\varepsilon'_{ij}\}, \quad \text{Siendo } i, j=1, 2, 6. \quad (7)$$

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{Bmatrix}=\begin{bmatrix}\overline{Q_{11}} & \overline{Q_{12}} & \overline{Q_{11}}\\ \overline{Q_{21}} & \overline{Q_{22}} & \overline{Q_{11}}\\ \overline{Q_{16}} & \overline{Q_{26}} & \overline{Q_{66}}\end{bmatrix}\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix}. \quad (8)$$

En las ecuaciones de la 9 a la 14 se muestra el cálculo para cada variable anterior de acuerdo con la orientación de las fibras respecto al eje local [5].

$$\overline{Q_{11}}=Q_{11}c^4+Q_{22}s^4+2(Q_{12}+2Q_{66})c^2s^2, \quad (9)$$

$$\overline{Q_{12}}=(Q_{11}+Q_{22}-4Q_{66})c^2s^2+Q_{12}(c^4+s^4), \quad (10)$$

$$\overline{Q_{16}}=(Q_{11}-Q_{12}-2Q_{66})c^3s-(Q_{12}-Q_{22}+2Q_{66})c\,s^3, \quad (11)$$

$$\overline{Q_{22}}=Q_{11}s^4+Q_{22}c^4+2(Q_{12}+2Q_{66})c^2s^2, \quad (12)$$

$$\overline{Q_{26}}=(Q_{11}-Q_{12}-2Q_{66})s^3c-(Q_{12}-Q_{22}+2Q_{66})sc^3, \quad (13)$$

$$\overline{Q_{66}}=(Q_{11}+Q_{22}-2Q_{12}-2Q_{66})c^2s^2+Q_{66}(c^4+s^4). \quad (14)$$

Las ecuaciones anteriores se obtuvieron tomando en cuenta el esquema mostrado en la figura 2.

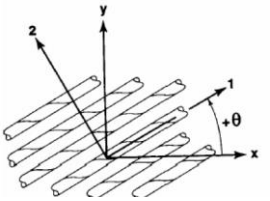



**Fig. 2.** En las expresiones anteriores: c=cosθ y s=senθ.

Según la teoría clásica de las placas laminadas, la ecuación constitutiva se puede escribir como en se muestra en la ecuación 15.

$$\begin{Bmatrix} N \\ M \end{Bmatrix} = \begin{bmatrix} A & B \\ B & D \end{bmatrix} \begin{Bmatrix} \varepsilon^0 \\ k \end{Bmatrix}, \tag{15}$$

donde:

$\varepsilon^0$=Vector de deformaciones en el plano medio

$k$=Curvaturas en la lámina

La forma completa de la anterior expresión se observa en la ecuación 16.

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{21} & A_{22} & A_{26} & B_{21} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{21} & B_{22} & B_{26} & D_{21} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \\ k_x \\ k_y \\ k_{xy} \end{Bmatrix}. \tag{16}$$

Las direcciones de las cargas y momentos aplicados a cada lamina que constituye el árbol de transmisión se muestran en la figura 3 [6].

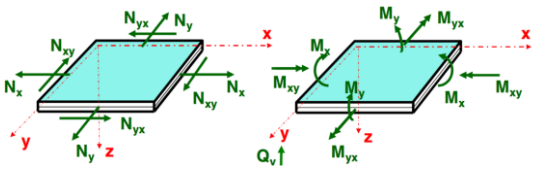



**Fig. 3.** Las fuerzas y los momentos están acoplados por la matriz $[B]$. Están definidos por unidad de longitud del lado sobre el que actúan.

Para el caso de estudio se considera un laminado simétrico, para que la pieza no se pandee después del proceso de fabricación, como se muestra en la figura 4.

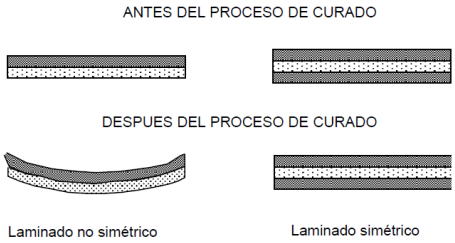



**Fig. 4.** Laminado simétrico.

Al considerar un laminado simétrico, resulta que $[\boldsymbol{B_{ij}}]$=0, por lo que se simplifica la ecuación 16 y se obtiene las ecuaciones 17 y 18.

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{21} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_Y^0 \\ \gamma_{xY}^0 \end{Bmatrix}, \tag{17}$$

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{21} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} k_x \\ k_y \\ k_{xy} \end{Bmatrix}, \tag{18}$$

Donde para calcular los coeficientes A, B y D, se ocupan las ecuaciones de la 19 a la 20.

$$[A_{ij}] = \sum_{k=1}^{n} \left[\overline{Q_{ij}}\right]_k (Z_k - Z_{k-1}) = \sum_{k=1}^{n} \left[\overline{Q_{ij}}\right]_k t_k \,, \tag{19}$$

$$[B_{ij}] = \sum_{k=1}^{n} \left[\overline{Q_{ij}}\right]_k (Z_k^2 - Z_{k-1}^2) \,, \tag{20}$$

$$[D_{ij}] = \sum_{k=1}^{n} \left[\overline{Q_{ij}}\right]_k (Z_k^3 - Z_{k-1}^3) \,. \tag{21}$$

Las deformaciones en cada lamina están dadas por las ecuaciones 21 y 22.

$$\begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_Y^0 \\ \gamma_{xY}^0 \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{16} \\ a_{21} & a_{22} & a_{26} \\ a_{16} & a_{26} & a_{66} \end{bmatrix} \begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix}, \tag{21}$$

$$\begin{Bmatrix} k_x \\ k_y \\ k_{xy} \end{Bmatrix} = \begin{bmatrix} d_{11} & d_{12} & d_{16} \\ d_{21} & d_{22} & d_{26} \\ d_{16} & d_{26} & d_{66} \end{bmatrix} \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix}, \tag{22}$$

donde:

$$\begin{bmatrix} a_{11} & a_{12} & a_{16} \\ a_{21} & a_{22} & a_{26} \\ a_{16} & a_{26} & a_{66} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{21} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix}^{-1}, \tag{23}$$

$$\begin{bmatrix} d_{11} & d_{12} & d_{16} \\ d_{21} & d_{22} & d_{26} \\ d_{16} & d_{26} & d_{66} \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{21} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix}^{-1} . \quad (24)$$

A partir de los cálculos anteriores, se calculan las tensiones en cada lámina en ejes globales, según las ecuaciones 25 y 26:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_k = \begin{bmatrix} \overline{Q_{11}} & \overline{Q_{12}} & \overline{Q_{11}} \\ \overline{Q_{21}} & \overline{Q_{22}} & \overline{Q_{11}} \\ \overline{Q_{16}} & \overline{Q_{26}} & \overline{Q_{66}} \end{bmatrix}_k \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}_k , \quad (25)$$

donde:

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}_k = \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_Y^0 \\ \gamma_{xY}^0 \end{Bmatrix} + Z \begin{Bmatrix} k_x \\ k_y \\ k_{xy} \end{Bmatrix} . \quad (26)$$

En la ecuación 26, el primer término corresponde al de deformaciones en el plano medio y el segundo al vector de curvaturas.

### 3.2. Optimización por enjambre de partículas

La optimización por enjambre de partículas (PSO) es una técnica de búsqueda desarrollada por el Dr. Eberhart y el Dr. Kennedy en 1995, inspirada en el comportamiento social de las aves, y que corresponde al tipo de inteligencia de enjambre, tiene sus raíces en la vida artificial, psicología social, ingeniería y ciencias de la computación [7].

De acuerdo con [8] PSO es un método adaptativo que utiliza agentes o partículas que se mueven a través de un espacio de búsqueda utilizando los principios de: Evaluación, Comparación e Imitación:

*Evaluación*: La tendencia al estímulo de evaluar, es la principal característica de los organismos vivos. El aprendizaje no ocurre a menos que el organismo pueda evaluar, pueda distinguir características del medio ambiente que atraen o características que repelen. Desde este punto de vista, el aprendizaje puede definirse como un cambio que posibilita al organismo mejorar la evaluación promedio de su medio ambiente.

*Comparación:* Los estándares del comportamiento social se realizan mediante la comparación con otros.

*Imitación:* Lorenz [8], asegura que solo los seres humanos y algunas aves son capaces de imitar. La imitación es central para la adquisición y mantenimiento de las habilidades mentales.

El proceso comienza con una población inicial aleatoria, y la búsqueda de la solución óptima se realiza según avanzan las generaciones; sin embargo, a diferencia de los algoritmos genéticos (AG), la optimización por enjambre de partículas (PSO) no necesita operadores de cruce y de mutación. Las posibles soluciones, llamadas partículas, vuelan por el espacio de búsqueda siguiendo las partículas óptimas actuales. Cada partícula realiza un seguimiento de sus coordenadas en el espacio de búsqueda,

que se asocian con la mejor posición actual (*fitness*). El valor de la mejor posición actual se denomina *pbest*, y será almacenado en una base de datos. Cuando una partícula vecina de la posición óptima actual encuentra una posición mejor, almacena la nueva posición óptima con el nombre de *lbest*. A medida que se descubren nuevas y mejores posiciones, éstas pasan a orientar los movimientos de las partículas. Cuando una partícula ha recorrido todo el espacio de búsqueda, la posición óptima global se almacena con el nombre de *gbes*t.

El método de PSO consiste en que, en cada paso de tiempo, se produce un cambio de la velocidad de cada partícula. La aceleración de cada partícula se genera a partir de un término aleatorio. En los últimos años, la optimización por enjambre de partículas ha sido aplicada en muchas investigaciones y casos ingenieriles. Está demostrado que la optimización por enjambre de partículas obtiene mejores resultados de forma más rápida y más económica comparado con otros métodos [4].

En el algoritmo 1 se muestra el pseudocódigo de la operación del algoritmo PSO general [9].

En PSO, en cada iteración, cada $x_i$ se renueva dependiendo de dos valores, el óptimo local $x_i$, y el óptimo global $x$. la renovación se realiza según las ecuaciones 27 y 28.

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_1\left[p_{i,j} - x_{i,j}(t)\right] + c_2 r_2\left[p_{g,j} - x_{i,j}(t)\right], \tag{27}$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1), j = 1,2,\dots,n\acute{u}mero\ de\ laminas\ . \tag{28}$$

**Algoritmo 1. Algoritmo de Optimización por enjambre de partículas (PSO)**

1. **BEGIN** /*Inicio del algoritmo*/
2. Crear una población inicial aleatoria, cada partícula de la población representa una posible solución.
3. Evaluar cada posición de las partículas de acuerdo con una función objetivo.
4. **WHILE NOT** Terminando **DO**
5. Si la posición actual de una partícula es mejor que las previas, actualícela.
6. Determinar la mejor partícula (de acuerdo con las mejores posiciones previas).
7. Actualizar las velocidades de las partículas según la ecuación 27.
8. Mover las partículas a sus nuevas posiciones según la ecuación 28.
9. **END WHILE**
10. Imprime la mejor posición encontrada

### 3.3. Planteamiento del problema de optimización

Diseñar un árbol de transmisión de materiales compuestos (figura 5), en donde las variables a optimizar son el número de láminas y la secuencia de apilamiento, suponiendo que el laminado es simétrico. La función objetivo establecida para este problema es el peso del árbol de transmisión, como se muestra en la ecuación 29.

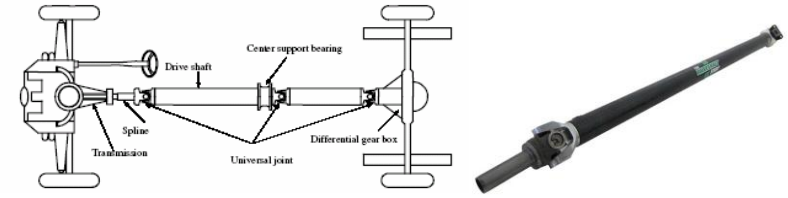



**Fig. 5.** Árbol de transmisión de materiales compuestos.

$$min \qquad f(x) = m = \rho \frac{\pi}{4}(d_0^2 - d_i^2)L, \tag{29}$$

$$vector\ de\ las\ variables\ de\ diseño \quad x = [n, \theta_k, d_i].$$

Sujeto a las siguientes restricciones de diseño:

| | |
|---|---|
| $n > 1$, $\quad donde\ n \in números\ enteros$ | Donde: |
| $-90 \leq \theta_k \leq 90$ | $\rho$= densidad del material |
| $Tcr \geq Tmax$ | $n$=número de láminas |
| $Ncr \geq Nmax$ | $\theta_k$=orientación de las fibras |
| Si se ha producido rotura $\rightarrow$ $m = \infty$ | $d_i$=espesor del árbol |
| | $d_o$=diámetro exterior |
| | $m$=masa |
| | $Tcr$=resistencia a deformación por torsión |
| | $Tmax$=par de servicio |
| | $Ncr$=frecuencia natural |
| | $Nmax$=velocidad de giro |

### 3.4 Fuerzas externas que actúan sobre el sistema

Las fuerzas o cargas externas que sufre un árbol de transmisión con sección hueca, sometido al torque aplicado y considerando la fuerza centrífuga (véase figura 6), resultan como se describe en las ecuaciones 30, 31 y 32.

$$Nx = 0, \tag{30}$$

$$Ny = 2t\rho r^2 w^2, \tag{31}$$

$$Nxy = \frac{T}{2\pi r}. \tag{32}$$

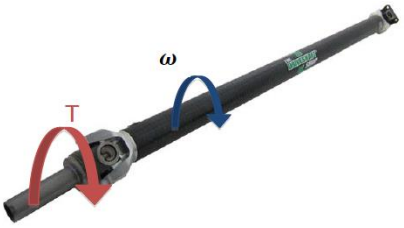



**Fig. 6.** Fuerzas externas que actúan sobre el árbol de transmisión.

Donde:
$\rho$= densidad del material
$\omega$= velocidad angular
$t$=espesor de la pieza
$r$=radio de la sección
$T$=torque aplicado
$N_x$=carga axial

Se propone el material mostrado en la tabla 1 para realizar el diseño del árbol, se muestra las propiedades mecánicas del material compuesto de Carbono/Epoxi USN150. Las direcciones materiales de una lámina se muestran en la figura 7 [6].

**Tabla 1.** Material propuesto para el diseño del árbol de transmisión.

| Concepto | | Unidad | Carbono/Epoxi (USN150) |
|---|---|---|---|
| Módulo de elasticidad en dirección 1 | $E_1$ | GPa | 131,6 |
| Módulo de elasticidad en dirección 2 y 3 | $E_2, E_3$ | GPa | 8,20 |
| Módulo de cortante en plano 23 | $G_{23}$ | GPa | 3,5 |
| Módulo de cortante en plano 13 y 12 | $G_{13}, G_{12}$ | GPa | 4,5 |
| Coef. de Poisson en plano 12 y 13 | $\upsilon_{12}, \upsilon_{13}$ | --- | 0,282 |
| Coef. de dilatación térmica en dirección 1 | $\alpha_1$ | $\times 10^{-6}$/°C | -0,9 |
| Coef. de dilatación térmica en dirección 2 | $\alpha_2$ | $\times 10^{-6}$/°C | 27 |
| Coef. de expansión higroscópico en dirección 1 | $\beta_1$ | --- | 0 |
| Coef. de expansión higroscópico en dirección 2 | $\beta_2$ | --- | 0,4 |
| Resistencia a la tracción en dirección 1 | $S_1^t$ | MPa | 2000 |
| Resistencia a la compresión en dirección 1 | $S_1^c$ | MPa | 1400 |
| Resistencia a la tracción en dirección 2 y 3 | $S_2^t, S_3^t$ | MPa | 61 |
| Resistencia a la compresión en dirección 2 y 3 | $S_2^c, S_3^c$ | MPa | 130 |
| Resistencia al cortante en plano 23 | $S_{23}$ | MPa | 40 |
| Resistencia al cortante en plano 13 y 12 | $S_{13}, S_{12}$ | MPa | 70 |
| Densidad | $\rho$ | kg/m³ | 1550 |
| Espesor de la lámina | $t_{lámina}$ | mm | 0,125 |

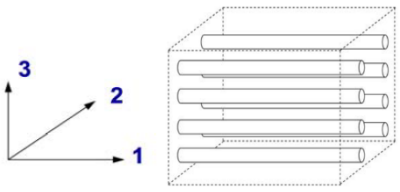



**Fig. 7.** Direcciones materiales en una lámina.

## 4. Implementación computacional

La implementación del algoritmo se programó en el entorno de MATLAB® R2015a, y las corridas se llevaron a cabo en una plataforma computacional con las siguientes características: procesador Intel Core i3 @ 1.80 GHz, con 12GB de memoria RAM y Sistema operativo Windows 10.

Los datos de entrada al algoritmo PSO se muestran en la tabla 2, en donde se consideran los parámetros de fabricación de la pieza, del algoritmo, geométricos y cargas externas.

**Tabla 2.** Datos de entrada al algoritmo PSO.

| **Datos de entrada:** | | |
|---|---|---|
| Velocidad máxima de giro | 9000 rpm | *Cargas externas*<br>*Parámetros geométricos* |
| Par máximo transmitido | 3500 Nm | |
| Diámetro exterior | 0.1 m | |
| Longitud del árbol | 1.5 m | |
| Temperatura de curado | 120 °C | *Parámetros de fabricación de la pieza* |
| Temperatura de servicio | 20 °C | |
| Contenido de humedad | 0.5 % | |
| Coeficiente de seguridad | 2.5 | |
| Tamaño de población | 50 | *Parámetros del algoritmo* |
| Número máximo de iteraciones | 100 | |

Consideraciones basadas en la literatura para obtener el número óptimo de láminas mediante el uso del algoritmo PSO [4]:

- 10-32 (experimentalmente se determina que por debajo de 10 láminas es imposible tener un diseño factible, mientras que con 32 se puede garantizar la existencia del diseño factible).
- N=tamaño de la población, para PSO normalmente es entre 40 y 100.
- w= factor de inercia, típicamente 0.5.
- c1 y c2=factores de aprendizaje, normalmente se toma 2.
- r1 y r2=son números aleatorios entre 0 y 1.

## 5. Resultados y discusión

Estos resultados son valores promedio de haber realizado múltiples corridas del algoritmo y de una selección de muestras de estudio de 30 corridas, las cuales convergieron a la solución. La tabla 3 muestra un resumen de los parámetros de diseño obtenidos por el algoritmo para un árbol de transmisión hecho de fibra de carbono/epoxi USN150, mientras que en la figura 8 se observa la evolución de la búsqueda, mencionando que a cada iteración se aumenta una lámina para realizar una búsqueda exhaustiva, y de las soluciones factibles tomar la de menor valor de función objetivo.

**Tabla 3.** Resultados del algoritmo PSO para el diseño de un árbol de transmisión hecho de fibra de carbono/epoxi.

| i | do (mm) | L (mm) | tk (mm) | n | t (mm) | Secuencia Óptima ($\theta_k$) | Tmax (Nm) | Nmax (rpm) | m (kg) | Tc curado (°C) | Ts servicio (°C) | % humedad | CS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 100 | 1500 | 0.125 | 22 | 2.75 | [18/84/13/-51/9/62/-50/-30/38/-7/62]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |
| 2 | 100 | 1500 | 0.125 | 22 | 2.75 | [1 /-86 /16/-7/-71/33/37/7/-47/33/58]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |
| 3 | 100 | 1500 | 0.125 | 22 | 2.75 | [-21/16/-15/-21/37/-14/0/-82/45/-20/-22]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |
| 4 | 100 | 1500 | 0.125 | 22 | 2.75 | [-23/-20/-58/7/38/-26/-22/55/-82/53/-]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |
| 5 | 100 | 1500 | 0.125 | 22 | 2.75 | [0/11/-20/-48/47/15/53/-41/-22/35]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |
| 6 | 100 | 1500 | 0.125 | 22 | 2.75 | [1/-29/50/-47/-35/74/20/8/-57/-11/-70]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |
| 7 | 100 | 1500 | 0.125 | 22 | 2.75 | [49/-51/-38/-5/-65/15/0/-28/-6/-35/-17]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |
| 8 | 100 | 1500 | 0.125 | 22 | 2.75 | [-4/-62/-87/34/-47/-47/-31/5/-20/-5/5]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |
| 9 | 100 | 1500 | 0.125 | 22 | 2.75 | [-10/-5/-13/-22/-22/-56/-2/21/-17/32/6]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |
| 10 | 100 | 1500 | 0.125 | 22 | 2.75 | [14/46/-27/82/18/-13/-6/76/47/-27/-36]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |
| 11 | 100 | 1500 | 0.125 | 22 | 2.75 | [46/-6/-23/-26/-42/-11/22/83/-61/-59/-22]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |
| 12 | 100 | 1500 | 0.125 | 22 | 2.75 | [49/-27/20/-4/41/14/46/-5/43/-69/14]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |
| 13 | 100 | 1500 | 0.125 | 22 | 2.75 | [-76/30/-14/-51/87/35/-21/35/-41/-49/34]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |
| 14 | 100 | 1500 | 0.125 | 22 | 2.75 | [2/-66/5/34/-21/13/-25/47/10/36/-44]s | 3500 | 9000 | 1.953417 | 120 | 20 | 0.5 | 2.5 |

donde:

$d_o$=*diámetro exterior*
$L$= *longitud*
$t_k$=*espesor de una lámina*
$n$=*número de láminas*
$t$=*espesor total de la pieza*
$\theta_k$=*orientación de las fibras*
$Tmax$=*par de servicio*
$Nmax$=*velocidad de giro*
$m$=*masa*
$Tc$=*temperatura de curado de la pieza*
$Ts$=*temperatura de servicio*
%=*porcentaje de humedad*
$CS$=*factor de seguridad en el diseño*

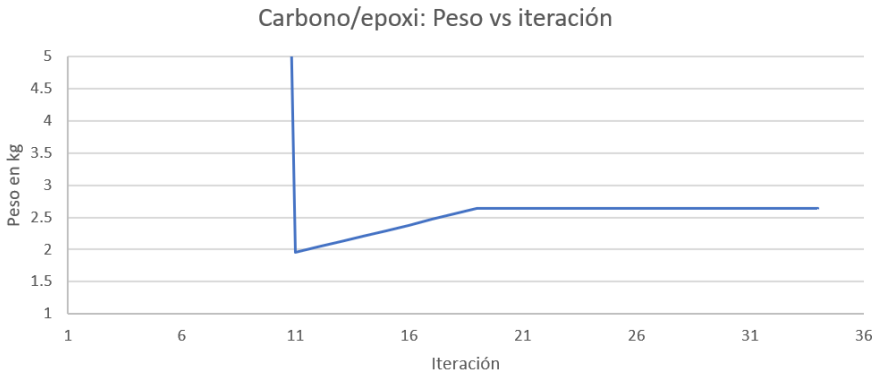



**Fig. 8.** Variación del peso del árbol de transmisión respecto al número de iteración.

Respecto a los resultados mostrados en la tabla 3, la nomenclatura de la orientación de las fibras o *secuencia óptima* de apilamiento es simétrica, es decir la *s* al final del corchete significa que es la primera parte de los laminados respecto al eje medio y que la siguiente mitad tendrá las mismas orientaciones de forma simétrica.

Además, se puede observar que mediante el algoritmo PSO se ha obtenido un diseño óptimo que tiene 22 láminas y un peso de 1.954 kg, para todas las corridas que convergen al óptimo global, sin embargo, las secuencias de apilamiento cambian, esto es razonable ya que se trata de un algoritmo que crea soluciones aleatorias y las va mejorando a cada iteración, además de que existen demasiadas combinaciones posibles que llevan a obtener el objetivo buscado, que para este caso es el menor peso de material para soportar las cargas a las que estará sometida la pieza, por lo que se puede establecer que todas las secuencias de apilamiento son factibles, siempre y cuando converjan al mínimo peso y al número de láminas óptimo.

El costo computacional en este caso no es prioritario, ya que lo que interesa es el resultado, para converger este algoritmo aproximadamente tarda 8 minutos, mientras que en el trabajo presentado en [4] tarda 80 minutos, además la secuencia de apilamiento es muy teórica, por los valores obtenidos, ya que si se quiere fabricar la pieza es difícil obtener las orientaciones sugeridas, en la práctica se suele restringir el diseño a ángulos de 0°, $\pm 45°$ y 90° para facilitar la fabricación.

En trabajos previos, como en [4], para parámetros similares de diseño se han obtenido el mismo número de 22 láminas y un peso similar de 1.95 kg, pero solo reportan una secuencia de apilamiento, sin embargo, ya se explicó que puede haber varias secuencias validas que cumplen con los parámetros establecidos. Mientras que en [5] utilizando parámetros similares, pero no iguales, por ejemplo, el material propuesto, y mediante algoritmos genéticos obtienen 17 láminas y un peso de 4.44 kg. Estos resultados dan certidumbre a los obtenidos por la implementación propuesta en este trabajo.

## 6. Conclusiones y trabajo futuro

Se ha logrado implementar un algoritmo que es capaz de optimizar la secuencia de apilamiento y el número de láminas de fibra de carbono/epoxi para el diseño de un árbol de transmisión, el cual estará sometido a cargas especificas en su funcionamiento, de esta manera, el diseñador podrá ahorrar tiempo en el proceso de diseño cuando requiera de un análisis de este tipo de elementos, variando dentro del código del algoritmo los parámetros necesarios como, la longitud árbol de transmisión, la velocidad de servicio, el diámetro exterior, la temperatura de curado, la humedad del material, así como las características de otro material propuesto. Se puede establecer que la implementación de metaheurísticas en el diseño mediante materiales compuestos es de gran utilidad y con resultados favorables, debido a la complejidad de este tipo de diseños y al gran número de posibles soluciones existentes.

Con el fin de continuar con el desarrollo alcanzado por este trabajo, se propone implementar otros algoritmos evolutivos y de inteligencia colectiva para el diseño con materiales compuestos en diversas aplicaciones, como por ejemplo en el diseño de prótesis de fibra de carbono, con el objetivo de tener un parámetro de comparación entre diferentes metaheurísticas y así el diseñador pueda hacer uso de los resultados con base en la aplicación específica.

**Agradecimientos.** Los autores agradecen al Instituto Politécnico Nacional, a la Sección de Estudios de Posgrado e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Azcapotzalco, al Centro de Innovación y Desarrollo Tecnológico en Cómputo, al Tecnológico de Estudios Superiores de Ecatepec y al Consejo Nacional de Ciencia y Tecnología por el apoyo brindado.

## Referencias

1. Suresh, S., Sujit, P. B., Rao, A. K.: Particle swarm optimization approach for multi-objetive composite box-beam desing. Composite Structures 81, pp. 598–605 (2007)
2. Vázquez Castillo, V., Hernández Lara, D., Merchán Cruz, E. A., Rodríguez Cañizo, R. G., Portilla Flores, E. A.: Implementación de algortimos genéticos para el diseño, optimización y selección de vigas. En: *CORE 2017* (2017)
3. Reguera, F., Cortínez, V. H.: Diseño óptimo de vigas de pared delgada PRF sometidas a cargas dinámicas. Mecánica computacional, vol. XXXII, pp. 3575–3595 (2013)
4. Shengyu, W.: Uso de materiales compuestos en el diseño de un árbol de transmisión. España: Universidad Carlos III de Madrid (2014)
5. Rangaswamy, T., Vijayrangan, S.: Optimal sizing and stacking sequence of composite drive shafts. Materials science, vol. 11 No. 2. India (2005)
6. Navarro, C.: Elasticidad y resistencia de materiales II. España (2014)
7. Montaña, A. C. L. y J. L.: Algoritmos bioinspirados: la evolución biológica en la computación. Cantabria: Departamento de Matemáticas, Estadística y Computación, Universidad de Cantabria (2011)
8. Kennedy, J., Eberhart, R. C.: Intelligent Swarm Systems. New York, USA: Academic Press (2000)
9. Parsopoulos, K. E., Vrahatis, M. N.: Particle Swarm Optimization and Intelligence, Advances and Aplications. United States of America: Information Science Reference, (2010)

ISSN 1870-4069

# Selecting an Optimization Algorithm for the Administration of Human Resources Process in the Production Line of a Textile Enterprise

Alejandro Ernesto González Alegrant[1], Yenny Villuendas-Rey[2], Jarvin Alberto Antón Vargas[3], Cornelio Yáñez-Márquez[4]

[1] Centro Provincial de Información de Ciencias Médicas, Ciego de Ávila, Cuba
alegrant.gonzalez@gmail.com
[2] Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Mexico City, Mexico
yenny.villuendas@gmail.com
[3] Universidad "Máximo Gómez Báez" de Ciego de Ávila, Ciego de Ávila, Cuba
jarvinalberto@gmail.com
[4] Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico City, Mexico
coryanez@gmail.com

**Abstract.** In this paper, we use Particle Swarm Optimization, with the aim of facilitating and optimizing the Human Resources administration process along the lines of production for the "Confecciones Trébol" textile enterprise of Ciego de Ávila city, Cuba, making use of historical data and metaheuristic algorithms. The evident need to optimize the process of allocation of Human Resources in the Production Line of "Confecciones Trébol", led us to analyze the different aspects of the problem, and to use a bio-inspired algorithm to solve it. The validity of the mathematical models of optimization in the solution of a practical problem was determined; contributing to the efficient use of material re-sources. The implementation of a bio-inspired algorithm (PSO) is effectively achieved by complying with the constraints of the defined mathematical model, with which it is possible to obtain feasible solutions to reduce the production time of a textile product.

**Keywords:** optimization, bio-inspired algorithms, textile enterprises.

## 1 Introduction

Within Artificial Intelligence, the branch of bio-inspired algorithms are characterized by emulating the behavior of natural systems and, from them, designing non-deterministic heuristic methods of search, optimization, learning, recognition, simulation and characterization [1].

Hence, Artificial Intelligence is the future of computing and artificial systems in general; bio-inspired algorithms give naturalness to systems that, little by little, will be refined more, in order to achieve greater similarity with natural methods.

The textile industry is an important sector today in society and is constantly developing, which cannot be alien to new technologies [2]. In Cuba, there are several companies related to this area of the economy that have decided to participate in this process. Among them is the "Confecciones Trébol" enterprise of Ciego de Ávila.

This textile enterprise is dedicated to the manufacture of garments and other products and is dependent on work orders that are managed within it. The production of each of the garments is guided by different operations (cutting, sewing, manual work), due to the non-use of information that is managed administratively (average time of completion of the operation by operators); sometimes they don't take the best strategic decisions to optimize the time of making a garment. There are several insufficiencies in the planning of Human Resources in the production line of the textile "Confecciones Trebol "enterprise. To address such issues, we intended to apply a bio-inspired algorithm, in order to optimize the human resources allocation process.

The rest of the paper is as follows: Section 2 gives a panoramic view of the "Confecciones Trébol" enterprise. Section 3 addresses the materials and methods, while Section 4 presents the results. The article ends with conclusions and future works.

## 2 The "Confecciones Trébol" Enterprise

The "Confecciones Trébol" enterprise is an entity that is dedicated to the manufacture of garments and other products and is dependent on work orders that are managed within it. This enterprise is composed of three fundamental workshops: cutting, sewing and manual work; which all depend on the service order.

Once the service order is made, it is taken to the cutting shop, where it is analyzed thoroughly, and depending on the products to be elaborated, the sizes are distributed and the cloths are stretched and then cut. Once the cloths are cut, they are passed to the sewing workshop, where the same process is carried out; but in this case, the cloths cut by tailors, and by the operation that is going to be carried out, are distributed. After finishing in the sewing workshop, the next process is the manual workshop. The manual work workshop includes the operations of thread cutting, final iron, packing, opening seam, pocket plate; it is important to understand that the operations of manual work are of the intercalated type, because they are in the order of the product to be elaborated, and the moment when it is needed.

This entity has a client-server system that covers the activity of Human Resources, Production and pre-payroll management, but the system does not have access, nor does it operate with any information after biweekly or brief closing, and the address cannot make decisions, or analyze past events. The information is stored incrementally in cycles of 15 days in a database, with which important decisions could be made.

Due to the non-use of this information, the enterprise is not able to determine if there has been an indiscriminate use of standards in consumer operations for the same product in different 15 day cycles. Also, it is not possible to determine which seamstresses best perform a certain function or access sequences of operations that have been made in each cycle. There is no access to statistical information of the payments made for each product manufactured. In addition, it is not possible to determine the

seamstresses taking more time to execute the same operation, nor to determine the optimum value of the number of seamstresses operating the same product in relation to those that supply the packages and are responsible for supplying all the positions in the workshop. The analysis of productivity and production levels is made on physical documents instead of accessing this digital information that remains unused.

# 3 Materials and Methods

## 3.1 Particle Based Optimization Algorithm (PSO)

The "Particle Cumulus Based Algorithm" or Swarm Optimization Particle is a population-based metaheuristic technique inspired by the social behavior of flocks of birds in flight as well as the movement of schools of fish. The original PSO algorithm was developed by the psychologist and sociologist James Kennedy and the electronic engineer Russell Eberhart in 1995, from an approach known as the "social metaphor" [3], which describes this algorithm and summarizes the following form: individuals who live in a society have opinions that are part of a "set of beliefs" (the search space) shared by all possible individuals.

The particle cluster (swarm) is a multi-agent system, the particles are simple agents that move through the search space and save (possibly also communicate), the best solution they have found. Each particle has its own "fitness", position and vector velocity that directs its "movement". The movement of the particles in space is guided according to the optimal particles at the present time.

The PSO algorithm is an iterative and stochastic process, which operates on a cluster of particles; the position of each particle is the representation of a potential solution of the problem to be solved, see Table 1.

**Table 1.** Terms of the PSO Algorithm.

| | |
|---|---|
| **Particle** | A person from the cloud. |
| **Position** | Coordinates of an agent in an N-Dimensional space that represents a solution to the problem. |
| **Cloud** | A whole collection of agents. A population of individuals. |
| ***Fitness*** | A number that indicates the quality of a given solution (represented by a location in the solution space). |
| **Pbest** | The best location obtained by a certain particle throughout the process. |
| **Gbest** | The best location of a particle in the whole cloud. |
| **Vmax** | The maximum speed allowed in a given direction. |

The cluster is initialized from the generation of the positions and the initial velocities of the particles. It is possible to generate the positions randomly in the search space (perhaps with the help of a construction heuristic), on a regular basis, or by combining both forms; Once the positions are generated, the fitness of each is calculated and the values of ***fitness_xi*** and ***fitness_pBesti*** are updated, [4].

The speeds are generated randomly, with each component in the interval ***[-vmax; vmax]***, where ***vmax*** will be the maximum speed that a particle can take in each movement); it is not convenient to fix them to zero, because good results are not obtained [5].

According to Sancho Caparini in [6], each particle (individual) has a position in the search space that is determined by a vector, and a speed with which it moves through the search space, equally determined by a vector. Like all particles of the real physical world, they have a capacity of inertia that keeps them in the same direction of movement, and an acceleration (change of speed) dependent on two characteristics mainly:

– Each particle is attracted to the best location that they, personally, has found in their story (best staff).
– Each particle is attracted to the best location that has been found by the set of particles in the search space (better global).

The global version, is where the distance of the social component is given by the difference in the position of the current particle and the position of the best particle found in the complete swarm$\boldsymbol{gBest_i}$, in Figure 1, the pseudocode of the global version of the algorithm is given.

```
1- E= Initialize Population
2- While no stop condition do
3-    For i = 1 to size(E) do
4-       Calculate fitness of each individual
5-       If fitness(Xi) > fitness (pBesti) then
6-          pBest = Xi
7-          fitness (pBesti) = fitness (Xi)
8-       End If
9-       If fitness(pBesti) = fitness (gBest) then
10-         gBest = pBesti
11-         fitness (gBesti) = fitness (pBesti)
12-      End If
13-   End For
14-   For j = 1 to size(E) do
15-      Calculatespeed (Xi)
16-      CalculatePosition (Xi)
17-   End For
18- End While
```

**Fig. 1.** Pseudocode of the PSO algorithm (Taken from [1]).

The global version of the PSO algorithm tends to converge faster because the visibility of each particle is better and closer to the best of the swarm, [1, 14]. S. Hillier and J. Lieberman states in [7] that the mathematical model of the problem is the system

of equations and related mathematical expressions that describe the essence of the problem.

In "Confecciones Trébol" textile enterprise of Ciego de Ávila city, it is necessary to optimize the production operations, and minimize the time. It must be specified that what is sought to be optimized is the production time of a textile product, taking into account that each product is made using a set of operations, carried out by the textile enterprise operators group. The decisions that should be made are subject to the number of operators per job in the production line, adjusted to the average time per unit of product and the types of products to be developed. According to authors in [8], an optimization model is constructed following three fundamental steps:

1. Define the essential or decision variables.
2. Construction of the system of restrictions.
3. Construction of the objective function.

Each decision variable is identified with each of the activities in which the problem that is studied is broken down, due to this the conceptual description for the variables in this problem are given in Table 2.

**Table 2.** Conceptual description of the decision variables for a product.

| Variables | Description |
|---|---|
| $O_n$ | Number of operators per job (Unknown) |
| $T_n$ | Average time per product unit |
| $n \geq 1$ | Initial conditions for each job |

The optimization problem is subject to several restrictions, described as follows.

Regarding the condition of work, it is based on the fact that there must be at least one operator per job who intervenes in the production:

$$O_n \geq 1. \tag{1}$$

The workshop currently has 64 operators that work in the textile enterprise, therefore the following restriction is obtained:

$$\sum_{i=1}^{n} O_i \leq cant_{oper}. \tag{2}$$

In the workshop different operations are carried out that are adjusted to a time constraint determined by the work to be done in 60 minutes (1 hour); all this, in an 8-hour day (expressing it in minutes):

$$\frac{O_1}{Jt} + \frac{O_2}{Jt} + \frac{O_3}{Jt} + \ldots\ldots + \frac{O_n}{Jt} \leq 60. \tag{3}$$

The objective function directly expresses the objective or purpose pursued depending on the problem to be investigated. The objective function for this model is constituted in the following way:

$$\min \rightarrow f(O) = \frac{O_1}{T_1} + \frac{O_2}{T_2} + \cdots + \frac{O_n}{T_n}. \quad (4)$$

Table 3 summarizes the description of the optimization problem to be solved.

**Table 3.** Optimization problem.

| **Variables** | **Description** |
|---|---|
| $O_n$ | Number of operators per job (Unknown) |
| $T_n$ | Average time per product unit |
| $n \geq 1$ | Available jobs |
| **Restrictions** | **Description** |
| $\boldsymbol{O_n \geq 1}$ | Working condition: at least one operator per job, which is involved in the production |
| $\sum_{i=1}^{n} O_i \leq cant\_oper$ | Availability of workforce |
| $\frac{O_1}{Jt} + \frac{O_2}{Jt} + \frac{O_3}{Jt} + \ldots\ldots + \frac{O_n}{Jt} \leq 60$ | Work to be done in 1h |
| **Objective Function** | |
| $\min \rightarrow f(O) = \frac{O_1}{T_1} + \frac{O_2}{T_2} + \cdots + \frac{O_n}{T_n}$ | |

## 4 Using PSO to Solve the Human Resources Administration for "Confecciones Trébol" Textile Enterprise

The authors state in [9] which PSO potential solutions, called particles, are within the search space, following the optimal particle, which are updated with an internal speed. They also have a memory, which is an important aspect for this algorithm. In addition, while PSO is easy to implement and has few parameters to adjust, it has successful applications in various areas: optimization functions, training of artificial neural networks, and diffuse control system [9, 10].

Before coding and implementation, the authors state in [11, 12], to verify that the algorithm will use few resources, the most important being the time it takes to run and the amount of memory space required. The analysis of the algorithmic complexity is shown, being this of $O(n * m) \equiv O(n^2)$.

As a first step of the algorithm consistent with the generation of the population, it was generated randomly. The number of particles to be generated is given in a typical range of [20, 40]. This value is subject to previous experiments carried out during the implementation and adjustment of the algorithm, looking for the results to be consistent with expected results.

The PSO model can be purely cognitive if for the particle trend it depends on the best positions found in your personal past; or, it can be purely social if this tendency is proportional to the past of the cluster. If the model has both components (cognitive and social) the velocity of each particle can determine how they converge to the optimal value. That is why Shi and Eberhart [13] proposed an improvement of the basic algorithm, modifying the formula for updating the speed and introducing a new variable called the inertial factor ***w***.

Both the process of exploration and exploitation of the search space, as well as the control that allows the speed does not exceed its established limits, are balanced by the inertia factor. This factor regulates the speed, multiplying its weight by the previous speed. A large weight of inertia facilitates the global search, while a small weight of inertia facilitates local search. Generally the best value for ***w*** is dependent on the problem, by incorporating the inertial factor ***w*** the update of the speed is determined in the formula shown in Figure 2, where the variable ***V_id*** is the speed of the previous iteration of the particle ***i*** in dimension ***d*** (where ***d*** is the number of problem variables), $r_1$ and $r_2$ are random values in the range [0,1], $p_1$ and $p_2$ are the personal and social learning coefficients respectively, $pb_{id}$ (personal best) is the best personal position found by the particle ***i*** in the dimension ***d***, $pos_{id}$ is the current position of the particle ***i*** in the dimension ***d*** y ***gbd*** (global best) is the global position vector, best found by all the particles, obtained in the objective function and it is updated after each iteration of the algorithm. The value of the inertial factor ***w*** can remain fixed or it can be linearly decremented in each flight cycle, in the case of our investigation the inertial factor is fixed.

$$v_i \leftarrow \omega \cdot v_i + \varphi_1 \cdot rand_1 \cdot (pBest_i - x_i) + \varphi_2 \cdot rand_2 \cdot (lBest_i - x_i)$$
$$x_i \leftarrow x_i + v_i$$

**Fig. 2.** Formula proposed by Cárdona y Nieto in [1, 4].

In order to control the possible solutions generated in such a way that they were adjusted to the feasible solutions, the strategy of keeping the minimum value on the components that were below the allowed value was adopted; in the same way a respective strategy was adopted for the maximum values. Table 4 shows the values for each of the parameters of the algorithm.

Throughout the development of the system and adjustment of parameters of the algorithm, it was necessary to check the efficiency of the same; for this purpose, predefined and real data were used. Within the first runs of examples, a text-type file was created where the times were initialized by operation, parameters such as the number of operators, number of particles, the coefficient of cognitive or personal learning, the overall learning coefficient were adjusted and the values of the inertial factors.

**Table 4.** Parameters for the PSO algorithm.

| Parameters | Values |
|---|---|
| Number of particles | 40 |
| Number of iterations | 100 |
| Inertia factor value | 0,1 |
| Value of the personal learning coefficients | 0,1 |
| Value of social learning coefficients | 0,9 |

For the adjustment of the appropriate values for the inertial weights, it was taken into account that they depended mainly on the problem to be solved. In the process of trial and error, different values of weights were tested, each time closer to the final solution, where the set of values was analyzed, leading to the best results in the problem.

Based on the other part of what is addressed by Kowalski and Lukasik in [14] where *"the choice of the parameter is based on the results of the numerical simulation that indicate the quality of the solution: the average value of error with its standard deviation, the best result obtained by the swarm and, additionally, the speed of its convergence "*, it is decided to decrease the value of the weight inertia parameter. Finally, a sequence of the best global particles in each of the iterations generated by the algorithm, taking into account the lower fitness value of the particles is obtained.

After adjusting parameters of the model and the algorithm, it was necessary to check the efficiency of the same, for this purpose, predefined and real data were used. Within the first run of examples, a text file was created where the times were initialized by operation, parameters such as the number of operators, number of particles, the values of inertial factors, the coefficient of cognitive learning were adjusted or personal and the overall learning coefficient. In the test program, several iterations were obtained and each obtained results closer and closer to the final solution, which was only obtained when the stop condition was met (maximum number of iterations). Finally a sequence of the best global particles in each of the iterations generated by the algorithm taking into account the lower fitness value of the particles. An example of this process can be seen in Figure 3.

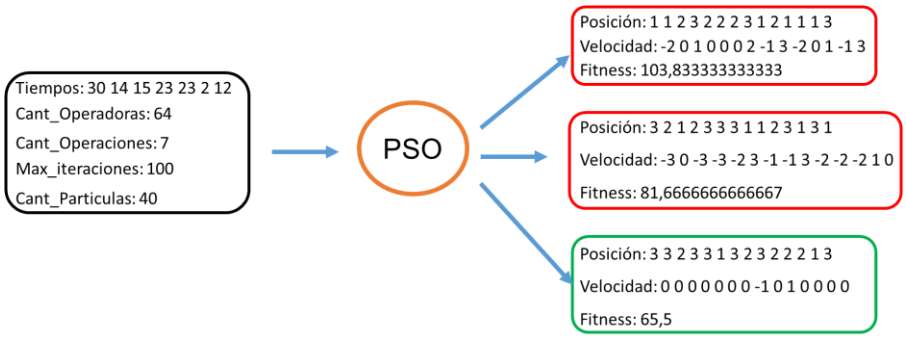



**Fig. 3.** Examples Algorithm PSO in several iterations.

Now with real data, as shown in Figure 4, once the operations to be carried out by a garment have been loaded, the possibility of choosing how to carry them out in three priority groups is offered.

When the priorities of these operations have been delimited, the algorithm will executed it, and will be able to offer an optimal solution of how to organize the human resources of the workshop in order to optimize the time of making a garment.

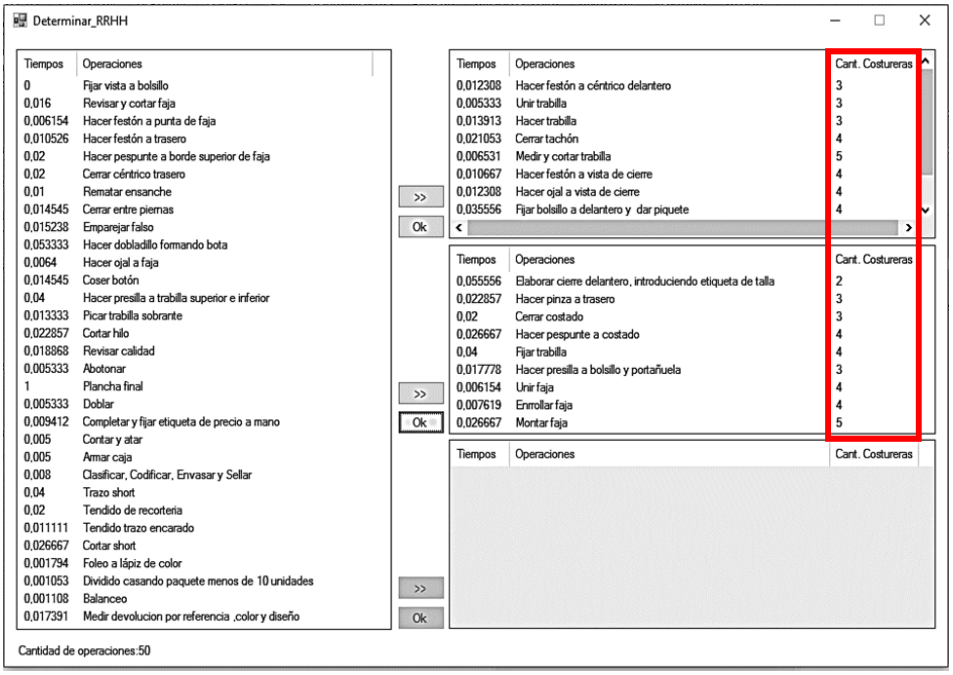



**Fig. 4.** Final result of the allocation of Human Resources.

Therefore, the algorithm correctly satisfies the constraints defined in the model, returning values within the range of acceptable values. Once again, fulfilling the condition of stoppage and the objective of determining the allocation of number of operators per operation of a product to be elaborated it in the textile enterprises. An analysis of the process of allocation of Human Resources of the "Confecciones Trébol" textile enterprises was achieved, which allowed defining a mathematical model of optimization adjusted to the characteristics of the process of making a textile product, taking into account the inherent restrictions of the process and the real capabilities of the entity.

## 5 Conclusions

The evident need to optimize the process of allocation of Human Resources in the Production Line of the "Confecciones Trébol" textile enterprise, led to analyzing the

different aspects of each of the methods and heuristic techniques to select the appropriate one according to the speed and certainty of the answers.

The validity of the mathematical models of optimization in the solution of a practical problem was determined, contributing to the efficient use of material resources. The implementation of a bio-inspired algorithm (PSO) was effectively achieved by complying with the constraints of the defined mathematical model, with which it is possible to obtain feasible solutions to reduce the production time of a textile product.

Finally, future work will continue with the study of the historical data of the textile enterprise, for a future adjustment of the model, achieving a greater accuracy in the response of the PSO something-rhythm.

## References


1. Cárdenas Cardona, A.: Inteligencia artificial, métodos bio-inspirados: un enfoque funcional para las ciencias de la computación. Universidad Tecnológica de Pereira (2012)
2. Becerra Días, L.: Metodologías para fortalecer los mecanismos evaluadores de la efectividad económica a través de la reducción de los costos de producción en la empresa de confecciones Trébol. Ciego de Ávila, Cuba (2012)
3. Eberhart, R. C., Shi, Y., Kennedy, J.: Swarm Intelligence. The Morgan Kaufmann Series in Evolutionary Computation (2001)
4. Nieto, J. G., Polo, G. J. L.: Algoritmos basados en cúmulos de partículas para la resolución de problemas complejos. (2006)
5. Kennedy, J. F., Kennedy, J., Eberhart, R. C., Shi, Y.: Swarm intelligence. Morgan Kaufmann (2001)
6. Sancho Caparini, F.: Swarm Inteligence. PSO: Optimización por enjambres de Partículas, Sevilla, Dpto Cienc. Comput. E Intel. Artif. Univ. Sevilla (2015)
7. Hillier, F. S. L., Hillier, G. J. F. S., Lieberman, G. J.: Introducción a la Investigación de Operaciones. McGraw-Hill (1989)
8. López Gutiérrez, N., Albelo Martínez, M., del Valle Cruz, A., Ruíz de Zárate del Cueto, J.: Elementos de Álgebra lineal y Programación lineal. 2da ed. La Habana: Felix Varela (2008)
9. Eberhart, R. C., Shi, Y.: Comparison between genetic algorithms and particle swarm optimization. In: International Conference on Evolutionary Programming, pp. 611–616 (1998)
10. Gudise, V. G., Venayagamoorthy, G. K.: Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks. In: Swarm Intelligence Symposium, SIS'03, Proceedings of the 2003 IEEE, pp. 110–117 (2003)
11. Mañas, J. A.: Análisis de Algoritmos: Complejidad. Disponible en: http://www.lab.dit.upm.es/~lprg/material/apuntes/o/index.html [Accedido: 04-jun-2017]
12. Ruz Valenzuela, V.: Manual Análisis de Algoritmos (2003)
13. Shi, Y., Eberhart, R. C.: Parameter selection in particle swarm optimization. In: International Conference on Evolutionary Programming, pp. 591–600 (1998)
14. Kowalski, P. A., Lukasik, S.: Experimental study of selected parameters of the krill herd algorithm. In: Intelligent Systems' 2014, Springer, pp. 473–485 (2015)

ISSN 1870-4069

# Generating Trading Strategies in the Mexican Stock Market: A Pattern Recognition Approach

David Ricardo Montalván Hernández, Ricardo Barrón Fernández, Salvador Godoy Calderón

Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico City, Mexico
{davidricardo888, barron2131, sgodoyc}@gmail.com

**Abstract.** With the digitalization of financial markets, namely, stock markets the development of algorithms and computational techniques in order to determine trading strategies has gained relevance as much as in academia as in the industry. This article explores the use of pattern recognition techniques (Support Vector Machines, Multilayer Perceptron and C4.5) as tools for finding trading strategies in the Mexican Stock Market. Our results show that, statistically speaking, the methods proposed here, cannot outperform the so called Efficient Market Hypothesis in its weak version. Nonetheless, this paper presents a labeling method for financial time series, which permits future investigations using other supervised learning techniques.

**Keywords:** trading strategies, Mexican stock market, pattern recognition, support vector machines, multilayer perceptron, C4.5.

## 1 Introduction

The increase in computing power, the digitalization of financial markets and the opportunity of generate big profits, have motivated to a large degree the research and development of computational algorithms whose purpose is the guidance in the investment decision process.

Even when the basic idea is: "buy low and sell high", given the uncertainty of financial markets, this research has used mathematical and computational techniques in order to create models that help in the trading decisions (when is the right time to buy or sell). Namely, the use of techniques in the field of artificial intelligence have gained notoriety.

The objective in this paper is the proposal of a method or algorithm in order to generate trading signals for the Mexican stock market.

We will try to find a set of trading signals that will generate profits (losses) bigger (minor) than the ones generated by the *Buy and Hold* strategy (the main benchmark in this kind of research), which is discussed in Section 3 and consists basically in the following actions:

- Fix a time period $[0, T]$.
- Buy at time 0 at price $P_0$.
- Sell at time $T$ at price $P_T$.
- The percetange profit (loss) is given by $\frac{P_T - P_0}{P_0}$.

This work also compares the performance of various pattern recognition methodologies when applied to the generation of trading strategies.

Lastly, two contributions are considered. The first is the analysis of Mexican stock market from a pattern recognition perspective (to the best of our knowledge, this is the first time that this market is analyzed using such perspective), the second is the proposal of a method for labeling financial time series in order to apply supervised learning techniques.

The rest of the paper is organized as follows. A brief review of related works is presented in Section 2. Some basic financial concepts are described in Section 3. Section 4 gives the experiments and results obtained, as well as the data used. Finally, Section 5 supplies the conclusions and future work to be done.

## 2 State of the Art

In order to find trading strategies that consistently beat the *Buy and Hold* strategy, several artificial intelligence techniques have been explored, for example, one of the first works using such techniques is [1], in which genetic programming is used in order to create the strategies In this work the U.S. stock market is analyzed using *S&P 500* stock index as a benchmark; they consider transaction costs but do not obtain favorable results.

In [6], chart heuristics are used to detect a chart pattern called *bull flag* They beat *Buy and Hold* strategy but do not consider transaction costs.

In [9], the work from [1] is taken up, omitting transaction costs and considering another way of creating the trees. They obtain positive results in stable and bearish (trending down) markets, but not in bullish (trending up) markets.

The authors in [8] also use genetic programming in order to determine the strategies, being the main difference the use of monthly prices (not daily as is the usual practice) They consider transaction costs and beat *Buy and Hold* strategy.

[7] use *Perceptually Important Points* (PIP) and *Symbolic Aggregate Approximation* (SAX) in order to reduce the dimensionality of the data and express it using symbols. Once they have these symbols they use a genetic algorithm for obtaining the trading strategy. They obtain positive results but do not consider transaction costs.

In [4] an event-based time scale is considered and *directional changes* are defined. Using this concept they are able to generate buy and sell signals that beat *Buy and Hold* strategy even when considering transaction costs and risk adjusted performance.

The authors in [3], propose the use of biclustering mining to discover effective technical trading patterns that contain a combination of indicators from historical financial data series. A modified K nearest neighborhood method is applied

to classification of trading days in the testing period. They outperform *Buy and Hold* strategy but do not consider transaction costs.

Finally in [12] , an evolutionary trend reversion model, based on an extension of the XCS (extended classifier system) algorithm, is proposed. They beat *Buy and Hold* strategy and obtain favorable results analyzing several risk-adjusted performance measures. In this work the model obtained is a set of *if..then* rules.

## 3 Basic Concepts in Finance

### 3.1 Technical Analysis

According to [5], in its basic form, technical analysis is the study of historical prices and volume from a stock series in order to determine future trends on its price. The basic assumptions for this kind of analysis are:

- Prices are uniquely determined by the interaction between supply and demand.
- Prices move following trends.
- Changes in supply and demand cause trend reversions.
- Changes in supply and demand can be detected using charts.
- Patterns in charts tend to repeat.

Technical analysis also makes the assumption that the information of all factors (including psychological factors such as greed, fear, miss information, etc...) affecting supply and demand curves, is already reflected in the stock's price.

### 3.2 Efficient Market Hypothesis (EMH)

This hypothesis, proposed by Nobel prize winner Eugene Fama in the 1960's, states that all the observed changes in the prices are caused only by the new available information, that is, historical data (of any kind) has no relevance when determining future trends. In particular, this hypothesis tell us that the use of technical analysis is unprofitable. There are three versions of EMH:

**Weak version of EMH** In its weak version, the Efficient Market Hypothesis, states that historical prices do not affect future price movements, thus, technical analysis is futile for the generation of trading strategies. This version only refers to historical prices and volume and leaves the door open for other types of data such as financial statements reports or news.

**Semi-strong version of EMH** In its semi-strong version, the Efficient Market Hypothesis, states that publicly available historical information (prices, financial statements reports, news, etc..) is useless when predicting future price movements. Thus, only private/classified information might be useful for predicting future trends.

**Strong version of EMH** Finally, in its strong version, the hypothesis states that even private/classified information cannot be used to outperform the market.

### 3.3 Buy and Hold strategy (BH)

This is the strategy proposed by the Efficient Market Hypothesis, and consists of the following actions:

- Fix a time period $[0, T]$.
- Buy at time 0 at price $P_0$.
- Sell at time $T$ at price $P_T$.
- The percetange profit (loss) is given by $\frac{P_T - P_0}{P_0}$.

According to the EMH, the profit (loss) obtained by the BH strategy is the maximum (minimum) profit (loss) that one can obtain in a systematic way. Hence, this strategy will be used as a benchmark when comparing with our proposed algorithms.

### 3.4 Titles Referenced to Shares

According to Mexican Stock Exchange's website [1], the Mexican market has Titles Referenced to Shares (TRAC's), which are participation certificates representing equity investment trusts. Their primary objective is to replicate the behavior of the stocks or portfolio which they are referred to (underlying), that is, TRAC's are Exchange Traded Funds (ETFs).

The most important TRAC is the one that represents the Mexican Stock Market as a whole and it is called NAFTRAC.

Thus, our objective is finding trading strategies able to beat BH strategy using NAFTRAC data.

## 4 Experiments and Results

### 4.1 Datasets

We used daily price data (open price, maximum price, minimum price, adjusted close price) from Yahoo Finance [2] for NAFTRAC for a time period between February 4th 2014 up to April 5th 2018.

It is worth mentioning that this is an unlabeled dataset, hence, we first need to find a way to label it in order to use supervised pattern recognition techniques. The approach taken is based on the idea of "given the historical prices, what should one have had to do in order to make profits?"

[1] https://www.bmv.com.mx/en/markets/instruments
[2] https://finance.yahoo.com/

**Training set and test set** For obtaining the training and test sets, the data was divided in three-months periods, starting on the day February 4th 2014. Following a three-months training period, there is a three-months test period, which later will become the new three-months training period, that is, we use a rolling window to separate the dataset as shown in the table below.

**Table 1.** Training and test set separation.

| Start training | End training | Start test | End test |
|---|---|---|---|
| 2014-02-04 | 2014-04-30 | 2014-05-02 | 2014-07-31 |
| 2014-05-02 | 2014-07-31 | 2014-08-04 | 2014-10-31 |
| 2014-08-04 | 2014-10-31 | 2014-11-03 | 2015-01-30 |
| 2014-11-03 | 2015-01-30 | 2015-02-03 | 2015-04-30 |

Using the procedure above, we were able to obtain 16 training/test datasets.

### 4.2 Labeling Process

As mentioned before, we need to label each observation in the training datasets according to one of the three possible actions: buy, sell or hold. To achieve this we used an Estimation Distribution Algorithm (EDA, [11]), namely, we used a version of an Univariate Marginal Distribution Algorithm (UMDA).

This algorithm tries to find the best strategy for the given training period, that is, the strategy that would have generated a bigger (minor) profit (loss) compared to the BH strategy.

Each individual in the population was encoded as a vector, $\mathbf{x}$, representing a trading strategy and having length equal to the number of trading days in the training period. Each entry in the vector takes a value in the set $\{-1, 0, 1\}$, where $-1, 0, 1$; represents a sell, hold and buy signal respectively. Thus the *i-th* component is the decision taken on day $i$. The algorithm finds the combination maximizing the profit, which is measured as the **Excess Return** over the BH strategy, that is, the return generated by the vector $\mathbf{x}$ minus the return generated by BH.

### 4.3 Features

For each day, the open, minimum, maximum and adjusted close prices were used as features.

### 4.4 Results

For every training and test set, the following models were tested[3]:

[3] We chose these models since they are among the most popular ones used in the pattern recognition literature (see [2] and [10] for their mathematical description) .

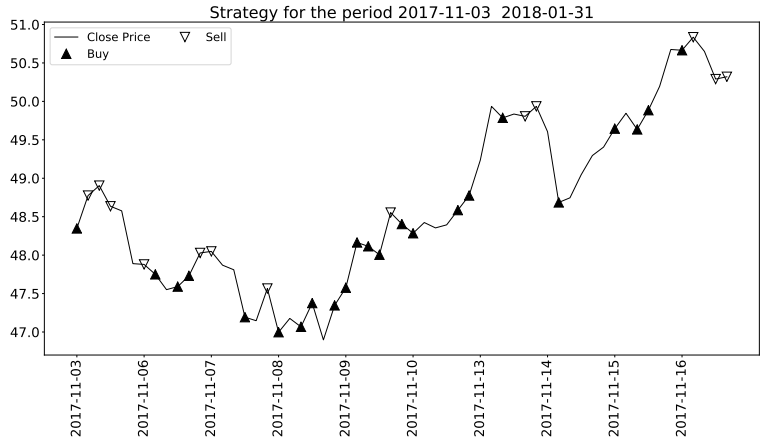



**Fig. 1.** Results from the labeling process.

- Support Vector Machine with gaussian kernel and assigning class weigths of $0.45, 0.10, 0.45$ for classes $-1, 0, 1$ respectively.
- Multilayer perceptron with a hidden layer with 10 neurons and *ReLu* as activation function.
- C4.5 tree with max depth of 5.

As described above, the performance measure was **Excess Return** which is calculated using the following assumptions:

- No short-sales allowed, this means that one can only sell if a buy action occurred in the past (we cannot sell something that we do not own).
- Once a trading signal is activated, we have to wait until we see a different one, so no repetitions of the same signal are allowed. This avoids excessive buys or sells.
- Since we are using end of day data, if we have a buy or sell signal on day $t$, then the buy or sell price (execution price) is the average between the minimum and maximum price at day $t + 1$.
- The cost of every transaction is equal to a 0.25% over the total cost. For example, if a stock was bought at a price of \$10, then the actual monetary amount paid for it is equal to $\$10(1 + 0.0025)$; likewise if a stock was sold at a price of \$10 we end up receiving. a monetary amount of $\$10(1 - 0.0025)$.

For every test set and every model we obtained the following results.

## 5 Conclusions

As we can observe, among the three models used, the best results were obtained by the Support Vector Machine.

**Table 2.** Results obtained for every model.

| | Excess Return | | |
|---|---|---|---|
| Test set | SVM | MLP | C4.5 |
| 2014-05-02 / 2014-07-31 | -0.013 | -0.067 | -0.067 |
| 2014-08-04 / 2014-10-31 | 0.0 | -0.006 | -0.006 |
| 2014-11-03 / 2015-01-30 | 0.0 | 0.0 | 0.0 |
| 2015-02-03 / 2015-04-30 | -0.058 | -0.066 | -0.07 |
| 2015-05-04 / 2015-07-31 | 0.0 | 0.0 | -0.039 |
| 2015-08-03 / 2015-10-30 | 0.02 | 0.0 | 0.025 |
| 2015-11-04 / 2016-01-29 | 0.113 | 0.054 | 0.091 |
| 2016-02-02 / 2016-04-28 | -0.053 | -0.025 | -0.065 |
| 2016-05-02 / 2016-07-29 | 0.046 | -0.025 | -0.024 |
| 2016-08-01 / 2016-10-31 | 0.003 | -0.027 | -0.027 |
| 2016-11-01 / 2017-01-31 | 0.028 | 0.004 | 0.01 |
| 2017-02-01 / 2017-04-28 | -0.008 | -0.045 | -0.047 |
| 2017-05-02 / 2017-07-31 | -0.029 | -0.029 | -0.014 |
| 2017-08-01 / 2017-10-31 | 0.016 | 0.054 | 0.045 |
| 2017-11-03 / 2018-01-31 | -0.033 | 0.032 | -0.06 |
| 2018-02-01 / 2018-04-05 | 0.0 | 0.0 | 0.069 |
| Overall sum | 0.32 | -0.174 | -0.179 |
| Average | 0.002 | -0.01 | -0.011 |
| Number of positive excess returns | 6 | 4 | 5 |
| Number of negative excess returns | 6 | 8 | 10 |

Unfortunately, even though when in this model we obtained a positive average excess return, using a t-test for the mean (with a confidence level of 95%) we found that is not possible to reject the null hypothesis (the true mean is zero) thus, statistically speaking, we cannot conclude that this method beats the BH strategy systematically.

Nonetheless its worthwhile mentioning that thanks to the labeling method we can further explore other types of supervised learning techniques whether using a symbolic or sub-symbolic approach.

The future directions for this work might be:

- Explore symbolic approaches for supervised learning, such as Extended Classifier System (XCS).
- Analyze the inclusion of technical indicators.
- Use another set of features.
- Use other sampling frequencies.
- Use risk-adjusted performance measures.

## References

1. Allen, F., Karjalainen, R.: Using genetic algorithms to find technical trading rules. Journal of Financial Economics 51(2), 245–271 (1999)

2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, 1st edn. (2006)
3. Huang, Q., Wang, T., Tao, D., Li, X.: Biclustering learning of trading rules. IEEE Transactions on Cybernetics (2015)
4. Kampouridis, M., Otero, F.E.: Evolving trading strategies using directional changes. Expert Systems with Applications 73, 145–160 (2017), http://dx.doi.org/10.1016/j.eswa.2016.12.032
5. Kirkpatrick, C.D., Dahlquist, J.R.: Technical Analysis The Complete Resource for Financial Market Technicians. Pearson Education, Inc., 2nd edn. (2011)
6. Leigh, W., Modani, N., Purvis, R., Roberts, T.: Stock market trading rule discovery using technical charting heuristics. Expert Systems with Applications 23(2), 155–159 (2002)
7. Leitao, J.: Combining rules between PIPs and SAX to identify patterns in financial markets (2016)
8. Lohpetch, D., Corne, D.: Outperforming Buy-and-Hold with Evolved Technical Trading Rules: Daily, Weekly and Monthly Trading (2010)
9. Potvin, J.Y., Soriano, P., Maxime, V.: Generating trading rules on the stock markets with genetic programming. Computers and Operations Research 31(7), 1033–1047 (2004)
10. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, 1st edn. (1993)
11. Simon., D.: Evolutionary Optimization Algorithms. Biologically Inspired and Population-Based Approaches to Computer Intelligence. John Wiley & Sons, 1st edn. (2013)
12. Zhang, X., Hu, Y., Xie, K., Zhang, W., Su, L., Liu, M.: An evolutionary trend reversion model for stock trading rule discovery. Knowledge-Based Systems (2015)

ISSN 1870-4069

# Comparison of Three Data Expansion Algorithms for Air Pollution Data in Irregularly Placed Measuring Stations

Hiram Calvo

Instituto Politécnico Nacional, Center for Computing Research, Mexico City, Mexico
hcalvo@cic.ipn.mx; hiramcalvo.com

**Abstract.** In some major cities there are stations for measuring atmospheric pollutants. These stations are often distributed in an irregular pattern. In order to predict pollutant's behavior, it is necessary to order data in a regular, uniform grid. For this, we employ expansion data algorithms. Our work centers on the software implementation and evaluation of three of these algorithms: Cressman, Voronoi and Kriging. For evaluation, we use real data of atmospheric pollutants, including the actual position of stations that measure air pollutants in Mexico City. We use actual values taken from different pollutants.

**Keywords**: air pollution, data expansion, interpolation, Cressman, Voronoi, Kriging, OpenGL.

## 1. Introduction

In some major cities there are stations for measuring atmospheric pollutants. These stations are often distributed in an irregular pattern. In order to predict pollutant's behavior, it is necessary to order data in a regular, uniform grid. Such an interpolated field is critical for calculus such as wind field divergence, reduction, data value edge, and initialization of calculus of pollutant transport [13]. In real world, it is impossible to obtain exhaustive values of data at every desired point because of practical restrictions. Because of this, the interpolation is important to graph, to analyze and to understand bi-dimensional data [5].

There are expansion data algorithms well known for data interpolation. These algorithms are used in a wide range of applications, but they have not been compared for the particular problem of interpolating scattered data of air pollutants. In this work we describe the software implementation and evaluation of three of these algorithms: Cressman, Voronoi and Kriging.

For evaluation, we use real data of atmospheric pollutants, including the actual position of stations that measure air pollutants in Mexico City. We use actual values taken from different pollutants.

Firstly, we present a review of the current situation with the software that allows data interpolation. We particularize on the software related to geological applications. In Section 2, we describe the mathematical fundamentals of each of the algorithms selected (Cressman, Voronoi and Kriging) and present some details on their implementation in Section 2. Afterwards, we compare the algorithms using the actual

data on the air pollution in Mexico City, including the exact positions of the measuring stations. For comparison, we follow the procedure of removing one point from the actual data set; then we interpolate the data for this point, and after that we compare the interpolated value with the actual value for this point. We apply the procedure repeatedly to obtain a measure of the quality of interpolation of our algorithm. This is explained in detail in Section 4.

We show the implementation details in Section 3 which allow interactive visualization of surfaces. For the visualization purposes, we use OpenGL and the Graphics Library Utility Toolkit (GLUT).

### 1.1. Related Software

To our knowledge, no particularly specialized software currently exists for the air pollution modeling. However, there are programs for geologic modeling that include at least one of the algorithms presented. One example of these programs is **GMS: Submarine Surface Modeling System**. This program has the ability to perform surface interpolation using the Kriging algorithm in two variants:

**Zonal Kriging.** A variogram can be defined for each stratographic area. Zones are defined by the material identifiers associated with the cells of a three-dimensional mesh which surrounds scattered points. When a cell is interpolated in a zone, only the scattered points found in the same zone of the mesh are used to interpolate the cell.

**Indicator simulation.** Indicator Kriging is a form of Kriging used for interpolating integral or zonal information instead of scalar values. For each scattered point a material identifier is assigned, and material identifiers are interpolated to the mesh cells. This makes possible to establish stratographic zones for a model of submarine surface using Kriging.

Other programs including the algorithms studied are those related to mesh processing. There are many programs for mesh processing. All these programs are for general mesh processing. These programs have interpolating algorithms implemented, but they are not specialized in data expansion, much less particularly in the problems of air pollution. Examples of these programs are: Algor, ARGUS MeshMaker, EarthVision, FEMAP, Finite-Octree, GOCAD, GRIDGEN, Hypermesh, ICEM CFD, Patran, STRATAMODEL, and TrueGrid.

Among the data expansion algorithms used we will focus on three of them (namely, Cressman, Kriging and Voronoi), which have been traditionally used in the problems similar to the one we are trying to solve [10]. We describe those algorithms in the following section.

## 2. Algorithms

In this section, we describe three algorithms for data expansion: Cressman [4], Kriging [11], and an algorithm based in Voronoi triangulation [8, 2, 1].

### 2.1. Cressman

A common way of addressing scattered data for its interpolation into a regular grid is to assume that the grid value is a weighted value of the surrounding data values, that is:

$$C_{ij} = \frac{\sum_{k=1}^{n} C_k W_k(r)}{\sum_{k=1}^{n} W_k(r)}, \tag{1}$$

where $C_k$ is the value measured at the $k$-th measuring station, $W_k(r)$ is the weighting function, and $r$ is the distance from the grid point to the station.

Cressman proposed a procedure for height surface analysis, in which he used the following weighting factor [4]:

$$W(r) = \frac{R^2 - r^2}{R^2 + r^2}, \tag{2}$$

where $R$ is the distance from which the weighting factor tends to zero; that is, the 'influence ratio'. This weighting technique helps to the interpolation procedure in scattered data areas. Decreasing values of $R$ are used in consecutive tests to analyze a scale spectrum. Values obtained from each test are averaged to produce the final field.

Endlich and Mancuso [7] combined the polynomial adjustment and the distance weighting in their interpolation technique. An adjustment of minimal-quadratic to a first order polynomial was performed using the closest five stations, in correspondence with:

$$W(r) = \frac{a}{(r + r^*)^2 + a}, \tag{3}$$

where $a$ is a constant, $r$ is the distance to the station, and $r^*$ is a distance factor ($0 \leq r^* \leq r$) which depends on the observation. It can be top-down ($r^* = r$) o crossing ($r^* = 0$) from the viewpoint of the grid.

### 2.2. Kriging

#### 2.2.1. Fundamentals

Kriging technique is known also as "optimal prediction". It is a method of interpolation that predicts unknown values of observed data in certain places. This method uses a variogram to express the spatial variation, and minimizes the error of predicted values that are estimated by the spatial distribution of the predicted values.

Ordinary Kriging estimates the unknown value using linear combinations weighted from the available sample [12]:

$$\hat{v} = \sum_{j=1}^{n} w_j * v \qquad \sum_{i=1}^{n} w_i = 1 \tag{4}$$

The estimated *i*-th error, $r_i$, is the difference between the estimated value and the true value at the same location:

$$r_i = \hat{v} - v_i \,. \tag{5}$$

The average error in a set of k estimated values is:

$$m_\tau = \frac{1}{k}\sum_{i=1}^{k} \tau_i = \frac{1}{k}\sum_{i=1}^{k} \hat{v}_i - v_i \,. \tag{6}$$

The error variance is:

$$\delta_R^2 = \frac{1}{k}\sum_{i=1}^{k} (\tau_i - m_R)^2 = \frac{1}{k}\sum_{i=1}^{k} \left[ \hat{v}_i - v_i - \frac{1}{k}\sum_{i=1}^{k} (\hat{v}_i - v_i) \right]^2 \,. \tag{7}$$

### 2.2.2. Behaviour

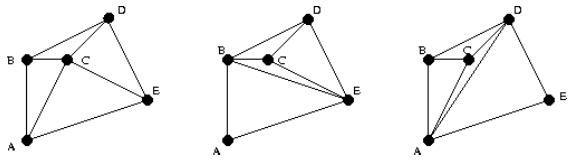
Many properties of Earth's surface vary in an apparently random, although spatially correlated, manner. Using Kriging for interpolation allows us to estimate confidence in every interpolated value in a better way than other methods [9].

Kriging is also the method which is associated with the acronym B.L.U.E. (best linear unbiased estimator). It is linear because the estimated values are the weighted linear combinations of the available data. It is unbiased because the average of error is 0. It is the 'best' because it tends to minimize the error variance. The difference between Kriging and the other estimation methods is its tendency to minimize the error variance.

## 2.3. Voronoi Algorithm

### 2.3.1. Delaunay Triangulation

There are several ways of triangulating, given a set of points:




Sometimes it is necessary to perform a triangulation of the points with certain properties. One of the most common and useful triangulations is the Delaunay

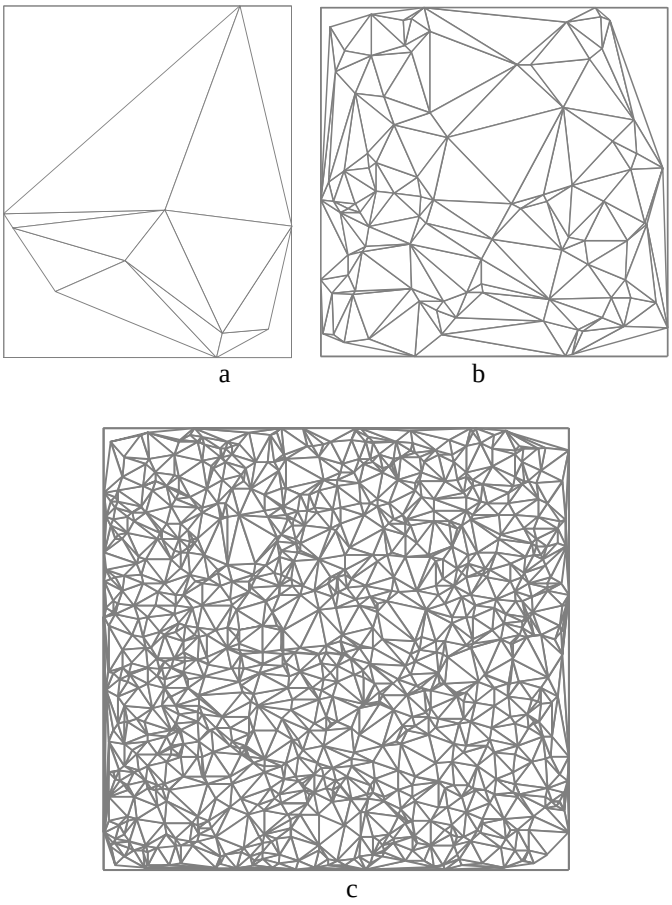



**Fig. 1**. Examples of Voronoi diagrams with (a) 10 points (b) 1000 points and (c) 10,000 points.

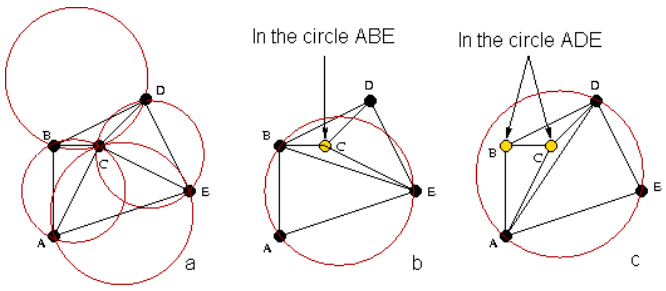



**Fig. 2**. A Delaunay Triangulation (a) and two non-Delaunay triangulations (b, c).

triangulation, named after the Russian mathematician Boris Delaunay. Delaunay's triangulation of a set of points is given by the following property:

AB is an edge of Delaunay's triangulation if and only if, there exists a circle that passes through A and B in such a way that any other point in the set of points, C, where C is different from A and B, is outside of the circle.

Equivalently, all circles in the Delaunay's triangulation for a set of points will have empty circumscribed circles. That is, there are no points inside the circumference of any triangle (see Fig. 2).

We can see immediately that the first triangulation is a Delaunay triangulation, as all the circumscribed circles are empty.

Delaunay triangulation always exists for any set of points in two dimensions. It is always unique as long as it does not happen that four points in the set of points are co-circular. Because it minimizes small angles and circumscribed circles, Delaunay's triangulation is geometrically convenient, and in general, appealing to sight.

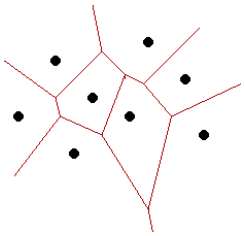
To generate the Delaunay's triangulation, we chose to implement the algorithm of 'divide and conquer' presented by Guibas and Stolfi in [6].

Some examples of the Delaunay triangulations are presented in Fig. 1.

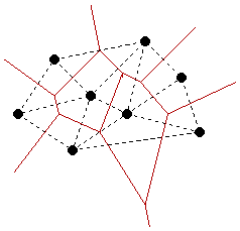
### 2.3.2. Voronoi Diagrams

Named after the Russian mathematician Georges Voronoi, Voronoi diagrams are also known as Thiessen polygons or the Blum's Medial Axe Transformation. Cells are called Dirichlet Regions, Thiessen polytopes, or Voronoi polygons.

The Voronoi diagram for a set of points, *S*, in two dimensions (assuming that there are no three collinear points or four co-circular points) is a plane division in polygons. Each point in S is inside some polygon, and each polygon contains exactly one point inside. Each polygon cuts the region that it is closer than any other point in *S*, to its contained point.



Voronoi diagram is the linear dual of Delaunay's triangulation. This means that we can change from Voronoi diagram to Delaunay's triangulation drawing the perpendicular edges to the region limits, and *vice versa*.



Now that we have the unique triangulation of Voronoi, we face the problem of knowing if a point is inside a triangle, so that this way, and using the equations that

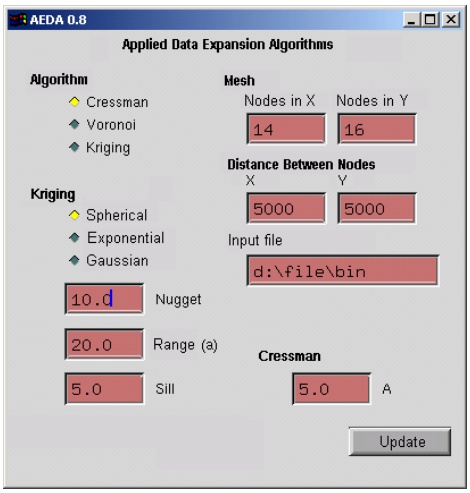
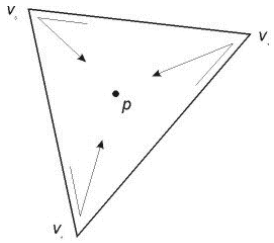



**Fig. 3**. Program's main interface.

describe a plane in the space, we can obtain the values of the regular grid that overlaps to the generated triangles.

### 2.3.3. Test of the Inclusion of a Point in a Triangle

Given a triangle ($v_1$, $v_2$, $v_3$) and a point $p$, the test of inclusion of $p$ in the triangle is:




If we travel across points $v_1$, $v_2$, $v_3$ and the point is inside, we will always look to the point at the same side of the segment we are visiting. If $v_1$, $v_2$, $v_3$ are arranged counter-clockwise, the points outside it will always be to the left of the segments. If the point is outside, at least one of the segments of the point will be on the right side (see figure). If the vertices are arranged clockwise, our reasoning is identical, except that a point that is inside the triangle will be always to the right of the segment we are visiting.

This way, to determine if the point $p$ is inside the triangle ($v_1$, $v_2$, $v_3$), we must obtain the rotation directions along triplets ($v_1$, $v_2$, $p$), ($v_2$, $v_3$, $p$) y ($v_3$, $v_1$, $p$). The point is inside if, and only if, the three directions are the same.

Now that we have an efficient way of knowing if a point is inside a triangle [3], we obtain the value of z using known equations of analytic geometry, which describe a plane.

# 3. Implementation

## 3.1. OpenGL and GLUT

For visualizing the results, we chose OpenGL because it is a free graphic library, it is efficient, it is available for every platform which can compile a C program, and it has a wide range of users.

OpenGL is a low level specification of graphic libraries. It gives the programmer a small set of primitive geometric objects: points, lines, polygons, images, and bitmaps. OpenGL provides with a set of commands that allow the specification of geometric objects in two and three dimensions, using the provided primitives, along with commands that control how these objects are rendered.

GLUT is a set of utilities designed for OpenGL. GLUT means OpenGL Utility Toolkit. It is a programming interface which links with ANSI C and FORTRAN to write OpenGL programs independent of the window system being used.

## 3.2. Interface

We implemented our software in C language, using the OpenGL library for graphical output, and the μ-UI (Micro User Interface) library for the user interface. μ-UI takes advantage of the bi-dimensional drawing in OpenGL. This way, the application can be easily ported to other operating system with a C compiler installed.

The main program's window allows to use the desired data expansion algorithm; some parameters particular to each one of the implemented algorithms, the size and spacing of the mesh desired, and the input file (see Fig. 3).

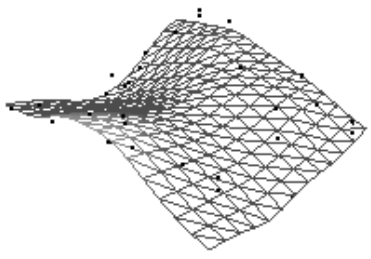
When the 'Update' button is pressed, a window with the mesh rendering will appear. For demonstration purposes, when the input file is not valid, or it does not exist, the program uses the original default functions:

$$x^2 + y^2,$$

or

$$x^2 - y^2.$$

From these default functions, 30 samples are taken in randomly chosen points, and afterwards, as it is done with normal data input, the mesh is reconstructed using the chosen data-expansion algorithm:



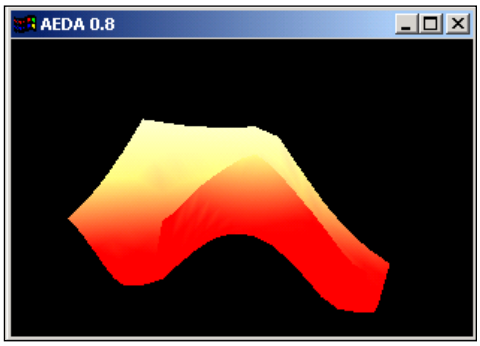
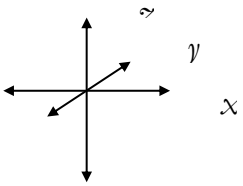



**Fig. 4**. Sample function reconstructed with the sample points hidden in solid surface mode.

In this image, we show the mesh generated with the Cressman algorithm, with parameter A = 5.0. The points where the height sample was taken appear in black dots. These points are used to reconstruct the surface.

Axis convention is as follow:




Rotating movements by keyboard can be done also with mouse, clicking somewhere in the window and dragging the mouse.

| | | |
|---|---|---|
| 28660 | 41020 | 0.101 |
| 24540 | 43990 | 0.171 |
| 45800 | 50030 | 0.136 |
| 28521 | 49233 | 0 |
| 42000 | 49090 | 0.251 |
| 37475 | 37620 | 0.292 |
| 28553 | 26673 | 0.013 |
| 42150 | 27800 | 0.091 |
| 41200 | 37370 | 0.143 |
| 33260 | 31870 | 0 |
| 31711.8 | 62477 | 0 |
| 23440 | 54620 | 0 |
| 39516.6 | 62932 | 0 |
| 28950 | 31340 | 0 |

(a)

| | | |
|---|---|---|
| 15000 | 35000 | 0.161997 |
| 15000 | 40000 | 0.166417 |
| 15000 | 45000 | 0.154865 |
| 15000 | 50000 | 0.129265 |
| 15000 | 55000 | 0.084798 |
| 15000 | 60000 | 0.034853 |
| 15000 | 65000 | -0.018312 |
| 15000 | 70000 | -0.053776 |
| 15000 | 75000 | -0.074067 |
| 20000 | 0 | -0.202291 |
| 20000 | 5000 | -0.200391 |
| 20000 | 10000 | -0.190037 |
| 20000 | 15000 | -0.140032 |
| 20000 | 20000 | -0.058771 |
| 20000 | 25000 | 0.011562 |
| 20000 | 30000 | 0.060174 |

(b)

**Fig. 5**. Fragment of input file and fragment of output file *mesh.out*

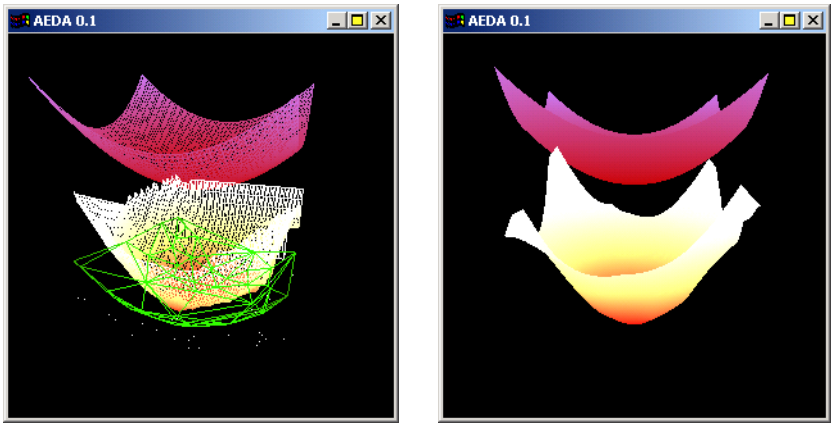
Input files must be in text format (ASCII). The first two values are for the position of the measuring station (*X* and *Y* coordinates, respectively). Values are separated by spaces. The third value is the value of the measuring. The first two values are integers. The last value is real.

The number of measuring stations (rows) must be less than 65535. Output file format of *mesh.out* has the same format.

## 4. Experiment and Results

We can judge subjectively the best data expansion algorithm if we examine the accuracy shown for the reconstruction of the original function, because we already know its original form (see Fig. 6 and 7).


**Fig. 6**. Cressman and Voronoi algorithms.




(a) (b)

**Fig 7**. (a) Voronoi in wire frame with overlapping triangulation; (b) Kriging.

For Cressman, we can see the influence rations of the measuring points. This produces an irregular surface, because this algorithm works locally.

Voronoi produces a uniform surface, although we can see spaces at the edges. This is because they are not included in the triangulation, so that they are undefined. In the next image we show the triangulation overlapped in green color.

We can see that Kriging tends to adjust closer to the original function.

Now we will evaluate quantitatively the performance of each data expansion algorithm using the real data of air pollution measuring stations. The measured pollutants are: NO, $NO_2$, Ozone, $RCHO_1$, $RCHO_2$, $RCHO_3$, $RH_1$, $RH_2$, and $RH_3$. To obtain the deviation error with respect to the original values, we compared the original data within a cell defined by 4 nodes of the mesh, with the value of the closest node of the reconstructed mesh. The sums of differences are shown in Table 1.

**Table 1**. Absolute sums of differences of original points with reconstructed points.

| | Cressman | Kriging | Voronoi |
|---|---|---|---|
| NO | 0.466401 | 0.313273 | 0.371092 |
| $NO_2$ | 0.109421 | 0.102081 | 0.147478 |
| Ozone | 0.024482 | 0.022138 | 0.132038 |
| $RHCO_1$ | 0.121287 | 0.244838 | 0.302807 |
| $RHCO_2$ | 0.47808 | 0.376325 | 0.351358 |
| $RHCO_3$ | 1.399525 | 0.983684 | 0.99482 |
| $RH_1$ | 0.11386 | 0.074988 | 0.076807 |
| $RH_2$ | 0.464336 | 0.323694 | 0.319748 |
| $RH_3$ | 1.856715 | 1.294978 | 1.280415 |

From this table we can see that in most cases, Kriging has the smallest difference values, followed after Voronoi. In some cases, Voronoi presents a better approximation, as for $RH_2$ y $RH_3$.

Table 2 shows the normalized values displayed as a percentage.

**Table 2**. Comparison of the three algorithms based on the differences of reconstructed points vs. original points

| | Cressman | Kriging | Voronoi |
|---|---|---|---|
| NO | 76.68% | 84.34% | 81.45% |
| $NO_2$ | 94.53% | 94.90% | 92.63% |
| Ozone | 98.78% | 98.89% | 93.40% |
| $RHCO_1$ | 93.94% | 87.76% | 84.86% |
| $RHCO_2$ | 76.10% | 81.18% | 82.43% |
| $RHCO_3$ | 30.02% | 50.82% | 50.26% |
| $RH_1$ | 94.31% | 96.25% | 96.16% |
| $RH_2$ | 76.78% | 83.82% | 84.01% |
| $RH_3$ | 7.16% | 35.25% | 35.98% |
| | **72.03%** | **79.24%** | **77.91%** |

It can be seen from this table that the Kriging is the best data expansion algorithm, in terms of the differences it presents with respect to random sampling points.

## 5. Conclusion

Kriging is the best linear estimator, although adjusting its parameters is a crucial task for achieving good performance. In addition, it has a longer response time. For real-time applications, we found that the Voronoi algorithm is the best, which in addition, requires no tuning of parameters. Voronoi has a small answer to local alterations. It tends to soften surface's form. Kriging and Cressman, by the contrary, can respond actively to small local alterations. In the case of the gas' diffusion in the atmosphere, the presence of local alterations decreases as altitude increases.

**Acknowledgements**. Work done under partial support of Mexican Government (CONACyT, SNI, PIFI-IPN, CGEPI-IPN), and RITOS-2, and the PAPIIT UNAM grant IN100405.

## References

1. Aurenhammer, F., Klein, R.: Voronoi Diagrams. Ch. 5. In: Handbook of Computational Geometry (Eds. J.-R. Sack and J. Urrutia). Amsterdam, Netherlands: North-Holland, pp. 201–290 (2000)
2. Aurenhammer, F.: Voronoi diagrams – a survey of a fundamental geometric data structure. ACM Computing Surveys, 23(3), pp. 345–401 (1991)
3. Bernardini, F., Bajaj, C., Chen, J., Schikore, D.: Triangulation – based object reconstruction methods. In: ACM Sympos. Comput. Geom., pages 481–484 (1997)
4. Cressman, G.P.: An operational objective analysis system. Mon. Wes. Rev., 87, pp. 367–374 (1959)
5. Devillers, O.: Improved incremental randomized Delaunay triangulation. In: 14th Ann. ACM Sympos. Comput. Geom, pp. 106–115 (1998)
6. Edelsbrunner, H., Mücke, E.P.: Three-dimensional alpha shapes. ACM Trans. Graph. 13(1), pp. 43–72, Ene. (1994)
7. Endlich, R. M., Mancuso, R. L.: Objective analysis of environmental conditions associated with severe thunderstorms and tornadoes. Mon Wea. Rev., 96, pp. 342–350 (1968)
8. Eppstein, D.: Nearest Neighbors and Voronoi Diagrams. http://www.ics.uci.edu/ ~eppstein/ junkyard/nn.html.
9. Melkemi, M.: A-shapes and their derivatives. In: 13th Ann. ACM Sympos. Comput. Geom., pp. 367–369 (1997)
10. Moreno, C.: Efficient 2-D geometric operations. http://www.mochima.com/ articles/ cuj_geometry_article/ cuj_geometry_article.html
11. Oliver, M. A., Webster, R.: Kriging: a method of interpolation for geographical information system. Int. J. Geographical Information Systems, 1990, vol. 4, No. 3, pp. 313–332 (1990)
12. Sethian, J.A.: Level Set Methods. Cambridge University Press (1996)
13. Watson, D.F.: Contouring: A guide to the Analysis and Display of Spatial Data. Pergamon (1992)

ISSN 1870-4069

# Sistema de reconocimiento de placas vehiculares haciendo uso de modelos asociativos

Luis Edgar Alanís Carranza, Moisés Vicente Márquez Olivera,
Viridiana Gudelia Hernández Herrera Olivera, Octavio Sánchez García

Instituto Politécnico Nacional, Centro de Investigación e Innovación Tecnológica,
Ciudad de México, México

## Vehicle Plates Recognition System using Associative Models

**Abstract.** This study used of the associative memories alpha-Beta (AMαβ) applied to the problem of the recognition of vehicle plates. We propose using a Haar Cascade classifiers with the AdaBoost in the detection of the plate finally proposes to use the AMαβ as a classifier to determine the alpha-numeric characters inside the plate. The AMαβ are trained with characteristic vectors extracted from images of an own database. In the detection stage the system can locate 98.3% of 707 car images and for the recognition stage the accuracy was 93%.

**Keywords:** inteligencia artificial, placas vehiculares, memorias asociativas, Haar Cascade, reconocimiento de patrones, caracteres alfanuméricos.

## 1. Introducción

Hoy en día la visión artificial ha tenido una amplia gama de aplicaciones en el reconocimiento de objetos, una de ellas son los sistemas de reconocimiento de placas vehiculares (LPR), los cuales buscan reconocer las placas de forma no intrusiva; es decir, que emplean dispositivos externos al auto para monitorear y ubicar el auto. El dispositivo por excelencia de estos sistemas son las cámaras, las cuales adquieren imágenes estáticas o dinámicas; en ocasiones las cámaras son mejoradas con tecnología infrarroja para oprimir factores que afectan a la detección, como son las condiciones ambientales o la hora del día. Idealmente, un sistema LPR debería estar optimizado con algoritmos de reconocimiento que le permitan trabajar en tiempo real y bajo condiciones reales del campo de aplicación, logrando así brindar una respuesta asertiva e instantánea del análisis realizado en una escena vial. Favorablemente, los avances tecnológicos en hardware, en conjunto con los nuevos modelos de Inteligencia Artificial, han dado como resultado el desarrollo de nuevos algoritmos, nuevos e híbridos, que buscan la solución del problema tratando diferentes problemas como iluminación, ángulos, tiempo real, seguimiento, entre muchos otros [1, 2].

Tabrizi y Cavus [1] utilizaron el método de detección de bordes utilizando un filtro Prewit y posteriormente utilizaron un algoritmo híbrido de inteligencia artificial para el

reconocimiento de los caracteres SVM y KNN, finalmente obtuvo un índice de asertividad del 94 al 97 %. Patel [2] utilizó el filtro de la mediana para eliminar ruido que pudiera afectar el reconocer la placa vehicular, posteriormente utilizó las redes neuronales para reconocer los caracteres. Ktata et al. [3] desarrollaron un sistema de reconocimiento de placas utilizando el algoritmo skew correction para detectar zonas que posiblemente serían una placa vehicular y finalmente usaron redes neuronales para reconocerla, obtuvieron un índice de asertividad del 90.78%. Sedighi y Vafadust [4] emplearon filtro Gaussiano para la eliminación de ruido y redes neuronales para tratar de reconocer un total de 500 imágenes con placas vehiculares, de las cuales solo el 90.05% fueron reconocidas.

Este trabajo describe el desarrollo de un Reconocimiento automático de matrículas (ANPR) utilizando memorias asociativas Alpha-Beta. El sistema propuesto consta de seis etapas. La primera etapa es la adquisición de la imagen, donde se obtendrá la imagen del automóvil. En la segunda etapa, se desarrollan algoritmos de procesamiento de imágenes digitales para mejorar la calidad de la imagen, posteriormente, se aplica el algoritmo de Haar Feature-based Cascade (HFC) con AdaBoost para detectar dentro de la imagen dónde se encuentra la placa. La tercera etapa es la segmentación que permitirá extraer la imagen de la matrícula, así como los caracteres intrínsecos. La cuarta etapa consiste en extraer información relevante sobre los caracteres segmentados y aquellos que pueden clasificarse, en la siguiente etapa se busca reconocer de forma particular los caracteres alfanuméricos para que en la última etapa se haga una interpretación general de la identificación de la placa vehicular.

## 2. Metodología

### 2.1. Proceso General

Los sistemas LPR de forma general presentan fases claramente definidas, los cuales fueron resumidas por Gonzales y Woods [7], en esta sección se mencionan los pasos a seguir, siendo la adquisición el proceso en donde se realiza la obtención de las imágenes estáticas o dinámicas (video), posteriormente durante el preprocesamiento se busca mejorar la calidad; la segmentación es la encargada de extraer de la imagen mejorada solo las área que son de interés para analizar, consecuentemente, se realiza la extracción de características esenciales de los caracteres a reconocer, los cuales son procesados por el algoritmo de Inteligencia Artificial (IA) supervisado previamente entrenado, el modelo de IA da como resultado cuál es el carácter que se encuentra contenido en cada una de las imágenes segmentadas, finalmente y una vez terminando con el análisis de reconocimiento a la salida se tiene la interpretación de placa objetivo.

#### 2.1.1. Adquisición

Como se muestra en la Fig. 1, el primer paso es la adquisición, por lo que para el presente trabajo se propuso crear una base de datos propia, la cual está integrada con 2000 imágenes con modo de color RGB tomadas con una cámara GoPro HERO4 Session con un tamaño de 1280x720 pixeles, las cuales contienen automóviles que portan placas vehiculares de la zona metropolitana.

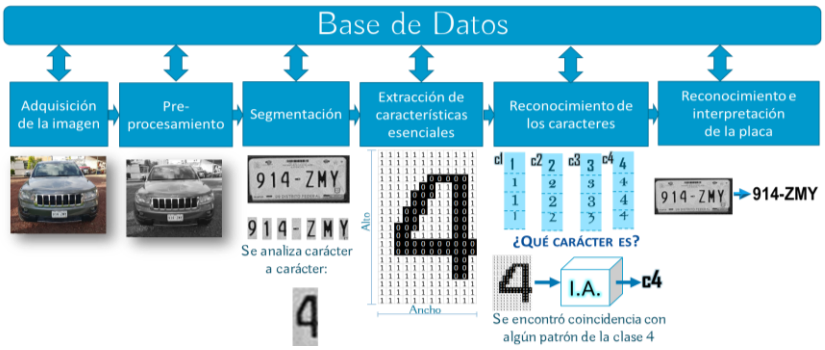



**Fig. 1.** Proceso general para el reconocimiento de placas vehiculares.

**Fig. 1.** Imagen del automóvil incluido en la base de datos creada.

### 2.1.2. Preprocesamiento

En esta etapa se utilizarán todos aquellos algoritmos para mejorar la calidad de la imagen y obtener un mejor resultado durante la etapa de segmentación. El primer preprocesamiento propuesto convertir de modo de color RGB a escala de grises para ello se utilizó el modelo YIQ o YUV [6], el cual consiste en multiplicar cada canal de color de la imagen por un cierto porcentaje y guardar todos esos valores en uno solo canal como se muestra en la siguiente expresión:

$$Y = R * 0.3 + G * 0.59 + B * 0.11. \quad (1)$$

Finalmente se obtendrá como salida una imagen a escala de grises con una profundidad de 8 bits (Fig. 2). Esto reducirá el costo computacional, debido a que los procesamientos que se realicen solo trabajaran en un solo canal.

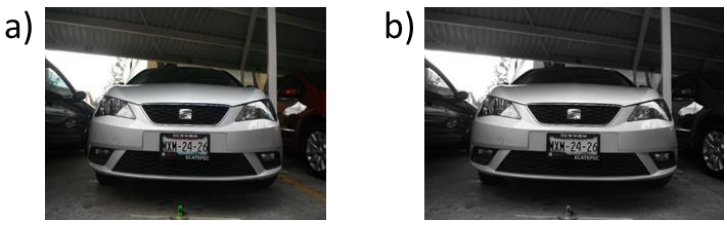



**Fig. 2.** Conversión escala de grises. a) Imagen original, b) Imagen en escala de grises.

***Ecualización***

Ahora bien, existen factores que puedan afectar en el reconocimiento de la placa y uno de ellos es el contraste ya que la imagen puede estar muy obscura o con brillo y esto se debe a que la imagen no tiene una buena distribución de los pixeles más obscuros o blancos en toda la imagen y no sería posible llegar a ver la placa vehicular al momento de buscarla por el algoritmo de segmentación. Por lo cual se utilizará el algoritmo de ecualización usado en [7], el cual consiste en mover los pixeles más oscuros o con más brillo en toda la imagen (Fig. 3).

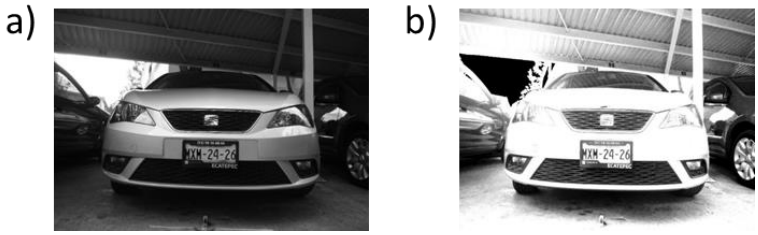



**Fig. 3.** Ecualización. a) Imagen en escala de grises, b) Imagen en escala de grises ecualizada.

***Filtro gaussiano***

La imagen puede llegar a tener ruido debido a los posibles defectos que llegue a tener la cámara, para poder llegar a eliminar parte de ese ruido y suavizar la imagen es necesario utilizar una máscara o también conocido en la literatura como Kernel. Este se desplazará por toda la imagen y realizará una operación matemática para tener un nuevo valor el cual será agregado a una nueva imagen.

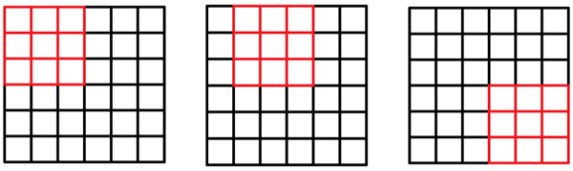


**Fig. 4.** Imagen de 6 x 6 usando un Kernel de 3 x 3.

En la Fig. 4 se puede observar que hay una imagen con resolución de 6 x 6 y tiene colocada sobre ella un Kernel de 3 x 3. No obstante, antes de realizar este procedimiento es necesario obtener los valores del Kernel y se pueden obtener por medio de la siguiente ecuación.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{-x^2+y^2}{2\sigma^2}}. \quad (2)$$

En este caso se utilizó una varianza ($\sigma$) de 0.25 ya que es la más utilizada por la literatura y las coordenadas $x, y$ son las del Kernel. Finalmente se tiene un Kernel con los siguientes valores Fig. 5.

| | | |
|---|---|---|
| 0.0183 | 0.1353 | 0.0183 |
| 0.1353 | 1 | 0.1353 |
| 0.0183 | 0.1353 | 0.0183 |

**Fig. 5.** Kernel con los valores

Una vez obtenido este Kernel se realiza la siguiente ecuación para la obtención de los nuevos pixeles de cada recorrido:

$$g(x,y) = h(x,y) \times f(x,y) = \sum_{i=0}^{i=2} \sum_{j=0}^{j=2} f(i,j)h(x,y). \quad (3)$$

Teniendo finalmente como salida una imagen con filtro Gaussiano Fig. 6.

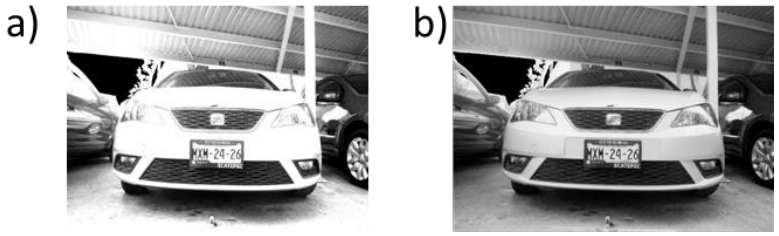



**Fig. 6.** Filtro gaussiano. a) Imagen ecualizada, b) Imagen con filtro Gaussiano.

## 2.2. Segmentación

En la etapa de segmentación se utilizará un algoritmo para detectar la zona que posiblemente contenga una placa vehicular. Para ello se utilizó el algoritmo Haar Feature-based Cascade (HFC) creado por Viola y Jones [8], el cual consiste en la extracción de características por medio de una imagen integral. Una vez que se tengan todas las características llamadas Haar, estas serán utilizadas para los clasificadores pequeños de AdaBoost para generar un clasificador fuerte. Este discriminará imágenes que no contengan una placa vehicular, a lo cual le llamaremos zonas candidatas. Adicionalmente, para mejorar los resultados de la clasificación se utilizó un árbol de clasificadores el cual es la combinación de varios AdaBoost (véase la Fig. 7).

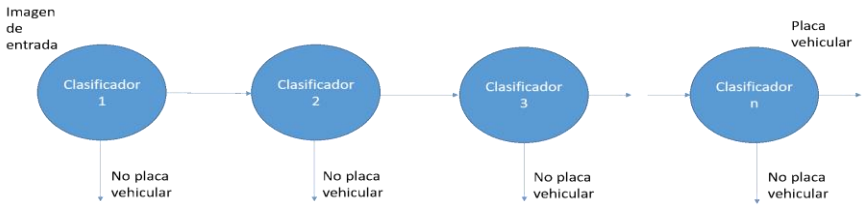



**Fig. 7.** Árbol de clasificadores (AdaBoost).

El algoritmo HLF fue programado en C# bajo la IDE de Visual Studio, el cual fue entrenado con 2,000 imágenes positivas (placas vehiculares) y 4,000 imágenes negativas las cuales no son placas; por ejemplo, letreros, retrovisores, árboles, etc. Una

vez entrenado el algoritmo, el resultado de detección obtenido se presenta en la Fig. 8, en donde el que el HLF detecta la placa dentro de la imagen.

**Fig. 8.** Detección de placa vehicular con HLF.

### 2.3. Extracción de características

Para comenzar con el proceso de extracción característica se propone realizar una binarización utilizando un umbral dinámico y después se realiza un escalamiento para normalizarla utilizando el método de interpolación lineal (Fig. 9)


**Fig. 9.** Imagen con Binarización y Rescalamiento.

Posteriormente, se buscan los caracteres para segmentarlos, para ello se propone realizar una búsqueda por contornos que cumplieran con las características de un carácter considerando la proporción existente entre el alto y el ancho, bajo este parámetro de normalización, es posible detectar caracteres más altos o anchos dependiendo del tamaño de la placa vehicular segmentada, el resultado se muestra en al Fig. 10.

Una vez detectados y segmentados los caracteres de forma individual se extrae el vector característico, el cual se forma a partir de concatenar los valores binarios (0 y 1) que tienen cada pixel de la imagen del carácter (véase la Fig. 11).


**Fig. 10.** Placa vehicular con caracteres segmentados.

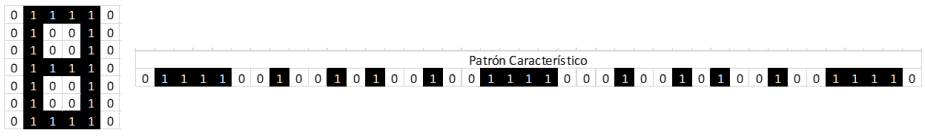



**Fig. 11.** Extracción de características y formación de patrón característico.

### 2.4. Reconocimiento

Para esta etapa se implementó el algoritmo de memorias asociativas Alfa-Beta [10] [11] [12] para reconocer cada uno de los caracteres. La idea básica de una memoria asociativa αβ es que es una matriz que almacena información de patrones que previamente aprendió usando el operador α y posteriormente los recupera utilizando el operador β (véase la Fig. 12).


**Fig. 12.** Matriz $M$ generada por patrones de entrada X.

La memoria asociativa αβ está constituida de tres fases.

**Fase de aprendizaje:** En esta fase se genera la memoria asociativa o la matriz $M$ por medio de los patrones de entrada; para ello se utilizará el patrón concatenado representados como $x_i^n$, con lo que se alimenta la memoria. Al comenzar con la generación de esta matriz se obtienen las matrices transpuestas de cada uno de los patrones (véase la Fig. 4).

$$[\mathrm{M}']^{\mathrm{n}} = [\mathrm{x}^{\mathrm{n}}\ \alpha\ (\mathrm{y}^{\mathrm{m}})^{\mathrm{t}}] = \begin{bmatrix} \mathrm{x}_1^{\mathrm{n}} \\ \mathrm{x}_2^{\mathrm{n}} \\ \vdots \\ . \\ \mathrm{x}_{\mathrm{l}}^{\mathrm{n}} \end{bmatrix} \alpha [\mathrm{y}_1^{\mathrm{m}}\ \ \mathrm{y}_2^{\mathrm{m}}\ \ \ldots\ \ \mathrm{y}_{\mathrm{l}}^{\mathrm{m}}]. \tag{4}$$

**Tabla 1.** Operador binario α.

| **X** | **Y** | $\boldsymbol{\alpha(x,y)}$ |
|---|---|---|
| **0** | 0 | 1 |
| **0** | 1 | 0 |
| **1** | 0 | 2 |
| **1** | 1 | 1 |

Una vez que se tiene la transpuesta de cada uno de los patrones se procede a utilizar el operador α de la Tabla 1 para obtener una nueva matriz por cada patrón.

Finalmente se busca el número más grande utilizando el operador max (V) de cada una de las matrices para obtener solamente una sola, esta será la memoria asociativa.

$$\mathrm{v}_{\mathrm{ij}} = \bigvee_{\mathrm{n}=1}^{\mathrm{p}} [\boldsymbol{\alpha}(\mathrm{x}_{\mathrm{i}}^{\mathrm{n}}, (\mathrm{y}_{\mathrm{i}}^{\mathrm{m}})^{\mathbf{t}})]^{\mathrm{n}}, \tag{4}$$

$$\boldsymbol{M} = \begin{bmatrix} v_{11} & v_{12} & \ldots & v_{1l} \\ v_{21} & v_{22} & \ldots & v_{2l} \\ . & . & \ldots & . \\ . & . & \ldots & . \\ . & . & \ldots & . \\ v_{l1} & v_{2l} & \ldots & v_{ll} \end{bmatrix}. \tag{5}$$

**Fase de recuperación:** Recupera patrones previamente aprendidos usando la matriz de la ecuación (6), una de las ventajas de este algoritmo es su factor de olvido del 0%. Esto quiere decir que no olvidan el patrón previamente aprendido. Se realiza una comparación del patrón de entrada con la matriz de aprendizaje utilizando el operador β de la Tabla 2. Una vez que se tienen la matriz resultante del operador β ecuación (7) se procede a obtener el operador min (⋀) de cada fila de la matriz para así obtener como salida un patrón recuperado ecuación (8).

**Tabla 2.** Operador β.

| X | Y | $\boldsymbol{\beta(x,y)}$ |
|---|---|---|
| **0** | 0 | 0 |
| **0** | 1 | 0 |
| **1** | 0 | 0 |
| **1** | 1 | 1 |
| **2** | 0 | 1 |
| **2** | 1 | 1 |

$$\boldsymbol{y_i^m} = \begin{bmatrix} \boldsymbol{\beta}(v_{11}, x_1^n) & \boldsymbol{\beta}(v_{12}, x_1^n) & \dots & \boldsymbol{\beta}(v_{1l}, x_1^n) \\ \boldsymbol{\beta}(v_{21}, x_2^n) & \boldsymbol{\beta}(v_{22}, x_2^n) & \dots & \boldsymbol{\beta}(v_{2l}, x_2^n) \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \boldsymbol{\beta}(v_{l1}, x_l^n) & \boldsymbol{\beta}(v_{2l}, x_l^n) & \dots & \boldsymbol{\beta}(v_{ll}, x_l^n) \end{bmatrix}, \tag{6}$$

$$\boldsymbol{y_i^m} = \begin{bmatrix} \bigwedge_{i=1}^{l} \boldsymbol{\beta}(v_{1i}, x_i^n) \\ \bigwedge_{i=1}^{l} \boldsymbol{\beta}(v_{2i}, x_i^n) \\ \cdot \\ \cdot \\ \cdot \\ \bigwedge_{i=1}^{l} \boldsymbol{\beta}(v_{li}, x_i^n) \end{bmatrix}. \tag{7}$$

**Fase de prueba:** La memoria asociativa tratará de asociar un patrón no aprendido con algunos de los que aprendió previamente, utilizando el mismo método de la fase de recuperación.

## 3. Resultados

Se obtuvieron resultados mostrados en las tablas 3 y 4 como parte de la detección y reconocimiento respectivamente de placas vehiculares.

**Tabla 3.** Resultados de la parte de detección.

| Número de AdaBoost | Imágenes positivas | Imágenes negativas | Imágenes probadas | Placas detectadas | Índice de asertividad |
|---|---|---|---|---|---|
| **16** | 2000 | 4000 | 707 | 585 | 82.7 % |
| **18** | 2000 | 4000 | 707 | 695 | 98.3 % |
| **20** | 2000 | 4000 | 707 | 590 | 83.4 % |
| **22** | 2000 | 4000 | 707 | 572 | 80.9 % |

En la Tabla 4 se puede apreciar que al utilizar 18 AdaBoost se obtuvieron los mejores resultados al momento de la detección de la placa. En la base de datos se probaron 707 imágenes de carros y el 98 .3% de las placas fueron detectadas.

**Tabla 4.** Resultados de la parte de reconocimiento.

| Número de patrones probados | Tamaño del patrón | Resolución | Tiempo de procesamiento | Índice de asertividad |
|---|---|---|---|---|
| **3200** | 1, 024 | 32 x 32 | 0.254s | 82 % |
| **3200** | 20, 000 | 100 x 200 | 0.615s | 93 % |

Se utilizaron 100 patrones por cada carácter existente en una placa vehicular en México. En la Tabla 4 se puede observar que mientras mejor resolución tenía la imagen del patrón con el que se alimentaba la memoria asociativa se obtenían mejores resultados, no obstante, el tiempo de procesamiento es mayor.

### 3.1.1. Preprocesamiento

Todos los algoritmos fueron programados en lenguaje C# bajo el entorno de Visual Studio en una PC con Windows 8, 8GB de RAM, y un procesador Intel Core i7-4700HQ a 2.40GHz. Se realizó una interfaz gráfica para mostrar los resultados del sistema la cual se muestra en la Fig. 13.


**Fig. 13.** Matriz *M* generada por patrones de entrada X.

## 4. Conclusiones

Durante la fase de detección de placas empleando el HFC se obtuvo un índice de asertividad máximo de 98.3%, para ello fue necesario probar con diferente número de Adaboost, esto para determinar el entrenamiento y sobre entrenamiento del algoritmo.

El HFC depende del número y tipo de imágenes positivas y negativas con las que se entrenan influyen en el índice de precisión del algoritmo durante la fase de prueba.

Las memorias asociativas Alfa-Beta son capaces de reconocer caracteres alfanuméricos en imágenes vehiculares al entrenar el algoritmo con el método de validación K fold cross validation con k=10, en el que se obtuvo un índice de exactitud máxima de 93% utilizando una resolución de 100 x 200 pixeles.

La resolución de la imagen influye directamente en el tiempo de procesamiento, no obstante, si se selecciona una resolución muy baja para las imágenes buscando mejorar el tiempo de respuesta, el índice de precisión obtenido por las memorias Alfa-Beta también se ve reflejado de forma negativa, ya que la pérdida de información en el patrón da como resultado que las memorias se confundan entre caracteres similares.

## Referencias


1. Atiwadkar, A., Mahajan, S., Lande, T., Patil, K.: Vehicle License Plate Detection: A Survey. International Research Journal of Engineering and Technology (IRJET), vol. 2, nº 8, pp. 354–360 (2015)
2. Du, S., Ibrahim, M., Shehata, M., Badawy, W.: Automatic license plate recognition (ALPR): A state-of-the-art review. IEEE Transactions on circuits and systems for video technology, vol. 23, nº 2, pp. 311–325 (2013)
3. Tabrizi, S. S., Cavus, N.: A hybrid KNN-SVM model for Iranian license plate recognition. Procedia Computer Science, vol. 102, pp. 588–594 (2016)
4. Patel, S. G.: Vehicle license plate recognition using morphology and neural network. International Journal on Cybernetics & Informatics, vol. 2, pp. 1–7 (2013)
5. Ktata, S., Khadhraoui, T., Benzarti, F., Amiri, H.: Tunisian License Plate Number Recognition. Procedia Computer Science, vol. 73, pp. 312–319 (2015)
6. Sedighi, A., Vafadust, M.: A new and robust method for character segmentation and recognition in license plate images. Expert Systems with Applications, vol. 38, pp. 13497–13504 (2011)
7. Gonzalez, R., Wood, R. E.: Tratamiento digital de imágenes. vol. 3, Addison-Wesley New York (1996)
8. Zhang, X., Xu, F., Su, Y.: Research on the License Plate Recognition based on MATLAB. Procedia Engineering, vol. 15, pp. 1330–1334 (2011)
9. Shidore, M. M., Narote, S. P.: Number plate recognition for indian vehicles. IJCSNS International Journal of Computer Science and Network Security, vol. 11, pp. 143–146, (2011)
10. Viola, P., Jones, M. J.: Robust real-time face detection. International journal of computer vision, vol. 57, pp. 137–154 (2004)
11. Yánez-Márquez, C.: Associative memories based on order relations and binary operators (in Spanish). Phd Thesis (2000)
12. Yáñez, C., Díaz de León, J.: Memorias Autoasociativas Morfológicas max: condiciones suficientes para convergencia, aprendizaje y recuperación de patrones. IT-I77, Serie Azul, *CIC-IPN*, Mexico (2003b) ISBN, pp. 970–36 (2003)
13. Yáñez, C., Díaz-de-León, J. L.: Introducción a las memorias asociativas. Research in Computing Science, México (2003)

ISSN 1870-4069

# Regular Activity Patterns in Spatio-Temporal Events Databases: Multi-Scale Extraction of Geolocated Tweets

Pablo López-Ramírez, Alejandro Molina-Villegas, Oscar Sánchez-Siordia, Mario Chirinos-Colunga, Gandhi Hernández-Chan

Conacyt - Centro de Investigación en Ciencias de Información Geoespacial, Mexico

**Abstract.** This paper proposes a new technique for the extraction of regular activity patterns at different scales (resolution levels), mined from the microblogging platform Twitter. The approach is based on the recursive application of the DBSCAN clustering algorithm to the geolocated Twitter feed. The proposed technique includes a novel way to obtain 'averaged' regular activity zones based on the rasterization and aggregation of the Concave Hull of the clusters identified at each resolution level. This technique uses only the spatio-temporal characteristics of the geolocated Twitter feed and does not depend on the data content; therefore it can be extended to work with different spatio-temporal event sources such as mobile telephone records. An experiment was carried out to demonstrate the effectiveness of our technique in the extraction of known activity patterns in the Mexico City Metropolitan Area.

**Palabras clave:** social media, urban activity, geographic data mining.

## 1 Introduction

Spatio-Temporal analysis is a rapidly growing field within Geographical Information Science (GIS). The rate of increase in the amount of information gathered every day, the pervasiveness of Global Positioning System (GPS) enabled sensors, mobile phones, social networks and the Internet of Things (IoT), demand for robust and efficient analysis techniques that can help us find meaningful insights from large spatio-temporal databases. Within these new sources of information, the digital breadcrumbs left behind by social media users, have proven to be a valuable resource. They allow us to examine different aspects of crowd behavior, for example, the role of this new media in the Arab Spring [16], or the way people react to hazardous events in general [24] or to more specific occurrences, like terrorist attacks [23] or earthquakes [5].

In the GIS field, one of the main lines of research has been the detection of events [3] or the characterization of zones through social media activity [11], [20]. In both cases, extraction and characterization of regular activity patterns is very important. With this in mind, in this paper we propose a technique for the

extraction of regular patterns of activity that relies only on the spatio-temporal features of the Tweeter feed. The purpose of this is, on the one hand, improve on the current available techniques [21], [11] and, on the other hand, to be as less dependent as possible from the nature of the Twitter feed.

The proposed technique is based on the observation that the regular activity patterns exhibit a wide range of scales [2], and that the current methods for determining this activity from Twitter messages do not consider this. The proposed approach is based on the recursive application of a clustering algorithm to extract patterns of activity across several scales or resolution levels. This approach demands the development of a novel way for 'averaging' the spatial patterns (clusters) extracted from the data.

The rest of the paper is organized as follows: in Section §2 we will establish the basic concepts and perform a general review of the available literature. In Section §3 we will explain in detail the proposed technique. Section §4 presents an experiment extracting the regular activity patterns at several scales from the geolocated Twitter feed in Central Mexico. Finally, Section §5 concludes this document.

## 2 Spatio-Temporal Events and Crowd Activity Detection

In the context of spatio-temporal events, as described by Kisilevich et. al [18], we refer as Crowd Activity to the collective aggregated patterns observed in some spatio-temporal events datasets, specially in data describing some aspect of the behavior of human populations. Although not formally defined, this concept underpins most of the work that we are going to review in the rest of this section.

Moving on to the subject of using Twitter as a source of geographical insights, there is a substantial body of work on techniques for event [1] detection using Twitter. Atefeh and Khreich [3] present a survey of such techniques. It is interesting to note that most of the work done on the subject has been particularly focused on extracting information from the content of the messages; this is natural since it is to be expected that the written messages contain valuable information that can be analyzed to extract meaningful insights. On the other hand, extracting meaningful information from the messages on Twitter can be a daunting task, not only because of the complications associated with analyzing natural language, but also because the Twitter feed is known to be polluted with meaningless messages, rumors, bots, and other kinds of spurious content [14],[20].

There is also an important amount of work done on using geolocated tweets to extract information about the geographic environment of the users. Gabrielli, et. al. [13] propose a framework for mining anomalous mobility patterns using geolocated tweets, they track the movement of users and semantically enrich their trajectories with information about the users and the kinds of activity present at the different locations each user has visited. Kim, et. al. [17] provide

[1] In this context, event refers to 'real world occurrences that unfold over space and time' [3], which is different to the use of the term on spatio-temporal databases.

a methodology for finding user clusters on nearby locations; they use both the geographic location and the content of the messages for detecting spontaneous events associated with topics of interest. In the same line, Boettcher and Lee [4] present a refinement of local spontaneous event detection that uses historic data to assess the regular topics related to a specific area.

All of the works cited above have in common the use of both the geographic location and the semantic content of the messages. However, there is also some work done on the extraction of meaningful patterns using only the spatio-temporal properties of the data and disregarding the semantic dimension completely. In this latter category, Frias-Martinez, et. al. ([10] and [11]) propose a technique for detecting land use by analysing geolocated tweets. This work is particularly relevant because it involves the extraction of regular patterns of activity from the geolocated Twiter feed and will be discussed in Section 3.

On the topic of unusual activity detection, Lee, et. al. [21], Fujisaka, et. al. [12] and Lee and Sumiya [20], propose successive refinements of a technique for detecting unusually crowded places extracting the regular pattern of activity by performing K-Means clustering over the geolocated tweets and then characterizing each cluster by the number of users, the amount of messages and a measure of the mobility of users in each cluster. Next, the unusual activity is detected by comparing the regular pattern with the characteristics of a specific moment.

From this brief review we can infer some generalities involved in the construction of regular patterns of activity from the geolocated Twitter feed:

- Time is segmented in intervals and the definition of this intervals is arbitrary. In [11], each day is divided in 20 minutes intervals, while in [21], each day is segmented in four six hours intervals.
- The geographic space is partitioned in a flat cluster hierarchy. Frias-Martinez, et. al use a Self Organizing Map [19] to obtain a tessellation of the study area, while Lee, et. al. use K-Means to obtain a similar tessellation.

This work will be focused on the second general characteristic: the way in which the space is partitioned to obtain regular activity zones. In both cases, the partition algorithm returns a flat hierarchy of zones which is a Voronoi tessellation around the cluster centroids identified. This partition reflects the differences in event (point) density across the whole space but, since it is flat (it has a single hierarchic level) it cannot represent the structures found at different scales, this means that such partitions mix the whole range of scales of the underlying processes into a single tessellation.

However, when addressing the regular activity patterns from a geographic perspective, the issue of scale is evident: the underlying processes that generate the observed spatio-temporal distribution of events are organized as a hierarchy of scales. In the case of geolocated tweets, the underlying process is the daily pattern of activity, whether in a whole region (as in [21]) or in a urban zone (in [11]). These patterns are closely related to the general fabric of the city or the region and those have been shown to exhibit different properties when analysed at different scales [2]. This suggests that the techniques for defining the regular

activity patterns can be improved by explicitly incorporating the concept of scale.

## 3 Multi-Scale Regular Activity Extraction

The main idea behind the technique proposed in this paper, is that the regular patterns of activity exhibit a range of scales, and that these scales cannot be represented by a flat tessellation. To overcome this limitation, we propose the use of a recursive algorithm to extract a hierarchy of clusters; this hierarchy will represent the structures apparent at different scales in the spatio-temporal events database.

### 3.1 Recursive Clustering

The use of *BigData* Clustering is a very powerful approach to extract knowledge about human activities in order to help decision making in big areas [9].

There are several clustering algorithms that produce a hierarchical structure of point samples, Amoeba [8], Chameleon [15], OPTICS [1] and HDBSCAN [22] are examples of such algorithms. In general, hierarchical clustering produces a complete hierarchy: from a single cluster containing all observations to separate clusters for each observation. In practice, most applications using this family of algorithms will cut the hierarchy trough a threshold value (a scale) to obtain a flat representation [22].

In this paper we propose a different approach, instead of using an algorithm that produces a whole hierarchy of clusters, we will use an algorithm that produces a flat representation and then recursively apply it to each cluster found. For this purpose we will use the algorithm DBSCAN [7], which finds density based clusters on databases with noise (that is, observations that do not belong to any cluster).

The decision to use DBSCAN is based on the following remarks:

- The signal coming from the event database will contain samples that represent regular activity and samples that are just the outcome of random processes in the study area (in the case of the geolocated Twitter feed, the latter could be interpreted as the noise commonly reported in Twitter). In this case, the ability of DBSCAN to detect clusters in noisy samples is of importance.
- Regular activity clusters are arbitrarily shaped. There is no reason to assume a particular shape for the regular activity zones, this makes the use of DBSCAN more appropriate to detect the shape of the regular activity zones than K-Means or any other Voronoi based tessellation, since these latter assume that the clusters are convex.
- Cluster algorithms that return a hierarchical structure are generally geared towards finding the most relevant structures in the data, regardless of scale [22]. This means that the flat representation needed to evaluate regular

activity will only contain clusters that are significant along a wide range of scales. In the case of the technique we are proposing here, it is important to have clusters representative of each scale.

The main disadvantage of using DBSCAN is the introduction of the parameters $eps_0$ and $MinPoints$. For each iteration of the recursive clustering algorithm it is necessary to set appropriate values for these parameters, and this can only be done heuristically and not in a fully tractable way. The heuristic proposed in [7] for determinating suitable $eps_0$ and $MinPoints$ consists on the examination of the *Sorted K-distance graph* [2]. This limitation will be further discussed in sections 4 and 5.

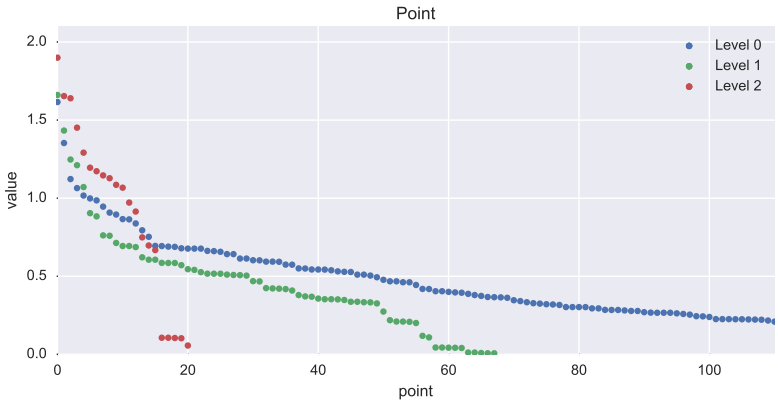



**Fig. 1.** K-distance graphs for three successive cluster levels. The distance value of the "valley" used to estimate $eps_0$ drops with each iteration.

On each iteration of the recursive DBSCAN procedure, the $eps_0$ value passed to the DBSCAN implementation is multiplied by a scaling factor (*decay*), this parameter represents the drop in the relative density of the clusters detected at each scale. Figure 1, shows the different *k-distance graphs* for each scale level for a sample from the experiment that will be discussed in Section 4. As can be seen from the graph, the distance of the "valley" drops as the scale increases (at greater resolution levels), this implies that we must use smaller $eps_0$ values at each iteration.

The process of regular activity patterns extraction begins, as in [21] and [11], by dividing each day in arbitrary time segments. Ideally, each of these intervals would represent periods where the processes producing the data are stable, for example, the morning commuting peak, working hours, etc. The general workflow

[2] The *Sorted K-distance graph* plots the number of points that exhibit a given distance to their first k-neighbors.

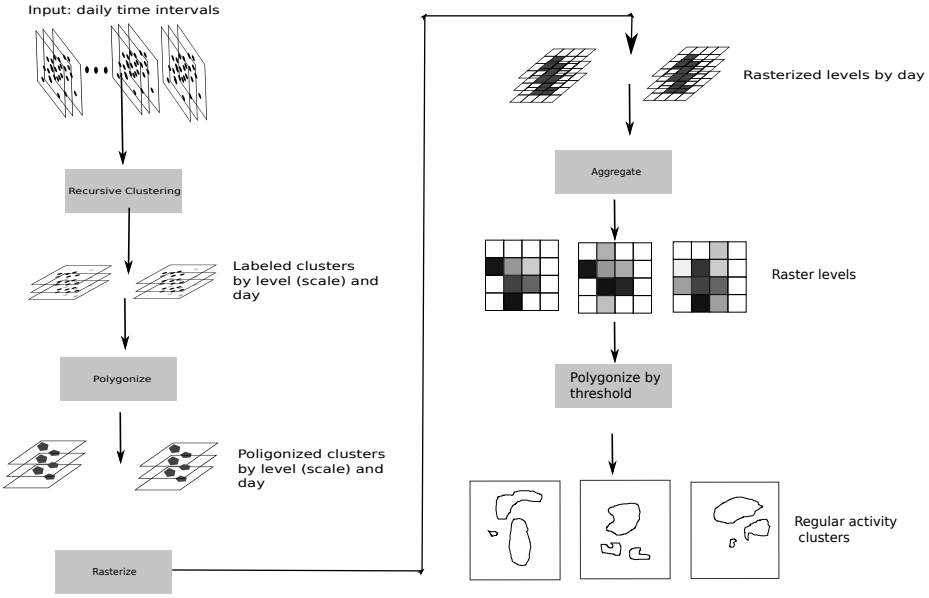



**Fig. 2.** Diagram showing the workflow for extracting regular activity polygons across multiple scales.

of the process to build the regular activity patterns across the scale levels is shown in Figure 2.

The recursive clustering procedure returns a list of labeled points, the labels represent the clusters to which each point belongs. The next step in the procedure is transforming these sets of labeled points into polygons for each cluster. In order to preserve the property of DBSCAN of producing arbitrarily shaped clusters, we polygonize the points using the Alpha Shape [6] instead of the Convex Hull. It is important to note that, although in general, Alpha Shape extraction involves a parameter determining how closely the polygons follow the underlying points, it is possible to extract an optimal alpha value subject to the following restrictions: 1) the number of connected components is given; 2) all points are either on the boundary or in the interior of the regularized version of the alpha-shape (no singular edges). The first condition implies that each cluster will be represented by a single polygon and the second one that such polygons will be simple.

Once the polygons for each day (and time segment) have been extracted, the next step is to "average" those polygons to find the regular zones of activity. To do this, the polygons are rasterized, i.e. converted to an image whose pixel values are 1 if the pixel lies within the polygon and 0 otherwise. This rasterization introduces another parameter, the *resolution*, that is, the pixel size in the rasterization process.

The rasterized polygons are then aggregated over the whole study period, thus obtaining images whose pixel values represent the number of days a given

pixel has been inside a cluster. Figure 6 shows examples of such images for the dataset used in the experiment discussed in section 4.

The images obtained represent the activity patterns of the spatio-temporal events at different resolution levels. Now, in order to characterize this patterns we need a way of assigning the characteristics of the underlying point distribution to the patterns extracted, in the same fashion as Lee, et al. characterize each Voronoi polygon with the count, diversity and movement variables or Frias-Martinez, et al., use the point count aggregated over twenty minute intervals.

In order to achieve this characterization, the raster images are polygonized. This is done by cutting by a threshold value. This threshold represents the number of days a pixel must belong to a cluster in order for it to be considered within zone of regular activity.

After this process, the zones of regular activity are obtained as sets of polygons for each resolution (scale) level. The final step is the characterization of these polygons. This characterization is not considered a part of the regular activity zones extraction since it depends on the objectives of the analysis and the general characteristics of the processes producing the events database.

Before moving on to the experiment that will demonstrate an application of the technique we are proposing, we will briefly discuss all the free parameters involved in this procedure:

$eps_0$ parameter: Threshold distance above which samples are considered noise in DBSCAN. The heuristic is to find the first "Valley" in the *k-distance graph*.

$MinPoints$ parameter: Number of points at which the $eps_0$ value crosses the *k-distance graph*. The heuristic is the Same as $eps_0$.

*decay* parameter: Drop in $eps_0$ value as the resolution increases. The heuristic is the observation of the different *k-distance graphs* for successive resolution levels.

$MinSamples$ parameter: Minimum number of points in a cluster to be a candidate for recursive clustering. The heuristic is that below this cluster size, there will be no further resolution levels, so this is should be set according to the minimum scale of detected activity clusters.

*resolution* parameter: Pixel size for the rasterization of each scale level where the size of the detected regular activity zones will be around 4 times the *resolution* size.

*threshold* parameter: Proportion of days a pixel must belong to a cluster to be considered part of the regular activity patterns. The heuristic explanation is that at larger *threshold* values the resulting polygons will be present in more individual (daily) samples, so this parameter should be set according to a statistical definition of the usual activity.

In the following section we will describe an experiment showing an application of the proposed technique to the extraction of the regular patterns of activity, using the geolocated Twitter feed, for the Central region of Mexico.

## 4 Experiments

As a first use case of the proposed technique, we will extract the regular patterns of activity in the Central Mexico area. The test database consists of all geolocated tweets from October 10 2014 to April 4 2015, there are 5,415,827 tweets within this period. Prior to the construction of the base scenario, we need to deal with the *pollution* commonly encountered in the tweeter feed [[14], [20]], this means that we must perform some preprocessing to clean up the database. In this case we want to filter out tweets by users that have more than one update within 100 meters of their original location in the same time period. The rationale behind this filtering is that this kind of behavior might be representative of bots or that it might artificially alter the shape of the clusters without representing the regular activity of the population.

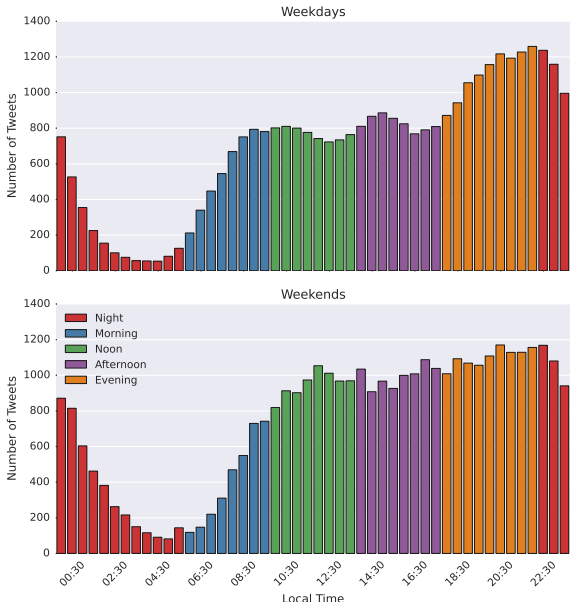



**Fig. 3.** Twitter activity for each time segment. The bars represent the amount of geolocated tweets aggregated over 30 minutes intervals.

The next step in the construction of the base scenario is the segmentation of time, in this case we will use intervals commonly used in several urban activity studies. Different regular activity patterns will be built for weekdays and weekends, since the patterns of activity are expected to be different. Each day will be segmented as follows: **Morning**: From 06:00 to 10:00, **Noon**: From 10:01 to 14:00, **Afternoon**: From 14:01 to 18:00, **Evening**: From 18:01 to 22:00

and **Night**: From 22:01 to 06:00.The resulting temporal activity patterns can be seen in Fig. 3.

Once we have defined the time segments, it is necessary to set appropriate values to the parameters defined in section 3. At this stage the significant parameters are $eps_0$, *decay* and $MinPoints$, since these will determine the number of scale levels and the clusters detected at each level. Following [7], it is possible to find suitable $eps_0$ and $MinPoints$ values by examining the *sorted k-distance* graph for the events population. Fig. 4 shows the *sorted k-distance* graphs for the different time intervals defined above (the Night interval is left out since it has very little activity).

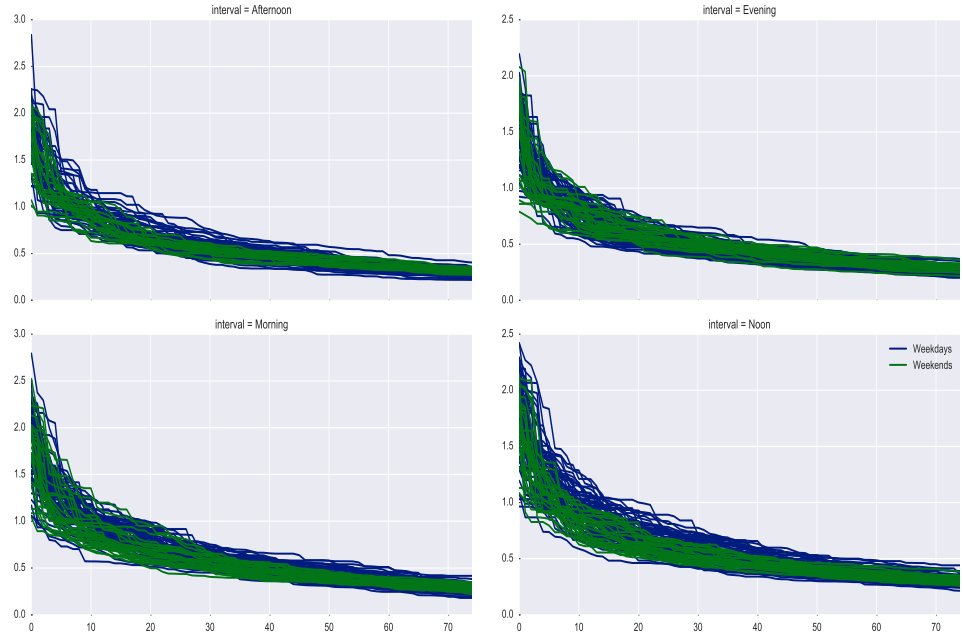



**Fig. 4.** Sorted k-distance plots for all time segments and every day in the experiment.

According to the heuristic proposed in [7], the value of $eps_0$ can be determined by finding the first "valley" in the *sorted k-distance* graph, that is, the distance at which we observe a sharp change in the decay rate of the nearest neighbor distance vs. the number of points. In the case of the present technique, we must find a value that is a suitable candidate for clustering every interval of every day within the study period. In Figure 5, we show a single interval *sorted k-distance* graph. As can be seen from the graph, every day shows a different "valley", so we end up having a range of suitable $eps_0$ values, which makes the selection somewhat arbitrary. Although this might seem a major obstacle, we will show that by choosing a value of $eps_0$ (and implicitly selecting $MinPoints$) around the

middle of the aforementioned range, we can find average (in the sense described in section 3) activity clusters that resemble the known spatio-temporal patterns of activity in the study area. For the rest of the present experiment we will use a value of 0.8 for $eps_0$ and 18 for $MinPoints$.

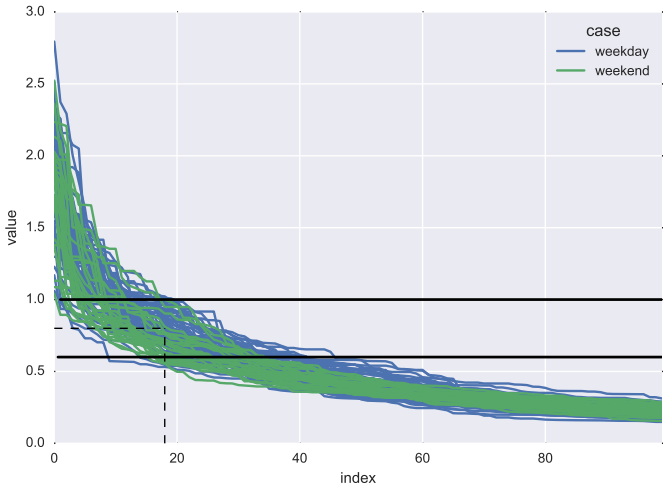
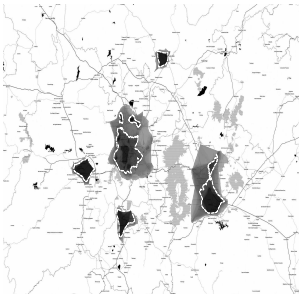



**Fig. 5.** Sorted k-distance plots for a single time segment. The horizontal lines show the range of "valley" candidates and the dashed lines show the selected values for $eps_0$ and $MinPoints$.

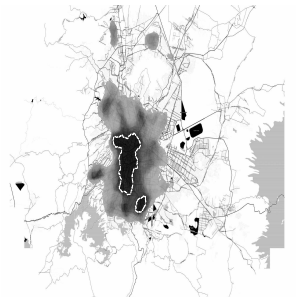
The next parameter we need to choose, as described in section 3, is $MinSamples$, the minimum number of points a cluster must have to be considered a candidate for having clusters of bigger scale. In the case of $MinSamples$, the selection criteria is based on the cluster size, in number of points, at the maximum resolution level, in the case of the current dataset we will set this value at 50.

The final parameter we need in order to perform recursive clustering with DBSCAN, is the *decay* value. This parameter represents the drop in the distance that separates the cluster samples from the noise samples in each DBSCAN iteration. As can be seen from Figure 1, the distance value of the "valley" drops from around 0.6 in the top level to around 0.1 in the bottom level, since the decay rate in the algorithm is constant we will use a *decay* value of 0.4 which represent a 0.8 drop across two levels (this value is set after examination of several point samples).

Finally, we need to set values for the *resolution* and *threshold* parameters. For the *resolution* parameter we will use a value o 100 meters, which means the the smaller cluster size we will detect as regular activity will be similar to a 400m by 400m square. The *threshod* value will be set at 0.75 which means that



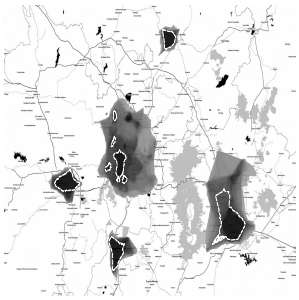
Evening Level 0.



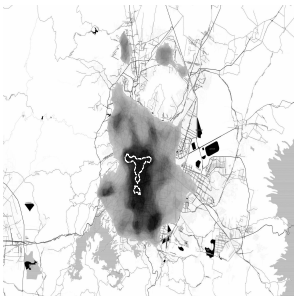
Evening Level 1.

Evening Level 2.

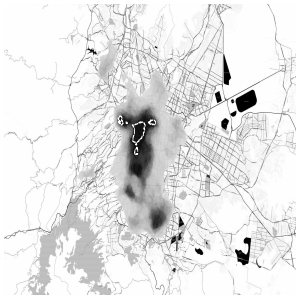
Evening Level 3.



Afternoon Level 0.



Afternoon Level 1.



Afternoon Level 2.

Afternoon Level 3.

**Fig. 6.** Aggregated activity rasters, with threshold cut polygons in dashed lines, for the Evening and Afternoon periods.

we require a cluster to be present in at least 75% of the sample in order to be considered a regular activity zone.

With all the parameters set, the regular activity zones are calculated, the resulting aggregated rasters are shown in Figure 6, together with the *threshold* cut resulting polygons.

### 4.1 Discussion of the Results

At the smaller scale (Level 0 in Figure 6), our technique is able to detect the greater metropolitan areas of Mexico City, Puebla, Toluca, Pachuca and Cuernavaca, in the Central Mexico region. As we increase the resolution, the sample density in the smaller cities (Puebla, Toluca, Pachuca and Cuernavaca), does not allow for the recursive algorithm to detect larger scale activity, the opposite is true for Mexico city, where we are able to detect up to three scales within the city.

By comparing the patterns found for the Afternoon and Evening intervals, we see that in the latter, the activity is more dispersed and Level 0 shows activity peaks in the northern low income housing suburbs. On the other hand, the activity for the Afternoon segment is more concentrated around the Central Buissnes District (CBD) of the city. This results are consistent with the known activity patterns for Mexico City. For example, Suarez and Delgado [25] performed a study in the Job-Housing ratio and found the same T-shaped pattern for the CBD. From the same study, we can see that the job to housing ratio of the northern low income suburbs is very low, which means it is mostly a residential area, this is in line with our results that show activity peaks for those areas only at the Evening intervals, that is when people are mostly at home.

## 5 Conclusions and Further Work

The technique presented in this paper represents an improvement on the available methods for determining the regular activity zones within the geolocated Twitter feed. Its main improvements are: 1) the ability to detect regular patters at different scale or resolution levels. This allows us to detect both major urban areas and activity zones within those areas that have a high enough activity density, 2) Using the Alpha Shapes to polygonize the clusters, allows us to account for the shape of the regular activity zones. This represents an improvement compared to the use of Voronoi tessellations.

The qualitative analysis of the obtained regular activity zones, show great accordance with the known activity patterns for the study area, mainly with the spatial distribution of the Job-Housing ratio. Albeit a formal quantitative validation of the activity patterns is missing, the fact that the technique is able to reproduce qualitatively the spatial distribution of activities within the city is very promising. The next step is the quantitative validation of the results obtained. For this, an updated Job-Housing ratio map must be built and the mobility patterns could be extracted from an Origin-Destination Survey (although the

must recent one available for Mexico City is from 2007, the government has announced plans to conduct a new study).

It is also necessary to find more tractable approximations to setting values for the parameters used in regular activity extraction. Ground truthing against measured activity distributions would provide basis for a calibration-validation approach. Finally, the multi-scale regular activity zones could be used to detect unusual crowd activity at various scales. The rationale behind this is that unusual events also exhibit scale differences. For example, it is known that important large scale events, such as the Super Bowl or the Arab Spring, produce a general increase of messages in the social networks, while localized small scale occurrences, such as festivals, demonstrations or accidents, produce small clusters of messages around the locations affected.

## References

1. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: Ordering Points to Identify the Clustering Structure. In: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data. pp. 49–60. SIGMOD '99, ACM, New York, NY, USA (1999)
2. Arcaute, E., Molinero, C., Hatna, E., Murcio, R., Vargas-Ruiz, C., Masucci, P., Wang, J., Batty, M.: Hierarchical organisation of Britain through percolation theory. arXiv:1504.08318 [physics] (Apr 2015)
3. Atefeh, F., Khreich, W.: A Survey of Techniques for Event Detection in Twitter. Computational Intelligence 31(1), 132–164 (Feb 2015)
4. Boettcher, A., Lee, D.: EventRadar: A Real-Time Local Event Detection Scheme Using Twitter Stream. pp. 358–367. IEEE (Nov 2012)
5. Crooks, A., Croitoru, A., Stefanidis, A., Radzikowski, J.: #Earthquake: Twitter as a Distributed Sensor System. Transactions in GIS 17(1), 124–147 (Feb 2013)
6. Edelsbrunner, H.: Smooth surfaces for multi-scale shape representation. In: Thiagarajan, P.S. (ed.) Foundations of Software Technology and Theoretical Computer Science, pp. 391–412. No. 1026 in Lecture Notes in Computer Science, Springer Berlin Heidelberg (Dec 1995)
7. Ester, M., Kriegel, H.p., S, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). pp. 226–231. AAAI Press (1996)
8. Estivill-Castro, V., Lee, I.: Amoeba: Hierarchical Clustering Based On Spatial Proximity Using Delaunaty Diagram (2000)
9. Estrada, R., Molina Villegas, A., Perez-Espinosa, A., Reyes-C, A., Quiroz, J., BravoG, E.: Zonification of heavy traffic in mexico city. In: Proceedings of the International Conference on Data Mining (DMIN). p. 40. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp) (2016)
10. Frias-Martinez, V., Soto, V., Hohwald, H., Frias-Martinez, E.: Characterizing Urban Landscapes Using Geolocated Tweets. In: Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom). pp. 239–248 (Sep 2012)

11. Frias-Martinez, V., Frias-Martinez, E.: Spectral clustering for sensing urban land use using Twitter activity. Engineering Applications of Artificial Intelligence 35, 237–245 (2014)
12. Fujisaka, T., Lee, R., Sumiya, K.: Detection of Unusually Crowded Places through Micro-Blogging Sites. In: 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA). pp. 467–472 (Apr 2010)
13. Gabrielli, L., Rinzivillo, S., Ronzano, F., Villatoro, D.: From Tweets to Semantic Trajectories: Mining Anomalous Urban Mobility Patterns. In: Nin, J., Villatoro, D. (eds.) Citizen in Sensor Networks, pp. 26–35. Lecture Notes in Computer Science, Springer International Publishing (Jan 2014)
14. Hurlock, J., Wilson, M.L.: Searching Twitter: Separating the Tweet from the Chaff. In: ICWSM. pp. 161–168 (2011)
15. Karypis, G., Han, E.H., Kumar, V.: Chameleon: hierarchical clustering using dynamic modeling. Computer 32(8), 68–75 (Aug 1999)
16. Khan, A.: The role social of media and modern technology in arabs spring. Far East Journal of Psychology and Business 7 No 1 Paper 4 April(4), 56–63 (2012)
17. Kim, T., Huerta-Canepa, G., Park, J., Hyun, S., Lee, D.: What's Happening: Finding Spontaneous User Clusters Nearby Using Twitter. In: 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom) (Oct 2011)
18. Kisilevich, S., Mansmann, F., Nanni, M., Rinzivillo, S.: Spatio-temporal clustering. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 855–874. Springer US (Jan 2010)
19. Kohonen, T.: The self-organizing map. Proceedings of the IEEE 78(9), 1464–1480 (Sep 1990)
20. Lee, R., Sumiya, K.: Measuring Geographical Regularities of Crowd Behaviors for Twitter-based Geo-social Event Detection. In: Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on Location Based Social Networks. pp. 1–10. LBSN '10, ACM, New York, NY, USA (2010)
21. Lee, R., Wakamiya, S., Sumiya, K.: Discovery of unusual regional social activities using geo-tagged microblogs. World Wide Web 14(4), 321–349 (2011)
22. Li, L., Xi, Y.: Research on Clustering Algorithm and Its Parallelization Strategy. In: 2011 International Conference on Computational and Information Sciences (ICCIS). pp. 325–328 (Oct 2011)
23. Oh, O., Agrawal, M., Rao, H.R.: Information control and terrorism: Tracking the Mumbai terrorist attack through twitter. Information Systems Frontiers 13(1), 33–43 (Sep 2010)
24. Starbird, K., Palen, L., Hughes, A.L., Vieweg, S.: Chatter on the Red: What Hazards Threat Reveals About the Social Life of Microblogged Information. In: Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work. pp. 241–250. CSCW '10, ACM, New York, NY, USA (2010)
25. Suarez, M., Delgado, J.: Is Mexico City Polycentric? A Trip Attraction Capacity Approach. Urban Studies 46(10), 2187–2211 (Sep 2009)

ISSN 1870-4069

# An Analysis of Demographic and Dietary Data with an Oral Health Approach: A Preliminary Study using Genetic Algorithms

Laura A. Zanella-Calzada[1], Carlos E. Galván-Tejada[1], Nubia M. Chávez-Lamas[2], María Del Carmen Gracia-Cortez[2], Jorge I. Galván-Tejada[1]

[1] Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica, Zacatecas, Mexico
{lzanellac, ericgalvan, gatejo}@uaz.edu.mx
[2] Universidad Autónoma de Zacatecas, Unidad Académica de Odontología, Zacatecas, Mexico
{nubiachavez, gacc005340}@uaz.edu.mx

**Abstract.** Oral health is one of the main components in the quality of life of people, since it is a determining factor in general health that affects the risk of suffering from other conditions, such as chronic diseases. Dental caries is the condition that most affects oral health worldwide, and occurs in about 90 % of people. The high prevalence in dental caries is caused by the diverse elements that interact simultaneously in their favor, such as the nutritional and socioeconomic elements. Based on this problem, this study proposes the analysis of a series of dietetic and demographic features compiled by the National Health and Nutrition Examination Survey 2013 – 2014, in order to obtain a model that allows the automatic classification of subjects according to their oral health status, presence or absence / restorations of caries. The methodology was carried out in three steps, starting with a data preprocessing, followed by a feature selection using a genetic algorithm and a final validation through a statistical analysis. The developed model is made up of five features (of the initial 188), and reached an area under the curve of 0.748, which is a statistically significant value. According to the results obtained, it was possible to conclude that the proposed model presents a preliminary result for the development of a low-cost support tool that helps the diagnosis of dental caries and the reduction of this condition.

**Keywords:** oral health, dental caries, classification, feature selection, genetic algorithm, statistical analysis.

## 1 Introduction

Chronic diseases is one of the main problems in public health and, nowadays, the pattern of diseases has changed, turning oral diseases into an important public health problem throughout the world. Oral diseases have a high incidence and

prevalence in all regions of the world, especially in disadvantaged and socially marginalized populations. According to the World Health Organization (WHO), the treatment of various oral conditions is extremely expensive and is not feasible in most low and middle income countries, being the fourth most expensive cause to treat [1]. Oral health presents an essential component in the quality of life due to its great influence on general health, since this condition can increase the risk of chronic diseases, such as: cardiovascular and cerebrovascular, diabetes mellitus and respiratory [2].

The most common condition in oral health is dental caries, which according to the WHO, affects between 60% and 90 % of children between five and 17 years. There is a series of determinants that favor this condition, such as carbohydrate consumption, food characteristics, plaque removal, among others, that interact simultaneously with variables that correspond to different orders of biological processes, complex historical-cultural structures, social relations, socioeconomic level, educational level, among others [3, 4].

Due to the difficulty of controlling the incidence of caries, which is caused by the large number of factors that influence, recent studies have implemented algorithms and performed analyzes based on computer-aided diagnosis (CADx) to develop prediction and classification models for preventive diagnosis and reduction of dental caries prevalence, looking for the main factors that affect this condition [5]. This study presents an analysis of two types of determinants that affect the important condition in oral health, dental caries. These determinants are demographic and dietary features that have been collected by the National Health and Nutrition Examination Survey (NHANES), 2013 – 2014. With a feature selection through a genetic algorithm, a multivariate model is developed that shows significant results in the classification of subjects with presence of this condition of subjects against absence, validated under a statistical analysis.

This work is organized as follows; Section 2 presents the methodology followed in three main steps, data preprocessing, feature selection and validation. In Section 3, the results obtained are shown and in Section 4 these results are discussed. Finally, conclusions are briefly described in Section 5.

## 2 Methodology

The methodology of this work is given in Figure 1, which is a flowchart of the stages that were followed for the development of this study.

A data preprocessing step (A) was performed to avoid any problem related to missing data or outliers that could affect the later stages. Then, in (B) a feature selection step is presented, which was carried out using the genetic algorithm "Galgo" to find the features that show the most significant behavior for the correct and automated classification of the subjects. Finally, (C) presents a validation step, where the features that were selected were subjected to a logistic regression, obtaining a general model for the classification of subjects that allows knowing the true positive and true negative rates through the calculation of

the receiver operating characteristic (ROC) curve and the area under the curve (AUC).

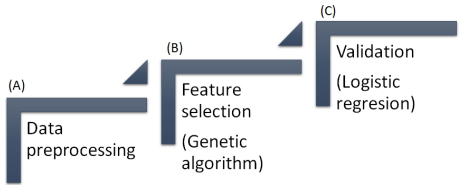



**Fig. 1.** Flowchart of the methodology followed.

All the methodology was performed using "R" (version 3.4.4) [6], which is a free software environment for statistical computing and graphics.

### 2.1 Data Description

NHANES is a program of studies designed to assess the health and nutritional status of children and adults in the United States of America (USA)and was founded by the Centers of Disease Control and Prevention (CDC) and the National Center for Health Statistics (NCHS). The surveys conducted by this program are unique, since they combine interviews and physical examinations [7]. NHANES collects information from different types of data and, in turn, this information is included in six main contexts; demographic, dietary, examination, laboratory, questionnaire and limited access. For this work, the demographic and dietary datasets were used;

- Demographic: it provides individual, family and household level information in different topics (income of households and families, size of households and families, pregnancy status, among others).
- Dietary: it provides detailed information on dietary intake, in order to estimate the types and amount of food and beverages consumed, in addition to estimating the intake of energy, nutrients, and other food components.

These datasets were contained by 188 features, and they were used as input features; while the condition of dental caries (absence or presence / restorations) was used as output feature. The subjects that were contained in those datasets belong to different counties in the USA and they were randomly selected with a computer algorithm by NHANES. The total number of subjects was 9812 (3690 controls / 6122 cases), 4982 females and 4830 males, and they were in a range age between zero and 89 years old.

### 2.2 Data Preprocessing

The two main purposes of the data preprocessing stage were to avoid problems of missing data and outliers.

Initially, all incomplete cases were manually removed, eliminating those subjects that presented $\geq$ 70 % of missing data. Of the rest of the subjects, all the missing values were imputed through the "nearestneighborimpute" function of the package "FRESA.CAD" (version 3.0.1) [8]. This function searches for any "NA" (not available) that is present in the dataset and looks for the row of complete observations that have the closest interquartile range normalized Manhattan distance to the rows that present missing values. In case that more than one row has similar minimum distances, the median value is used.

Then, the data was normalized using Z normalization, forcing the data to have a standard normal distribution. It was calculated with Equation 1, where $x_i$ refers to the data, $\mu$ to the mean value and $\sigma$ to the standard deviation:

$$Z_i = \frac{x_i - \mu}{\sigma}. \tag{1}$$

Finally, the singular values were removed, eliminating those columns that presented multiple values among themselves or the same value along the entire rows. This step was performed to avoid redundant or non-significant information.

### 2.3 Feature Selection

The feature selection was carried out through "Galgo" (version 1.1) [9], an R package for the selection of multivariate variable using genetic algorithms. In addition, Galgo presents a series of functions for the analysis of the population contained in the models and for the reconstruction and characterization of the models.

The Galgo procedure begins with a set of subsets of features or genes (named chromosomes) contained by data that was randomly selected. Then, each chromosome is evaluated based on its ability to predict a dependent variable, measuring its level of accuracy. The general idea is to replace the initial set of data with new data that conforms variants of chromosomes with a higher classification accuracy and repeat this process until achieving a desired level of accuracy. The progressive improvement of the chromosomes is carried out through a series of steps that resemble the process of natural selection, which consists of three steps; selection, mutation and crossover.

Looking for the increment of the solution space that is explored, independent chromosomes can evolve in isolated environments (named niches). Chromosomes can occasionally migrate between niches ensuring that that good solutions can recombine.

The classification methods that are included in Galgo are "k-nearest-neighbors", "nearest centroid", "support vector machines", "neural networks", "classification trees" and "discriminant functions" [10].

The main four steps that Galgo follows are briefly described.

1. **Setting-up the analysis.** This step is necessary to specify the input and the output features, the statistical model, the desired accuracy, the error estimation scheme and the parameters that decide the search environment of

the genetic algorithm. It is important to mention that the estimation of the error can be defined in two levels; a validation strategy for training / testing, and within the training process using k-fold cross-validation, random splits or re-substitution.

2. **Searching for relevant multivariate models.** Starting with a random set of chromosomes, the genetic algorithm procedure will look for a diverse collection of statistically significant local solutions. A sufficiently large number of chromosomes must be selected to have a good representation of the solution space.
3. **Refinement and analysis of the set of selected chromosomes.** The chromosomes that were selected have a length that was previously defined. Although the models have reached the desired classification accuracy, a refinement step is carried out, since there is a possibility that not all the model genes have a significant contribution to accuracy. Through a backward selection strategy, only those genes that effectively contribute to the classification task are selected.
4. **Development of a representative statistical model.** Finally, in this part of the analysis a single representative model of the selected chromosomes is obtained. For this step, a forward selection strategy was implemented based on the step-wise inclusion of the most frequent genes in the chromosome.

For this work, the settings used in this experimentation were the following: chromosomes composed of five genes, with 200 evolutionary processes in 200 big bangs. These settings were selected given the number of features and patients to avoid overfitting.

On the other hand, the classification model used was the nearest centroid, which is a supervised method that assigns to the observations the class label of the training samples whose mean or centroid is the closest to the observations. For each set of data points belonging to the same class, the centroid vectors are calculated. If there are $k$ classes in the training set, there are $k$ centroid vectors. The test samples are classified in the class with the nearest centroid.

For the training procedure, given the labeled training samples $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)$, with class labels $y_i \in Y$, the class centroids $\boldsymbol{\mu}_k$ are calculated with Equation 2, where $C_k$ is the set of indices of samples belonging to class $k \in Y$:

$$\boldsymbol{\mu_k} = \frac{1}{|C_k|} \sum_{l \in C_k \boldsymbol{x}_i} . \tag{2}$$

The prediction function $\hat{y}$, which is the class assigned to an observation $\boldsymbol{x}$ is calculated with Equation 3:

$$\hat{y} = min_{k \in Y} ||\boldsymbol{\mu}_k - \boldsymbol{x}||. \tag{3}$$

### 2.4 Validation

The validation stage was carried out in order to evaluate the multivariate model resulting from the feature selection. Initially, the model was subjected to a

Logistic Regression (LR) to obtain a general model for the classification of subjects using the set of selected features. LR is an analysis that consists of a statistical technique to model the relationship between the features. This method belongs to the statistical methods where the contribution of different factors in the occurrence of a simple event is measured. The main objective of LR is to model the influence of the probability of an event. The simplest representation of a model obtained by this method is presented in Equation 4, where $y$ from $logit(y = 1)$ is the dependent variable or the outcome feature that is necessary to be subjected to a logarithmic transformation ($logit$) because the initial equation of the model is of exponential type and by this transformation it is possible to use it as a lineal function, $w$ is an offset term that can be included, $\beta_1$ is the slope and $x$ is the independent variable or the analyzed feature. Models can be composed by the number of independent variables needed [12]:

$$logit(y = 1) = w + \beta_1 x. \tag{4}$$

On the other hand, the AUC value is a standard method used for the evaluation of the accuracy obtained by the model, and is calculated with the relationship between the specificity and the sensitivity [13].

The sensitivity parameter is referred to the proportion of data that belongs to a condition and it is classified as positive. This value is obtained with Equation 5, where $TP$ represents the quantity of true positives and $FP$ represents the quantity of false positives:

$$PPV = \frac{TP}{TP + FP}. \tag{5}$$

The specificity parameter is referred to the proportion of subjects without a condition that are classified as negative. This value is obtained with Equation 6, where $TN$ represents the quantity of true negatives and $FN$ represents the quantity of false negatives:

$$NPV = \frac{TN}{TN + FN}. \tag{6}$$

## 3 Results

The results obtained from the methodology performed are presented in this section.

From the preprocessing step, a series of features were eliminated according to the missing data and singular values that were present in the dataset, maintaining a total of 136 dietary and demographic features. Then, those features were submitted to the Galgo algorithm for a feature selection.

Figure 2 presents a graph where all the features are ordered according to their degree of appearance in the chromosomes throughout the iterations. The features shown in black have the highest frequency, while the features shown in gray have the lowest.

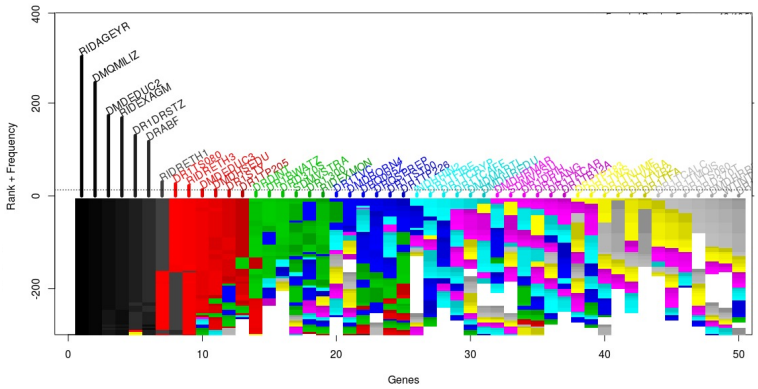



**Fig. 2.** Graph of the frequency with which the features appear on chromosomes, ordered descending. Vertical axis presents the frequency number and horizontal axis presents the chromosomes.

Figure 3 presents the performance graph of forward selection, showing the classification accuracy achieved for each feature included in the model. The features are arranged according to their frequency of appearance, from the highest to the lowest.

After the forward selection step, the backward elimination process was performed, obtaining a final model that presented the best accuracy without any redundant information. This model was contained by the five features that are described in Table 1.

**Table 1.** Features obtained from the backward elimination step.

| Feature | Description |
|---|---|
| RIDAGEYR | Age in years, at the time of the screening interview, is reported for survey participants between the ages of 1 and 79 years of age. |
| RIDEXAGM | Age in months of the participant at the time of examination. |
| DMQMILIZ | Have you ever served on active duty in the U.S. Armed Forces, Military reserves, or National Guard? |
| DMDEDUC2 | What is the highest grade or level of school you have completed or the highest degree you have received? |
| DR1DRSTZ | Dietary recall status. |

Finally, after the feature selection, the validation step was carried out, where the ROC curve presented in Figure 4 was obtained, having an AUC of 0.748.

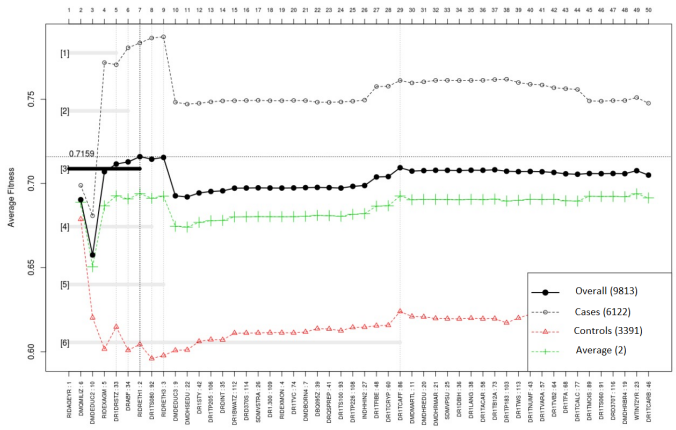



**Fig. 3.** Graph of the forward selection step. Vertical axis presents the accuracy value and horizontal axis presents the chromosomes ordered according with their frequency appearance.

## 4 Discussion

The results obtained show a multivariate model, contained by demographic and dietary features, which was obtained through the Galgo genetic algorithm, which presents a statistically significant performance in the classification of subjects who show absence of dental caries from those who have presence or restorations of them.

Figure 2 presents the graph of the frequency that each feature obtained according to the number of times they appeared in the different chromosomes that developed throughout the Galgo process, where it is possible to observe that from the total features, there where seven with a significant number of frequency, which are those presented in black; RIDAGEYR, DMQMILIZ, DMDEDUC2, RIDEXAGM, DR1DRSTZ, DRABF and RIDRETH1. Then, in Figure 3 the graph of the forward selection step is observed, where the accuracy obtained for each feature that is included in the model is presented; when the model is contained by the seven most frequent features, the best accuracy is achieved.

In addition, it is possible to observe that the classification of the cases (-○-) is much more accurate than the classification of the controls (-△-), making that the overall (-●-) classification reduces its accuracy. This may occur due to the low significant information that the features may be presenting for the control subjects or to the number of controls presented in the data set, which can be resolved by increasing the number of subjects and / or features.

Then, Table 1 includes the five features that were selected through the backward elimination step, eliminating two of the features that were selected in the previous step; DRABF and RIDRETH1. These features were removed because they didn't present any significant information for the classification of subjects or their contribution was redundant.

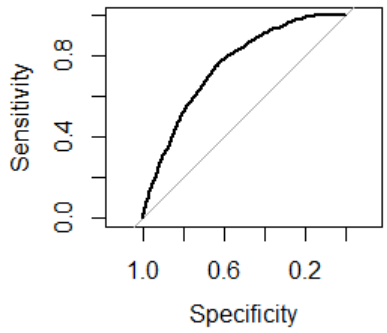



**Fig. 4.** ROC curve obtained from the classification of subjects.

The information presented in the multivariate model is related to the age of the subjects, the educational level, the state of the diet and the duty of activity in USA services. As mentioned above, one of the main factors that affect the state of oral health is age, with young people being the most affected by dental caries. On the other hand, the state of dietary recall is used to know the individual foods and the total intake of nutrients, which indicates the quality and integrity of the subjects, being one of the determinants of dental caries that was also mentioned above.

The educational level and the participation in the services of the USA are features related to the socioeconomic level. The educational level is significant because the low income regions have less educational opportunities; while participation in the services of the USA is an important feature taking into account that the main objective population of NHANES is the non-institutionalized civilian resident population of the USA, which may be people who decided to emigrate in search of better economic opportunities and one of the requirements for the Armed Forces of the USA, the Military Reserves and the National Guard is being a citizen or a permanent resident. Therefore, it is possible to justify the selection of those features for the developed model.

Finally, Figure 4 presents the ROC curve obtained according to the true positives and true negatives calculated with the developed model; its AUC value was 0.748, which is statistically significant since it means that from the total subjects, 74.8 % were correctly classified.

## 5 Conclusions

This paper presents the analysis of a series of demographic and dietary features, in order to develop a model that can present a tool for specialists in the preventive diagnosis of dental caries and the reduction of the incidence of this condition. This analysis was performed in three main stages, data preprocessing, feature

selection and validation. In the stage of feature selection, a multivariate model was developed that contained the features that provided the most significant information in the classification of control and case subjects, while the validation stage allowed the evaluation of this multivariate model, obtaining statistically significant results.

Therefore, the model proposed in this work may represent a low-cost preliminary tool that helps in the diagnosis of dental caries and in the possible prediction of them, both for high and low income regions, to reduce their high incidence and avoid the high cost treatments that this condition represents.

## References

1. World Health Organization (NCHS): Oral Health, http://www.who.int/oral-health/disease-burden/global/en/. Last accessed 05 June 2018
2. Ridao Marín, D.: Desarrollo de un Sistema de Ayuda a la Decisión para Tratamientos Odontológicos con Imágenes Digitales. Universidad de Málaga: Málaga, Spain, 10–12 (2017)
3. Espinoza Solano, M.; León-Manco, R.A.: Prevalencia y experiencia de caries dental en estudiantes según facultades de una universidad particular peruana. Rev. Estomatol. Hered. 25(3), 187–193 (2015)
4. Acuña Aguilar, L.D., Porras Cerón, D., Ríos Rueda, L.D.: Prevalencia de Lesiones Cariosas y Factores Asociados Presentes en Pacientes con SíNdrome de Down en las Fundaciones Fundown y san Luis Guanella de Bucaramanga. Universidad Santo Tomás: Bucaramanga, Colombia, pp. 12–16 (2017)
5. Gispert Abreu, E.D.L.Á., Castell-Florit Serrate, P., Herrera Nordet, M.: Salud bucal poblacional y su producción intersectorial. Rev. Cubana Estomatol.52, 62–67 (2015)
6. R: A Language and Environment for Statistical Computing, https://www.R-project.org/. Last accessed 17 June 2018
7. National Health and Nutrition Examination Survey Data. 2013–2014, http://www.cdc. gov/nchs/nhanes.htm. Last accessed 17 June 2018
8. FRESA.CAD: Feature Selection Algorithms for Computer Aided Diagnosis, https://CRAN.R-project.org/package=FRESA.CAD. Last accessed 17 June 2018
9. GALGO: an R package for multivariate variable selection using genetic algorithms, http://bioinformatica.mty.itesm.mx/galgo2. Last accessed 17 June 2018
10. Trevino, V., Falciani, F.: GALGO: an R package for multivariate variable selection using genetic algorithms. Bioinformatics 22(9), 1154–1156 (2006)
11. Das, T.: Machine Learning algorithms for Image Classification of hand digits and face recognition dataset. Machine Learning 4(12), 640–649 (2017)
12. Montgomery, D. C., Peck, E., Vining, G.: Introduction to linear regression analysis. John Wiley & Sons, Location (2015)
13. Berlanga Silvente, V., Vilà Baños, A.: AUC: a misleading measure of the performance of predictive distribution models. Global ecology and Biogeography 17(2), 145–151 (2008)

ISSN 1870-4069

# Credit Assignment: Using Resampling Methods for Dealing with the Class Imbalance Problem

Víctor D. de la Cruz-Galarza[1], Yenny Villuendas-Rey[1], Cornelio Yáñez-Márquez[2]

[1] Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Mexico City, Mexico
[2] Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico City, Mexico
mhwara@gmail.com, yenny.villuendas@gmail.com, coryanez@gmail.com

**Abstract.** Nowadays, credit assignment constitutes a way in which persons or entities access to money. However, bad clients can cause big distress to financial institutions. If there are appropriate data banks whose patterns contain financial information from the scope of the allocation of credits, the intelligent pattern classifiers are ideal candidates to solve the credit assignment problem. Nevertheless, working with data sets from credit environment has the disadvantage that, in most of the cases, have unbalanced classes. This situation represents a problem at the moment of work with this kind of datasets due to the fact that unbalanced classes, in general, create biased learning. The consequences of this are reflected during the testing phase because the biased learning causes the classifiers to just recognize appropriately the elements of the ruling class and therefore, give us inaccuracy results. In this paper, we tested some undersampling and oversampling algorithms, and we compared their performance, based on the Imbalance Ratio measure, over different well-known credit related datasets.

**Keywords:** credit assignment, sampling techniques, instance selection, imbalanced data.

## 1 Introduction

One of the main activities of banks is to provide loans to their clients. If a debtor does not pay the money that was loaned in the established term, it violates the trust that the creditor granted and possibly, will stop lending him money. There are companies whose purpose is to keep track of credit payment, through which it is known which people have fulfilled their obligations to pay and who have stopped doing so; these companies are known as Credit Information Companies [1]. That is why credit institutions must be very careful when granting loans, because in doing so they use the money that people have deposited in their bank accounts.

This scenario clearly demonstrates the difficulties that delinquent clients cause to financial institutions. In this sense, it is of great interest for credit companies to have the real possibility of intelligent tools that evaluate potential clients, with a certain acceptable degree of certainty; these tools should provide entities with valuable

information regarding potential clients and answer a question whose answer is crucial: is the potential client a good or bad payer?

The advantages that the correct answer of this question would bring to the financial system are translated into a scenario where, as far as possible, the expenses derived from non-payment of a bad client are avoided.

If there are appropriate data banks whose patterns contain financial information from the scope of the allocation of credits, the intelligent pattern classifiers are ideal candidates to solve the credit assignment problem. Nevertheless, working with data sets from credit environment has the disadvantage that, in most of the cases, they have unbalanced classes and mixed attribute types [2], for which is very important to choose the classification models in accordance with this situation.

A situation like unbalanced classes is present in a dataset when one of the classes has more elements than the others. This situation represents a problem while working with this kind of datasets because unbalanced classes, in general, creates biased learning. The consequences of this are reflected during the testing phase because the biased learning causes that the classifiers just recognize appropriately the elements of the ruling class and therefore, give us inaccuracy results.

In this article is an experimental work using different sampling algorithms with credit related datasets with the purpose to know which one has the best performance under the circumstances described above is presented.

The rest of the paper is organized as follows. Section 2 details some previous works and Section 3 offers a discussion about the results obtained. Finally, the paper ends with some conclusions and future research suggestions.

## 2 Previous Works

From the point of view of supervised classification, the problem of the assignment of credit is a problem of two classes (credit is assigned or not assigned to the requestor) and of an unbalanced nature. This imbalance occurs because, in practice, more credits are awarded than those that are rejected. However, the costs of classification are not the same for both classes, due to the very nature of the phenomenon. [3, 4].

For example, if a potential good applicant is denied credit, the financial institution loses that client. However, if a bad applicant is granted credit, the financial institution has monetary losses, and possibly expenses associated with legal actions that have to be taken to recover the money invested. That is why the class of greatest interest in this phenomenon is the detection of potential bad applicants, who should not be granted credit [5]. Paradoxically, this class of greatest interest is the minority class in this phenomenon, which adds complexity to the data banks of work that are involved in the search for solutions to the problem of credit allocation in the context of Intelligent Computing [6].

In the scientific literature of the state- of- the- art, it is possible to find research works that report attempts to solve the problem of credit allocation through the application of Intelligent Computing. In these investigations, various models of supervised classification have been used; among them, is highlighted the use of Support Vector

Machines [7], Artificial Neural Networks [8, 9] and Classifier ensembles [10, 11], among others [12, 13, 14]. The experimental comparisons made to determine the performance of the classifiers in terms of the allocation of credit [15, 16, 17], exhibit certain problems that prevent generalizing the results they have published.

On the one hand, the studies incorporate few data banks, and to complicate matters, many of the data banks used are not public, nor are they available for use; In addition, there are almost no common data banks in the different investigations. Additionally, in the documentary study of the state- of- the- art carried out in the framework of this work, it has been observed that, if a research group has used a certain supervised classifier, in other researches this is not taken into account, but rather they are used other supervised classifiers.

The No Free Lunch [18] theorems argue that there is no superiority of one classifier over others, over all data banks and all performance measures. However, recent studies point to the existence of a good performance of associative classifiers in the solution of problems of supervised classification of the financial environment [19].

It is a fact known to the scientific community that, on numerous occasions, the preprocessing of data contributes to the improvement of the performance of certain supervised classifiers; in particular, when data banks show an imbalance between classes [20, 21]. The literature reports several investigations that have been conducted in order to determine the impact of data preprocessing on improving solutions to the problem of granting credit [6, 22]. In particular, the computational problem related to the selection of instances (applicants) [4] has aroused great interest in the scientific community, so that in recent years emphasis has been placed on the study of techniques for the selection of classifiers for unbalanced data [3].

Moreover, in the comparative studies reviewed [5, 23], there is no consensus as to which are the best preprocessing techniques for the different classifiers in the allocation of credit. The previous considerations allow affirming, without a doubt, that the results of the mentioned experimental comparisons are hardly conclusive. With this investigation, it is intended to successfully attack this type of problem.

## 3 Results and Discussion

### 3.1 Datasets

In this section, we describe the data banks that will be used to evaluate the impact of the pre-processing of financial data on the performance of associative classifiers. These data banks are well known in literature, as well as being a reference, because they are widely used in many of the research works carried out so far. These data banks are known as "*Give me some credit*"[1], "*Iranian*"[2] and "*Polish bankruptcy*"[3], which are very interesting for this type of research, due to the nature of the data, because they include a wide variety of attributes, a high level of imbalance, in addition to having

[1] https://www.kaggle.com/c/GiveMeSomeCredit/data

[2] Personal shared by Hassan Sabzevari.

[3] https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data

missing data in many of the data banks. By way of summary, a description of the data banks used in the present investigation is shown in Table 1. The abbreviation IR represents the ratio of imbalance.

**Table 1.** Characteristics of the datasets used in this work.

| *Data set* | *Instances* | *Attributes* | *IR* | *Missing* |
|---|---|---|---|---|
| Give me credit | 150000 | 10 | 13.9611 | No |
| Iranian | 1002 | 28 | 19.0400 | Yes |
| Polish_year1 | 7027 | 64 | 24.9299 | Yes |
| Polish_year2 | 10173 | 64 | 24.4325 | Yes |
| Polish_year3 | 10503 | 64 | 20.2182 | Yes |
| Polish_year4 | 9792 | 64 | 18.0136 | Yes |
| Polish_year5 | 5910 | 64 | 13.4146 | Yes |

As shown, all data banks are very unbalanced (it is considered unbalanced from IR> 1.5 and all these data banks have IR> 13), and six contain absences of information. Note that in all cases have only two classes.

### 3.2 Algorithms to Compare

In a large number of works, novel methods have been proposed to address the problem of imbalance between classes. Those approaches are classified into two groups: approaches at the level of algorithms in which a new algorithm is created or one that already exists is modified and data-level approaches, in which data are modified in order to lessen the impact on the performance of classification algorithms when there is an imbalance in the distribution of classes.

In this section, the class balancing algorithms that will be evaluated in the present investigation are addressed. First, reference is made to the oversampling algorithms and, subsequently, to the sub-sampling algorithms (undersampling). In each case, its operation is detailed and a brief reference is made to its main characteristics, as well as its application or not to the financial field.

In the state- of- the- art, it is possible to find several articles [24, 25, 26, 27], where the pre-processing of data banks is addressed to reduce the impact caused by the distribution of classes. In those articles, it has been empirically demonstrated that the application of a preprocessing stage to balance the class distribution is usually a useful solution to improve the quality of the identification of new instances.

The data pre-processing techniques are divided into three groups: *Undersampling algorithms*, which are based on the elimination of instances of the majority class, *Oversampling algorithms*, which are based in the creation of instances of the minority class, by replication or modification of existing instances, and *Hybrid algorithms*, which are a combination of both over and under sampling techniques.

The oversampling algorithms seek to match the quantities of objects in each class by over-sampling the minority classes. In this way, the number of objects in these classes

will be artificially increased, ensuring that all classes have approximately the same number of objects. The main techniques of selection of instances by oversampling that will be used for the comparative analysis carried out in this work are listed and detailed below.

SMOTE: Synthetic Minority Over-sampling TEchnique [28], this method of preprocessing has become one of the most renowned in terms of oversampling techniques. The fundamental principle of this technique is based on creating synthetic instances for the minority class, through the nearest k neighbor of each of the instances of this class. The new instances are created by interpolating the sample vectors of the sample (example) of the minority class and its respective nearest neighbor. Each difference is multiplied randomly by zero or one. Then the non-zero characteristic vectors are taken as the new synthetic instances. This technique forces the decision region of the minority class to make it more general. The main disadvantage of this technique is that it can create instances that over train the classifier.

ADASYN: ADAptive SYNthetic Sampling [29], this method is based on the generation of instances adaptively for the minority class according to their distributions: the synthetic instances are generated for instances of the minority class that are more difficult to learn compared to those instances that are of the minority class and easier to learn.

ROS: Random over-sampling [24], This is a method that creates new synthetic instances in a random way. This is done until both classes contain the same number of instances.

ADOMS: Adjusting the Direction Of the synthetic Minority clasS examples [30], This method works similar to SMOTE. However, this method generates synthetic instances along the first principal component axis (PCA) of the local data of the distribution using the nearest k neighbors.

SPIDER: Selective Preprocessing of Imbalanced Data [31], this method combines local sampling of the minority class with filtering of difficult instances of the majority class. This method identifies which instances are labeled as noisy or difficult (misclassified) by the kNN classifier. Then, noisy instances can be duplicated, deleted or re-labeled depending on the option to be chosen (weak or strong).

As mentioned previously, the undersampling algorithms seek to equip the quantities of objects in each class, by sampling the major classes. Thus, the objects that are considered less relevant are eliminated, so that all classes have approximately the same number of objects. Next, the undersampling algorithms evaluated in the present investigation are explained.

TL: Tomek's modification of Condensed Nearest Neighbor [32], in this method if two instances form a Tomek link, then one of them is noise or the two instances are on the border. Prior to applying the condensed rule of the nearest neighbor (NN), this method obtains a set of objects containing only the objects near the decision boundaries.

RUS: Random under-sampling [24] is an algorithm that randomly selects instances of the majority class to be eliminated until both classes are balanced.

OSS: One Sided Selection [33], in this method, all the instances belonging to the minority class and instances of the misclassified major class are selected (by the 1-NN

classifier) in order to find Tomek links between them. Instances of the majority class participating in a Tomek link are removed.

CNNTL: Condensed Nearest Neighbor + Tomek's modification of Condensed Nearest Neighbor [24], In this algorithm the CNN and TL methods are combined. The main idea is to reduce the size of the original data set through the elimination of certain objects by applying CNN without significantly affecting the performance of the NN classification using the information provided by the Tomek link method.

NCL: Neighborhood Cleaning Rule [34], uses the ENN rule to remove objects from the majority class. ENN removes any object whose label class differs from the class of at least three of its five closest neighbors.

### 3.3 Discussion

Each algorithm was tested with the different datasets in the KEEL software [35] using the default parameters offered. We used a 5-fold cross validation procedure as model validation technique. Tables 2 and 3 show the results for the undersampling and oversampling algorithms, respectively. We use the Imbalance Ratio measure (IR) as performance measure.

**Table 2.** Imbalance Ratio for the undersampling algorithms.

| *Datasets* | CNNTL | NCL | OSS | RUS | TL | Original |
|---|---|---|---|---|---|---|
| Give me credit | 1.36 | 10.84 | 1.39 | 1.00 | 12.95 | 13.96 |
| Iranian | 1.47 | 14.53 | 2.18 | 1.00 | 17.91 | 19.00 |
| Polish_year1 | 1.40 | 21.14 | 2.50 | 1.00 | 23.70 | 24.93 |
| Polish_year2 | 1.41 | 20.31 | 2.56 | 1.00 | 22.97 | 24.43 |
| Polish_year3 | 1.33 | 16.33 | 2.45 | 1.00 | 18.86 | 20.22 |
| Polish_year4 | 1.25 | 14.42 | 2.22 | 1.00 | 16.72 | 18.01 |
| Polish_year5 | 1.15 | 10.23 | 1.62 | 1.00 | 12.28 | 13.41 |

As can be seen in Table 2, as expected, the Random undersampling Method (RUS) obtained a perfectly balanced dataset. In addition, the CNNTL algorithms obtained very good imbalance ratios, all of them very close to one. In a similar way, one Side Selection (OSS) obtained good results, although not as good as the ones by CNNTL. Neither NLC nor TL obtained good results, failing to obtain a balanced dataset.

As shown in Table 3, all oversampling algorithms but SPIDER obtained a perfectly balanced dataset. However, it was by significantly increasing the number of instances in the dataset. Figure 2 shows the differences according to instance amount among undersampling and oversampling methods. ADASYN, ADOMS, ROS and SMOTE algorithms obtained the same number of instances; then Figure 2 only depicts the results for ADASYN.

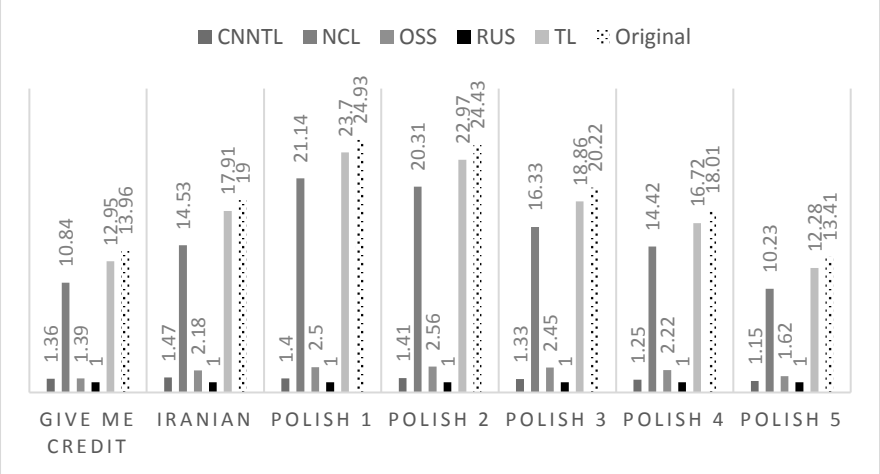



**Fig. 1.** Graphical representation of the Imbalance Ratio for undersampling methods.

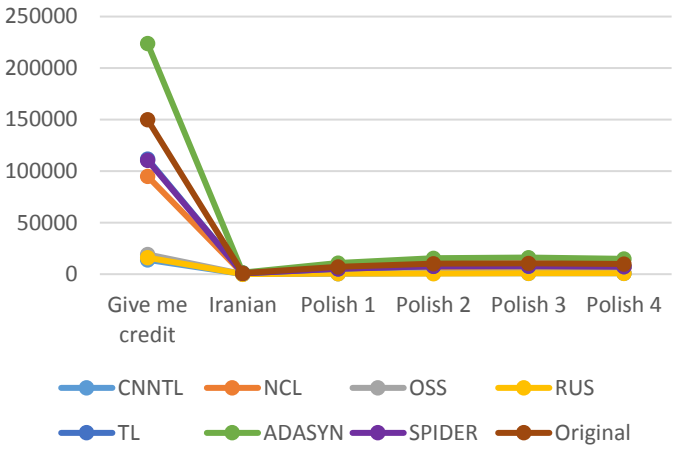



**Fig. 2.** Graphical representation of the amount of instances selected (X axis) for each considered dataset (Y axis).

**Table 3.** Imbalance Ratio for the oversampling algorithms.

| *Datasets* | ADASYN | ADOMS | ROS | SMOTE | SPIDER | Original |
|---|---|---|---|---|---|---|
| Give me credit | 1.00 | 1.00 | 1.00 | 1.00 | 4.22 | 13.96 |
| Iranian | 1.00 | 1.00 | 1.00 | 1.00 | 6.18 | 19.00 |
| Polish_year1 | 1.00 | 1.00 | 1.00 | 1.00 | 6.57 | 24.93 |
| Polish_year2 | 1.00 | 1.00 | 1.00 | 1.00 | 6.49 | 24.43 |
| Polish_year3 | 1.00 | 1.00 | 1.00 | 1.00 | 5.50 | 20.22 |
| Polish_year4 | 1.00 | 1.00 | 1.00 | 1.00 | 5.08 | 18.01 |
| Polish_year5 | 1.00 | 1.00 | 1.00 | 1.00 | 3.98 | 13.41 |

The experiments show that oversampling methods significantly increase the amount of instances, rising both the storage and execution computational costs. On the other hand, undersampling method obtain balanced datasets, and significantly reduce the amount of instances.

## 4 Conclusions and Future Work

In the credit environment, there are some datasets that can be considered important to test automated decision-making systems, but in most cases, these datasets have some characteristics (such as class imbalance) that make this task more complicated. In this work, we compared 10 different sampling techniques in credit environment using the imbalance ratio measure. Our studies showed that CNNNTL and RUS models turned out to be best sampling algorithm for almost all the datasets used in this work. As future work, we would address the classification performance of associative classifier over the original and balanced datasets.

**Acknowledgments**. The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, Comisión de Operación y Fomento de Actividades Académicas, Secretaría de Investigación y Posgrado, Centro de Investigación en Computación, and Centro de Innovación y Desarrollo Tecnológico en Cómputo), the CONACYT, and SNI for their economical support to develop this work.

## References

1. C. D. D. D. H. C. D. LA UNIÓN: Ley para regular las sociedades de información crediticia. Diario Oficial de la Federación. Ciudad de México (2002)
2. Abdi, L., Hashemi, S.: To combat multi-class imbalanced problems by means of over-sampling and boosting techniques. Soft Comput. 19(12), pp. 3369–3385 (2015)
3. Bischl, B., Kuhn, T., Szepannek, G.: On Class Imbalance Correction for Classification Algorithms in Credit Scoring. In: Oper. Res. Proc. 2014 Sel. Pap. Annu. Int. Conf. Ger. Oper. Res. Soc. (GOR). RWTH Aachen Univ. Ger. Sept. 2-5, 2014 (2016)
4. Garcia, V., Marques, M. I., Sanchez, J. S.: On the use of data filtering techniques for credit risk prediction with instance-based models. Expert Syst. Appl. 39(18), pp. 13267–13276 (2012)
5. Marqués, A. I., García, V., Sánchez, J. S.: On the suitability of resampling techniques for the class imbalance problem in credit scoring. J. Oper. Res. Soc. 64(7), pp. 1060–1070 (2013)
6. Banasik, J., Crook, J., Thomas, L.: Sample selection bias in credit scoring models. J. Oper. Res. Soc. 54(8), pp. 822–832 (2003)
7. Danenas, P., Garsva, G.: Selection of Support Vector Machines based classifiers for credit risk domain. Expert Syst. Appl. 42(6), pp. 3194–3204 (2015)
8. Zhao, Z., Xu, S., Kang, B. H., Kabir, M. M. J., Liu, Y., Wasinger, R.: Investigation and improvement of multi-layer perceptron neural networks for credit scoring. Expert Syst. Appl. 42(7), pp. 3508–3516 (2015)
9. Khashman, A.: Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. Expert Syst. Appl. 37(9), pp. 6233–6239, Sep. (2010)

10. Xiao, H., Xiao, Z., Wang, Y.: Ensemble classification based on supervised clustering for credit scoring. Appl. Soft Comput. J. 43, pp. 73–86 (2016)
11. Wang, H., Xu, Q., Zhou, L.: Large unbalanced credit scoring using lasso-logistic regression ensemble. PLoS One 10(2) (2015)
12. Cao, V. L., Le-Khac, N. A., O'Neill, M., Nicolau, M., McDermott, J.: Improving fitness functions in genetic programming for classification on unbalanced credit card data. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 9597, pp. 35–45 (2016)
13. Zhang, Z., Gao, G., Shi, Y.: Credit risk evaluation using multi-criteria optimization classifier with kernel, fuzzification and penalty factors. Eur. J. Oper. Res. 237(1), pp. 335–348 (2014)
14. Tomczak, J. M., Zięba, M.: Classification Restricted Boltzmann Machine for comprehensible credit scoring model. Expert Syst. Appl. 42(4), pp. 1789–1796 (2015)
15. Louzada, F., Ara, A., Fernandes, G. B.: Classification methods applied to credit scoring: Systematic review and overall comparison. Surv. Oper. Res. Manag. Sci. 21(2), pp. 117–134, Dec. (2016)
16. Lessmann, S., Baesens, B., Seow, H. V., Thomas, L. C.: Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. Eur. J. Oper. Res. 247(1), pp. 124–136 (2015)
17. Brown, I., Mues, C.: An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Syst. Appl. 39(3), pp. 3446–3453, Feb. (2012)
18. Wolpert, D. H.: The supervised learning no-free-lunch theorems. Soft Comput. Ind., pp. 25–42 (2002)
19. Villuendas-Rey, Y., Rey-Benguría, C. F., Ferreira-Santiago, Á., Camacho-Nieto, O., Yáñez-Márquez, C.: The Naïve Associative Classifier (NAC): A novel, simple, transparent, and accurate classification model evaluated on financial data. Neurocomputing. Jun. (2017)
20. López, V., Fernández, A., García, S., Palade, V., . Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Inf. Sci. (Ny) 250, pp. 113–141, Nov. (2013)
21. Dal Pozzolo, A., Caelen, O., Bontempi, G.: When is Undersampling Effective in Unbalanced Classification Tasks? In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9284, pp. 200–215 (2015)
22. Piramuthu, S.: On preprocessing data for financial credit risk evaluation. Expert Syst. Appl. 30(3), pp. 489–497, Apr. (2006)
23. Crone, S. F., Finlay, S.: Instance sampling in credit scoring: An empirical study of sample size and balancing. Int. J. Forecast. 28(1), pp. 224–238, Jan. (2012)
24. Batista, G. E. A. P. A., Prati, R. C., Monard, M. C.: A study of the behaviour of several methods for balancing machine learning training data. Sigkdd Explor. 6(1), pp. 20–29 (2004)
25. Batuwita, R., Palade, V.: Efficient resampling methods for training support vector machines with imbalanced datasets. In: Proceedings of the International Joint Conference on Neural Networks (2010)
26. Fernández, A., del Jesus, M. J., Herrera, F.: On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets. Inf. Sci. (Ny). 180(8), pp. 1268–1291 (2010)
27. Fernandez, A., Garcia, S., del Jesus, M. J., Herrera, F.: A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. Fuzzy Sets Syst. 159(18), pp. 2378–2398 (2008)

28. Chawla. N., Bowyer, K.: SMOTE: Synthetic Minority Over-sampling Technique Nitesh. J. Artif. Intell. Res. 16, pp. 321–357 (2002)
29. He, H., Bai, Y., Garcia, E. A., Li, S.: ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of the International Joint Conference on Neural Networks, pp. 1322–1328 (2008)
30. Tang, S., Chen, S.P.: The generation mechanism of synthetic minority class examples. In: 5th Int. Conf. Inf. Technol. Appl. Biomed. ITAB 2008 conjunction with 2nd Int. Symp. Summer Sch. Biomed. Heal. Eng. IS3BHE 2008, pp. 444–447 (2008)
31. Stefanowski, J., Wilk, S.: Selective pre-processing of imbalanced data for improving classification performance. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5182. LNCS, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 283–292 (2008)
32. Tomek, I.:Two Modifications of CNN. IEEE Trans. Syst. Man Cybern. 6, pp. 769–772, (1976)
33. Kubat, M., Matwin, S.: Addressing the Curse of Imbalanced Training Sets: One Sided Selection. Icml 97, pp. 179–186 (1997)
34. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. Proc. In: 8th Conf. AI Med. Eur. Artif. Intell. Med., pp. 63–66 (2001)
35. Alcalá-Fdez, F. H. J., Sánchez, L., García, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, C., Rivas, V.M., Fernández, J.C.: KEEL (software) (2009)

ISSN 1870-4069

# Predicting Academic Performance of Engineering Students after Approving a Mathematics Leveling Course using Decision Trees

Silvia B. González-Brambila, Lourdes Sánchez-Guerrero, Irma Ardón-Pulido, Josué Figueroa-González, Beatriz A. González-Beltrán

Universidad Autónoma Metropolitana, Unidad Azcapotzalco, Mexico City, Mexico
{sgb,lsg,ifap,jfgo,bgonzalez}@azc.uam.mx

**Abstract.** Academic performance of recently accepted students is one of the main issues in Higher Level Institutions since first scholar periods trend to be the most difficult ones for students. Some institutions offer leveling courses to develop students basic knowledge for later courses. However, it is not clear if these help students in more advanced courses. This work presents an analysis, using decision trees, for predicting marks in two mathematics courses based on different criteria of the performance on a previous leveling course. This allows finding the factors that impact in the marks obtained in posterior courses and determining if the leveling one is helping students to improve their academic performance.

**Keywords:** classification techniques, decision trees, educational data mining, predictive techniques, students performance.

## 1 Introduction

Data Mining (DM) is defined as the science of analyzing big volumes of data for finding interesting patterns which can lead to knowledge about certain aspects or phenomena [4]. DM uses concepts of machine learning and statistic and considers different kinds of analysis: clustering, associative and predictive (which can be divided into numerical and categorical). Since its appearance, DM has been applied in several areas such as medicine, commerce, finance, business, etc. From year 2000, the concepts and algorithms of DM have been applied in an educative environment, analyzing many aspects related with education, this branch of DM is known as Educational Data Mining (EDM) [8].

One of the most common problems studied in EDM is the academic performance of students, specially using classification or predictive techniques.

The academic performance of students is one of the main issues in Higher Level Institutions (HLI), especially in the recently admitted ones. Transition from High School to College has a great impact in students. Moreover, sometimes the knowledge acquired is not enough to face complex courses and students fail in their first courses. Considering this, HLI implements some leveling courses

whose goal is to give students foundational knowledge for taking more complex subjects.

In the Mexican Universidad Autónoma Metropolitana Azcapotzalco (UAM-A), since year 2008, in all the study plans of the ten engineering programs, a leveling course called Mathematics Workshop (MW) [9] has been implemented whose goal is to level students in basic mathematics concepts so they can have a better performance in posterior and more complex courses. At UAM-A, programs offered at the bachelor level are twelve trimesters (four years of full time studies); then the existence of MW course increases the time of a student for finishing their studies, at least in one trimester.

For approving MW, all admitted students present a diagnosis exam. If their marks are good enough (8 of 10), they do not have to take it and can access to the next mathematics courses Complements of Mathematics (CM) and Introduction to Calculus (IC) which is part of the General Branch Level at UAM-A [10], so all engineering students must take them. If their marks are not good enough, then, students must take MW, expecting that at the moment of approving it, they have the adequate level for taking the next courses.

However, it is not clear whether MW is helping students to have a better performance in CM and ICS. Several students need more than one trimester for approving it, and once they approve it, they do not have a good performance in CM and IC. For this reason, an analysis is necessary which allows us to classify or predict the students' marks in CM and IC, from their performance in MW, and finding the most important factors in students performance in CM and IC.

The structure of the paper is as follows: Section 2, gives a general description of the predictive technique decision trees. Section 3 presents works that have used predictive analysis for studying academic performance. Section 4 contains the steps followed for processing data and obtaining the classification models. Section 5 shows obtained results and models and their analysis. Finally, Section 6 contains conclusions and future works.

## 2 Predictive Techniques and Decision Trees

### 2.1 Predictive Techniques

Predictive, also called classification or supervised learning techniques, considers a learning scheme where a set of data is divided in two subgroups; one with a certain classification for teaching or training (called training data) which according to the technique and algorithm used, generates a model that can predict or classify data of the second group, a set of non classified data. Common techniques involved in predicting or classifying data are: decision trees, neural networks and Naïve Bayes classifier. Predictive techniques consider two kind of criteria, that used for classifying (independent variables) and the one to be classified or predicted (decision). Once a model is generated using a predictive technique, its efficiency is tested considering a set of data (test data) which already has a classification. The model must classify this data correctly, assigning

to it the correct value to the decision variable. The percentage of correct classified cases represents the level of efficiency of the model. The accuracy of a classifier model is the probability of correctly predicting the class of an instance, normally is estimated in different ways.

### 2.2 Decision Trees

Decision trees is one of the most used predictive techniques in DM [12]. As a tree structure is composed by branches and nodes, nodes can be: a root, node or leaf. A leaf represents the assigned classification. Other nodes represent a set of characteristics which lead, through a branch, to other nodes or a leaf. Root is called the best predictor, which is the most important criteria for giving a certain classification. Once the tree is constructed, the predicted or classified value is obtained traveling from the root to a leaf according to the characteristics (independent variables) of the case that is being analyzed.

When a decision tree is generated, the nearer the root is to a node, the more it is relevant for assigning a classification. This means that it has more importance in terms of information quality than other criteria. There are several decision tree algorithms such as ID3 [6], J48 [5] and CART [1].

## 3 Related Work

Predicting students performance is one of the most studied topics in Educational Data Mining.Several works use prediction techniques for predicting or classifying the performance or marks on exams, courses or scholar periods. In [7], applied predictive techniques over scholar data. They tested at Delhi Technological University data for classifying aspects like: students enrollment preferences, actual demand of certain courses, students that would like to be transferred, satisfaction level and future marks considering several factors. They applied decision trees and artificial neural network techniques. They presented results for enrollment decision and branch predictions showing that C5.0 algorithm of decision trees obtained the best accuracy.

The work of [2] predicted students' performance by using linear regression and matrix factorization approaches. They predicted students' next-term course grades and within-class assessment performance. In particular, they investigated four methods: the course-specific regression (CSpR), the personalized linear multi-regression (PLMR) methods, the standard matrix factorization (MF) and the MF method based on factorization machines (FM) to predict the grade that a student would achieve in a specific course. The results showed that PLMR and MF can predict next-term grades with lower error rates than traditional methods. PLMR were also useful for predicting grades on assessments within a traditional class or online course.

In [3], authors present a predictive analysis for the performance of students from Brazilian public schools. They considered several criteria for two stages, the first before students entered school, and academic criteria was added for

the second stage. Data used consisted in registers of students from year 2015 and 2016. The study determined the causes of student failure for both years. Techniques used involved Gradient Boosting Machine (GBM). Results showed that marks and assistance rates are relevant academic criteria, but social criteria, neighborhood, their previous schools and their age also have an important impact in students performance.

## 4 Decision Trees Generation

### 4.1 Obtaining Data

For making the analysis, the CRISP-DM methodology [11]was consideed and data was gathered from two sources, the General File of Students (AGA from its acronym in Spanish) which contains information about students, the entrance trimester being used, and the historical record of marks (called kardex) at UAM-A which contains the obtained marks for every student in every course they have taken. In particular, for analyzing how the performance in MW can predict the mark in CM and IC, the marks from students that entered since 2008 and that have already passed MW were considered. Initially, 5,181 students were considered. Marks at UAM-A are assigned with letters: MB (Very Good), B (Good), S (Sufficient) and NA (Not Approved) It is considered that the minimum mark for approving the diagnostic exam is B (Good).

### 4.2 Generating Sets of Data

After assessing the group of students, their information was processed for obtaining sets of data composed by the following criteria (in parenthesis, the name used for processing it):

- The way a student approved MW, through diagnosis exam or taking it, if MW was taken, the total of chances needed for approving it (WAPR)
- The mark obtained in MW (MMW)
- Time (in quarters) passed after approving MW and taking CM and IC (TAMW)
- The mark obtained the first time CM and IC was taken (approved or not) (MCM and MIC)

Two sets of data were generated, one for the relationship between MW and CM with 5,090 students, and another for MW and IC with 4,228. The difference is because not all students that approved MW have taken CM or IC. Classification variables were MCM and MIC for each set of data respectively. Possible values and their meaning for each variable are presented in Table 1.

Initially, the marks considered at UAM-A were used, but the efficiency of the generated models was very poor (less than 30%). The main problem was that several approving marks (MARK_MB, MARK_B and MARK_S) for CM and IC were classified as not approving marks (MARK_NA). For this reason, the values of the classification variables were changed to the approved mark (MARK_A), that considers all the approved and not approved marks. (MARK_NA).

**Table 1.** Possible values and meaning for each variable of the data set.

| Variable | Possible Values | Description |
|---|---|---|
| WAPR | EXAM | Approved through diagnosis exam |
| | FIRST | Approved through taking the course once |
| | SECOND | Approved taking the course twice |
| | MORE_TWO | Approved taking the course more than two times |
| MMW | MB | Approved with MB |
| | B | Approved with B |
| | S | Approved with S |
| TAMW | NONE | Took CM or IC the same trimester it approved MW |
| | NEXT | Took CM or IC the next trimester it approved MW |
| | ONE | Took CM or IC one trimester after approving MW |
| | TWO | Took CM or IC two trimesters after approving MW |
| | MORE_TWO | Took CM or IC two or more trimesters after approving MW |
| MCM or MIC | MARK_MB | Approved with MB |
| | MARK_B | Approved with B |
| | MARK_S | Approved with S |
| | MARK_NA | Not Approved |

### 4.3 Trees Generation

Two decision trees were created, one for predicting the mark obtained in CM and another for the one in IC. For both trees 70% of data for training and 30% for testing their accuracy were used. 10 repetitions were performed considering random sets of data for training and testing. Trees were generated using the algorithms CART, ID3 and J48. Average accuracy and standard deviation for each algorithm is presented in the Section Results and Analysis. Also, trees and rules of best accuracy algorithm are shown.

## 5 Results and Analysis

### 5.1 Predicting Performance in "Complements of Mathematics"

After 10 repetitions, the average accuracy using CART algorithm was 61.85% with a standard deviation of 1.03. J48 algorithm had an average accuracy of 61.92% with a standard deviation of 1.077. Finally, CART algorithm produced an average accuracy of 61.38% and a standard deviation of 0.96.

Best accuracy using J48 algorithm was 63.19%. Using ID3 algorithm, 62.9% and using CART algorithm, best accuracy was 61.48%. Confusion matrix with the total of correctly and incorrectly classified of the best accuracy model for each algorithm is shown in Table 2.

**Table 2.** Amount of correct and incorrect predicted marks in "Complements of Mathematics" per algorithm.

| | J48 | | ID3 | | CART | |
|---|---|---|---|---|---|---|
| | MARK_A | MARK_NA | MARK_A | MARK_NA | MARK_A | MARK_NA |
| MARK_A | 706 | 203 | 723 | 186 | 714 | 185 |
| MARK_NA | 359 | 259 | 381 | 237 | 402 | 226 |

As Table 2 shows, all the algorithms have good results classifying approved marks correctly, but they failed in the non approving marks. The most efficient algorithm was J48 whose rules are presented in Algorithm 1 and its tree in Figure 1.

**Algorithm 1** Rules of J48 generated tree for predicting performance in "Complements of Mathematics".

```
if MMW == B then
  MCM ←MARK_A
end if
if MMW == MB then
  MCM ←MARK_A
end if
if MMW == S then
  if TAMW == NEXT then
    MCM ←MARK_NA
  end if
  if TAMW == ONE then
    if WAPR == FIRST then
      MCM ←MARK_A
    end if
    if WAPR == SECOND then
      MCM ←MARK_A
    end if
    if WAPR == MORE_TWO then
      MCM ←MARK_NA
    end if
  end if
  if TAMW == MORE_TWO  then
    MCM ←MARK_NA
  end if
  if TAMW == TWO then
    MCM ←MARK_A
  end if
end if
```

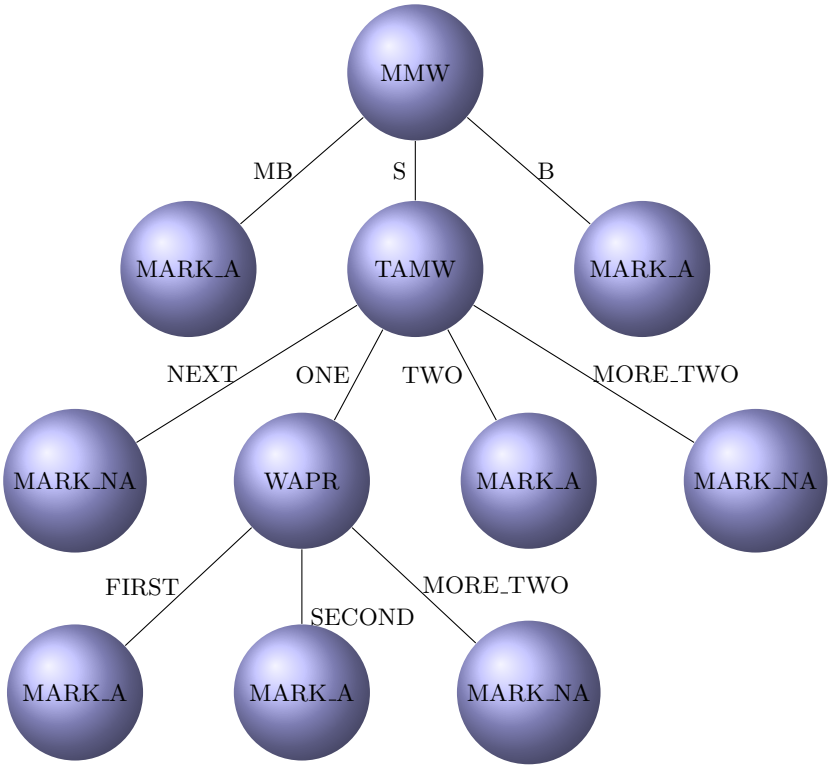



**Fig. 1.** Decision Tree for predicting the mark in "Complements of Mathematics".

### 5.2 Predicting Performance in "Introduction to Calculus"

Average accuracy using CART algorithm was 60.59% with a standard deviation of 1.18. J48 algorithm had an average accuracy of 61.2% with a standard deviation of 0.95. Finally, CART algorithm produced an average accuracy of 60% and a standard deviation of 1.04.

Best accuracy using J48 algorithm was 61%, ID3 60.75% and CART 60.67%. Confusion matrix with the amount of correctly and incorrectly classified instances for the best accuracy of each algorithm is presented in Table 3.

**Table 3.** Amount of correct and incorrect predicted marks in "Introduction to Calculus" per algorithm.

| | J48 | | ID3 | | CART | |
|---|---|---|---|---|---|---|
| | MARK_A | MARK_NA | MARK_A | MARK_NA | MARK_A | MARK_NA |
| MARK_A | 268 | 359 | 269 | 358 | 259 | 376 |
| MARK_NA | 136 | 506 | 140 | 502 | 123 | 511 |

Opposite to CM, the algorithms have a better performance classifying correctly a non approved mark. Similarly with CM, the most efficient algorithm was J48. Its rules for classifying marks in IC are presented in Algorithm 2 and its tree in Figure 2.

**Algorithm 2** Rules of J48 generated tree for predicting performance in "Introduction to Calculus".

```
if MMW == MB then
  MIC ←MARK_A
end if
if MMW == S then
  MIC ←MARK_NA
end if
if MMW == B then
  if WAPR == EXAM then
    MIC ←MARK_A
  end if
  if WAPR == FIRST then
    MIC ←MARK_NA
  end if
  if WAPR == SECOND then
    MIC ←MARK_NA
  end if
  if WAPR == MORE_TWO then
    MIC ←MARK_NA
  end if
end if
```

From the results, it can be seen that the algorithm that obtained the best results was J48 for predicting students performance in both topics. However, the obtained accuracy (63.19% and 61%) is not big enough for determining if MW is really helping or not students in CM and IC. Analyzing the trees, there exists some problems, specially with the CM tree, in particular three branches:

- **S** in **MMW**, **ONE** in **TAMW**, **FIRST** in **WAPR** which leads to an Approved Mark (**MARK_A**)
- **S** in **MMW**, **ONE** in **TAMW**, **SECOND** in **WAPR** which leads to an Approved Mark (**MARK_A**)
- **S** in **MMW**, **TWO** in **TAMW**, which leads to an Approved Mark (**MARK_A**)

It is expected that approving MW with the lowest approving mark (S) and waiting for one or two scholar periods after taking the next courses, will lead to a non approving mark. However, the tree shows the opposite.

Decision trees for predicting performance in IC do not have these kinds of contradictions. Here, approving MW with the lowest mark is related with non approving IC (branch S in MMW). Meanwhile, obtaining the best mark (BM) leads to approve IC (branch BM in MMW).

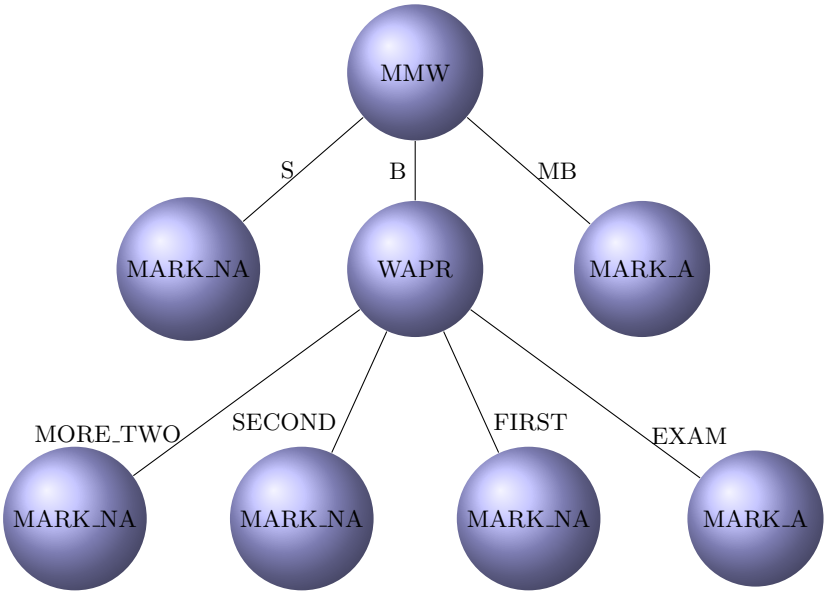



**Fig. 2.** Decision Tree for predicting the mark in "Introduction to Calculus".

Besides the confusion matrices, the Sensitivity and Specificity for each algorithm was obtained. Sensitivity and Specificity are two measures commonly used at the moment of predicting and classifying binary cases (with only two possible classified values). Sensitivity or true positive rate measures the proportion of positives cases that are classified as positive. Specificity or true negative rate measures the proportion of negative cases that are classified as negative. Results for each algorithm are presented in Table 4.

**Table 4.** Sensitivity and Specificity for each algorithm.

| | "Complements of Mathematics" | | "Introduction to Calculus" | |
|---|---|---|---|---|
| | Sensitivity | Specificity | Sensitivity | Specificity |
| J48 | 0.662 | 0.560 | 0.663 | 0.584 |
| ID3 | 0.654 | 0.560 | 0.657 | 0.583 |
| CART | 0.639 | 0.549 | 0.678 | 0.576 |

## 6 Conclusions and Future Work

The aim of this paper was to predict the marks in two mathematics courses according to the performance in a leveling course for determining if it helps students in later courses. However, the obtained models did not have enough

accuracy for being considered reliable. From the analysis, some problems were identified. Having four possible values for assigning in the prediction with decision trees produced a very low accuracy, so, it was necessary to group the approved marks for raising the accuracy. This avoided making a more specific analysis of the obtained marks in "Complements of Mathematics" and "Introduction to Calculus" and was only reduced to determine if the student approved or not. Also, all the algorithms had problems classifying correctly either approved marks ("Introduction to Calculus") and non approved marks ("Complements of Mathematics"). The Decision tree of "Complements of Mathematics" has some branches which have an unexpected behavior leading to approving marks where it is supposed to expect non approving ones.

Low prediction accuracy could be due to it not being very clear the difference between cases with MARK_A and the ones with MARK_NA, specially in the values of WAPR, MMW and TAMW criteria.

Despite the low accuracy, results from the impact of "Mathematics Workshop" over "Complements of Mathematics" showed that students that took and approved "Mathematics Workshop" with MB or B mark, also approved "Complements of Mathematics" and "Introduction to Calculus". However, the tree did not give information about the way "Mathematics Workshop" was approved. Students that approved with S and took "Complements of Mathematics" or "Introduction to Calculus" the following period or waited more than two periods, did not approve in their first attempt.

Future approaches for this work include testing other predictive techniques and adding new criteria to the analyzed ones, such as characteristics of the students before entering university. Also, applying some algorithms for leveling the amount of data could improve the accuracy of prediction.

## References

1. Breiman, L.: Classification and regression trees. Routledge (2017)
2. Elbadrawy, A., Polyzou, A., Ren, Z., Sweeney, M., Karypis, G., Rangwala, H.: Predicting student performance using personalized analytics. Computer 49(4), 61–69 (Apr 2016)
3. Fernandes, E., Holanda, M., Victorino, M., Borges, V., Carvalho, R., Van Erven, G.: Educational data mining: Predictive analysis of academic performance of public school students in the capital of brazil. Journal of Business Research (2018)
4. Hand, D.J.: Principles of data mining. Drug safety 30(7), 621–622 (2007)
5. Patil, T.R., Sherekar, S.: Performance analysis of naive bayes and j48 classification algorithm for data classification. International Journal of Computer Science and Applications 6(2), 256–261 (2013)
6. Peng, W., Chen, J., Zhou, H.: An implementation of id3 — decision tree learning algorithm (01 2009)
7. Rajni, J., Malaya, D.B.: Predictive analytics in a higher education context. IT Professional 17(4), 24–33 (2015)
8. Romero, C., Ventura, S.: Educational data mining: a review of the state of the art. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 40(6), 601–618 (2010)

9. Universidad Autónoma Metropolitana Azcapotzalco, División de Ciencias Básicas e Ingeniería: http://cbi.azc.uam.mx/es/CBI/Tronco_de_Nivelacion_Academica
10. Universidad Autónoma Metropolitana Azcapotzalco, División de Ciencias Básicas e Ingeniería: http://cbi.azc.uam.mx/es/CBI/Planes_Programa_Estudio_Com
11. Wirth, R., Hipp, J.: Crisp-dm: Towards a standard process model for data mining. In: Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining. pp. 29–39. Citeseer (2000)
12. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Philip, S.Y., et al.: Top 10 algorithms in data mining. Knowledge and information systems 14(1), 1–37 (2008)

ISSN 1870-4069

# A Parallel Implementation on CUDA for Solving 2D Poisson's Equation

Jorge Clouthier-Lopez[1], Ricardo Barrón Fernández[2], David Alberto Salas de León[3]

[1,2] Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico
[3] Universidad Nacional Autónoma de México, Instituto de Ciencias del Mar y Limnología, Mexico

clouthier@gmail.com[1], rbarron@cic.ipn.mx[2], david.alberto.salas.de.leon@gmail.com[3]

**Abstract**. A parallel iterative Finite Difference (FD) method for solving Poisson's equation on CUDA is implemented. The aim of this paper is to give a detail explanation about the parallel solution of a Partial Differential Equation (PDE). To examine the performance of the implemented iterative algorithm, a number of experiments were tested. The performance shows the benefit of using the implemented approach on GPU devices in terms of execution time.

**Keywords:** numerical schemes, partial differential equations, GPU, CUDA.

## 1 Introduction

In parallel computing many calculations are carried out simultaneously in one or in different hardware environments. In this paper, we focus on the implementation of a parallel numerical scheme over GPU NVIDIA devices, as well as on the detailed explanation on the process of discretization and parallelization of a PDE. Parallel computing is applied to solve large problems over High Performance Computing (HPC). These large problems are mainly present in the research and forecasting of weather, climate (*e.g.* simulations of extreme events where atmosphere-ocean interactions have a great relevance, [1] and [2]), planetary sciences, and astronomy; as well as in engineering (*e.g.* computer, civil, and mechanical), where the reduction in computation time and the improvement in speedup is of fundamental importance.

More and more research has focused on solving numerical models applying different numerical schemes over HPC involving GPUs. With the latest advances on GPU cards, new numerical models on self or hybrid hardware have arisen, *e.g.*, MPI and CUDA combined, *e.g.* [3]. This of fundamental importance since large scale problems, represented by sets of PDEs and their parametrizations, *e.g.* source and sink terms that cannot be resolved directly [4], require high resolutions to reproduce many characteristics, that today are not possible. These large scale problems need substantial computation time and memory to run on ordinary computers. This means that powerful capabilities are required to obtain the numerical simulations in a reasonable period time.

FD schemes have been employed in many numerical models that are applied in different branches of geophysics. The content of this paper is based on the fundamentals

that are being acquired to develop and implement a geophysical model with applications on oceanography and climate for the Ph.D. research of the first author. This model will be solved applying numerical hybrid methods based on its variational formulation.

The Poisson's equation is present in many geophysical models. In the solution of this equation, a numerical method is defined for a rectangular domain. As a result, a linear system of equations is found. Then, the system of equations is implemented on a CUDA parallel algorithm as a numerical linear algebra problem. Finally, the CUDA program is executed and the results are analyzed.

Numerical schemes based on FD are classical and straightforward ways to solve numerically both PDEs and systems of PDEs. It is well known that FD schemes are applied to solve systems of PDEs with many nodes on their structured domains [5]. The resulting algebraic systems are solved using iterative methods due to the fact that direct methods have disadvantages to calculate inverse matrices [6]. To reach higher accuracy and faster convergence rate, modifications have been made over iterative implicit methods, such as Jacobi, Gauss-Seidel and SOR, to have parallel algorithms, *e.g.* [7]. Moreover, pre-conditioning has also been applied for ill conditioned matrices that have large condition numbers.

In this paper we use the worth and simplicity that FD schemes have to solve the 2D Poisson's equation in parallel as a numerical modeling problem using CUDA C. A two-colored domain decomposition is applied in the FD discretization. The discrete equation is solved applying the Gauss-Seidel iterative method.

This paper is organized as follows. In Section 2, both the modeling problem and the 2D domain decomposition method are introduced. In Section 3, the parallel iterative finite difference algorithm and its implementation are presented. In Section 4, a brief overview about CUDA and the hardware architecture is provided. In Section 5, the numerical experiments and discussions are given.

## 2 Modeling Problem and Discretization

The 2D Poisson's equation is solved in the rectangular region Ω=[0≤x≤m]X[0≤y≤m]. This PDE is written as:

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = f(x, y), \tag{2.1}$$

with boundary conditions:

$$\varphi(x, y) = g(x, y), \quad (x, y) \in \partial\Omega. \tag{2.2}$$

The domain was divided uniformly into a number of shares with mesh grid size h, where

$$x_i = ih, \text{ and } y_j = jh, \quad i, j = 1,2,3, \dots, n \tag{2.3}$$

with

Assuming a homogenous mesh, see Figure 1, each second derivative in (2.1) is approximated with centered FD according to the following expressions:

$$\frac{\partial^2 \varphi}{\partial x^2} \approx \frac{\varphi(x+h,y)-2\varphi(x,y)+\varphi(x-h,y)}{h^2}, \tag{2.4}$$

$$\frac{\partial^2 \varphi}{\partial y^2} \approx \frac{\varphi(x,y+h)-2\varphi(x,y)+\varphi(x,y+h)}{h^2}. \tag{2.5}$$

Substituting (2.4) and (2.5) into (2.1) we have

$$\varphi(x+h,y)+\varphi(x-h,y)+\varphi(x,y+h)+ \\ +\varphi(x,y-h)-4\varphi(x,y)=h^2 f(x,y). \tag{2.6}$$

Writing (2.6) with subscripts

$$\varphi_{i+1,j}+\varphi_{i-1,j}+\varphi_{i,j+1}+ \\ +\varphi_{i,j-1}-4\varphi_{i,j}=h^2 f_{i,j}\ . \tag{2.7}$$

Expression (2.7) is the discretized Poisson's equation. This expression will result in an algebraic system with $n^2$ equations of the form

$$AX=b. \tag{2.8}$$

where $b$ is a known $n^2$-vector (its elements are boundary conditions in (2.2)), $X$ is a $n^2$-vector to be determined (its elements are the function $\varphi(x,y)$ evaluated on the $n^2$ nodes of the mesh), and $A$ is a $n^2 \times n^2$ matrix. $A$ has the following form

$$A=\begin{bmatrix} C & E \\ F & D \end{bmatrix}, \tag{2.9}$$

with

$$C=D \text{ and } E=F^T,$$

where

$$C_{k,l}=-4 \text{ if } k=l,$$

and

$$C_{k,l}=0 \text{ if } k\neq l.$$

In order to arrive at the system (2.8), the first $n^2/2$ rows are obtained applying (2.7) on all nodes where $i+j$ is even and the following $n^2/2$ rows are obtained applying (2.7) on all nodes where $i+j$ is odd.

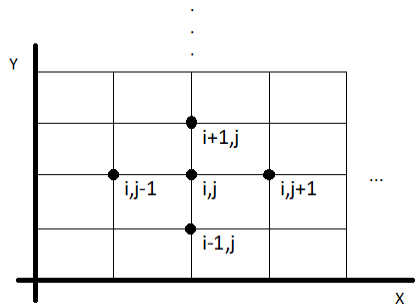



**Fig. 1.** The homogeneous mesh and the dependency of node $i,j$ on the nearest ones.

# 3 The Iterative Algorithm

## 3.1 Sequential Gauss-Seidel

Before going directly to the parallel Gauss-Seidel (GS) method, we must first discuss the sequential one. The GS is an improvement of the Jacobi algorithm. It can be written in matrix and iterative notation. GS in the first form is expressed as

$$X^{k+1} = -(D + L)^{-1}UX^k + (D + L)^{-1}b, \tag{3.1}$$

where $L$, $D$, and $U$ are the lower, diagonal, and upper-triangular parts of $A$, respectively.

In the second form GS is written as

$$X_i{}^l = \frac{1}{a_{ii}}(b_i - \sum_{j<i} a_{i,j}X_j{}^l - \sum_{j>i} a_{i,j}X_j{}^k), \tag{3.2}$$

where $l = k + 1$.

In (3.2) GS corrects the $i$th component of the vector $X_i{}^k$, where $i$= 1, 2, $\cdots$ $n^2$. The approximation solution is updated immediately after the new component is determined.

## 3.2 Parallel Gauss-Seidel

In (2.8) $A$ is a diagonal-dominant matrix, meaning that convergence will occur for the algorithm in (3.2). In applying (3.2) directly to (2.8), the $X_i{}^{k+1}$ for the first $n^2/2$ rows of the system of equations are calculated at the same time. Then the $X_i{}^{k+1}$ for the following $n^2/2$ rows are calculated also at the same time. The reason is that the $X_i{}^{k+1}$ for the last $n^2/2$ rows depend on the previous $X_i{}^{k+1}$, that were calculated in the first $n^2/2$ rows of the system of equations. This procedure is repeated many times until a desirable tolerance or approximation is reached. If the mesh ordering was not applied, the parallel procedure would not be possible.

Kernel Pseudocode

1: Define indices for the corresponding block and thread
2: Assign the index value to the corresponding thread
3: Select the number and order of the rows of *A* to work with
4: Determine *index_i* and *index_f* according to the balance of rows in the grid
5: **for** (from *index_i* to *index_f*)
6: Assign indices *i* and *j* to each node in the corresponding assigned rows of A, where *i+j* is even, in order to not calculate zeroed elements
7: Calculate $\varphi_{i,j}$ for the assigned rows, where *i=j*
8: Calculate the local error
9: **end**
10: synchronize
11: **for** (from *index_i* to *index_f*)

12: Assign indices *i* and *j* to each node in the corresponding assigned rows of A, where *i+j* is odd, in order to not calculate zero elements
13: Calculate $\varphi_{i,j}$ for the assigned rows, where *i=j*
14: Calculate the local error
15: **end**

The CUDA kernel, which is executed in the device, is called from the host (CPU) many times until the desirable tolerance is reached.

## 4 CUDA Programming and Hardware Architecture

CUDA is a general purpose parallel computing architecture that exploits the parallel compute structure in NVIDIA GPUs to solve many complex computational problems. CUDA C extends C by defining C functions that are executed in the device (in the GPUs). Each function, also called a kernel function, is mapped to all the threads on the device. All the threads in a GPU make up a grid. The GRID is divided into blocks. Threads within a block can communicate with each other and synchronize together, while threads from different blocks cannot share information between them. This means that a kernel function is executed from all the threads in a grid. In other words, kernel functions are copied to all threads to be executed simultaneously [8].

GPUs were originally designed for graphics rendering. Due to their unique hardware architecture, they have become a powerful and suitable tool for general purpose computing. Each thread reads data in different memory locations when executing a kernel function and has its own registers and local memory. Each block has the same shared memory of its own and all threads in a grid can access the data in global memory. Additionally, there are five kinds of memory: register, shared, local, global, and constant [8].

## 5 Results

In this section, we evaluate experimentally the performance of the implemented algorithm on CUDA C. In the experiments we used 10 different node densities or problem sizes (36, 64, 100, 400, 1 600, 3 600, 6 400, 10 000, 40 000, and 160 00) for the mesh of the domain.

First, considering m=1, we solve (2.1) for four different cases where the numerical solutions in the Figures 2 to 5 are obtained considering a problem size of 3 600. This means that a higher value for the problem size would increase the resolution, while a lower value would present a poor resolution in the depiction of the solution of the PDE, as well a low quality of the numerical solution. Then we present the performance of the implemented algorithm.

The first three cases are solved taking into account homogeneous boundary conditions $(x = 0\ y = 0,\ \ (x,y) \in \partial\Omega)$ and the last case is solved with nonhomogeneous boundary conditions. For each case a different $f(x,y)$ in (2.1) is used.

i) First case

$$f(x,y) = \cos(2xy). \tag{5.1}$$

In Figure 2 the graph of the solution of (2.1) in the selected domain is presented.

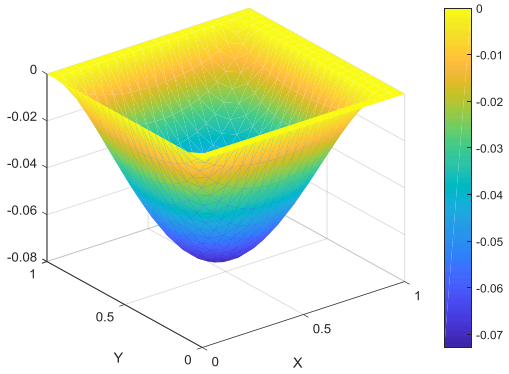



**Fig. 2.** Solution of Poisson's equation with $f(x,y) = \cos(2xy)$ and homogeneous boundary conditions.

ii) Second case

$$f(x,y) = \cos(2\pi x). \tag{5.2}$$

The solution for (2.1) for this case is presented in Figure 3.

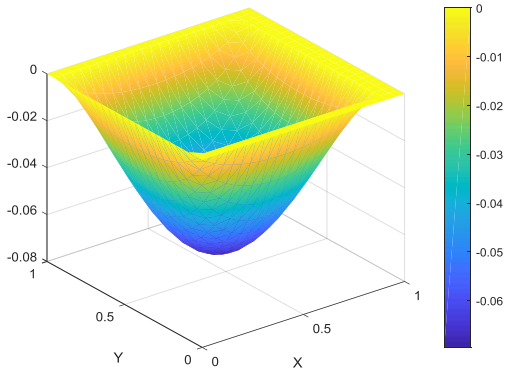



**Fig. 3.** Solution of Poisson's equation with $f(x,y) = \cos(2\pi x)$ and homogeneous boundary conditions.

iii) Third case

$$f(x,y) = 0. \tag{5.3}$$

The solution of (2.1) is presented in the following figure:

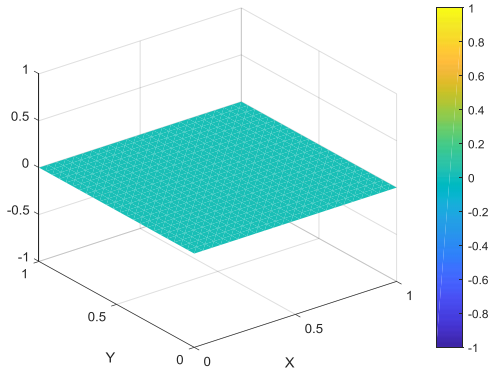



**Fig. 4.** Solution of Poisson's equation with $f(x, y) = 0$ and homogeneous boundary conditions.

iv) Fourth case

$$f(x, y) = 2x^2 + 2y^2 + 6x, \tag{5.4}$$

with the following boundary conditions:

$$x^3 + x^2y^2 \ (x, y) \in \partial\Omega.$$

The graph of the solution, in the selected domain, for this case is presented in the following figure:

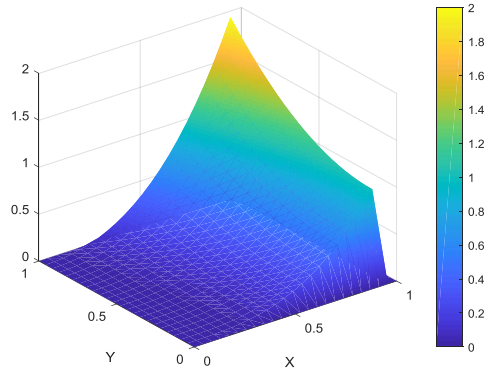



**Fig. 5.** Solution of Poisson's equation with $f(x, y) = 2x^2 + 2y^2 + 6x$ and nonhomogeneous boundary conditions equal to $x^3 + x^2y^2$.

In Figure 6 the execution time, in seconds, of the parallel algorithm for different problem sizes or node densities is presented. The sizes are obtained using 4, 9, 64, 100, and 200 threads. The results show that the parallel algorithm performs better as the domain resolution increases. In order to calculate the execution time, the fourth case is used. The reason is that it considers non-homogeneous boundary conditions.

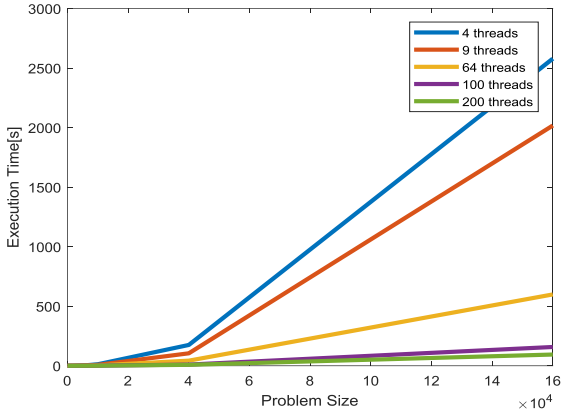



**Fig. 6.** The relationship between the execution time and the problem size.

Besides the execution time, the speedup also is calculated, see Figure 7. It is calculated selecting four different problem sizes: 3600, 10 000, 40 000 and 160 000 nodes. It can be seen that a better performance is obtained when large problems are considered.

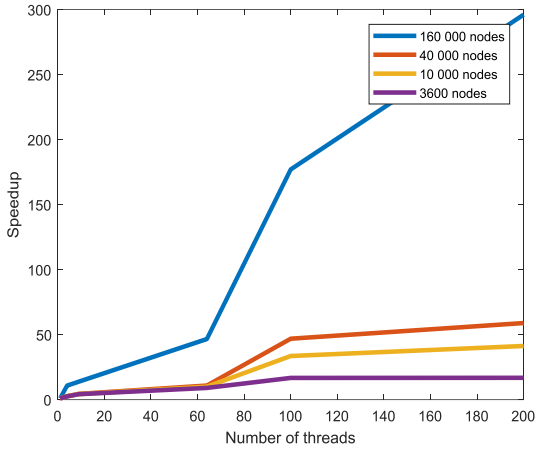



**Fig. 7.** The speedup for 3 600, 10 000, 40 000 and 160 000 nodes.

## 6 Conclusions

In this paper, we have implemented and analyzed a parallel implementation of the Gauss-Seidel algorithm with two-colored grid ordering to solve the 2D Poisson's equation on a GPU device using CUDA. The implemented parallel algorithm presents a good performance when the number of nodes of the mesh (discrete domain of the PDE) increases. This means that the behavior of the resulting parallel scheme is acceptable and good when the spatial resolution of the discrete domain is high.

The parallel algorithm takes advantage of the iterations to solve the linear system that results as a consequence of the discretization of the PDE.

**Acknowledgements.** The first and second author would like to thank to Instituto Politécnico Nacional for the support provided for the development of this work through the project SIP: 20181698

## References

1. Skamarock, W., Klemp, J., Dudhia, J., Gill, D., Barker, D., Duda, M., Huang, X. Y., Wang, W., Powers, J.: A description of the Advanced Research WRF Version 3, NCAR Technical Note TN-475+STR, http.//www.mmm.ucar.edu/wrf/ (2008)
2. Regional Oceanic Modeling Syste: [Online]. Available: https://www.myroms.org/ (2018)
3. Salgueiro, D. V., Silvestre, N., Conde, D. A. C., Ferreira, R. M. L.: Implementation and experimental benchmark of a two-layer CPU+ GPU hydrodynamics model. Geophysical Research Abstracts, 20, EGU2018-18718 (2018)
4. Solano-Quinde, L., Gualan-Saavedra, R., Zuñiga-Prieto, M.: Multi-GPU implementation of the Horizontal Diffusion method of the Weather Research and Forecast Model. Proceedings of the 7th International Workshop on Programming Models and Applications for Multicores and Manycores, pp 98-103, Barcelona, Spain (2016)
5. Vázquez-Báez, V., M., Rubio-Arellano, A. B., García-Toral, D., Rodríguez-Mora, I: Model and Solution of Darcy's Law: Homogeneous and Inhomogeneous Media. arXiv:1802.00890 [physics.geo-ph].
6. Al-Towaiq, M H.: Parallel Implementation of the Gauss-Seidel Algorithm on k-Ary n-Cube Machine. Applied Mathematics, 4, pp 177–182 (2013)
7. Olszewski, L.: A Timing Comparison of the Conjugate Gradient and Gauss-Siedel Parallel Algorithms in a One- Dimensional Flow Equation Using PVM. In: Proceedings of the 33rd Annual on Southeast Regional Conference, Clemson, pp. 205–212 (1995)
8. NVIDIA, CUDA C Programming Guide. [Online]. Available: http://docs.nvidia.com/cuda/cuda-c-programming-guide/ (2018)

ISSN 1870-4069

# Transmission and Reception of Images via Visible Light

Sergio Sandoval-Reyes

Instituto Politécnico Nacional, CIC, Mexico City, Mexico
sersand@cic.ipn.mx

**Abstract.** Communication by light or VLC by its acronym in English (Visible Light Communication), uses visible light from light emitting diodes (LEDs) to transmit information. Using a computing device and some hardware, the transmission of information is performed driving and modulating the light emitted by the LEDs. In the receiver side, the information carried by the modulated light is demodulated through a photo-detector, which is usually connected to a similar computing device for the final recovering of the information. In this article we describe an application based on VLC using OOK (On-Off Keying) modulation, to transmit color images from a RaspBerry Pi computer (using Python as the programming language), and several modules (LEDs and a sensor light) from LittleBits.

**Keywords:** VLC, image transmission, Raspberry Pi, Python, OOK.

## 1 Introduction

Visible Light Communication (VLC) [1, 2], can be used to transmit audio, voice and data. It uses laser light or light from emitter diodes (LEDs) and light detectors at the transmitter and receiver ends respectively (Fig. 1). It works in the 380 nm to 780 nm optical band which is visible light and hence the name VLC [3, 4, 5].

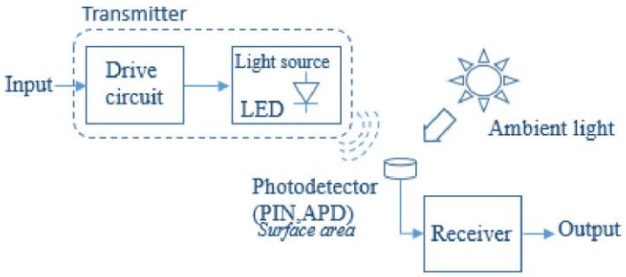



**Fig. 1.** A basic VLC link structure.

To convey information, this one has to be encoded, and then the light has to be modulated and demodulated at the transmitter and receiver sides. There are several methods to do this, some are briefly discussed in the following. Then, the received information has to be decoded and processed to recover it fully. The success of this

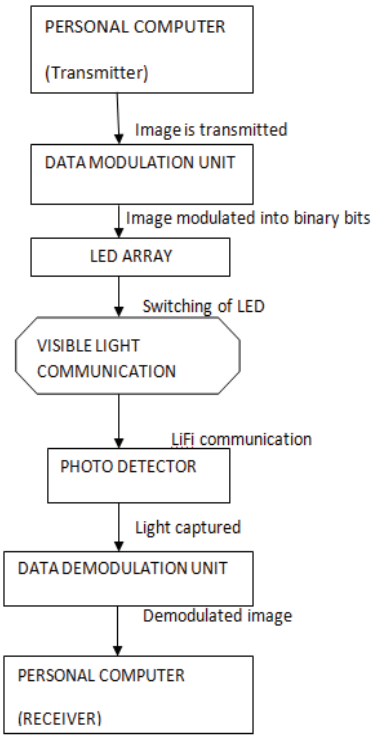



**Fig. 2.** A flow chart for VLC image transmission.

recovery depends of several factors, among them: 1) The number, shape, and wavelength of the LEDs employed; 2) The number and type of light detectors (photo-resistor, photo-diode, reverse-biased LED, etc.) used; 3) The encoding method (RZ, NRZ, NRZP, etc.); 4) The modulation scheme (OOK, WPM, VWPM, PPM, OFDM, etc.), and 5) The synchronization and distance between the LEDs and the light detector. Remember that VLC is a technology that among other things, it requires line-of-sight between emitter and receiver [6].

This paper describes an application based on VLC using OOK (On-Off Keying) modulation, to transmit as-a-proof of concept color images using a RaspBerry Pi 3 computer as the data source and sink (to simplify the synchronization problem between emitter-receiver), Python as the programming language, and several modules (LEDs and a sensor light) from LittleBits, to easy the hardware implementation.

The remainder of this paper is organized as follows: Section 2 gives a summary of works related to the transmission of digital images using VLC. Section 3 describes the design of our VLC application. Section 4 describes the experiments realized, the results obtained, and their analysis. Finally, our conclusions and future work are presented in Section 5.

## 2 Related Work

Several research works on VLC technologies to transmit images have been proposed. The most important are described in the following.

### 2.1 How Based VLC Image Systems Work

Typical based VLC image systems are implemented using an intensity modulation and direct detection (IM/DD) scheme with a line-of-sight (LOS) configuration [7]. In the transmitter, IM is implemented through the modulation of the transmitted signal into the instantaneous optical power of the LED by controlling the radiant intensity with the forward current through the LED (High modulation frequencies are used to avoid flicker). In the receiver, the transmitted signal is recovered using direct detection (DD). In this simple method, a photodiode (or array) is used to convert the incident optical signal power into a proportional current. Figure 2 shows a general flow chart for VLC image transmission [8].

### 2.2 VLC Image Transmitter

A typical based VLC image transmitter contains an image generator (a PC with Matlab to convert the image into bits), an interface (usually a USB cable) to send the bits to a microcontroller (for coding and modulation), and outputted through one of its ports to the LED driver and the LED optics. See Figure 3. [9]. The modulated signals are used to switch on-an-off the LEDs at desired frequencies using LED drivers. These drivers rely on trans-conductance amplifiers to convert voltage signals into corresponding current signals to excite the LEDs array for communication purposes.

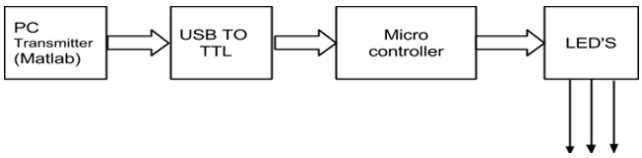



**Fig. 3.** VLC image transmission.

#### 2.2.1 VLC Modulation

Although there are different modulation schemes for VLC, mainly, on-off keying (OOK), variable pulse-position modulation (VPPM), color shift keying (CSK) and orthogonal frequency division multiplexing (OFDM) [10], OOK is the most popular. OOK is the most commonly used IM/DD modulation scheme in VLC due to its simple implementation. In this method basically the LED intensity is changed between two distinguishable levels corresponding to the data bits (1 or 0). See Figure 4. A modified OOK, called Variable OOK (VOOK) can provide dimming. It is achieved by changing the data duty cycle through pulse-width modulation (PWM), with only 1 bit of information carried per symbol period.

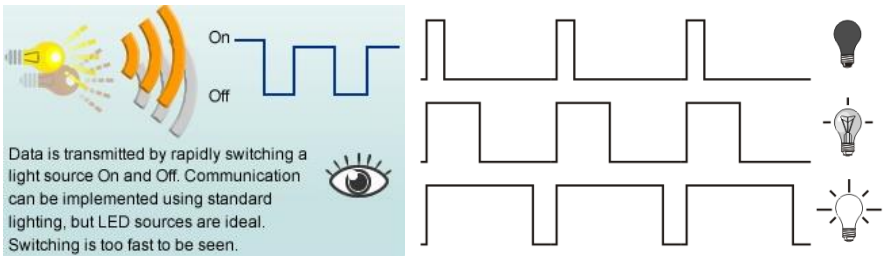



**Fig. 4.** On-Off Keying with PWM.

### 2.3 VLC Image Receiver

A typical optical image receiver consists of a photo detector followed by an amplifier. The photo detector can be a photo transistor, a reverse-biased LED, or a Light Detect Resistor (LDR). The light captured by the photo transistor which acts as a sensor, passes the output to the comparator which compares the binary input, and similarly the original image is recovered using Matlab software in the PC. See Figure 5 [8, 9].

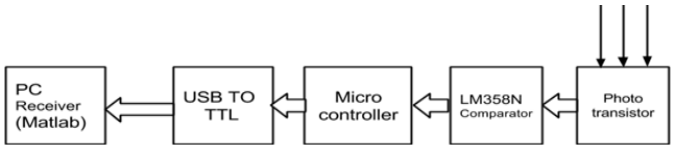



**Fig. 5.** VLC image reception.

## 3 Image Transmission and Recovering Using VLC

In the following we develop a VLC application to transmit and receive in real time an image using two LEDs in the transmitter, and a photo detector in the receiver.

In order to do that, we will use in the transmitter as a data source, a Raspberry Pi 3 (RBPi) computer, and five LittleBits bit-modules [12]: power, button, proto, split, and two bright LED bits. See Figure 6. The RBPi using a software written in Python will read an image byte by byte, and will send each byte using the SPI MOSI (Master Output Slave Input) output port (Pin 19), toward the proto module, OOK signals to drive the two LEDs.

While as in the receiver we will use three LittleBits components: power, light sensor, and another proto bit. See Figure 7. The light sensor captures the light emitted by the two LEDs and converts it into a digital signal which is fed to the proto module. The proto module in turn outputs this signal and with a wire connector, feeds this signal toward the MISO (Master Input Slave Output) input port (Pin 21) of the RBPi.

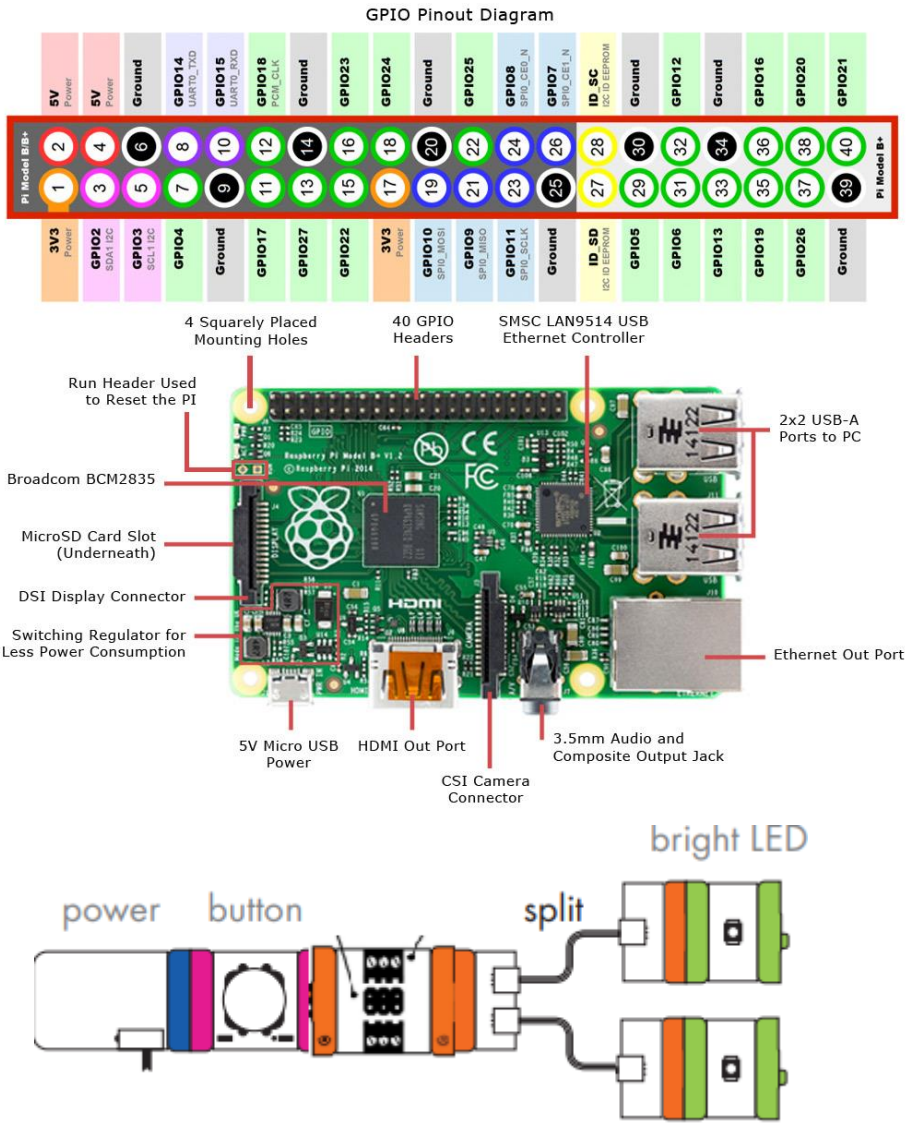



**Fig. 6**. Raspberry Pi 3, and five bits: power, button, proto, split and two LEDS.


**Fig. 7.** Receiver components: Light sensor and proto bits.

### 3.1 Transmitting an Image Via VLC

To transmit an image via VLC, the RBPi with a script written in Python, opens a picture file ("lenna.jpeg") and it reads it into a bytearray "b". Then with a "for" loop reads byte by byte of the picture and it send them to the SPI MOSI output port 19, using the spi.xfer () directive. This MOSI output is fed using two wire connectors (signal and ground), into the input of the "proto" module (lower middle connector in figure 7). The proto module outputs and split the OOK signals to drive the two LEDs. The Python code to execute the above mentioned is shown in Figure 8.

```
# Python code to read, transmit, and recover a picture via Visible Light
import spidev
from PIL import Image, ImageFilter
import io
from array import array
spi = spidev.SpiDev()
spi.open(0,0)
spi.max_speed_hz = 4000000
b = bytearray()
buffer = bytearray()
try:
    # Load an image from Raspberry Pi
    with open ("lenna.jpeg", "rb") as img:
        f = img.read()
        b = bytearray(f)
    # Sending with SPI and converting the image, byte by byte into visible light
    for byte in b:
        rx = spi.xfer([byte])
        rx_data = rx[0]
        buffer.append(rx_data)
    # Recovering the bytes array and back into an image
    img_rec = Image.open(io.BytesIO(buffer))
    img_rec.save(´lena_img_recovered.png´)
    img_rec.show()
except:
    print "Unable to recover image"
```

**Fig. 8.** Python code to read, transmit and recover a picture via VLC.

In this code, it is necessary to import the following libraries: "SPI", "PIL" (Python Image Library), and "Array". The use of the LittleBits modules simplified very much the hardware implementation. The power module was fed with a 9-V battery, and this was necessary because the outputs of the RBPi are low-voltage (3.3 volts) and low-current (Individual pins must not pull more than 16 mA and the entire GPIO must not source more than 50 mA), which are no good enough to drive two bright LEDS [13]. These LEDs are simple yellow LEDS with a wavelength of 550-to-600 nm, luminous flux of 4-to-5 lm, and consume around 16-to-20 mA each, with an aperture angle of about 120 degrees. See Figure 9.

**Fig. 9**. LittleBits bright LED.

### 3.2 Receiving an Image via VLC

As was mentioned, the picture was sent via VLC as LED light. This light is received through a light sensor module which then sent it back through out another proto module, to the MISO port 21 of the RBPi. This light sensor not only receives the OOK light signal but also has a trans-impedance amplifier for high speed operation. The light sensor has 2 modes, Figure 7. In LIGHT mode, as the light shining on the sensor gets brighter, more signal passes through it. In DARK mode, the signal increases as it gets darker. Furthermore, the light sensor has a sensitivity dial or slide dimmer to adjust how much light it takes to change the signal, and has a spectral sensitivity range from 500-to-600 nm similar to the LEDs wavelength. See Figure 10.

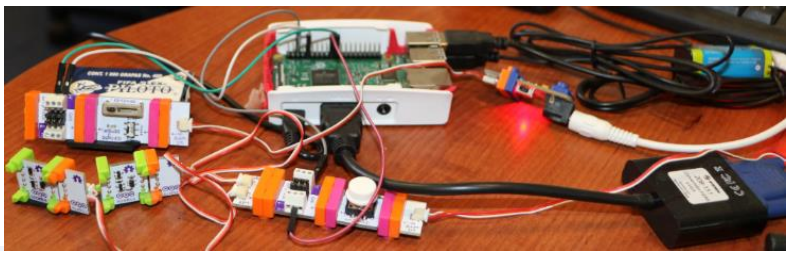


**Fig. 10**. Receiving a picture via VLC.

The recovered picture from the SPI MISO input is stored into a buffer, saved and displayed, as can be seen from the last 9 Python code lines of Figure 8.

## 4 Experiments and Results

For the experiments we use as was mentioned a RaspBerry Pi 3 computer with several LittleBits components. The whole setup is shown in Figure 11.

Figure 11 shows the RaspBerry Pi 3 computer connected through out bus connections to a HP display, keyboard and mouse. The figure also shows the recovered picture after the execution of the line command "***sudo python lena_spi.py***". Figure 12.

Figure 12 also shows that the recovered picture was not perfect. That was due to the presence of noise, mainly: Fluorescent light from ceiling lamps, misalignment and distance between LEDs and light sensor, and low sensitivity to light from the sensor.

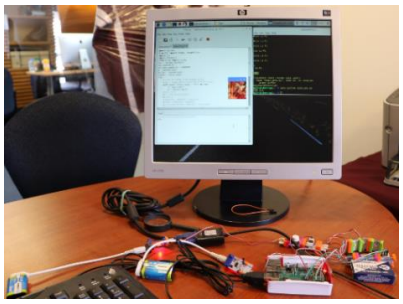


**Fig. 11.** Setup for the transmission and reception of images using VLC.

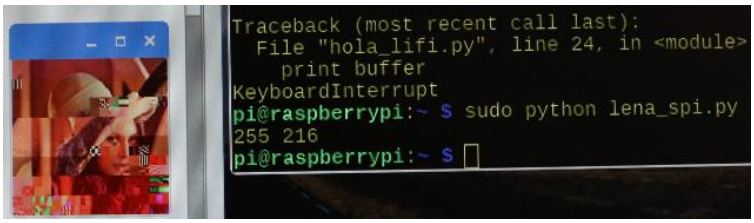



**Fig. 12.** Recovering of picture via VLC.


**Fig. 13.** Recovered picture via VLC after some adjustments.

### 4.1 Discussion of Results

After several adjustments to the setup and a few intents, the picture was finally well recovered. The whole transmission and recovering of a 225 x 225 pixel color image weighting 8 Kbytes, took less than a second. See Figure 13. Naturally, the noise cannot completely eliminated and increases when the misalignment and the distance between the LEDs and the light sensor is larger. Also the brightness of the LEDs influence the performance. That was the main reason for using two LEDs in parallel to increase the amount of light sent to the light sensor.

### 4.2 Metrics of the Recovered Picture via VLC

Because the transmission (LEDs) and reception (light sensor) link is made using Raspberry SPI, two questions arise: 1) What percentage of the image was recovered correctly? and 2) How many times the image was well recovered, versus the number of times the image could not be recovered?

**1) Percentage of the image that was recovered correctly**: As was shown in Figure 12, initially without any kind of adjustments, the recovery of the image was around 50 %. Notice from this figure the pixelation of the image in certain areas. In contrast with the adjustments mentioned before, the image was 100 % recovered as is shown in Figure 13.

**2) Number of times the image was well recovered**: Even though the adjustments performed to the transmitter-receiver setup, the link was shaky, requiring many tries to obtain a 100 % image recovery. The number of times the image was well recovered, was 3-out-of-10.

### 4.3 Contributions

This work differs with respect to similar works as in [8], [9] and [11], in the following. In [8] only a design model is proposed without any proof of implementation. In [9], a hardware setup is showed, but again, there is not any proof of sending and receiving any picture. It is also required to use two microcontrollers (one for the transmission and one for the reception). The main problem of using microcontrollers is that not all of them have enough RAM memory to store a medium to large image sent by the PC. Furthermore, the Matlab software have to reside at both PCs to connect with the microcontrollers and to process the picture downloading and uploading. In [11] a Windows PC with Matlab and an Arduino Uno microcontroller is used for the serial transmission and reception of two 24 x 25 color pictures, but without using visible light communication.

## 5 Conclusions

An image transmission and reception application using VLC was developed using a Raspberry Pi 3 computer, two bright LEDs and a light sensor from LittleBits, OOK modulation, and the Python Image Library. The picture in color, was recovered acceptably well although with a small presence of noise. It should be noted that this noise is due to environment light, and the distance between the LEDs and the light sensor.

Also, the performance of the application depends on the brightness of the LEDs and the sensitivity of the light sensor with respect to the light received from the LEDs and the surrounding light (in Figure 7 it can be seen that the light sensor has a control for graduating the sensitivity in the presence of high or low light). Additionally, the alignment between the LEDs and the light sensor influences the reception, and consequently the quality of the image reproduction.

## 6 Future Work

This application could be improved by: 1) Increasing the number and/or power of the LEDs to increase the reception distance; and 2) Implement and include a continuous

synchronization mechanism to improve the LED-to-light sensor VLC transmission-reception.

## References


1. Haas, H.: Wireless data from every light bulb. In: TED Ideas worth spreading (2011)
2. Tsonev, Dobroslav, Videv, Stefan; Haas, H.: Light fidelity (Li-Fi): towards all-optical networking. In: Proc. SPIE (Broadband Access Communication Technologies VIII) 9007 (2) (2013)
3. Sherman, J.: How LED Light Bulbs could replace Wi-Fi. Digital Trends (2013)
4. Haas, H.: High-speed wireless networking using visible light. SPIE Newsroom (2013)
5. Vincent, J.: Li-Fi revolution: internet connections using light bulbs are 250 times faster than broadband (2013)
6. Wikipedia: Location awareness (2016)
7. Jovicic, A., Li, J., Richardson. T.: Visible light communication: opportunities, challenges and the path to market (2013)
8. Mahendran, R.: Integrated Lifi (Light Fidelity) For Smart Communication Through Illumination. In: International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), pp. 53–56 (2016)
9. Vyom S. et al: 2D Image Transmission using Light Fidelity Technology, Communication. International Journal of Innovation & Advancement in Computing Science, IJIACS, ISSN 2347-8616, Volume 4, Issue No. 4, pp. 121–126 (2015)
10. Kwonhyung, L., Hyuncheol. P.: Modulations for Visible Light Communications With Dimming Control. IEEE Photonics Technology Letters, Vol. 23, Issue 16 (2011)
11. Cubas Perfecto, G., Santiago Godoy, R., Anzueto Rios, A.: Transmisión de imágenes de Matlab a Arduino vía Puerto Serial. Boletín UPIITA-IPN, 644 CyT, No. 52 (2016)
12. LittleBits: https://www.littlebits.cc/ (2017)
13. Raspberry Pi: Raspberry Pi input and output pin voltage and current capability. Mosaic Documentation Web (2018)

ISSN 1870-4069

# Monitor de signos vitales con comunicación inalámbrica Wi-Fi para unidad de cuidados intensivos desarrollado en LabVIEW y la tarjeta myRIO-1900

Héctor García Estrada[1], Angelo Pastrana Manzanero[1], Omar Alejandro Linares Escobar[1], Jeroan García Vázquez, María Guadalupe Ramírez Sotelo[2], Agustín Ignacio Cabrera Llanos[1]

[1] Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Biotecnología, Departamento de Bioprocesos, Ciudad de México, México

[2] Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Biotecnología, Departamento de Bioingeniería, Ciudad de México, México

aicllbuda@yahoo.com

**Resumen**. Se presenta el diseño y desarrollo de un monitor de signos vitales con conexión inalámbrica, destinado para una unidad de cuidados intensivos considerando las variables de electrocardiografía, oximetría, neumografía, temperatura, frecuencia cardiaca, frecuencia respiratoria y saturación parcial de oxígeno. Este proyecto funciona en tres etapas: adquisición de las señales, procesamiento y transmisión. Se diseñaron circuitos de adquisición para las señales de electrocardiografía y neumografía; para la electrocardiografía se usó un amplificador, seguido de un circuito de aislamiento óptico y un filtro rechaza banda, para la neumografía se utilizó un transductor de temperatura; para la señal de temperatura se utilizó otro transductor de temperatura y para la oximetría se utilizó el sensor pulsesensor. El procesamiento de las señales se realizó mediante la tarjeta myRIO-1900 programada en LabVIEW filtrando digitalmente las señales, y obteniendo a partir de las señales de electrocardiografía, neumografía y oximetría, la frecuencia cardiaca, la frecuencia respiratoria y la saturación parcial de oxígeno, respectivamente. Finalmente, la transmisión de la información se realizó mediante una red Wi-Fi generada con la myRIO mediante direccionamiento IP y programando una interfaz en LabVIEW para su despliegue gráfico en una PC.

**Palabras clave:** LabVIEW, monitor de signos vitales, unidad de cuidados intensivos, myRIO-1900.

# Vital Signs Monitor with Wireless Wi-Fi Communication for the Intensive Care Unit Developed in LabVIEW and the myRIO-1900 card

**Abstract**. The design and development of a vital signs monitor with wireless connection intended for an intensive care unit considering the variables of electrocardiography, oximetry, pneumography, temperature, heart rate, respiratory

rate and partial oxygen saturation is presented. This project works in three stages: signal acquisition, processing and transmission. Acquisition circuits were designed for electrocardiography and pneumography signals; an amplifier was used for the electrocardiography, followed by an optical isolation circuit and a band rejection filter, for the pneumography a temperature transducer was used; for the temperature signal another temperature transducer was used and for the oximetry the pulsesensor sensor was used. Signal processing was carried out using the myRIO-1900 card programmed in LabVIEW, digitally filtering the signals, and obtaining the electrocardiography, pneumography and oximetry signals, heart rate, respiratory rate and partial oxygen saturation, respectively. Finally, the transmission of the information was done through a Wi-Fi network generated with the myRIO through IP addressing and programming an interface in LabVIEW for its graphic display on a PC.

**Keywords:** LabVIEW, vital sign monitor, intensive care unit, myRIO-1900.

# 1. Introducción

## 1.1. Monitor de signos vitales

En el área de la salud es de gran importancia el conocer a cada instante la evolución fisiológica del paciente, lo cual permite a médicos y enfermeras valorar las condiciones generales y específicas de este y tomar decisiones. Ante esta necesidad, surgió un avance tecnológico conocido como monitor de signos vitales [1].

Los monitores de signos vitales, dependiendo de su configuración, adquieren, amplifican, procesan, registran y despliegan señales y/o información numérica para varios parámetros fisiológicos. Las variables por medir suelen depender del uso que se le dará al monitor, así como de las especificaciones técnicas [2].

– Monitor de signos vitales Básico: los parámetros que despliega son electrocardiograma, frecuencia cardiaca, frecuencia respiratoria, temperatura, presión no invasiva y oximetría de pulso.

– Monitor de signos vitales Intermedio: despliega los parámetros de un electrocardiograma, frecuencia cardiaca, frecuencia respiratoria, temperatura, presión no invasiva, oximetría de pulso y monitoreo de segmento ST, siendo opcionales el monitoreo de la presión invasiva, el gasto cardiaco por termodilución, cinografía y otros parámetros, de acuerdo con la especialidad en la que se instale.

– Monitor de signos vitales Avanzado: los parámetros que despliega son electrocardiograma, frecuencia cardiaca, frecuencia respiratoria, temperatura, presión no invasiva, presión invasiva, oximetría de pulso, cinografía, monitoreo de segmento ST, y en algunos casos se agregan los parámetros de índice biespectral, gasto cardiaco, espirometría y otros parámetros, de acuerdo con la especialidad en la que se instale.

– Monitor de signos vitales de Transporte: despliega los parámetros de un electrocardiograma, frecuencia cardiaca, frecuencia respiratoria, temperatura, presión no invasiva y oximetría de pulso.

– Monitor de signos vitales de Anestesia Básico: los parámetros que despliega son electrocardiograma, frecuencia cardiaca, frecuencia respiratoria, temperatura, presión no invasiva, presión invasiva, gasto cardiaco cinografía, oximetría de pulso, monitoreo de segmento ST, gases anestésicos y respiratorios.

En caso del monitor desarrollado, a excepción de la medición de la presión arterial no invasiva, se cumplen los requisitos de un monitor de signos vitales básico.

### 1.2. NI myRIO-1900

La tarjeta de adquisición y control de señales NI myRIO-1900, es un dispositivo de diseño embebido, el cual cuenta con múltiples entradas y salidas para adquisición y envío, analógicas y digitales, canales de audio, una alimentación de salida, entre otras cosas. Esta se conecta a un ordenador por medio de un cable tipo USB o mediante una red inalámbrica propia, permitiendo así una integración rápida y fácil en aplicaciones remotas y embebidas. La tarjeta de control NI myRIO-1900 posee tres puertos principales, los cuales son nombrados mediante las letras A, B y C, de los cuales, los dos primeros son conocidos como puertos de expansión (MXP), mientras que el último es llamado puerto del mini sistema (MSP). Estos cuentan con una descripción del tipo de señales que pueden recibir o emitir [3].

### 1.3. LabVIEW

El uso de la tarjeta myRIO-1900 permite una facilidad de programación en el software LabVIEW, el cual es un lenguaje gráfico de programación utilizado como estándar en el desarrollo de aplicaciones de test y medida, control de instrumentación y sistemas de adquisición de datos por medio de la generación de VI (Virtual Instrument). National Instruments ha ido desarrollando desde hace cinco años nuevas áreas estratégicas, relacionadas con nuevos campos de trabajo como Simulación, Diseño de Control, sistemas embebidos en tiempo real (FPGAs, DSPs, microprocesadores), algoritmos matemáticos avanzados, entre otras cosas.

LabVIEW cuenta con una caja de herramientas exclusivas para la tarjeta myRIO-1900, en la cual encontramos los bloques de adquisición o generación de las señales, ya sean de tipo analógicas o digitales.

## 2. Metodología

### 2.1. Diseño del circuito

Se diseñó y construyó un circuito impreso en placa por software, considerando las etapas de cada una de las variables a medir en nuestro monitor. El circuito contempla tres alimentaciones separadas, una de más menos 15 V para el módulo de filtro del ECG y el acondicionador de señal de neumografía, otra de más menos 9 V para el circuito de amplificación del ECG y una 5 V para el oxímetro de pulso y los transductores de temperatura.

Las señales del sensor de pulso y el medidor de temperatura se toman directamente y se ingresan a la myRIO, mientras que la señal de ECG se somete a un proceso de acondicionamiento de señal, así como la señal de neumografía, de manera que ambas se trabajen adecuadamente por la myRIO. Esto se abordará con más detalle posteriormente.

## 2.2. Diseño del canal de ECG

En el diseño y desarrollo del canal de ECG se contemplaron tres módulos: amplificación y prefiltrado, adecuación y aislamiento, y filtrado de la señal. Para la amplificación y prefiltrado se utilizó un amplificador de instrumentación AD620 [4], con un arreglo en la ganancia realizado en un circuito RC, como se muestra en la Fig. 1.

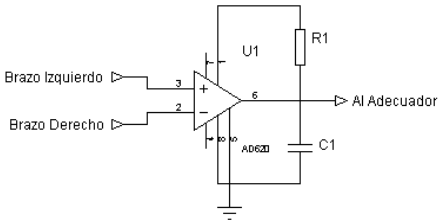



**Fig. 1**. Circuito de amplificación para el canal de ECG.

El arreglo RC que se colocó modifica la función de transferencia, generando un filtrado pasa banda en la ganancia, como se puede observar en la ecuación 1.

$$G = 1 + \frac{49.4\ k\Omega Cs}{R_G Cs + 1}. \tag{1}$$

En el caso del módulo de adecuación y aislamiento se utilizó un amplificador diferencial para colocar la señal en los parámetros de operación del opto acoplador utilizado (4N25), para la transferencia de la señal se polarizó el transistor del 4N25 [5], como emisor común obteniendo la señal de la base del transistor, como se puede observar en la Fig. 2.

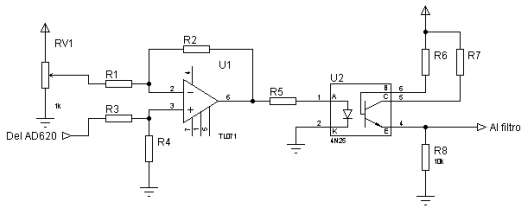



**Fig. 2.** Circuito de adecuación y aislamiento.

El circuito utilizado para la filtración fue de tipo notch, similar al que se muestra en la Fig. 3, logrando un filtrado rechaza banda a 60 Hz con un ancho de banda de rechazo de 6 Hz, eliminando el ruido de la línea.

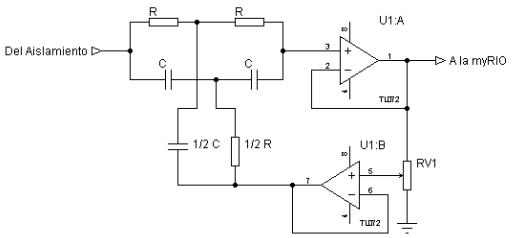



**Fig. 3.** Filtro de topología Notch.

La adquisición de la señal de ECG por la myRIO se programó en uno de los canales diferenciales debido a la capacidad de estos al leer valores de voltaje negativos. Una vez digitalizada la señal, se le aplicaron tres filtros: pasa bajas a 150 Hz, pasa altas a 0.5 Hz y rechaza banda a 60 Hz. La frecuencia de corte de los filtros pasa bajas y pasa altas se determinaron debido a que las señales de interés de un ECG, en un adulto se encuentran entre el ancho de banda de 0.5 a 120Hz, con un intervalo de hasta 50Hz [6].

## 2.3. Señal de pletismografía parcial de oxígeno no invasiva

Para la medición de la pletismografía y la saturación parcial de oxígeno se empleó el oxímetro de pulso Pulse Sensor modelo SEN-11574 [7], este funciona mediante una fuente de luz y un fototransistor. Cuando el sensor se encuentra en contacto con la punta del dedo o con el lóbulo de la oreja hay un cambio en la reflexión de la luz cuando la sangre es impulsada a través de los tejidos generando una señal analógica fluctuante. Esta señal es medida por la myRIO mediante un canal analógico referenciado a tierra.

Ya digitalizada la señal se somete a un proceso de filtrado digital con la finalidad de eliminar ruido de alta frecuencia. Posteriormente, de la señal de pletismografía se calcula la saturación parcial de oxígeno, a partir de la ley de Lambert-Beer (ecuación 2) donde a, b, c y d son constantes, R se obtiene a partir de la ecuación 3 aplicando el método de pico-valle [8].

$$SPO_2 = 100\,\frac{a-bR}{c-dR}, \tag{2}$$

$$R = ln\frac{V_{min}}{V_{max}}. \tag{3}$$

## 2.4. Neumografía y temperatura

Para la obtención de la señal de neumografía y temperatura se usó el transductor LM35 [9], el cual entrega una salida lineal de voltaje con respecto a temperatura a una razón de 10 mV por grado centígrado.

Para la temperatura se obtuvo la señal analógica directamente del sensor y se multiplica en el programa en LabVIEW para obtener el valor en grados centígrados. En lo que corresponde a la señal de neumografía, se le aplicó una ganancia a la señal del LM35 correspondiente por medio de un amplificador no inversor (Fig. 4); esto se

realizó debido a que la señal de neumografía se registró como los pequeños cambios de temperatura provocados por las inhalaciones y las exhalaciones.

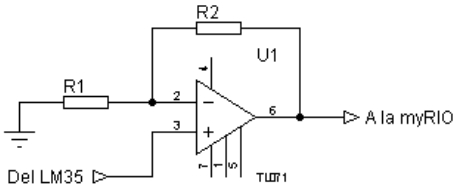



**Fig. 4.** Amplificador no inversor.

Después se establece la adquisición de la señal por medio de uno de los canales de entradas analógicas de la myRIO ubicados en el puerto C (MSP). Dicha señal es promediada y llevada a la siguiente etapa de nuestro algoritmo: la frecuencia respiratoria. Para ello se toma el valor máximo y mínimo dentro de un muestreo de 500 datos y se toma el promedio para delimitar el umbral que servirá como valor de referencia de nuestro contador. Esta operación se agrupa de igual manera en una estructura regida por un temporizador de 30 s, almacenando todas las elevaciones de nuestra señal detectadas. Por último, se multiplica este valor por 2 para obtener el número de respiraciones por minuto. El siguiente diagrama de flujo describe de manera general el proceso que se lleva a cabo en el cálculo de la frecuencia respiratoria (Fig. 5):

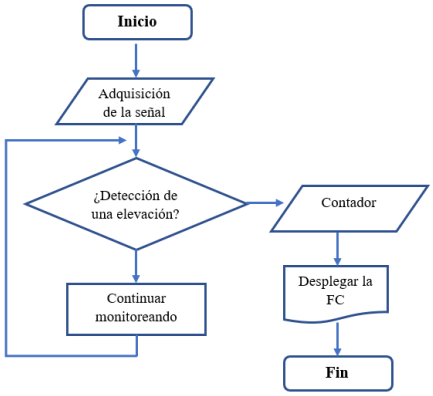



**Fig. 5.** Diagrama de flujo del algoritmo de frecuencia respiratoria.

Para la medición de temperatura corporal también se empleó un sensor LM35 por su salida en voltaje linealmente proporcional a la temperatura en grados centígrados, su baja impedancia de salida y su precisa calibración inherente de este dispositivo. También se encuentra clasificado para operar en un rango de -55 °C a 150 °C entrando en este rango el valor de temperatura corporal normal presente en el cuerpo humano.

La alimentación del sensor es llevada a cabo por el circuito antes mencionado, alimentando nuestro sensor con 5 V por medio de la señal de salida del conector C (MSP) de la myRIO, posteriormente la señal es tomada y llevada a través del circuito

hacia uno de los canales de entrada analógica presentes en el conector A (MXP) de la myRIO. La señal es compensada mediante un control numérico, esto debido a la escala de la señal de salida con la que opera nuestro sensor (10 mV/°C).

## 2.5. Creación de una red inalámbrica

Para crear una red Inalámbrica en una tarjeta myRIO-1900 es necesario contar con el programa NI-MAX, el cual nos permite conocer las configuraciones existentes en la tarjeta, así como modificarlas.

En ella podemos observar los sistemas remotos conectados a la computadora personal, al seleccionar un sistema en específico en la parte derecha de la pantalla se despliega la configuración existente de la tarjeta.

En el apartado Ethernet Adapter usb0 (Primary) se despliega la dirección IP, así como la configuración de esta y la sub-máscara de la tarjeta (Fig. 6).

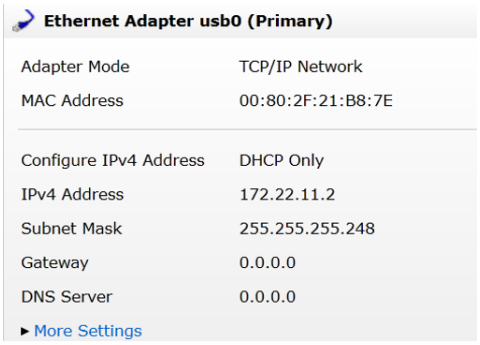



**Fig. 6.** Ethernet Adapter usb0 (Primary).

En el apartado *Wireless Adapter wlan0* tenemos las configuraciones inalámbricas de la tarjeta, la cual nos da tres opciones de configuración, para crear la red se selecciona la opción *Create Wireless network*, una vez seleccionada dicha opción se verifica que el país corresponda, en el caso particular, México.

Posteriormente, aparecerá la opción para configurar el nombre de la red, en la cual se le asignará el nombre deseado y se podrá elegir el tipo de configuración de la dirección IP, DHCP Only, lo cual nos genera de manera automática la dirección IP y la sub-máscara, de la red creada.

## 2.6. Posicionamiento de electrodos y sensores

Para la determinación del correcto funcionamiento del proyecto, fue necesaria la realización de pruebas a distintos pacientes.

En la Fig. 7 P se muestra el oxímetro de pulso conectado al dedo índice del paciente y fijado con velcro para evitar falsos contactos y alteraciones por movimiento de la señal de pletismografía.

En la Fig. 7 N se puede visualizar la conexión del sensor de neumografía en el labio superior adherido con microfibra y colocado en la boca de la fosa nasal para sensar los

cambios de temperatura por respiración y a partir de ahí obtener la frecuencia respiratoria.

Para la conexión al módulo del ECG, se emplean latiguillos conectados a un Jack 3.5 con electrodos de campana (Fig. 7 E) conectados al paciente con base al triángulo de Einthoven para las derivaciones bipolares. A partir de esta señal se obtiene la frecuencia cardiaca del paciente

En la región axilar izquierda está colocado el sensor (Fig. 7 T) para obtener el valor de la temperatura corporal de la piel del paciente.

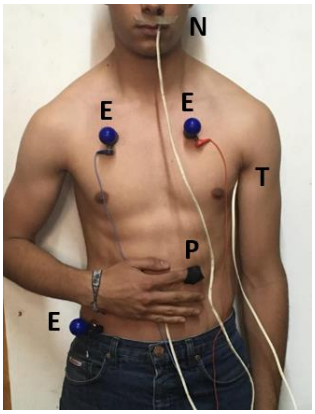



**Fig. 7.** Conexiones conjuntas de los sensores.

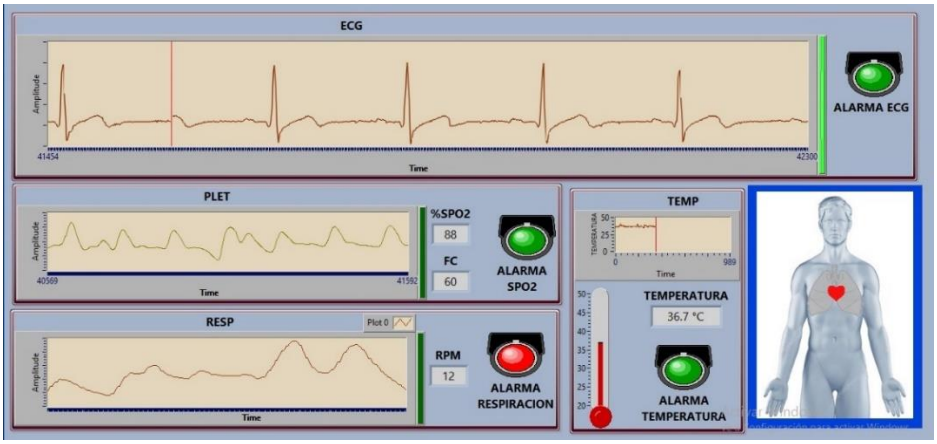



**Fig. 8.** Se muestra la visualización de la adquisición de las señales.

## 3. Resultados

Para la realización de pruebas, se despliega en el panel frontal la interfaz desarrollada en LabVIEW de la monitorización de las señales fisiológicas como son: ECG, neumografía, pletismografia, respectivamente, como valor numérico la Frecuencia

Cardiaca, Frecuencia Respiratoria y Saturación Parcial de Oxígeno y finalmente el indicador gráfico (termómetro) correspondiente al valor de la temperatura corporal (Fig. 8).

En la Fig. 9 se muestra el proceso de medición de los signos vitales de un paciente, conectando las señales analógicas a la tarjeta myRIO la cual inalámbricamente transmite los datos al ordenador, el cual es proyectado en una pantalla.

**Fig. 9.** Tarjeta myRIO-1900, con los sensores y circuitos diseñados mostrando en LabVIEW las señales monitoreadas.

## 4. Conclusiones

Se realizó un monitor de signos vitales en el cual se adquirieron, adecuaron y filtraron las señales fisiológicas tomadas de un paciente para visualizarlas en la interfaz de LabVIEW en tiempo real. El módulo Wi-Fi de la tarjeta myRIO permite el envío de datos vía inalámbrica Wi-Fi lo que posibilita la obtención a distancia de los parámetros del paciente a diferencia de los monitores comerciales. La realización de este trabajo puede abrir paso a una central de monitoreo más extenso en la que se haga una transmisión de señales vía inalámbrica Wi-Fi de distintos pacientes en tiempo real para una unidad de cuidados intensivos.

## Referencias

1. Jiménez-Ortiz, M.: Centro Nacional de Excelencia Tecnológica en Salud. Obtenido de http://www.cenetec.salud.gob.mx: http://www.cenetec.salud.gob.mx/descargas/biomedica/guias_tecnologicas/24gt_centrales_monitoreo.pdf (02 de Junio de 2006)
2. Secretaria de Salud: CENETEC. Retrieved from Guía Tecnológica No. 13: Monitor de Signos Vitales.: http://www.cenetec.salud.gob.mx/descargas/biomedica/guias_tecnologicas/13gt_monitores.pdf (enero, 2005)
3. National Instruments: Manuales de Productos. Retrieved from National Instruments: http://www.ni.com/pdf/manuals/376047c.pdf (febrero 27, 2018)
4. Analog Devices: Low Cost Low Power AD620. Retrieved from Technical Documentation, Datasheets: http://www.analog.com/media/en/technical-documentation/data-sheets/AD620.pdf (2011)

5. Vishay Semiconductor: 4N25, 4N26, 4N27, 4N28 Optocoupler, Phototransistor Output, with Base Connection. Retrieved from Vishay Semiconductor Docs: https://www.vishay.com/docs/83725/4n25.pdf (2017)
6. Kligfield, P., Gettes, L. S., Bailey, J. J., Childers, R., Deal, B. J., Hancock, E. W., Mirvis, D. M.: Recommendations for the Standardization and Interpretation of the Electrocardiogram. Journal of the American College of Cardiology, pp. 1109–1127 (2007)
7. World Famous Electronics llc: Pulse Sensor, Easy to use heart rate sensor and kit. Retrieved from Pulse Sensor: https://cdn.shopify.com/s/files/1/0100/6632/files/Pulse_Sensor_Data_Sheet.pdf?1435879 2549038671331 (2016)
8. Mazón, A., Rojas, S., Sánchez, E., Ramírez, G., Cabrera, A.: Oxímetro de pulso para monitoreo no invasivo aplicado en el monitoreo atlético. En: Memorias del VII Congreso Nacional de Tecnología Aplicada a Ciencias de la Salud, (2016)
9. Texas Instruments: LM35 Precision Centigrade Temperature Sensors. Retrieved from Texas Instruments Datasheets: http://www.ti.com/lit/ds/symlink/lm35.pdf (2017)

ISSN 1870-4069

# Spatio-Temporal Assessment of "*Chlorophyll a*" in Banco Chinchorro Using Remote Sensing

Hugo E. Lazcano-Hernandez[1], Javier Arellano-Verdejo[2], Hector A. Hernandez-Arana[2], M. Susana Alvarado-Barrientos[3]

[1] CONACYT-ECOSUR Chetumal, Laboratorio de estructura y función del bentos, ERIS, Mexico
[2] ECOSUR Chetumal, Laboratorio de estructura y función del bentos, ERIS, Chetumal, Quintana Roo, Mexico
[3] Instituto de Ecologa A.C., Red de Ecologa Funcional, Xalapa, Veracruz, Mexico
hlazcanoh@ecosur.mx

**Abstract.** Quantitative assessments of the temporal variances of physical and biological phenomena are very useful for the understanding of ecosystem functioning. Marine ecosystems are very complex, and regarding biodiversity and fishing resources, Banco Chinchorro (BCh) is one of the most important in the south of the Yucatan Peninsula. Additionally, BCh is an important hotspot of assimilation and release of carbon, for which hurricanes play a supporting role by mixing deep and superficial water masses affecting nutrient mixing and distribution. Here, the concentration of *chlorophyll a* (*Chl a*) was quantitatively linked to the occurrence of the four most recent hurricanes that affected BCh utilizing time-series analysis of satellite-derived (AQUA-MODIS) datasets. Interestingly, different *Chl a* concentrations between the south and north of BCh were confirmed quantitatively, which points to differential conservation efforts. The aim of this study was also to provide a proof-of-concept for the development of long-term monitoring methodology using remotely sensed data so that it may be replicated in other regions and with other satellite databases.

**Palabras clave:** remote sensing, time series analysis, AQUA MODIS, chlorophyll *a*, Banco Chinchorro, Hurricanes.

## 1 Introduction

*In-situ* observation and data collection from coastal and marine ecosystems in tropical areas remain a challenging endeavor. Coastal and marine tropical systems are complex in nature and subjected to several environmental forcing that drives its structure and function. This complexity is particularly evident in the Mexican Caribbean, a coastal and marine region with a wide ecosystem diversity that ranges from flooded forests, flood plains, wetlands, mangroves, freshwater lakes, coastal lagoons, coral reef lagoons, coral reef ecosystems and an oceanic realm [1].

To generate scientific knowledge and operational remote-sensing products a collaborative framework is required such that specialists in different areas of knowledge contribute. Spatio-temporal analysis of *Chl a* on BCh requires computational analysis of satellite datasets that can later be interpreted by a specialist in marine biology.

BCh has been studied by marine biological sciences for many years, however, the satellite remote sensing analysis has not been applied in depth in the region yet. The aim of this research is to create a *Chl a* foundation knowledge, to create a *Chl a* index to the region as [2]

The main aspects of different knowledge areas that were required for the analysis of *Chl a* in BCh are discussed in the present paper. In subsection 1.1 the stages that constitute the remote sensing processes are listed. Subsection 1.2 presents a brief summary of the satellite platforms and some technical aspects of AQUA. In subsection 1.3 some technical aspects of AQUA-MODIS sensor are presented. In subsection 1.4 some technical aspects of AQUA-MODIS datasets are explained. Subsection 1.5 shows the importance of hydro-meteorological phenomena for marine ecosystems. Later in Section 2, the ecological and economical importance of BCh for the region is summarized, as well as the relationship between hydro-meteorological phenomena and BCh. Subsequently in Section 3, the computer procedures and mathematical techniques that were applied in *Chl a* dataset are explained. Finally in Section 4, the contribution of this research is given.

### 1.1 Remote Sensing

Currently, satellite remote sensing is one of the main sources of quantitative spatial and temporal data, used to expand all aspects of Earth Sciences. The remote sensing process includes several very important subprocesses such as: satellite platform observation, data reception at a ground station, raw data preprocessing, data corrections, high level data analysis and final interpretation of the data by the end user. A schematic of remote sensing process is shown in Figure 1. The work done for this paper corresponds to the high level data analysis and final interpretation stages.

### 1.2 AQUA Platform

Currently, very different types of space missions (i.e. civil, commercial, communications, Earth observation) operate around the world. In the area of civil Earth observation, the AQUA mission is highlighted given the prolonged time that it has remained in operation and for the number of products that it offers. AQUA, or EOS-PM, was launched in May 2002. Its orbit is helio-synchronous and quasi-polar with an inclination of 98° and an average altitude of 705 km [3]. AQUA passes from south to north over the equator at 1:30 pm, which allows observation scenes of the Caribbean Sea with the sun very close to the zenith.

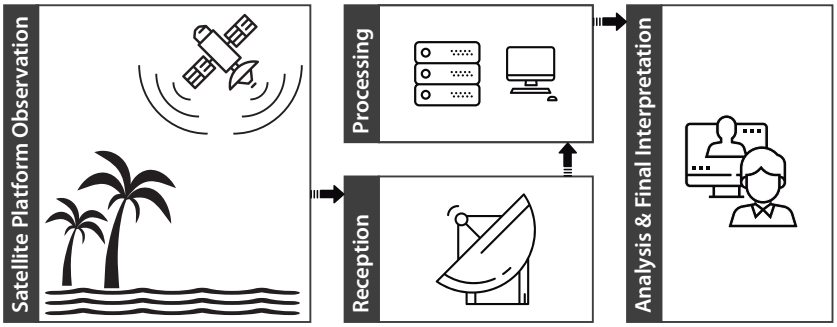



**Fig. 1.** Remote sensing

### 1.3 MODIS Sensor

One of the instruments that travels on board AQUA is the MODIS sensor. It is remarkable that all MODIS sensor products are freely available for download, through various NASA portals [4]. The MODIS instrument has 12-bit radiometric sensitivity and 36 spectral bands with a wavelength ranging from 0.4 $\mu$m to 14.4 $\mu$m [3]. It has a high geometric quality that allows the precise monitoring of the alterations of the terrestrial surface (RMS error of less than 50 m). The MODIS sensor is a sweeping scanner, that is, a mobile mirror that oscillates perpendicularly to the direction of the trajectory with an angle of $\pm$ 55° allowing the exploration of a strip of land on both sides of the satellite trace, whose width is 2,330 km. The optical system is a telescope with two mirrors outside its focal axis that direct the incident radiation to four reflective optical systems, one for each spectral region (visible, near infrared, medium and thermal). A silicon photodiode technology is used for the visible and near-infrared bands and mercury-Cadmium Teluret (HgCdTe) detectors are used for the thermal infrared[3].

### 1.4 *Chlorophyll a* Datasets

MODIS products are divided into five levels (0 to 4) depending on the processing level [3]. For the present study, level 3 Chlorophyll *a* concentration (*Chl a*) from AQUA-MODIS satellite platform, were used [5]. *Chl a* data gives near-surface concentration of *Chl a* in $[mg/m^3]$, calculated using an empirical relationship derived from *in situ* measurements of *Chl a* and blue-to-green band ratios of *in situ* remote sensing reflectances (Rrs) [4]. Bands 9 (443 nm), 10 (488 nm) and 12 (551 nm) were used to calculate *Chl a* [6]. Each data has a pixel size of 4.6 km per side, therefore 108 pixels were needed to complete one scene of BCh. The available information from AQUA starts in July 2002 and ends on April 2018, thus 190 matrices were analyzed.

*Chl a* datasets are the result of a remote sensing processing algorithm used to calculate the amount of phytoplankton biomass in the sea [5], because as

green plants, phytoplankton contains chlorophyll, which is the most abundant photosensitive molecule in nature and through which photosynthesis is possible. Phytoplankton photosynthesis is fundamental for various biogeochemical processes in the oceans and thus, constitutes the foundation of life in the oceans [7]. One of the potential benefits of using satellite information to derive chlorophyll concentration is that it allows continuous monitoring of phytoplankton.

### 1.5 Hurricanes

Tropical storms and hurricanes are a recurrent hydro-meteorological phenomena in the Caribbean basin and constitute the major physical disturbance for coastal areas [8]. Tropical storms and hurricane effects have been evaluated for both natural and human systems highlighting their negative impact on coral reefs, mangroves, wetlands, forest and on human lives and infrastructure. However, tropical storms and hurricane also possess a positive feedback on natural systems by causing the mixing of deep and superficial water masses with cascading effects in nutrient mixing and distribution. It has been observed that several weeks after a passing storm, the concentration of *Chl a* is increased and, as a consequence, carbon fixation and phytoplankton biomass increase [9]. Therefore, life is activated. In order to assess whether the same ecosystem-level and short-term response can be observed using available satellite data for a small area such as the BCh reef, the spatio-temporal distribution of *Chl a* was compared with the occurrence of extreme hydro-meteorological events that directly affected BCh, during the study period: Dean, Karl, Ernesto and Earl hurricanes (Table 1).

**Table 1.** Hurricanes with impact on Banco Chinchorro.

| Name | Date over BCh | Maximum cathegory | Cathegory over BCh | Displacement Vel km/h | Maximum winds over BCh km/h |
|---|---|---|---|---|---|
| **Dean** | 19 Ago 2007 | H5 | H5 | 32 | 232 |
| **Karl** | 15 Sep 2010 | H3 | TT | 13 | 90 |
| **Ernesto** | 08 Ago 2012 | H2 | TT-H1 | 24 | 116 |
| **Earl** | 04 Ago 2016 | H1 | H1-TT | 22 | 119 |

## 2 Study Area

The Banco Chinchorro reef (BCh; Figure 2) was selected for this analysis, due to its great ecological and economic importance for the south of the Mexican Caribbean. BCh is a coral reef of irregular oval form, with a length of 43 km in its longest part, and 18 km in its widest part. It is located southeast of the Yucatan Peninsula (18° 36' 15" N, 87° 21' 15" W), at 30.8 km of the coast, separated by a channel with a maximum depth of 1000 m. In the windward there is a very well developed reef, which confers a gentle slope, whereas in the

leeward, the slopes are abrupt reaching down to 500 m of depth [10, 13]. The depth of the study area ranges from 7 to 9 m in the southern portion and 2 m in the north, where marine grasses abound such as *Thalassia testudinum* [13]. The cays are surrounded by a mangrove ecosystem dominated red mangrove (*Rhizophora mangle*) and black mangrove (*Avicennia germinans*). The area is influenced by natural disturbances such as storms and hurricanes that damage mangrove vegetation. During the study period the eastern coast of the Yucatan Peninsula has been affected by two tropical storms and 12 hurricanes with direct impact to BCh in 2007 by Dean, in 2010 by Karl, in 2012 by Ernesto and in 2016 by Earl (Table 1). The area is also vulnerable to global climate change effects such as rising mean sea level and ocean acidification. Also, bleaching due to increased sea temperature threat to reduce coral cover [10]. Other local threats to the BCh ecosystems include overfishing. For instance, the commercially important threatened species pink snail (*Strombus gigas*) maintained abundant densities in the reserve but have began to decline since 1990 [13]. Likewise, the spiny lobster (*Panulirus argus*) is under a temporary fishing ban [14].

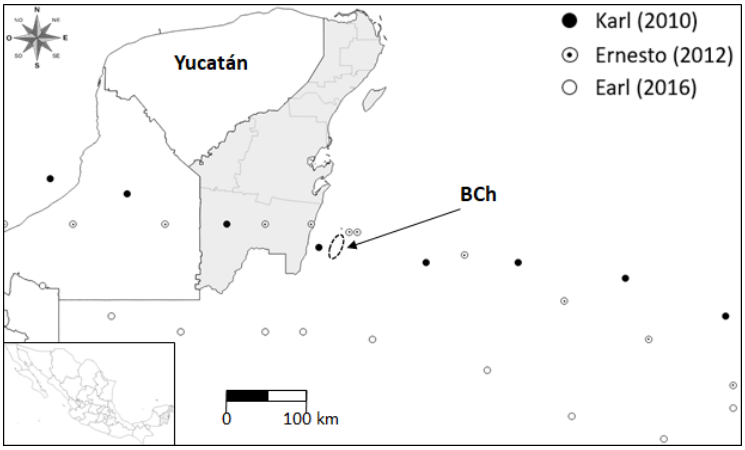
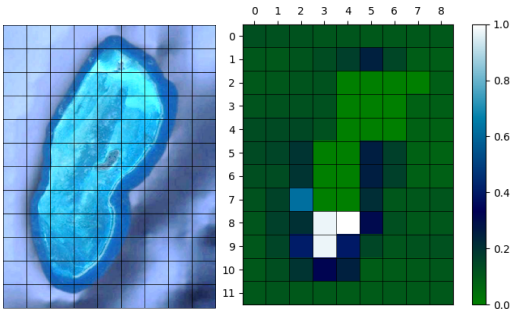



**Fig. 2.** Location map of Banco Chinchorro (BCh) and paths of hurricanes Karl, Ernesto and Earl.

Therefore, it is relevant to develop and evaluate methods to remotely monitor and study the dynamics of *Chl a* concentration at BCh because the food chain starts with phytoplankton and any event that quickly changes its concentration can trigger a cascade of ecosystem-level responses. As such, spatio-temporal knowledge of *Chl a* concentration is critical for the development of environmental preservation and fishing management proposals based on quantitative data and scientific information.

## 3 Data Analysis

Monthly *Chl a* datasets from AQUA-MODIS [4], were used and cropped out only to analyze the BCh area. Resulting bitmap files used were 9 pixels wide by 12 long (Figure 3). In total, 190 bitmap files were used in this study (from July 2002 to April 2018).




**a)** BCh Satellite view

**b)** Concentration map of *Chl a* in BCh

**Fig. 3.** Banco Chinchorro coral reef.

A time-series (Fig. 4) and spatial analysis per pixel of each file were applied (Fig. 8). For a more precise analysis, the time series were divided into its components: trend ($T$), seasonal($S$) and remainder component ($R$) [15]. Equation 1, shows the components of a dataset ($Y$), for $v = 1 \; to \; N$.

$$Y_v = T_v + S_v + R_v. \tag{1}$$

In this way, it was possible to analyze only the trend information, which in our case, is the most valuable, as the periodicity is already known and the noise contains no desirable information (Fig. 5).

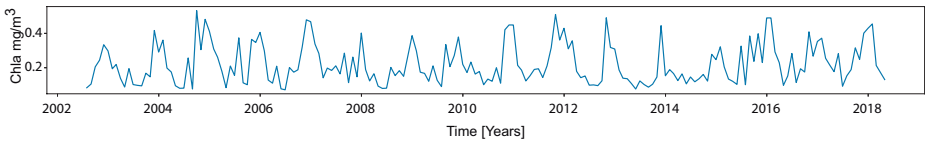



**Fig. 4.** Time series of the mean concentration of *chl a* in Banco Chinchorro.

To know *Chl a* levels in BCh before, during and after a hurricane, the four most recent hurricanes that affected BCh were located over the time series. Every hurricane has very specific features, which were considered in the analysis. Such features at the moment of hitting BCh are shown in Table1.

## 4 Results

The *Chl a* time-series was analyzed by a twelve-months mobile mean. Focused only on the trend information, only two of the four hurricanes were related with the lowest concentration values of *Chl a* (Figure 5**b**). It was expected that the occurrence of all events would be related to the minimum *Chl a* values, but that was not observed with a twelve-months mobile mean periodicity. From the computational point of view, it was not clear how to find the adequate periodicity to an accurate analysis. However, from the point of view of nature, it is obvious. It is known that hurricanes in the northern hemisphere (Atlantic Ocean) only occur during the months of May to November. Therefore, the mobile mean of the analysis was adjusted at six months. Fig. 6 clearly shows that hurricanes Earl and Ernesto also occurred at the lowest levels of *Chl a*, while this was not evident from the annual periodicity analysis.

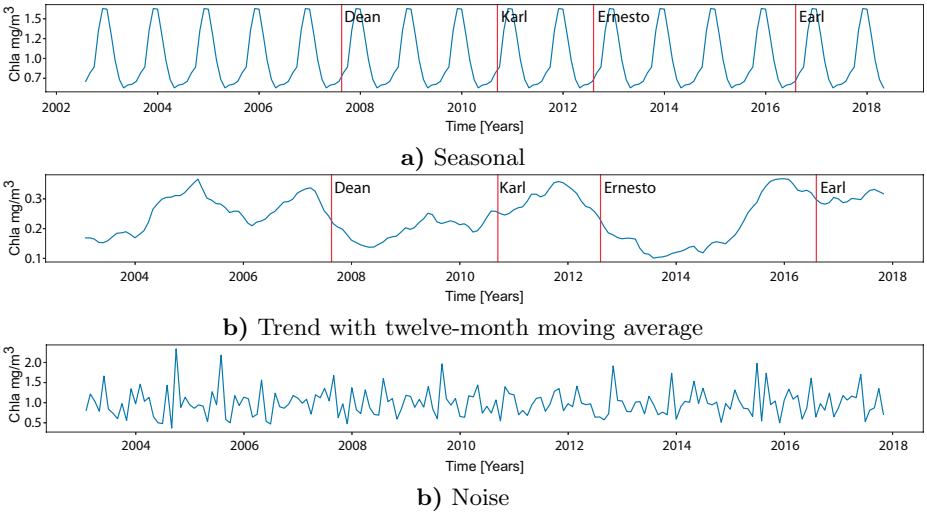



**Fig. 5.** Decomposition of the chlorophyll concentration time series.

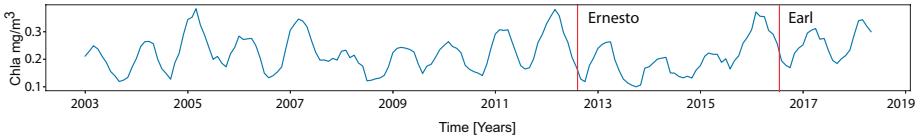



**Fig. 6.** Trend with 6-month moving average.

Because of the large amount of water and clouds over the sea during the occurrence of hurricanes, the MODIS optical sensor is practically blind. Thus, the sensor is unable to observe the sea surface, therefore it cannot produce any *Chl a* measurement. As an example, satellite images of Ernesto and Earl are shown in Figure 7**a** and 7**d**, respectively.

AQUA-MODIS sensor can observe the sea surface again some days after the hurricane passes due to such factors as: clouds and sensor technical issues. On the other hand, the increments of *Chl a*, it is observable and measured until the second month after a hurricane passes due to biological processes. As an example, see Figures: 7**b** and 7**c** which exemplify Ernesto, and Figures 7**e** and 7**f** which exemplify Earl.

Another important aspect seen in Figures 7**b** and 7**e**, is that concentrations of *chl a*, are higher in the southern of BCh compared to the north. Ergo, before the subsidy of nutrients and energy driven by the hurricane is used, in the southern zone of BCh, a very high concentration of *chl a* is observed.

Figure 8 strengthens the aforementioned. Regarding Ernesto, Figure 8**a** shows that *Chl a* in the southern zone of BCh is higher compared to the north. After that, from the next month, the increase of *Chl a* in whole BCh is shown in Figure 8**b**. This suggests that southern BCh is a bastion for the entire reef. Therefore, from the point of view of this analysis, southern zone of BCh, is placed as a strategic region for the preservation of the whole reef.

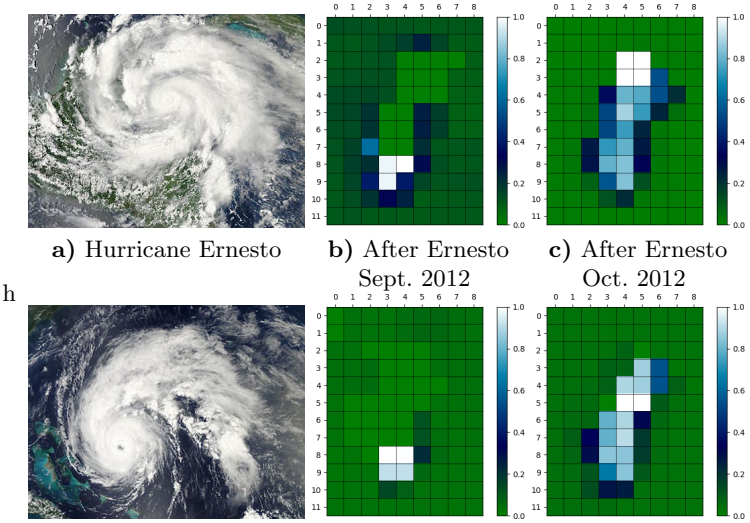
Regarding Earl, the same situation described for Ernesto was observed, (figures 8**c** and 8**d**). Therefore, the conclusion that, the southern zone is a bastion for the reef is confirmed. On the other hand, it is acknowledged that four events are probably not a large enough sample. However, the hurricanes that have affected BCh from 2002 to 2018 (the time period of AQUA-MODIS operation), have only been four.

## 5 Future Work

Analysis with SATMO [6] datasets is the natural next step in this research, because SATMO offers a spatial resolution of one kilometer per pixel. Additionally applied *Chl a* algorithms in optical bands of other satellite missions as "Sentinel 2" or landsat is missing. Finally *in situ* measurements of *Chl a* are still missing to calibrate satellite dataset.

## 6 Conclusions

Through a time-series and spatial analysis AQUA-MODIS datasets, the concentration of *chlorophyll a* was quantitatively linked to the four most recent hurricanes that affected BCh. Additionally, very important *Chl a* concentration differences between the south and north of BCh were confirmed. Compared to the north of BCh, the southern zone always maintains *chlorophyll a* concentrations an order of magnitude higher.

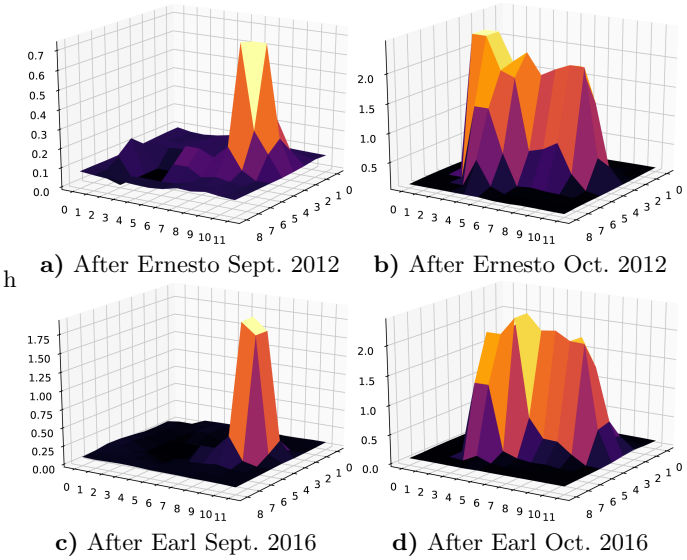


**a)** Hurricane Ernesto **b)** After Ernesto Sept. 2012 **c)** After Ernesto Oct. 2012

h

**d)** Hurricane Earl **e)** After Earl Sept. 2016 **f)** After Earl Oct. 2016

**Fig. 7.** Hurricanes and BCh raster images of *Chl a* $[mg/m^3]$.



**a)** After Ernesto Sept. 2012 **b)** After Ernesto Oct. 2012

h

**c)** After Earl Sept. 2016 **d)** After Earl Oct. 2016

**Fig. 8.** Spatial *Chl a* concentration on BCh. Axis: $z[mg/m^3]; x, y$ length $[pixel]$.

This study presents remotely sensed evidence of the influence of hydrometeorological events such as hurricanes on the generation of important phytoplankton blooms which are involved in the production of critical resources for the food chain, in particular for the sustainability of fisheries. The present study also hints at methodology development towards long-term monitoring of *chlorophyll a* temporal and spatial trends and studies to quantify the frequency and intensity of hurricanes' impact on coral reefs as well as related effects of climate change. Download, pre-processing, and spatio-temporal analysis of satellite-derived *Chl a* datasets was possible due to the computation work.

**Acknowledgments**. The authors acknowledge the Mexican National Council of Science and Technology (CONACYT) for funding through a Catedras project for young scientists (2015-526), to the satellite ground station ERIS for the access to the MODIS historical data collection and to El Colegio de la Frontera Sur (ECOSUR) campus Chetumal for additional support.

## References

1. Hernandez-Arana, H.-A., Vega-Zepeda, A., Ruiz-Zarate, M.-A. Falcon-Alvarez, L.-I., Lopez-Adame, H., Herrera-Silveira, J., Kaster, J. Transverse coastal corridor: from freshwater lakes to coral reefs ecosystems. In: Biodiversity and Conservation of the Yucatan Peninsula. Editors. Eslebe G., Calm S., Len-Cortes J. L., Schmook B. Springer Switzerland (2015) https://doi.org/10.1007/978-3-319-06529-8
2. Lacava T., Ciancia E., Di Polito C., Madonia A., Pascucci S., Pergola N, Piermattei V, Satriano V., Tramutoli V.:Evaluation of MODISAqua Chlorophyll-a Algorithms in the Basilicata Ionian CoastalWaters. Remote Sens. 2018, 10, 987, doi:10.3390/rs10070987
3. Jean-Franois M. (coordinator): Aplicaciones del sensor MODIS para el monitoreo del territorio. 2nd edn. SEMARNAT, Mexico (2011)
4. NASA Goddard Space Flight Center, Ocean Ecology Laboratory, Ocean Biology Processing Group: MODIS-Aqua Ocean Color Data; NASA Goddard Space Flight Center, Ocean Ecology Laboratory, Ocean Biology Processing Group (2014)
5. Hu Ch., Lee Z.,Franz B.: Chlorophyll a algorithms for oligotrophic oceans: A novel approach based on three-band reflectance difference. JOURNAL OF GEOPHYSICAL RESEARCH,doi:10.1029/2011JC007395, 117(5), 1–25 (2012)
6. Cerdeira-Estrada S., Lopez-Saldaa G.,A novel satellite-based Ocean Monitoring System for Mexico. Ciencias Marinas 37(2), 237–247 (2011)
7. Allen, J.F.: How does protein phosphorylation regulate photosynthesis?, Trends Biochem. Sci. 1(1), 12–17 (1992)
8. Farfan, L., DSa, E., Liu, K., Rivera-Monroy, V.: Tropical Cyclone Impacts on Coastal Regions: the Case of the Yucatn and the Baja California Peninsulas, Mexico. Estuaries and Coasts, 37, 1388–1402 (2014)
9. Hernandez de la Torre B. y Gilberto Gaxiola-Castro G. (Compiladores): Carbono en ecosistemas acuticos de Mxico. SEMARNAT, Cap.19, 279–292 (2016)
10. Vega-Zepeda, A., Hernandez-Arana, H., Carricart-Ganivet J.P.: Spatial Distribution and Health Condition of Acropora (Cnidaria: Scleractinia) Species in Chinchorro Bank, Mexican Caribbean: Implications for Management. Coral Reefs DOI 10.1007/s00338-007-0245-7. 26(3), 671–676 (2007)

11. Swain T.D, DuBois E., Goldberg S.J., Backman V., Marcelino L.A.: Bleanching response of coral species in the context of assemblage response. Coral Reefs 36(5), 395–400 (2017)
12. Perez-Santos I., Schneider W., Valle-Levinson A., Garces-Vargas J., Soto I., Montoya-Sanchez R., Melo-Gonzalez N., Muller-Karguer F. :Chlorophyll-a patterns and mixing processes in the Yucatan Basin, Caribbean Sea, Ciencias Marinas, http://dx.doi.org/10.7773/cm.v40i1.2320 40(1), 11–31 (2014)
13. Cala, Y.R., De Jesus-Navarrete A., Ocaa F.A., and Oliva-Rivera J. : Densidad, estructura de tallas y actividad reproductiva del caracol rosado Eustrombus gigas (Mesogastropoda: Strombidae) en Banco Chinchorro, Mxico. Biologa Tropical 61(4), 1657–1669 (2013)
14. Rios-Lara G.V., Espinoza-Mendez J.C., Zetina-Moguel C., Aguilar Cardozo C., Ramirez-Estevez A.: La pesqueria de langosta Panulirus argus en el Golfo de Mxico y mar Caribe mexicano In: Instituto Nacional de Pesca 50 Aniversario 1962-2012, Instituto Nacional de Pesca, Mxico (2013)
15. Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I.: STL: A Seasonal-Trend Decomposition. Journal of Official Statistics, 6(1), 3–73 (1990)

ISSN 1870-4069

# Control and Automation of an Industrial Food Dryer

Héctor García Estrada[1], Angelo Pastrana Manzanero[1], Lilia Leticia Méndez Lagunas[3], Juan Rodríguez Ramírez[3], María Guadalupe Ramírez Sotelo[2], Agustín Ignacio Cabrera Llanos[1]

[1] Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Biotecnología, Departamento de Bioprocesos, Mexico City, Mexico

[2] Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Biotecnología, Departamento de Bioingeniería, Mexico City, Mexico

[3] Instituto Politécnico Nacional, Centro Interdisciplinario de Investigación para el Desarrollo Integral Regional, Laboratorio de Tecnología Agroalimentaria, Oaxaca, Mexico
aicllbuda@yahoo.com

**Abstract**. In this paper, the development of a system of automation and control designed for the optimization of the operation of an industrial dryer is presented. The system was divided in three stages, such as: acquisition of variables, design of interfaces and the programming of the system. A series of transducers for the variables were used in the first stage. The development of the electronic circuits was used as an interface with the acquisition and processing of data card NI myRIO-1900. The stage of the development of the programming was made through the software known as LabVIEW, using a PID controller for the development of each control giving: temperature, relative humidity and air velocity. Controlling and visualizing these 3 parameters required a correct evolution on the dynamics of the drying process. The evolution of the variables described with the adjusted parameters is shown for PID controllers.

**Keywords:** industrial dryer, control PID, myRIO-1900, LabVIEW.

## 1 Introduction

The food industry has developed many techniques the drying food for diverse purposes. One technique is convective drying, where the heat is transferred to the solid by a hot air stream, which is driven by fans [1]. The dryer that was automated is located at the facilities of the Agro-alimentary Technology Laboratory of the Interdisciplinary Research Center for Regional Integral Development Oaxaca Unit (CIIDIR), of the National Polytechnic Institute.

This dryer has modules of temperature, humidity and air velocity. Each of these modules has its own acquisition and regulation system.

**Fig. 1.** Continuous flow dryer.

### 1.1 Transducers

The transducers consist of a group of instruments with dedicated microprocessors. The HMP230 transmitter series of Vaisala perform measurements of relative humidity and temperature. Output signals can be configured at measurement ranges within certain limits. It also has 2 types of analog outputs, current output from 0 to 20 mA and voltage output from 0 to 10 V maximum [2].

**Fig. 2.** Model HMP230 transducer of the Vaisala brand.

The AVT55, AVT65 and AVT75 air speed transducers are ideal at temporary and permanent installations for measuring air velocity. One of the advantages of using these models is that they contain calibration curves that provide output signals with linear response, which allows them to be adaptable to data acquisition systems [3].


**Fig. 3.** Alnor AVT55 air speed transducer.

### 1.2 Proportional Integrative Derivative Controller (PID)

A PID is a second order controller, with an integrator. PID controllers allow control actions from a measured value and a desired value, allowing a quick and easy application in industrial processes, since they only need a sensor that acquires the process variable and an actuator that reacts to the control output. The equation of a controller with this combined action is found in the following Equation (1):

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau)\, d\tau + K_D \frac{de(t)}{dt}. \quad (1)$$

Where *u(t)* represent the output control, *Kp*, *Ki* and *Kd* the gains in each stage and *e(t)* is the inherent error in the system. Empirical researches show that the structure of the PID usually has enough flexibility to achieve excellent results in many control applications [4].

## 2 Methodology

The automation of the dryer is divided in three stages, which are:

- Acquisition and measurement of the variables to be controlled.
- Design of the interface circuits required by the system.
- Programming of the controls in LabVIEW.

Below is a general outline of the designed system followed by the development of each of the three mentioned stages and the results.

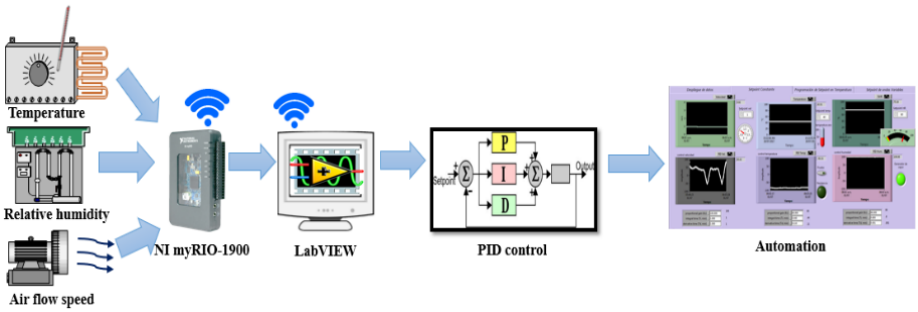



**Fig. 4.** General scheme of the designed system.

### 2.1 Variables Acquisition

For the acquisition of the variables, both transducers were set-up in order to acquire the output signals in voltage, avoiding the use of adequacy circuits.

For the air velocity transducer, the output signal and the measurement range were in a range of 0 to 5 volts and 0 to 5 meters per second, which achieved a 1:1 ratio in terms of voltage output and air velocity.

For the transmitter of relative humidity and temperature, the output signals were of 0 to 5 volts. The measurements intervals in both variables were from 0 to 100, in percentage for relative humidity and degrees Celsius for temperature, thus making a slope adjustment to obtain the variables.

## 2.2 Interface Circuits

For each process there was a device that was responsible for compensating the variables. In the case of the air speed of the dryer, there is a two-phase alternating current motor as a fan; the regulation of the rotation speed of this engine was implemented by the frequency converter SAMI 018MD2. As the air speed depends directly on the speed of the motor rotation, the output of the speed control is the frequency of operation of the converter.

The input of the frequency converter in voltage was of 0-10 V. This input was achieved using one of the myRIO's analog outputs. Since the analog outputs of the mmyRIO are in the range of 0-5 V, a circuit was necessary to amplify the signal and equalize it with the interval of the input of the frequency converter. For this, an OPAM array was used composed of two inverting amplifiers. The first gives the necessary gain while the second corrects the polarity. Because the frequency converter is a power system, a diode was added to the output of the OPAMS to protect both the circuit and the card against any return current.

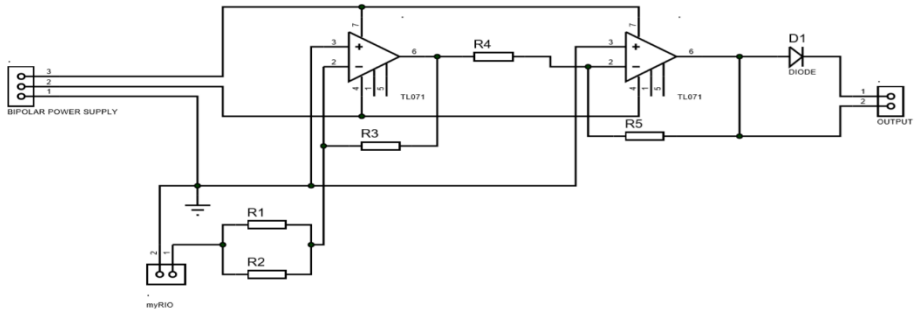



**Fig. 5.** Adequacy circuit of the myRIO analog output.

On the other hand, the output of the temperature control is a finned resistor for air heating fed to 220 V. To protect the system and prevent damage on the resistor, there is a safety stage that does not allow the activation of the resistor unless the fan is on. For this, the resistor is actioned by a solid-state relay (SSR3-440B), which can operate with an input and an output operation of 120 V and 450 V in alternating current [5].

To control the resistor, it was necessary to design an interface circuit between the myRIO and the relay. For this, the optocoupler model MOC4011 was used for its output in the form of phototriac [6]. To complete the activation circuit the triac BTA12 was used with an array of resistors, managing to control the operation of the resistance by means of a digital output of the myRIO [7].

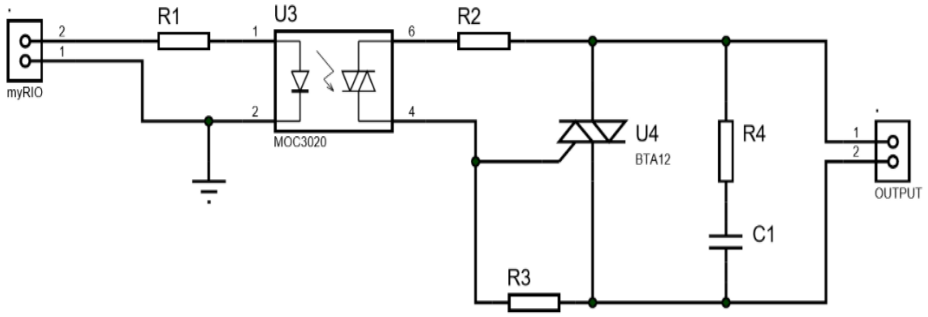



**Fig. 6.** Power circuit for the activation of the resistance and steam generator.

In the case of humidity variable, a steam generator was used. Within a control output, steam was injected to the dryer by this generator. This device operates with an alternating current of 220 V, controlling in this way the steam flow by electro valves. Because the circuit design applied to the resistor control operates within the same parameters as the steam generator, it is implemented again with the humidity generator.

### 2.3 Programming and Tuning of the Controls

The programming and tuning of the controls were made using LabVIEW with the "Control and Simulation" module. The type of controller employed is the PID controller. This is due to the ease of implementation. To perform the necessary tests on the controls, a virtual instrument was developed in LabVIEW where the parameters could be varied while the program was running.

The PID tuning was made with a heuristic approach, comparing the behavior when making changes in the gains. Each of the three controls was tuned separately obtaining different values for the constants of each. The control outputs of the system were programmed in different ways. In the case of the output of the speed control this was set-up to use an analog output of the myRIO together with the adequacy circuit.

For the temperature and humidity outputs, the outputs were set-up in such a way that the output of the control was positive, either the resistance or the electro valve of the steam generator were activated. Otherwise, the systems were turned off. This was achieved by the digital outputs of the myRIO.

## 3 Results

A continuous flow dryer was automated through the implementation of PID controls in the variables with a virtual instrument for set-point definition, the variables and control responses. The control outputs behaved in such way that the drying process was not affected. Tuning the controls by test and error, the gains were reached to values with a good performance as presented below:

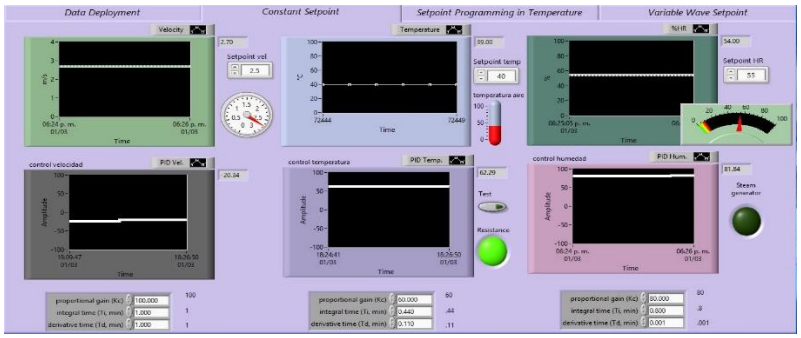



**Fig. 7.** Front panel virtual instrument for the programming of the controls.

In the case of speed control with gain values of $Kp$ = 100, $Ki$ = 1 and $Kd$ = 1, the following behavior was achieved. In the graph a set point of 2.5 m/s was used with an error in the steady state of ± 0.1 m/s.

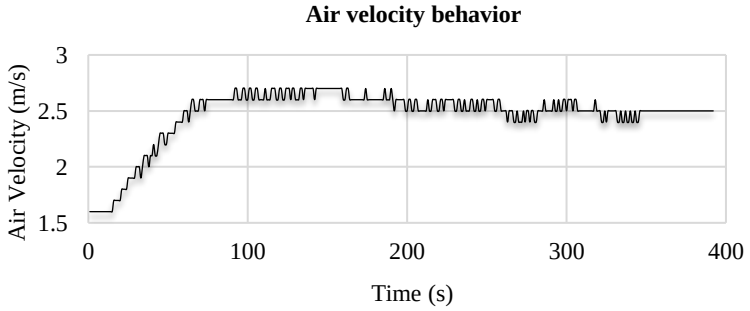



**Fig. 8.** Speed control behavior graph.

To control humidity with gain values of $Kp$ = 80, $Ki$ = 0.8, $Kd$ = 0.001 also a set point of 50% with an error in the steady state of ± 6.85 percent relative humidity; this value being acceptable for the experiments that require it.

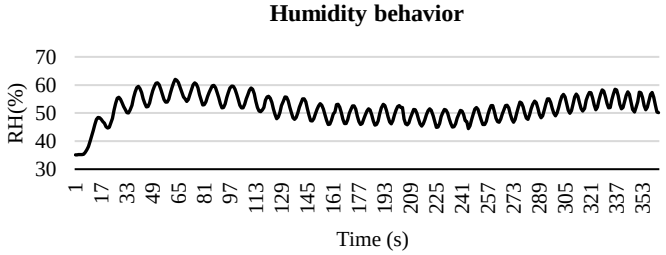



**Fig. 9.** Relative humidity control behavior graph.

Finally, the temperature variable with gain values of $Kp$ = 60, $Ki$ = 0.44, $Kd$ =0.11 and a set-point of 40 °C can be controlled with an error in the steady state of ± 1.65 degrees centigrade, an acceptable range for the system. The values of gain in each control can be modified according to the experimental conditions required by the researchers of the unit [6].

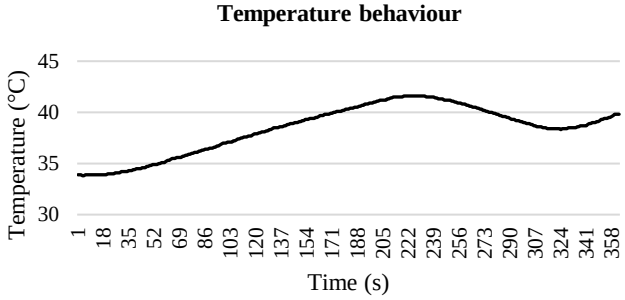



**Fig. 10.** Temperature control behavior graph.

We know that one of the main reasons to automate a system is the increase of productivity levels, reduction of the energy consumption, increase of the reliability of the result and, of course, the optimization of the human resource inside the facilities of the laboratory allowing the user to develop other activities while the process is taking place.

The maintenance programs for the NI myRIO and the interface circuits are minimal, since, only spare parts of the electronic components would be needed within the interface circuit. Finally, it is important to highlight that at most cases it is always cheaper to use new instruments than repair an older one because the dryer presented in this work used a FieldPOINT acquisition system of the FP-1000 series changing the maintenance cost of this system, and added to that, modules that are difficult to obtain in the market [7, 8].

Due to the characteristics of the PID controller and the form of tuning that was used, a fast and functional control system is obtained according to the characteristics for which the dryer is required. In addition, the ease of use of control through the LabVIEW module makes the PID control system a useful tool for this process, without ruling out the possibility of implementing in future work other more modern control systems [9]. Sometimes it is very difficult to obtain the model of the plant which is due to different factors. As future work the corresponding model of the system could be made, taking the time necessary for its construction.

## 4 Conclusions

The variables of temperature, relative humidity and air speed in a dryer tunnel were controlled by using PID-type controllers, achieving an evolution within acceptable parameters in the food drying process, allowing the drying process to comply with the conditions imposed by the researchers of CIIDIR Oaxaca. The realization of this project

implies an innovation in the system, allowing a control of the process, in a more effective way, due to the friendly handling of the parameters of the PID controllers, shown in the front window of the program carried out. The updating of the data acquisition system by the myRIO allowed the development of all the control algorithms, to be used in a single computer. Finally, by leaving the changes enabled gain values of each of the controls to perform different tests at the user's pleasure, thus observing different behaviors in the dynamics of the system.

## References

1. Fito-Maupoey, P., Andrés-Grau, A. M., Barat-Baviera, J. M., Albors-Sorolla, A. M.: Introducción al secado de alimentos por aire caliente. Universitat Politécnica de Valencia, Dorman (2016)
2. Vaisala: HMP230 Series Transmitters user's guide. Finland: Vaisala (2002)
3. Alnor: Air Velocity Transducers Models AVT55, AVT65, and AVT75. Estados Unidos: Alnor (2010)
4. Mazzone, V.: Controladores PID. Argentina: Universidad Nacional de Quilmes (2002)
5. Asiaon Relay: SSR3 Solid State Relay. China: Asian Relay (2010)
6. García, H., Pastrana, A., Méndez, L., Rodríguez, J., Ramírez, M., Cabrera, A.: Automatización de un secador de flujo continuo por medio de la tarjeta myRIO-1900 y LabVIEW. En: Congreso Internacional sobre Innovación y Desarrollo Tecnológico (2018)
7. Velásquez, J.: Cómo justificar proyectos de automatización. Industrial data, 7(1), pp. 7–11 (2004)
8. Gómez-Gómez, N.: Cinéticas de ácido pirúvico durante el proceso de secado constante y variable del ajo. Tesis en Ciencias en Conservación y Aprovechamiento de los Recursos Naturales, Oaxaca, Oax., Instituto Politécnico Nacional (CIIDIR) (2008)
9. Ruiz-Alcántara, A. H.: Control Difuso Vs Control PID: Análisis y Simulación Numérica. Tesis en Ciencia Básicas en Ingeniería, Pachuca de Soto: Universidad Autónoma del Estado de Hidalgo (2007)

ISSN 1870-4069

# Synesthetic Musical Composition using Computational Intelligence

Alan Garcia-Zambrano[1], Yenny Villuendas-Rey[1], Oscar Camacho-Nieto[2]

[1] Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Mexico City, Mexico
[2] Instituto Politécnico Nacional, Secretaría de Extensión e Integración Social, Mexico City, Mexico
alanz2706_garzia@hotmail.com, yenny.villuendas@gmail.com, ocamacho@ipn.mx

**Abstract.** Synesthesia is the combination of two senses in the same perceptive act; see the music, hear a red color or feel the texture of a green sound are some examples. The problem of a bad teaching in music theory turns music into a tedious and boring subject, causing in many cases a desertion. The musical composition with synesthesia will make it easier for the user to learn music through the association of sound with color and exploit a new way of learning with intelligent computing methods, where the advanced user creates visually pleasing musical compositions encouraging musical creativity. This article explains the use of evolutionary algorithms as a novel auxiliary model for musical innovation. In addition, it shows the existing tools of intelligent computation (IC) for musical composition and creation of computational art.

**Keywords:** computational intelligence, computational evolution, synesthesia, music composition, genetic algorithms.

## 1 Introduction

Musical language for centuries has been represented by musical signs embodied in manuscripts; systems of notation that determine the rules for the interpretation and together they integrate a melody [1]. Sometimes, the learning of *theoretical-musical* concepts is a rigorous process leading to ostracism. A study indicates that it takes about 10 thousand hours of practice in a musical instrument to be an expert; what represents practice for more than 10 years [2]. Recently, a large number of methods have been used of Computational Intelligence (CI) for compositions and expression of musical abilities. Genetic Algorithms (GA) [3], [4], Genetic Programming (GP) [5], Particle Swarm Optimization (PSO) [6]: metaheuristics for search and optimization. Its engine is the inspiration of nature as a biological metaphor, *selection*, *crossover* and *mutation*, proposed by Darwin and Mendel. Mitchell, Holland and Goldberg [7] who have managed to develop novel applications using CI for an endless number of scientific applications. Each individual in an evolutionary algorithm represents the solution to a problem, and those that offer better adaptability to the mechanisms of nature will

survive. The CI has managed to adapt models of artificial intelligence inspired by biological evolution; proving to be necessary in the study and development of new methodologies and applications as an excellent option to solve this type of problems.

We propose the intelligent modelling of the music composition problem as an auxiliary resource for learning, which allows the joint assimilation of perceptive sensations (hearing and sight) to improve techniques of improvisation and musical learning with *Synesthesia*. Its purposes are: (1) that musical learning with synesthesia avoids ostracism. (2). highlight the importance of the exercise of the senses in the acquisition of knowledge and musical ability and (3) adapts the evolutionary computation to a requirement of the company. In addition, it would work as a very useful tool for experienced musicians who are in the so-called "*Creative Stagnation*" due to lack of inspiration, stress or depression; creating visual art based on compositions and interpretations of musical language, supported by the principles of genetic evolution.

The paper is organized as follows: In Section one: a brief introduction on the importance of the project is detailed; in Section two: related works and bases for technological development are mentioned, in Section three: the methodology used, materials and methods for development are established. In Section four: the solution proposal is detailed; in Section five: the expected results are shown. Finally, the conclusions and future work are mentioned.

## 2 Previous Works

Composers like J. S. Bach, and W.A. Mozart used algorithms as a technique of improvisation and musical writing: the "Music Dice Game" used randomness, inspiration and a pair of dice creating complex musical pieces [8] or the "BACH Motive", which was used in cryptography [9]. Hiller, and Isaacson, designed the first computer-aided music generation (ILLIAC) [10]. The musical composition uses different methodologies for composition such as mathematics based on stochastic and probabilistic processes to predict musical sequences based on "Markov chains" [11] and fractal geometry as a form of musical expression [12]. In addition, grammar, which encode multidimensional arrays to create patterns of rhythm and also harmony. The authors of [13] built a system for electric bass. With an input line connected to the computer, it feeds the population and agents cooperate to generate complete musical pieces as a solution to an optimization problem. [14].

He uses musical compositions based on AG to create cryptograms. The encryption algorithm encodes a message into another intelligible with music. The authors of [15] generate images from L-Systems using AGI; because of visual interpretation, the system models growth and develops visual constructions with music, balancing technology and music with visual expressions. In the current market, there are a variety of software applications for musical composition and assistance, such as GarageBand [16] an auxiliary software for professional musicians which supports numerous synthetic instruments so that the user can compose a melody easily, Guitar Pro [17] is a sheet music editor that supports the MIDI format. Chordbot [18] is an application for

complex chords and accompaniment styles to produce tracks and mixes of piano and bass. TonePad [19] is an application with an interactive interface; the user creates melodies by touching a matrix on the device's screen. However, perhaps the most notable example of algorithmic musical composition is the Iamus Computer Cluster [20], it is the first computer that simulates and recognizes human language to perform musical improvisation.

## 3 Coding, Structure and Configuration

In the following section, we present the methodology of the modelling in four sub-sections: Structure, configuration and coding of the phenotype, primordial elements of the genetic algorithm, representation of the solutions and calculation of objective function.

### 3.1 Genetic Coding

The genetic structure is composed of the genotype and phenotype with the following *genes*: (1) *Clef*, (2) *Octave*, (3) *Scale*, (4) *Measure*, (5) *Note* and (6) *Accidental*.

1. **Clef**: Determines the position in the musical staff with a binary coding [0,1].
2. **Octave:** Each musical octave is composed of twelve semitones with their accidentals, that is, seven fundamental notes plus five accidentals defined by the pitch or height of the sound in a musical instrument. Using Octave 2 - Octave 6 with the configuration [0 - 4] to delimit the range of sounds and colors is proposed.

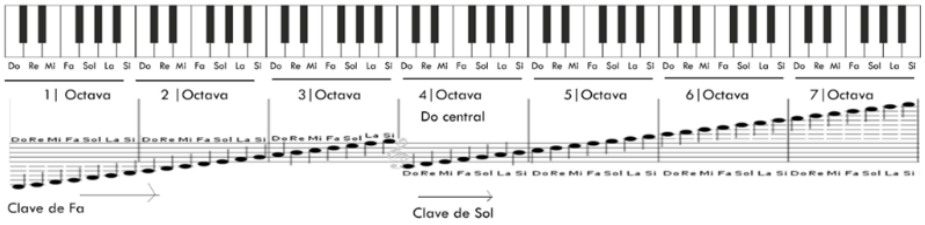



**Fig. 1.** Conventional keyboard of seven octaves differentiated by the clefs of G and F.

3. **Scale:** It contains the coding for seven musical notes distributed progressively (Figure 2).
4. **Measure**: Its phenotype distinguishes six figures; *Silence*, *Whole*, *Half*, *Quarter*, *eighth* and *sixteenth note* with coding: [0 - 6]. The spaces of the chromosome (5-36) determine time, duration, frequency and the number of elements.

A *Whole note* is equivalent to a structure of sixteen elements (*sixteenth note*) which is equivalent to two eighths of eight notes (*eighth note*). In addition, two figures of (*Half note*) are equivalent to four figures of one time (*Quarter*), that is: Half is worth two Quarters; the *eighth note* is worth two eighth notes. Therefore, the *Whole note* is equivalent to two *Half notes*, four *Quarters*, eight *eighth notes* (*sixteenth notes)*. This

scheme (Figure 3) is proposed to represent the duration and distribution of the musical notes in the *measure* with a speed of 80 beats per minute (bpm).

| C Major | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C | | D | | E | | F | | G | | A | | B | Silence | |
| 0 | 0 | 1 | 0 | 2 | 0 | 3 | 0 | 4 | 0 | 5 | 0 | 6 | 0 | 7 | 0 |

| D Major | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D | | E | | F# | | G | | A | | B | | C# | | Silence | |
| 1 | 0 | 2 | 0 | 3 | 1 | 4 | 0 | 5 | 0 | 6 | 0 | 0 | 1 | 7 | 0 |

| E Major | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E | | F# | | G# | | A | | B | | C# | | D# | | Silence | |
| 2 | 0 | 3 | 1 | 4 | 1 | 5 | 0 | 6 | 0 | 0 | 1 | 1 | 1 | 7 | 0 |

| F Major | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F | | G | | A | | Bb | | C | | D | | E | | Silence | |
| 3 | 0 | 4 | 0 | 5 | 0 | 6 | 2 | 0 | 0 | 1 | 0 | 2 | 0 | 7 | 0 |

| G Major | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| G | | A | | B | | C | | D | | E | | F# | | Silence | |
| 4 | 0 | 5 | 0 | 6 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 3 | 1 | 7 | 0 |

| A Major | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | | B | | C# | | D | | E | | F# | | G# | | Silence | |
| 5 | 0 | 6 | 0 | 0 | 1 | 1 | 0 | 2 | 0 | 3 | 1 | 4 | 1 | 7 | 0 |

| B Major | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B | | C# | | D# | | E | | F# | | G# | | A# | | Silence | |
| 6 | 0 | 0 | 1 | 1 | 1 | 2 | 0 | 3 | 1 | 4 | 1 | 5 | 1 | 7 | 0 |

**Fig. 2.** Genetic configuration of the notes by major scales with accidental and silence.

| Measure in 4/4 at a tempo of 80 bpm | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Whole Note | | | | | | | | Sixteenth Note | | | | | | | |
| 16 | | | | | | | | 1, 2 , 3 ... 16 | | | | | | | |
| 0 - 1920 ms | | | | | | | | 120, 240, 360, ... , 1920 ms | | | | | | | |
| Half Note | | | | | | | | Eighth Note | | | | | | | |
| 8 | | | | 16 | | | | 2 | | 4 | | 6 | | 8 | |
| 0 - 960 ms | | | | 960 - 1920 ms | | | | 0 - 240 ms | | 240 - 480 ms | | 480 - 720 ms | | 720 - 960 ms | |
| Quarter Note | | | | | | | | 10 | | 12 | | 14 | | 16 | |
| 1 | | 2 | | 3 | | 4 | | 960 - 1920 ms | | | | | | | |
| 0 - 480 ms | | 480 - 960 ms | | 960 - 1440 ms | | 1440 - 1920 ms | | | | | | | | | |

**Fig. 3.** The proposed compass is divided into four quarters of time with a total duration of 1920 *ms*; each *quarter* of time lasts 480 *ms*, in turn, it is divided into four segments of 120 *ms*.

5. **Musical Notes**: The notes are the interpretation of a frequency produced by the vibration of a sound, together they allow composing melodies. Its phenotypic representation is defined as (*C, D, E, F, G, A, B* and silence) with the coding [0-7].
6. **Accidentals**: They are the symbols that increase or decrease the intonation of the notes by semitone. Its phenotype is represented by sharp or flat (#, b), with coding [0,1] They are essential in the genetic mutation process.

### 3.2 Structure of the Chromosome

The genetic structure of the chromosome contains thirty-six spaces available to store the genes. The first four spaces belong to the **configuration** and the rest defines the **structure** of the musical system.

- **Configuration**: The genes in the chromosome (1 - 4) define the initial configuration and rules of the system.
- **Structure**: The genetic composition of a chromosome in the spaces (5-36) correspond to the musical notes and alteration. The odd spaces (5,7, ..., 35) determine specific I formation of the alleles corresponding to the musical notes (*C, D, E, F, G, A* and *B*), and the pairs (6,8, ..., 36) accidentals (#, b).

- **Genotype**: Contains the genetic configuration of the genes in a chromosome, represents the information contained in the genes with the characteristics that define each individual, these are transmitted from generation to generation by inheritance.
- **Phenotype**: Represents the visible characteristics of each gene in the chromosome, during development, such as hair color, eyes, height, weight, etc.

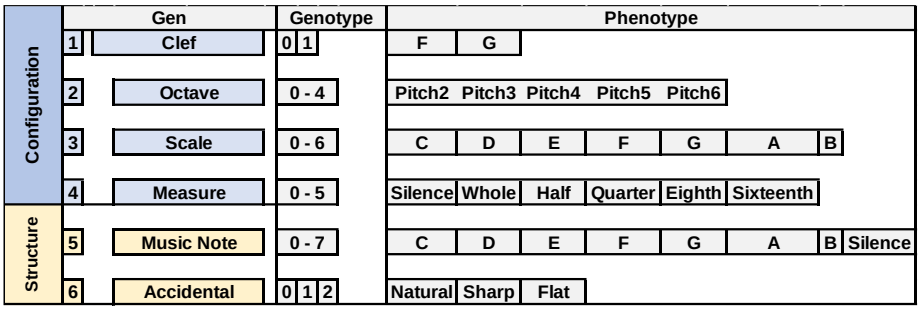



**Fig. 4.** General outline of the configuration and block structure of a chromosome.

## 4 Genetic Algorithms for Intelligent Musical Composition

Genetic Algorithms (GA) are metaheuristics of search and optimization inspired by nature's mechanics. They are used to solve problems or optimize solutions and are essential in the construction of Computational Intelligence. This technology is perfectly applicable in musical composition and computational creativity [21]. They result in the adaptability of individuals to better search regions for generations. The genetic selection, crossing and mutation operators carry out the evolution of the population and repeats until a stop condition is fulfilled [4].

**Selection**: The best parent individuals (P1, P2) are chosen from the initial population by a selection method such as roulette or tournament [22] that will determine their qualification. They seek to improve the quality of descendants of the population [23].

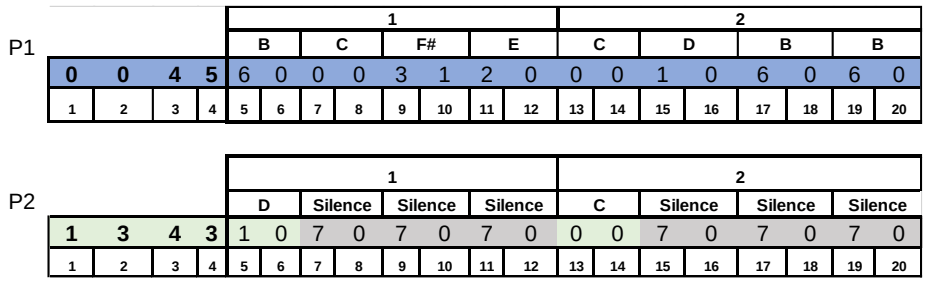



**Fig. 5.** Example of genetic structures to represent two quarters of best two individuals in the population.

**Crossover**: We propose to exchange information by crossing under the following conditions: (1) It is required to implement an elitist operator that does not touch the

configuration of the chromosome, because it would alter the structure producing amorphous individuals. (2) Place the cut points only in even spaces of the chromosome to preserve the note with its alteration. (3) The minimum cutting range must be made after the fourth space and (4) it is advisable to use the cut in two points; in this case the operator would represent improvement [24]. (5) The probability of crossing (Pc) must be high.

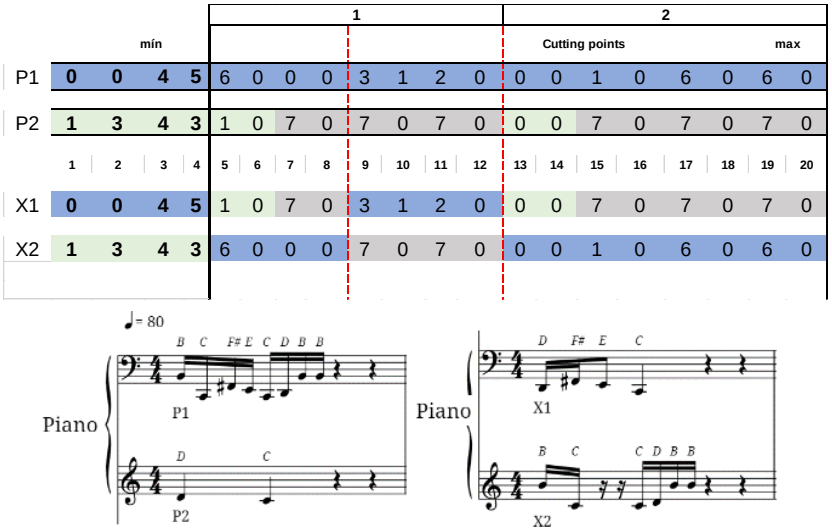



**Fig. 6.** Example of genetic structures (P1 and P2) to represent the product of crossing at two points and its musical staff.

**Mutation:** Occurs in the genes of both descendants after crossing, randomly changes one of the genes of the individual child. The mutability considers the following restrictions: (1) Probability of mutation (*Pm*) in each gene must be very small. (2) Applies only to: *Cliff*, *Octave*, *Music Note* or *Accidental*. (3) The compass and scale are omitted. As in the crossing, it would decompose the genetic structure.

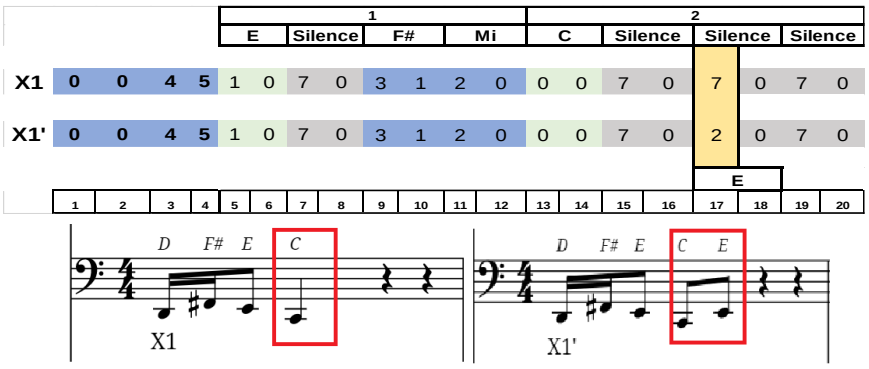



**Fig. 7.** Example of mutation (*X1*) The *Quarter Note* C becomes an *eighth note* (*X1'*).

### 4.1 Generated Individual and Objective Function

A population represents the group of individuals as possible solutions, and with the passage of generations they will manage to evolve until they reach an objective. Each generation must carefully perform the selection of the best individuals to cross them and in very few cases to mutate their elements so as not to stagnate the evolution.

**Generated Individual**: looks like an objective function, and has to calculate how it looks and what its distance function is.

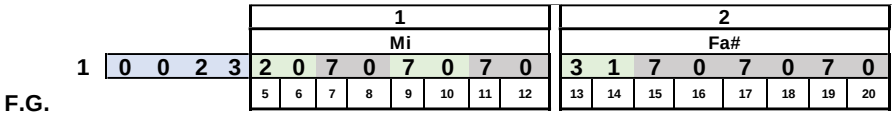



**Fig. 8.** Example of individual generated.

**Objective function:** seeks to reward the adaptability of their individuals towards a desired solution. The initial fragment of two-time melody "Imagine" by composer and musician John Lennon. Figure 9 represents the configuration of the phenotype with the individual coding for a compass.

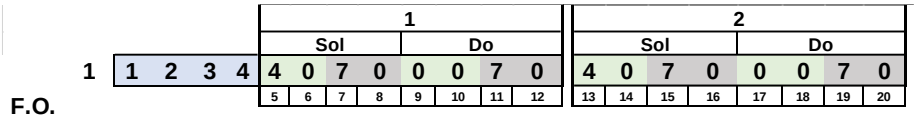



**Fig. 9.** Example of objective individual.

### 4.2 Objective Function Calculation

We propose to apply the distance calculation in: *Octave*, *Scale* and *Musical Note*. The distance calculation procedure for the values of each chromosome is made as follows:

**Calculation of Objective Function in the Octave:** The fundamental values of the notes in each musical octave are taken; these are measured in a fundamental frequency range (Hz) and their corresponding average is calculated by set of tones (Table 1).

The generated individual indicates that his Octave contains a coding [2] (Octave 4) and the target individual contains a value [4] (Pitch 6).

$$d_1(Pitch4, Pitch2) = \mathbf{360 - 90},$$
$$d1\ (360 - 90) = \mathbf{270}.$$

**Calculation of Objective Function in Scale:** We propose adding the coding of each major scale to obtain a real number (Table 2). The generated individual has a coding [3] (Scale 4) and the objective individual contains a value [2] (Scale 3).

$$d_1(Escala4, Escala3) = \mathbf{27 - 22},$$
$$d1\ (27 - 22) = \mathbf{2}$$

**Calculation of Objective Function in Musical Note:** Each musical notes according to the **protocol** of communications *MIDI* (*Musical Instrument Digital Interface*), represents a numerical value (36 - 95), depending on the octave in which it is located. Each octave is a group of twelve notes (Table 3). We proposed that the *silent note* is assigned a high value, otherwise it can be confused with another MIDI note and the same procedure is performed for distance calculation.

**Table 1.** Shows the corresponding average of frequencies by Octave.

| Coding | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Octaves | Pitch 2 | Pitch 3 | Pitch 4 | Pitch 5 | Pitch 6 |
| Frequencies | 65.4064 | 130.813 | 261.626 | 523.26 | 1046.52 |
| | 69.2957 | 138.591 | 277.183 | 554.37 | 1108.73 |
| | 73.4162 | 146.832 | 293.665 | 587.33 | 1174.66 |
| | 77.7817 | 155.563 | 311.127 | 622.26 | 1244.51 |
| | 82.4069 | 164.814 | 330 | 659.26 | 1318.51 |
| | 87 | 174.614 | 349 | 698.46 | 1396.91 |
| | 92.4986 | 184.997 | 370 | 739.99 | 1479.98 |
| | 97.9989 | 195.998 | 391.995 | 783.99 | 1567.98 |
| | 103.826 | 207.652 | 415.305 | 830.61 | 1661.22 |
| | 110 | 220 | 440 | 880.00 | 1760.00 |
| | 116.541 | 233.082 | 466.161 | 932.33 | 1864.66 |
| | 123.471 | 246.942 | 493.883 | 987.80 | 1975.60 |
| Average | 90 | 180 | 360 | 720 | 1440 |

**Table 2.** Shows the proposal of distance corresponding to the calculation of scales.

M2 =

| | | | E1 | E2 | E3 | E4 | E5 | E6 | E7 | ∑ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0 0 | 1 0 | 2 0 | 3 0 | 4 0 | 5 0 | 6 0 | 21 | 21 | 23 | 25 | 27 | 22 | 24 | 26 | |
| 0 | E1 | Do | 0 0 | 1 0 | 2 0 | 3 0 | 4 0 | 5 0 | 6 0 | 21 | 0 | 2 | 4 | 6 | 1 | 3 | 5 | 0 |
| 1 | E2 | Re | 1 0 | 2 0 | 3 1 | 4 0 | 5 0 | 6 0 | 0 1 | 23 | 2 | 0 | 2 | 4 | 1 | 1 | 3 | 1 |
| 2 | E3 | Mi | 2 0 | 3 1 | 4 1 | 5 0 | 6 0 | 0 1 | 1 1 | 25 | 4 | 2 | 0 | 2 | 3 | 1 | 1 | 2 |
| 3 | E4 | Fa | 3 0 | 4 0 | 5 0 | 6 6 | 0 0 | 1 0 | 2 0 | 27 | 6 | 4 | 2 | 0 | 5 | 3 | 1 | 3 |
| 4 | E5 | Sol | 4 0 | 5 0 | 6 0 | 0 0 | 1 0 | 2 0 | 3 1 | 22 | 1 | 1 | 3 | 5 | 0 | 2 | 4 | 4 |
| 5 | E6 | La | 5 0 | 6 0 | 0 1 | 1 0 | 2 0 | 3 1 | 4 1 | 24 | 3 | 1 | 1 | 3 | 2 | 0 | 2 | 5 |
| 6 | E7 | Si | 6 0 | 0 1 | 1 1 | 2 0 | 3 1 | 4 1 | 5 1 | 26 | 5 | 3 | 1 | 1 | 4 | 2 | 0 | 6 |

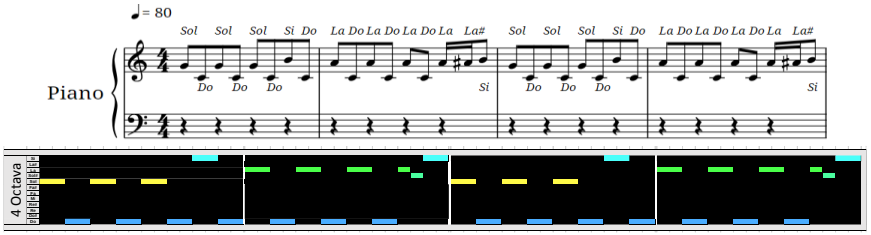
**Table 3.** Contains the MIDI values for this proposal is delimited from octave 2 to 6.

| Octave | C | C# | D | D# | E | F | F# | G | G# | A | A# | B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 1 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |
| 2 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 3 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 4 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 |
| 5 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 |
| 6 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| 7 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 |
| 8 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| 9 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 |

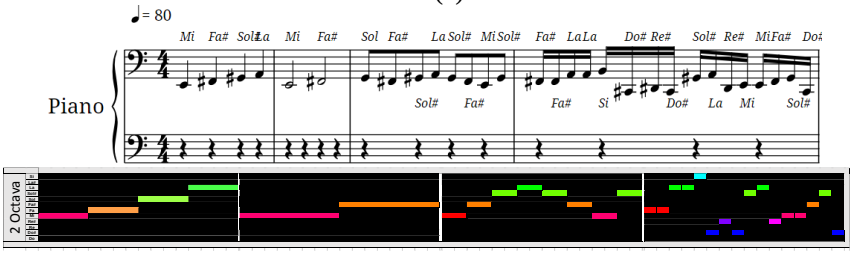
The difference between the objective function (*Fo*, Figure 9) and the individual function (*Fi*, Figure 8) is expressed by the following formula:

$$F(fi,fo) = \sum_{I=1}^{3+33} {}^{m}_{k}(fi,fo),. \quad (1)$$

$$F(fi,fo) = 1 + 270 + 2 + 27 + 140 + 25 + 140 + 23 + 140 + 26 + 140 = 934.$$




(a)




(b)

**Fig. 10.** Visual comparison between objective (a) and the generated individual (b).

The difference between both individuals is remarkable in sound as in color scale; although the calculation focuses only on the first two frames, the objective function shows harmony and the ordered combination of colors. While the generated individual is the opposite. However, with the passing of generations it may be able to adapt to achieve the objective.

## 5 Conclusions and Future Work

Music is a universal language that should not be difficult to understand or interpret, and with the appropriate software, it can be made even easier. The evolutionary algorithms prove to be ideal for the creation of better visual proposals that contribute to the creative process of musical learning. The user is the one who conducts the feedback process of the system, evaluating how much the individual generated is similar to the solution. The best solution generated could sound (or look) slightly different from the objective, but it is part of the purpose to stimulate the creativity process. As future work, we will implement its use with other configurations for the Genetic Algorithm.

**Acknowledgments**. The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP, CIDETEC and SEIS), the Consejo Nacional de Ciencia y Tecnología, and Sistema Nacional de Investigadores for their economical support to develop this work.

## References


1. Herrera, E.: Teoría musical y armonía moderna, 1st ed. Antoni Bosch editor (1995)
2. Ericsson, K. A., Krampe, R. T., Tesch-Römer, C.: The role of deliberate practice in the acquisition of expert performance. Psychol. Rev., vol. 100, no. 3, pp. 363–406 (1993)
3. Goldberg, D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning. MA, USA: Addison-Wesley (1989)
4. Holland, J. H.: Adaptation in Natural and Artificial Systems: An introductory Analysis with Applications to Biology, Control and Artificial Intelligence (1975)
5. Koza, J. R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: MIT Press (1992)
6. Kennedy, J.: Particle swarm optimization. In: Encyclopedia of Machine Learning, New York, USA: Springer, pp. 760–766 (2010)
7. Mitchell, M.: An introduction to genetic algorithms. MIT Press, vol. 32, no. 6, p. 162 (1996)
8. Acevedo, A. G.: Fugue composition with counterpoint melody generation using genetic algorithms. In International Symposium on Computer Music. Berlin, Heidelberg.: Springer, pp. 96–106 (2004)
9. Boyd, M., Butt, J.: J.S. Bach. (1999)
10. Hiller, L., Isaacson, L.: Experimental music: composition with an electronic computer. New York, USA: McGraw-Hill (1959)
11. Diaz-Jerez, G.: Algorithmic music: Using mathematical models in music composition. New York, USA: Manhattan School of Music. (2000)
12. Hsu, K. J., Hsu, A. J., Hspt, A. J.: Fractal Geometry of Music. In: Proc. Natl. Acad. Sci. United States Am. Phys., vol. 87, pp. 938–941 (1990)
13. Munoz, E., Cadenas, J. M., Ong, Y. S., Acampora, G.: Memetic Music Composition. IEEE Trans. Evol. Comput., vol. 20, no. 1, pp. 1–15 (2016)
14. Kumar, C., Dutta, S., Chakraborty, S.: Hiding messages using musical notes: A fuzzy logic approach. Int. J. Secur. its Appl., vol. 9, no. 1, pp. 237–248 (2015)
15. Rodrigues, A., Costa, E., Cardoso, A., Machado, P.: Evolving L-systems with musical notes. In: Lecture Notes in Computer Science, vol. 9596, pp. 186–201 (2016)
16. GarageBand: [Online]. Available: https://www.apple.com/mx/mac/garageband/.
17. GuitarPro: [Online]. Available: https://www.guitar-pro.com/en/index.php.
18. ChordBot: [Online]. Available: http://www.chordbot.com/.
19. TonePad: [Online]. Available: https://appadvice.com/app/tonepad/.
20. Iamus: [Online]. Available: http://www.melomicsrecords.com/.
21. Liu, C.-H., Ting, C.-K.: Computational Intelligence in Music Composition: A Survey. IEEE Trans. Emerg. Top. Comput. Intell., vol. 1, no. 1, pp. 2–15 (2017)
22. Miller, B. L., Goldberg, D. E.: Genetic Algorithms, Tournament Selection, and the Effects of Noise. Complex Syst., vol. 9, no. 3, pp. 193–212 (1995)
23. Gestal, M.: Introducción a los Algoritmos Genéticos. (2010)
24. Moujahid, A., Inza, I.: Algoritmos genéticos. Metod. Mat. en Ciencias la Comput., pp. 1–34 (2008)

ISSN 1870-4069

# A New Experimentation Module for the EPIC Software

Javier A. Hernández-Castaño[1], Yenny Villuendas-Rey[1], Oscar Camacho-Nieto[2], Carmen F. Rey-Benguría[3]

[1] Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Mexico City, Mexico
[2] Instituto Politécnico Nacional, Secretaría de Extensión e Integración Social, Mexico City, Mexico
[3] Universidad de Ciego de Ávila, Centro de Estudios Educacionales "José Martí", Cuba
javierhc92@gmail.com, yenny.villuendas@gmail.com, ocamacho@ipn.mx, carmenrb@sma.unica.cu

**Abstract.** In this paper, we introduce a new experimentation module for the recently developed EPIC software. EPIC is a tool for applying computational intelligence algorithms. The main advantages for our proposal concern the direct handling of mixed and incomplete data, the inclusion of several algorithms within the associative approach, and a very user-friendly graphical interface.

**Keywords:** computational intelligence, experimental tools, supervised classification.

## 1 Introduction

Intelligent Computing (IC) is an important branch of Computer Sciences, which has emerged recently as a scientific discipline [1, 2].

The introduction of IC is justified due to it bringing a computational solution to problems characterized as complex, due to the cost of obtaining the solutions, or due to the inexistence of exact solutions. For the development of new models and algorithms, it is necessary to compare them with respect to existing similar models; and for this task, several researching supporting tools have been developed. Among the most popular tools for supervised classification are WEKA [3, 4] and KEEL [5, 6]. Such tools hasten the researching process, as they include existing algorithms and procedures, and in some cases, they include ways to analyze the quality or performance of the algorithms under study. However, the researchers in the CI community suffer from numerous functionality insufficiencies exhibited in such tools.

The proposal of this research consists in the creation of a new experimentation module for the EPIC software [7]. With the inclusion of this new module, EPIC keeps the main functionalities and characteristics of the existent CI platforms and tools, and includes CI algorithms not considered by any other platform. In addition, the proposed module overcomes some of the deficiencies of the user interface shown by existing tools. EPIC software has a simple yet effective architecture, capable of fulfilling the needs of users, in particular the need of directly handling mixed and incomplete data,

without any data transforming or preprocessing, and the need of handling data belonging simultaneously to several decision attributes (multi-target classification). The main contribution of this paper is to develop a new module for EPIC, with a user interface to develop supervised classification experiments, which is friendlier and has more functionalities than the ones by other existing tools, such as WEKA and KEEL.

The rest of the paper is organized as follows. Section 2 details some of the previous works and Section 3 offers the description of the proposed module. Section 4 presents the discussion on the results obtained. Finally, the paper ends with some conclusions and future research suggestions.

## 2 Previous Works

From the point of view of the supervised classification, there are several tools to perform experiments. Among them, the mostly used are WEKA [3, 4] and KEEL [5, 6]. Both have a user interface, and allow the designing and execution of supervised classification experiments.

However, they have several disadvantages from the user point of view. Considering the above, and carrying out a deep analysis of both tools, we found that some of the drawbacks of WEKA experiment module are:

a. It does not allow the use of dissimilarity functions for mixed data descriptions (it only has distances, to be computed over real data).
b. It arbitrarily handles mixed and incomplete data (the architecture assumes that the feature values of instances are an array of doubles, and it converts the data to fulfill the architecture requirement).
c. It does not include any associative supervised classifier.
d. It does not allow other validation technique apart from Hold-Out and Cross Validation.
e. It does not cancel a single dataset while the experiment is running.
f. It does not serialize the results in a user-friendly way.
g. It does not serialize the results of each partition for each dataset.

On the other hand, although KEEL solves some the above mentioned drawbacks (a, d, g) the supervised classification experiment module of KEEL tool maintains drawbacks (b, c, e, f).

In order to solve the drawbacks 2, the new EPIC software [7] was developed. However, in its first release, it did not have a module for supervised classification experimentation. In this research, we develop such a module.

## 3 Supervised Experiments Module for EPIC

In this research, we decide to begin again, in order to develop an effective solution for the EPIC software, and to overcome the drawbacks shown by both WEKA and KEEL in their supervised experiment modules. We decide to use C# programming language, and the Integrated Development Environment (IDE) *Visual Studio Community* 2017,

due to the facilities they offer to create a tool with a very user-friendly interface, and as it was the language used by EPIC developers. Despite C# not being a multiplatform language, we consider that its use will not represent a difficult, due to the widely extension of Windows operating system in Mexico and the rest of the world.

For the standard supervised classification experiment module, we had the following requirements:

a. The module must include supervised classifiers within the associative approach.
b. The module must include several validation techniques.
c. The user can select the desired validation technique (k-fold cross validation, k-fold stratified cross validation, 5x2 cross validation, 5x2 stratified cross validation or Distribution optimally balanced stratified cross validation).
d. The user can select the desired dataset, either in .ARFF or .Dat format
e. The user can select the desired supervised classifier and to freely configure all of the parameters.
f. The module must serialize the results of each classifier, over each partition of each dataset, including the real and assigned labels for each instance, as well as the corresponding confusion matrix. Such results must be provided in a compatible, friendly way.
g. The module must serialize a summary file, including the average results (of all partitions) of the datasets and classifiers, according to several performance measures, suitable for both balanced and multi-class imbalanced scenarios. Such results must be provided in a compatible, friendly way.
h. The module must allow the user to be able to cancel, at any time, the execution of the experiment only in a desired dataset, without affecting the execution of the experiments in the other datasets.

Considering those requirements, we designed a user interface. In order to update the module with the potential inclusion of other validation techniques and supervised classifiers, we use the *Visual Studio Community* 2017 IDE functionalities of Assembly to create at execution time, all of the related user-interface controls, such as buttons, labels, combo boxes, and so on.

To make an efficient use of computational resources, we program the module using asynchronous threads. It also allows us to cancel the execution of the experiment only in the desired datasets, as well as to show the user the overall progress of the experiment. Figure 1 will show the first user interface of the Standard Supervised Classification Experiment developed. Note, *a* and *b* requirement are fulfilled.

The module (Figure 2) include eight classifiers of the associative approach: HACT [8], CHAT-OHM [9], Gamma [10], Gamma with Differential Evolution (GammaED) [11], NAC [12], NAC with Differential Evolution (NACED) [13], SNDAM [14] and ACID. It also includes the ALVOT classifier [15] from the logical Combinatorial Approach to Pattern Recognition. Neither WEKA nor KEEL includes such classifiers.

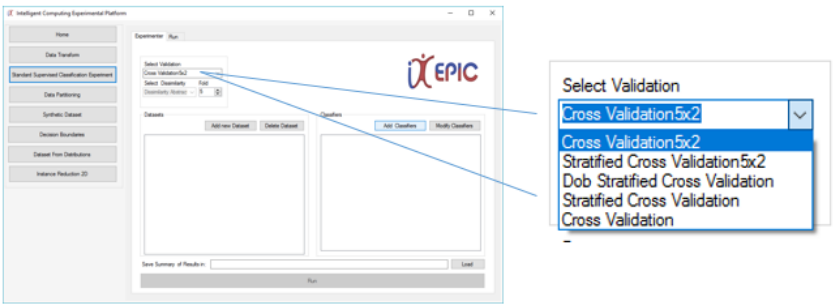



**Fig. 1.** Validation procedures available in the module.

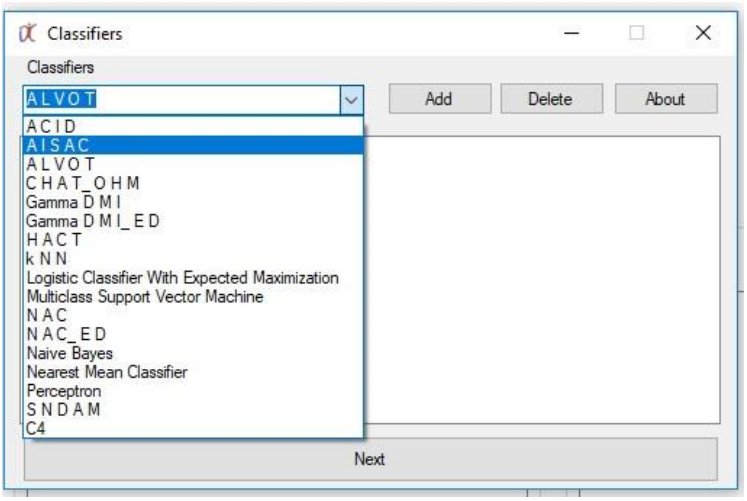



**Fig. 2.** Classifiers available in the module.

To address requirements *c* and *d*, the module automatically checks if the folder contains the files corresponding to the validation technique selected. If not, it shows an error message. Otherwise, it includes the desired dataset. An important advantage of this module, with respect to the ones by WEKA and KEEL, is that it allows including, in the same experiment, datasets in .ARFF format and datasets in .Dat format.

For requirement (*e*), the interface has the user controls according to the data types of the corresponding parameters. For example, if the parameter is a real number, the user interface will create a "Numeric Up Down" component, on the contrary, if the parameter is a dissimilarity function (for instance, for ALVOT classifier) the user-interface will create a "Combo box" component, and will fill this component, at execution time, with all the available classes having the IDissimilarity interface. It is important to highlight that the procedure for creating user-interfaces for parameter configuration is recursive (Figure 3).

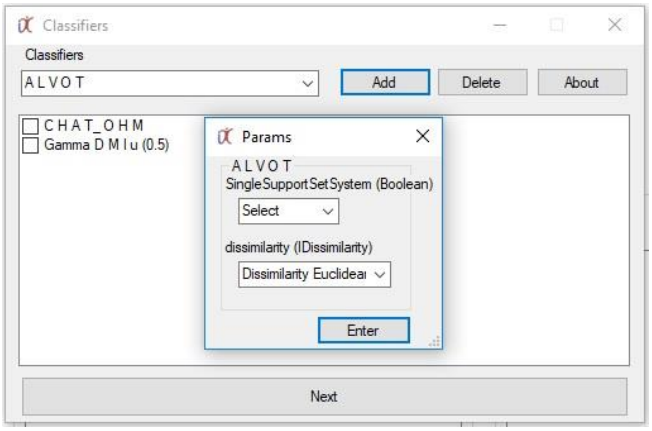
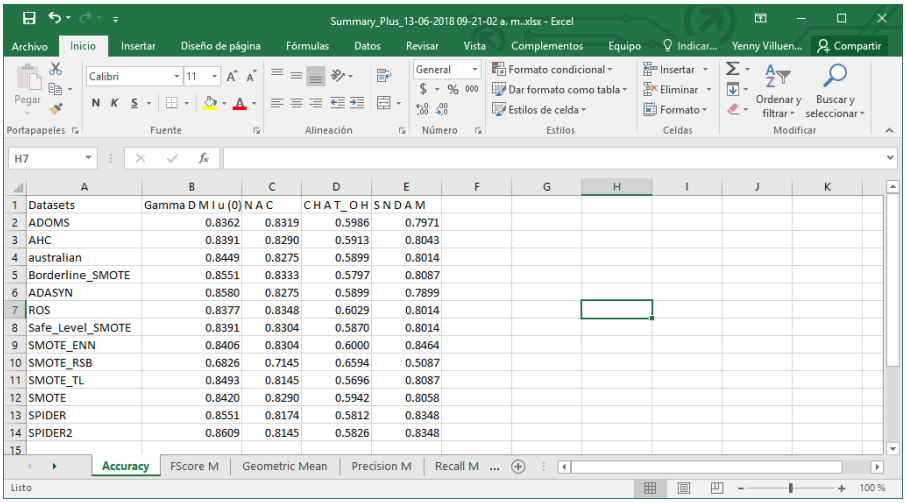



**Fig. 3.** Parameters of the classifiers detected at runtime.

For serializing requirements (*f* and *g*) the module creates a folder named "Results" in the same folder of each dataset. Into such folder, the module saves a file for each classifier. These files are .xlsx files, compatible with Microsoft Excel and Open Office. For each partition, the file has two sheets: one stating real and assigned labels for each test instance, and the other with the corresponding confusion matrix.




| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Datasets | Gamma DMIu (0) | NAC | CHAT_OH | SNDAM |
| 2 | ADOMS | 0.8362 | 0.8319 | 0.5986 | 0.7971 |
| 3 | AHC | 0.8391 | 0.8290 | 0.5913 | 0.8043 |
| 4 | australian | 0.8449 | 0.8275 | 0.5899 | 0.8014 |
| 5 | Borderline_SMOTE | 0.8551 | 0.8333 | 0.5797 | 0.8087 |
| 6 | ADASYN | 0.8580 | 0.8275 | 0.5899 | 0.7899 |
| 7 | ROS | 0.8377 | 0.8348 | 0.6029 | 0.8014 |
| 8 | Safe_Level_SMOTE | 0.8391 | 0.8304 | 0.5870 | 0.8014 |
| 9 | SMOTE_ENN | 0.8406 | 0.8304 | 0.6000 | 0.8464 |
| 10 | SMOTE_RSB | 0.6826 | 0.7145 | 0.6594 | 0.5087 |
| 11 | SMOTE_TL | 0.8493 | 0.8145 | 0.5696 | 0.8087 |
| 12 | SMOTE | 0.8420 | 0.8290 | 0.5942 | 0.8058 |
| 13 | SPIDER | 0.8551 | 0.8174 | 0.5812 | 0.8348 |
| 14 | SPIDER2 | 0.8609 | 0.8145 | 0.5826 | 0.8348 |
| 15 | | | | | |

**Fig. 4.** Results obtained in the Summary of the performance measures.

In addition, at the end of the experiment, the module saves a summary file (Figure 4). This file is also a .xlsx file, and has five sheets, one for each performance measure: accuracy, F-Score M, Geometric Mean, Precision m and Recall M measures. All these measures are given in a tabular form, with datasets in the rows and supervised classifiers in the columns. This summary allows the user to quickly report the results, and to easily include graphics, figures and other Microsoft Excel related elements.

For canceling requirement (*h*) the module has a user interface showing the current progress of the experiment (Figure 5) and it allows canceling a single dataset at any time (Figure 6).

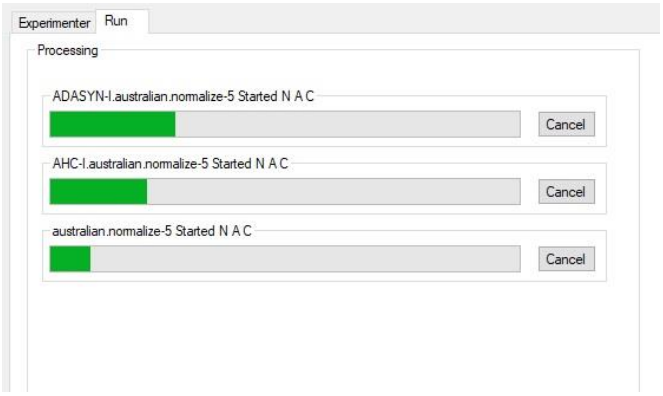



**Fig. 5.** Execution of an experiment.

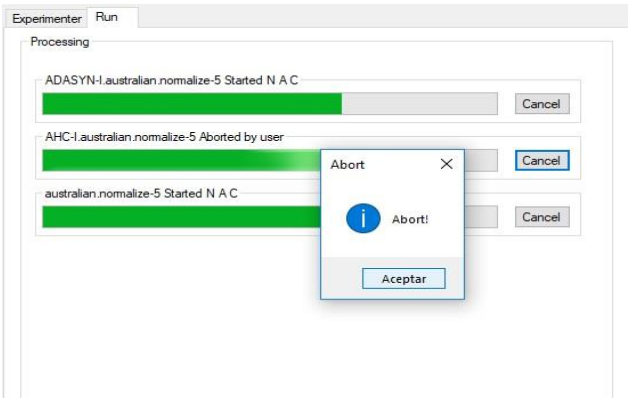



**Fig. 6.** Cancelling a dataset.

This functionality keeps the experiment working without the canceled dataset, and saves time and effort to the users.

## 4 Results and Discussion

In this section, we define several dimensions and indicators in order to evaluate the proposed module for supervised classification experiments. We consider four dimensions, from the user point of view. Each of these dimensions has several indicators of usability. Table 1 shows the dimensions and their corresponding indicators, while Table 2 addresses the corresponding comparison.

We consider four dimensions: Versatility, Execution, Serialization of results and Interpretation of results. All of them have several indicators. We emphasize from the

user point of view, for as researchers, we want a tool easy to use, user-friendly, as well as being intuitive and at the same time robust. For the research process, it is very good to have summaries; have all of the information available, and to store it for further use, as well as to be able to easily navigate for the information.

The most used formats for supervised classification datasets are .ARFF format (designed by the WEKA team) and the .Dat format (designed by the KEEL team). Also it is possible to convert .ARFF into .Dat and vice versa in the KEEL software (not in WEKA). These conversions lead to unnecessary computation and storage costs, particularly for big data. Thus, we consider as an important issue to be able to handle, simultaneously datasets for each format, increasing usability and versatility of the supervised classification experimental modules.

**Table 1.** Dimensions and indicators of module usability.

| *Dimensions* | *Indicators* |
|---|---|
| 1. Versatility | 1.1 Simultaneous handling of both .ARFF and .Dat datasets |
| | 1.2 Sampling procedures for classifier validation |
| | 1.3 Use of specific (serialized) partitions as a hole |
| | 1.4 Evaluation of datasets stored independently of the module |
| 2. Execution | 2.1 Automatic parametrization of classifiers |
| | 2.2 Cancelation of a desired dataset |
| | 2.3 Visualization of the experiment progress |
| 3. Serialization of results | 3.1 Serialization of the classification results for each classifier on each partition (real vs assigned labels) |
| | 3.2 Serialization of the confusion matrix for each classifier on each partition |
| | 3.3 Organization of the serialized information |
| 4. Interpretation of results | 4.1 Automatic calculation of average (by partition) performance measures |
| | 4.2 Performance measures |
| | 4.3 Intuitive interpretation of results |
| | 4.4 Existence of global (classifiers vs datasets) summaries of performance |

Another important issue for supervised classification experiments is the validation (or data partition) procedure. There are several well-known procedures, such as Leave One Out, Hold Out and Cross Validation. For the last two, there are stratified versions, able to divide the data considering class distribution. In literature, stratification is a "must be" for experimentation, with special relevance over imbalanced data.

In addition, due to the computational complexity of several algorithms, and the high volume of some datasets, the possibility of cancelling the execution of the experiment only in the desired datasets, without losing all the experiment is very important. Neither WEKA nor KEEL offers such functionality to the user. In both cases, if the user wants to cancel the execution of the experiment in a single dataset, it must cancel the entire

experiment, and must start over, losing all the computations made so far. On the other hand, the proposed module for EPIC software, allows the user to cancel a desired dataset, and the experiments continues executing in the remaining data. In addition, the final performance summaries are not affected by the cancellation. We believe that this is a huge improvement for researchers and students who use the tools for experimentation.

**Table 2.** Evaluation of the proposed module with respect to others.

| *Indicator* | *WEKA* | *KEEL* | *EPIC* |
|---|---|---|---|
| **1 Versatility** | | | |
| 1.1 Simultaneous handling of both .ARFF and .Dat datasets | No | No | Yes |
| 1.2 Sampling procedures for classifier validation | 2* | 3† | 5‡ |
| 1.3 Use of specific (serialized) partitions as a hole | No | Yes | Yes |
| 1.4 Evaluation of datasets stored independently of the module | Yes | No | Yes |
| **2 Execution** | | | |
| 2.1 Automatic parametrization of classifiers | Yes | Yes | Yes |
| 2.2 Cancelation of a desired dataset | No | No | No |
| 2.3 Visualization of the experiment progress | Partial | No | Yes |
| **3 Serialization of results** | | | |
| 3.1 Serialization of the classification results for each classifier on each partition (real vs assigned labels) | No | Yes | Yes |
| 3.2 Serialization of the confusion matrix for each classifier on each partition | No | Yes | Yes |
| 3.3 Organization of the serialized information | Bad | Bad | Good |
| **4 Interpretation of results** | | | |
| 4.1 Automatic calculation of average (by partition) performance measures | No | No | Yes |
| 4.2 Performance Measures (by partition) | 49§ | 2** | 5†† |
| 4.3 Intuitive interpretation of results | No | No | Yes |
| 4.4 Existence of global (classifiers vs datasets) summaries of performance | No | No | Yes |

Another important issue is to store the real and assigned labels for a dataset, and the corresponding confusion matrix. Such information is very useful for data analysis, and

* Hold-Out and k-fold Cross Validation

† 5x2 Cross Validation, k-fold Cross Validation and k-fold Distribution Optimally Balanced Stratified Cross Validation

‡ 5x2 Cross Validation, 5x2 Stratified Cross Validation, k-fold Cross Validation, k-fold Stratified Cross Validation and k-fold Distribution Optimally Balanced Stratified Cross Validation

§ In the authors' opinion, most of them are useless. In addition, WEKA computes several measures (such as Area under ROC Curve) under circumstances where it cannot be computed (for instance in multi-class datasets).

** Accuracy and Area under ROC Curve (the last for 2-classs datasets only, as it should be)

†† Accuracy, F Score M, Geometric Mean of Recall, Precision M and Recall M

having a confusion matrix allows the researcher to compute almost all desired performance measures, due to they are based on this matrix.

In addition, for analysis and reports, researcher need summarized information (for instance, the average results over the k-folds for each classifier over each dataset) of the desired performance measures. If the information is not summarized, there is a significant lack of time and effort to obtain such summaries. WEKA does not provide summarized files of results, just a single, big file with the information of the entire experiments. KEEL does provide the summaries, in an unstructured plain text file. For instance, to obtain a bar graphic of the summaries, it is necessary to copy the information, to format it, and then to organized into other program, such as Microsoft Excel, Numbers or similar. The proposed module gives the user a single .xlsx file, with five sheets, having the summary for the corresponding performance measure. From the user point of view, this is an important advance, gaining time and having the desired information without effort. Another aspect to highlight is that the information is well organized, intuitive, easy to use, and explain.

## 5 Conclusions and Future Work

This paper introduces a new module for standard experimentation on supervised classification, for the EPIC software. The proposal has several advantages with respect to the state- of- art, and includes significant contributions from the user point of view. In the future we will be working on other modules, for weighted supervised classification, as well as for data preprocessing.

**Acknowledgments**. The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, Comisión de Operación y Fomento de Actividades Académicas, Secretaría de Investigación y Posgrado, CIDETEC and SEIS), the Consejo Nacional de Ciencia y Tecnología, and Sistema Nacional de Investigadores for their economical support to develop this work.

## References

1. Teti, R., Kumara, S.: Intelligent computing methods for manufacturing systems. Cirp Annals 46, pp. 629–652 (1997)
2. Mandal, J.K., Paramartha, D., Mukhopadhyay, S.: Advances in Intelligent Computing. Springer (2019)
3. Holmes, G., Donkin, A., Witten, I.H.: Weka: A machine learning workbench. In: Intelligent Information Systems. Proceedings of ANZIIS '94, pp. 357–361. IEEE (1994)
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD explorations newsletter 11, pp. 10–18 (2009)
5. Alcalá-Fdez, J., Sanchez, L., Garcia, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M.: KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Computing 13, pp. 307–318 (2009)
6. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: data set repository, integration of algorithms and

experimental analysis framework. Journal of Multiple-Valued Logic & Soft Computing 17, (2011)
7. Hernández-Castaño, J.A., Camacho-Nieto, O., Villuendas-Rey, Y., Yáñez Márquez, C.: Experimental Platform for Intelligent Computing (EPIC). Computación y Sistemas 22, pp. 245–253 (2018)
8. Santiago-Montero, R.: Clasificador Híbrido de Patrones basado en la Lernmatrix de Steinbuch y el Linear Associator de Anderson-Kohonen. Centro de Investigación en Computación. Master Thesis. Instituto Politécnico Nacional. Mexico (2003)
9. Uriarte-Arcia, A.V., López-Yáñez, I., Yáñez-Márquez, C.: One-hot vector hybrid associative classifier for medical data classification. PloS one 9, e95715 (2014)
10. López-Yáñez, I., Sheremetov, L., Yáñez-Márquez, C.: A novel associative model for time series data mining. Pattern Recognition Letters 41, pp. 23–33 (2014)
11. Ramirez, A., Lopez, I., Villuendas, Y., Yanez, C.: Evolutive improvement of parameters in an associative classifier. IEEE Latin America Transactions 13, pp. 1550–1555 (2015)
12. Villuendas-Rey, Y., Rey-Benguría, C.F., Ferreira-Santiago, Á., Camacho-Nieto, O., Yáñez-Márquez, C.: The naïve associative classifier (NAC): a novel, simple, transparent, and accurate classification model evaluated on financial data. Neurocomputing 265, pp. 105–115 (2017)
13. Serrano-Silva, Y.O., Villuendas-Rey, Y., Yáñez-Márquez, C.: Automatic feature weighting for improving financial Decision Support Systems. Decision Support Systems 107, pp. 78–87 (2018)
14. Ramírez-Rubio, R., Aldape-Pérez, M., Yáñez-Márquez, C., López-Yáñez, I., Camacho-Nieto, O.: Pattern classification using smallest normalized difference associative memory. Pattern Recognition Letters 93, pp. 104–112 (2017)
15. Ruiz-Shulcloper, J., Ponce, E., López, N.: ALVOT, system of programs of voting algorithms for classification. Revista Ciencias Matemáticas (In Spanish) 7, pp. 41–67 (1986)

ISSN 1870-4069

# Implementación del protocolo DALI en FPGAs de bajo consumo de energía para uso en redes inalámbricas de sensores

Oscar Osvaldo Ordaz-García[1, 2,*], Manuel Ortiz-López[1],
Francisco Javier Quiles-Latorre[1], José Guadalupe Arceo Olague[2],
Francisco José Bellido-Outeiriño[1]

[1] Universidad de Córdoba, Depto. Ingeniería Electrónica y de Computadores, España
oscarord27@hotmail.com, {el1orlom, el1qulaf, fjbellido}@uco.es
[2] Universidad Autónoma de Zacatecas, Unidad Académica Ingeniería Eléctrica, México
arceojg@uaz.edu.mx

**Resumen.** Este trabajo expone el diseño, descripción e implementación del hardware requerido por el protocolo *DALI* (*Digital Addressable Lighting Interface*) usado en la comunicación de nodos en redes inalámbricas de sensores para control de luminarias. La descripción se realiza en el lenguaje descriptivo de hardware (*VHDL*) para ser embebido sobre *FPGAs* (*Field Programmable Gate Array*), con el objetivo de que funcione sobre un nodo inalámbrico basado en el estándar *IEEE* 802.15.4., con codificación *Manchester* diferencial *bi-phase*. El funcionamiento se comprobó con simulación y para contrastarlo se obtuvo una arquitectura de referencia implementada en cuatro plataformas de desarrollo con *FPGAs*, demostrando su correcta ejecución. La implementación del diseño en *FPGAs* de bajo consumo energético y económico, hace viable la disminución del consumo de energía generado en la comunicación, y realizar un prototipo de un nodo inalámbrico para su uso en una red inalámbrica de sensores.

**Palabras clave:** protocolo *DALI*, *FPGA*, redes inalámbricas de sensores, *VHDL*, codificación *Manchester* diferencial.

# Implementation of the DALI Protocol in FPGAs of Low Energy Consumption for use in Wireless Sensor Networks

**Abstract**. This work exposes the design, description and implementation of the hardware required by the DALI (Digital Addressable Lighting Interface) protocol used in the communication of nodes in wireless sensor networks for luminaire control. The description is made in the hardware descriptive language (VHDL) to be embedded on FPGAs (Field Programmable Gate Array), with the aim of operating on a wireless node based on the IEEE 802.15.4 standard, with Manchester differential bi-phase coding. The operation was verified with simulation and to contrast it, a reference architecture was obtained, implemented in four development platforms with FPGAs, demonstrating its correct execution. The implementation of the design in FPGAs of low energy and economic

consumption, makes viable the decrease of the energy consumption generated in the communication, and make a prototype of a wireless node for use in a wireless sensor network.

**Keywords**: DALI protocol, FPGA, wireless sensor networks, VHDL, Manchester differential coding.

## 1. Introducción

Los sistemas de comunicación para transmitir datos para la administración de la iluminación pública consumen gran cantidad de energía eléctrica. Además, estos sistemas dependen en gran medida de la regulación de la tensión y del mantenimiento [1], y en muchas ocasiones esto implica un mayor costo [2], en contraste con nuevos sistemas de ahorro de energía, como la iluminación basada en leds [3]. Una estrategia para reducir el consumo de energía es el protocolo de datos *DALI*, protocolo de comunicación desarrollado por empresas de equipamiento de iluminación, para asegurar interoperabilidad entre diferentes fabricantes, funcionalidad, simplicidad, ahorro de energía y bajos costos [4].

Aunado el protocolo *DALI*, con el uso de dispositivos lógicos programables para el control de la comunicación en sistemas de iluminación se puede reducir el alto consumo de energía que este genera. Un *FPGA* es un circuito integrado diseñado para que su interconexión y funcionalidad sea configurada por un lenguaje de descripción de hardware (*HDL*) [5]. VHDL es un lenguaje de descripción de hardware de VHSIC, utilizado para describir circuitos digitales y automatizar diseños electrónicos en el cual la programación se puede ejecutar en paralelo [6].

El objetivo de este trabajo es describir el protocolo de datos *DALI* en el lenguaje descriptivo de hardware *VHDL* para su funcionamiento en un nodo inalámbrico basado en el estándar *IEEE* 802.15.4, permitiendo realizar una arquitectura de referencia para implementarla en *FPGAs* de bajo consumo energético y económico.

En el documento se comentan los antecedentes y trabajos previos; posteriormente se explican los materiales y métodos del procesamiento y representación de la descripción de la interfaz; enseguida los resultados y finalmente las conclusiones.

## 2. Antecedentes y trabajos previos

En investigaciones referentes a redes inalámbricas de sensores, se menciona que el mayor consumo energético se debe a las comunicaciones; además, en muchas ocasiones esta energía se malgasta al mantener un nodo encendido a la espera de recibir algún mensaje (un módulo de comunicaciones típico de redes de sensores gasta la misma energía en modo emisor que en modo receptor) [7], por lo que Rosello *et al*. [7] propone una arquitectura de procesamiento para comunicaciones bajo demanda en *FPGA* de bajo consumo, evaluando la utilización de dispositivos de lógica programable para realizar el procesamiento de mensajes. Cárdenas *et al*. [8] metodológicamente describen a nivel cualitativo los protocolos de enrutamiento basados en la estructura de red jerárquica para la búsqueda de un equilibrio, donde los protocolos pretenden

mejorar los algoritmos de funcionamiento, procesamiento y transmisión de información. En [9] se diseña e implementa un servidor *Web* empotrado en un *FPGA* para proporcionar una interfaz remota y monitorizar una red inalámbrica de sensores.

Se han realizado trabajos para obtener mediciones de varios parámetros con dispositivos programables, como en [10] donde se identifica la necesidad de mejorar la sensibilidad, estabilidad y presentación de datos. De forma semejante, en [11] se utiliza un *FPGA* para resolver problemas de adquisición y presentación de datos; sin embargo, no los resuelven del todo, ya que se encuentran diferentes problemas de conversión, procesamiento y presentación. Se emplea en [12] un lenguaje de descripción de hardware de alto nivel para ser usado en placas de desarrollo de la familia *Zynq* de *Xilinx,* con el objetivo de diseñar e implementar un *MCA (Multi-Channel Analyzer)* de 4096 canales para espectrometría nuclear, obteniendo datos de un generador de funciones.

Hoy en día, antes de fabricar en silicio algún procesador, se realiza un riguroso análisis, diseño, implementación y pruebas en un dispositivo lógico reconfigurable; los *FPGAs* son utilizados para estos procesos [13]. Diferentes ejemplos se muestran en trabajos realizados en [14, 15, 16, 17]. Como se ha constatado, diversas investigaciones tratan de mejorar la obtención de datos sensibles a una magnitud y otras, su comunicación a través de dispositivos electrónicos, por lo que el uso de *FPGAs* para este fin y para el que se pretende en este trabajo es viable.

El desarrollo de redes inalámbricas de sensores implica el estudio de varios aspectos, como lo relacionado a plataformas de hardware que soporten sensores y comunicación, y lo referente al software. Este trabajo se enmarca en referencia al hardware, para lo cual se necesitan los materiales descritos en la siguiente sección.

## 3. Materiales y métodos

En el proyecto se describió en *VHDL*, el protocolo de datos *DALI* para el control de luminarias, con una estructura *master – slave*. La topología de bus está basada en un canal de comunicación serie de dos vías, donde hay al menos un controlador *master* y generalmente múltiples *slaves*. El *master* envía paquetes de datos de 16 bits y recibe respuestas con paquetes de datos de 8 bits en codificación *Manchester* diferencial *bi-phase*. Donde "01" corresponde a '1' y "10" corresponde a '0'. Se usan dos símbolos por cada bit de información, el *bit rate* está especificado en 1200 *bps* con un tiempo de error de ±10 %, el tiempo de bit es 833,33 $\mu s$, y la frecuencia a la cual funciona el canal es de 2400 *Hz*. El bit más significativo (*MSB*) se envía primero, como se ilustra en la Fig. 1. La sincronización indica que el tiempo medio por bit (*Te*) es de 416.67 $\mu s$, por lo que un paquete de envío tiene una duración de 38 *Te*, que es igual a 15.83 *ms*. Un *Backward frame* toma 22 *Te* o 9.17 *mseg* [18].

En terminología *DALI*, un *Forward Frame* es un paquete enviado del dispositivo de control al equipo receptor; y un *Backward Frame* es un paquete de respuesta enviado del equipo receptor al dispositivo de control [18]. El *Forward Frame* consta de un bit de inicio, ocho bits de dirección, ocho bits de datos y dos bits de alto, como se muestra en la Fig. 2, donde: *S* es el bit de inicio (1 lógico); *YAAA AAAS* es el *Byte* de dirección;

*XXXX XXXX* es el *Byte* de datos; y | | son dos bits de alto (línea inactiva) [19]. El *Backward Frame* consta de un bit de inicio, ocho bits de datos y dos bits de alto. Donde: *S* es el bit de inicio (1 lógico); *XXXX XXXX* es el Byte de datos; y | | son dos bits de alto (línea inactiva), como se observa en la Fig. 3 [18].

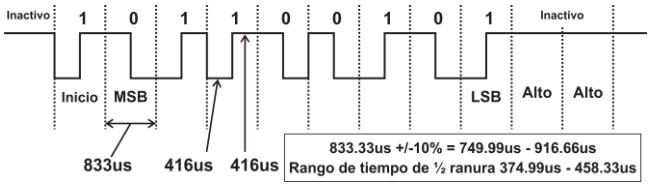



**Fig. 1.** Transmisión de *DALI.*

Como la transferencia de información es en codificación *Manchester* diferencial *bi-phase*, es necesario la decodificación para determinar la dirección y entonces procesar el mensaje. El código es un formato de codificación digital en el cual el símbolo '1' está representado por una transición ascendente (estado en bajo seguido de un flanco a alto) y el símbolo '0' está representado por un flanco descendente (estado en alto seguido de un flanco a bajo). Ambos pulsos, altos y bajos, tienen un periodo de tiempo igual a la mitad del período de bit.

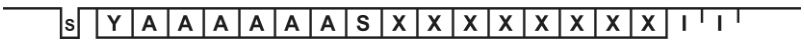



**Fig. 2.** *Forward Frame.*

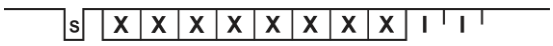



**Fig. 3.** *Backward Frame.*

Para implementar una arquitectura en un *FPGA* se diseña y describe su funcionamiento mediante el uso de diferentes herramientas, como diagramas esquemáticos, lenguajes descriptivos de hardware, o una combinación de los dos. Para describir el nodo de transmisión/recepción *DALI* se utilizó el lenguaje VHDL 93 estándar. El diseño, los bloques y su interconexión se muestran en la Fig. 4. La entidad principal nombrada *Rx_Tx_dali.vhdl* consta de dos bloques (*Transmisor_Manchester.vhdl y Receptor_Manchestre.vhdl*). Depende de seis entradas (*Clk, Reset, WR_Manch, Dato_in_Manch(15:0), RXD_Manch y Rx_ena_Manch*) y entrega cuatro salidas (*Ready_trans_Manch, TXD_Manch, Dato_rec_Manch y Dato_sal_Manch(7:0)*).

Con el objetivo de realizar el proceso de transmisión de datos, se desarrolló la entidad *Transmisor_Manchester.vhdl*, la misma consiste en una máquina de estados finitos (*FSM* por *finite state machine*) que se encarga de las fases necesarias para la transmisión, que es la sincronización de la frecuencia en *bps* del nodo de transmisión, la coordinación de la ejecución del protocolo con un codificador/decodificador *Manchester* y tiene un registro de datos paralelo/serie para el envío de bits.

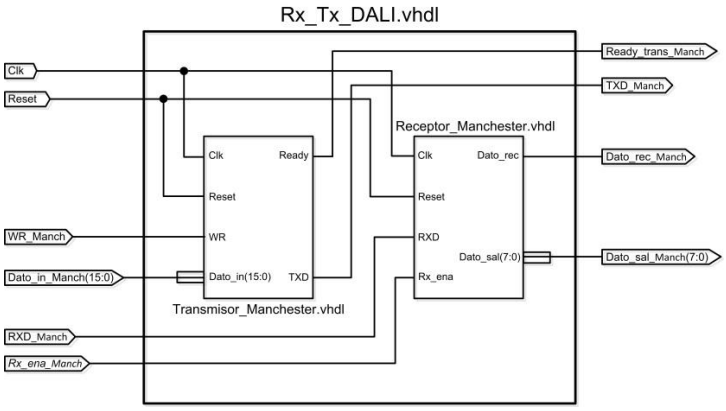



**Fig. 4.** Diagrama de la Descripción del Transmisor/Receptor para el protocolo DALI.

La entidad *Receptor_Manchestre.vhdl*, es descrita con una *FSM*; efectúa la recepción de datos por medio del puerto de entrada *RXD_Manch*, los 8 bit se reciben de forma secuencial e internamente con un registro se realiza la conversión de datos serie/paralelo para que el valor sea mostrado en el puerto de salida *Dato_sal_Manch(7:0)*, la terminal de salida *Dato_rec_Manch* indica el fin del proceso de conversión. La entidad, también sincroniza la velocidad para el dato de recepción que es de 1200 *bps*, realizando el muestreo del punto más preciso de *RXD_Manch* para obtener un valor válido del bit recibido, esto mediante la espera de un cuarto de bit una vez detectado el bit de "*inicio*" (*Rx_ena_Manch*) y después se espera un tiempo de bit de, que es el inverso de la velocidad de recepción y así detectar el valor del bit recibido de forma precisa, esto se realiza con un registro para determinar el instante de muestreo y un registro para contar el número de bit recibidos; además la entidad coordina la ejecución del protocolo con un codificador/decodificador *Manchester*.

## 4. Implementación del protocolo en *FPGAs*

Después de diseñar y describir la interfaz *DALI*, se realizó la simulación en el software de desarrollo *ISim* de *Xilinx ISE*. La simulación de la Fig. 5 comprueba que el diseño para la transmisión y la recepción de datos funciona adecuadamente. Para el proceso de transmisión, cuando la señal *wr* es activada (*wr* = 1) se cargan en un registro los datos que se desean enviar, en este ejemplo "1010101010101010". En seguida se codifican los datos para enviar el bit de inicio del protocolo *Manchester*. Después, se envían uno a uno los bits, tal como se observa en la señal de salida *txd*. En la fase de recepción, es necesaria la activación de *rx_ena*, para comenzar a recibir la serie de bits, que se visualizan en la señal *rxd*. Para la simulación, tanto el envío como la recepción son de 16 bits, dado que el puerto de salida *TXD* de la entidad *Transmisor_Manchester.vhdl*, es instanciado al puerto de entrada *RXD* de la entidad *Receptor_Manchestre.vhdl*, esto con la finalidad de que el dato de envío sea el mismo que se recibe, para comprobar que teóricamente existe sincronía en ambas entidades.

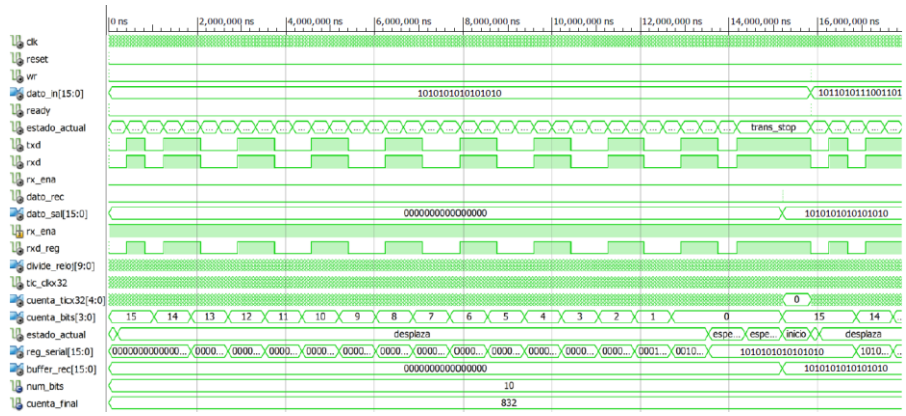



**Fig. 5.** Simulación de la Transmisión/Recepción de la Codificación *Manchester* Diferencial.

La descripción se implementó en cuatro diferentes placas de desarrollo, para comprobar su funcionamiento en cualquier *FPGA*, con reconfiguración respecto a la frecuencia del oscilador de cada placa de desarrollo. En la descripción se incluye el uso de "*generics"* para simplificar la reconfiguración de la implementación en las diferentes plataformas y adicionalmente hacer más compacto el diseño. Cabe mencionar que el proposito del presente trabajo es el desarrollo del hardware requerido para la transmision/recepción de los datos, dejando de lado el aspecto de corrección de errores, el cual está a cargo de un bloque de corrección de errores.

Una de las plataformas de computación reconfigurable en la que se implementó la descripción fue la *UCOS3E5*, que se desarrolló por académicos-investigadores de la Universidad de Córdoba, España, con propósitos educacionales y como plataforma de prototipado rápido en trabajos de investigación. La placa *UCOS3E5* contiene una *FPGA XC3S500E8* (*PQ208*) de la familia *Spartan-3E* de *Xilinx* semiconductor [19], un oscilador a 32 *MHz*, y varios componentes más.

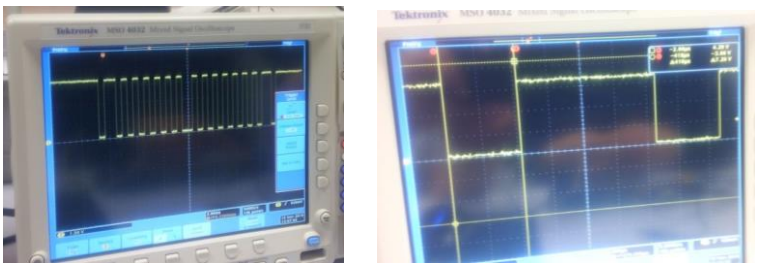


**Fig. 6.** Transmisión/Recepción del protocolo *DALI* implementado en la plataforma *UCOS3E5*.

Para la implementación de la descripción de la interfaz DALI en la plataforma *UCOS3E5*, los datos a transmitir se ingresaron con los *switch's*. Como los bits a transmitir son 16 y la placa solamente cuenta con 8 *switch's*, la descripción se reconfiguró para que al momento de hacer la carga al "*buffer*" de datos, los 16 bits correspondan al complemento de los *switch's* concatenado con el dato original de los

*switch's*. Los datos ingresados fueron "11111110", por lo tanto, los datos que se transmitieron fueron: "0000000111111110". En la Fig. 6 se puede observar la transmisión codificada en *Manchester* diferencial del dato mencionado. Se analizó la sincronización del envío y recepción de los datos, ya que como se indica de manera teórica, el tiempo medio por bit (*Te*) debe ser de 416.67 $\mu s$, y con la información obtenida por el osciloscopio, en la implementación es de 416 $\mu s$.

Otra plataforma para implementar la descripción del interfaz *DALI*, fue la *Basys 2* de *Digilent*, que contiene una *FPGA XC3S100E* (*CP132*) de la familia *Spartan-3E* de *Xilinx* [20]. Incluye un oscilador de silicio configurable, a frecuencias de 25 *MHz*, 50 *MHz* o 100 *MHz*, y más componentes que permiten implementar diferentes diseños, entidades y sistemas digitales.

La descripción se reconfiguró para hacer la implementación para enviar los 16 bits. Se usaron los 8 *switch's* para cargar los datos a enviar. Ahora la configuración de los datos se realizó con el dato de los *switch's* concatenado con su complemento. El valor de los *switch's* fue "10001001" por lo que la transmisión se generó para la siguiente cadena de bits "1000100101110110". En la Fig. 7 se corrobora el dato en codificación *Manchester* diferencial.

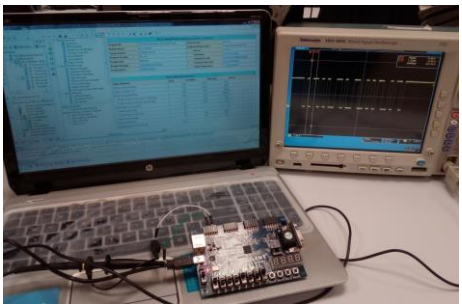


**Fig. 7.** Implementación del protocolo *DALI* en la plataforma *Basys 2 de Digilent*.

El tiempo de bit en teoría debe ser de 833,33 $\mu s$, y a través de una aproximación por medio de las líneas verticales de medición del osciloscopio en las transiciones de la codificación *Manchester*, se puede ver que el resultado obtenido es de 840 $\mu s$, valor aceptable dentro del espacio de tiempo para error que es del ±10 %. Todo esto se visualiza en la Fig. 8.

Al hacer una medición ajustada a los flancos, se puede observar en la Fig. 9 que el tiempo medio por bit (*Te*) es de 418 $\mu s$, y en comparación con el ideal, se determinó que solamente se tiene un 0.32 % de error.

En las plataformas de desarrollo *iCEblink40-HX1K* [21] y *iCEblink40-LP1K* [22] *Evaluation Kit* de *Lattice Semiconductor* se realizaron dos implementaciones más. Ambas placas tienen un oscilador de precisión (*LT1799*) con tres diferentes frecuencias configurables de 3.33 *MHz*, 333 *KHz* y 33.3 *MHz*.

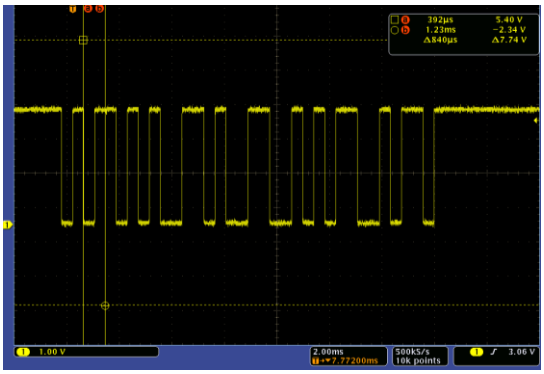


**Fig. 8.** Resultados de la transmisión del protocolo *DALI* en la plataforma *Basys 2* de *Digilent*.

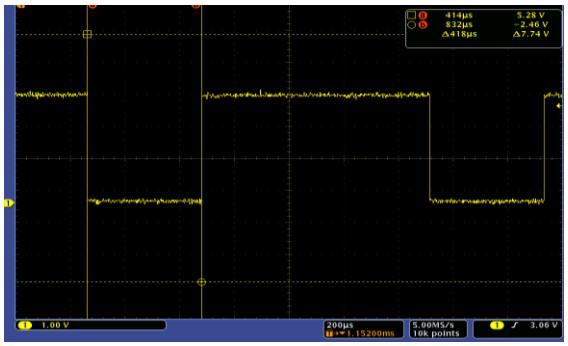


**Fig. 9.** Medición del tiempo medio por bit, del dato de recepción (*Basys 2* de *Digilent*).

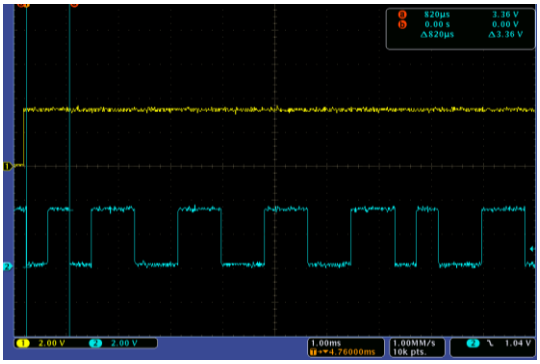


**Fig. 10.** Secuencia del protocolo *DALI* implementado en la plataforma *iCEblink40-HX1K*.

Para las implementaciones en las placas de *Lattice Semiconductor*, se utilizó un registro para el envío de los datos con un valor constante. En la Fig. 10 se muestra un fragmento de los datos enviados en codificación *Manchester* diferencial. Después de la transición del bit de inicio, se pueden observar transiciones referentes al dato con valor de "10101010010". Lo relativo a la sincronización se puede observar en un

acercamiento aproximado con las líneas de medición verticales del osciloscopio un tiempo por bit de 820 $\mu s$ en la transición inicial. Este valor, se encuentra por mucho dentro del porcentaje de error permitido.

## 5. Conclusiones

Se obtuvo la descripción en el lenguaje descriptivo de hardware *VHDL* de la interfaz *DALI* para transmisión y recepción de paquetes de datos en codificación *Manchester* diferencial; la cual puede funcionar sobre un nodo inalámbrico basado en el estándar *IEEE* 802.15.4. El objetivo principal se cumplió, ya que se realizó una arquitectura de referencia reconfigurable que se implementó en varias plataformas de desarrollo, en las cuales como dispositivo principal es un *FPGA*. Los resultados de la implementación muestran que el diseño y descripción cumplen con los requerimientos necesarios para la transmisión/recepción.

En particular, la implementación de la descripción en las plataformas de desarrollo de *Lattice Semiconductor* que tienen *FPGAs* de la familia *iCE40* que son de ultra bajo consumo energético, fabricadas con tecnología *CMOS* estándar de baja potencia de 40 *nm* y con altas prestaciones de desempeño, permite reducir el consumo de energía que generan los procesos de comunicación inalámbrica en los dispositivos que actualmente soportan el protocolo *DALI*, y entonces realizar el prototipo de un nodo inalámbrico o mote para su uso en una red inalámbrica de sensores asociado al bajo costo económico de éstas permitiría aumentar la cobertura de la red.

La cantidad de casos de prueba realizados permitió perfeccionar a detalle el diseño y la descripción para realizar las correcciones pertinentes. La variación de los valores obtenidos para la sincronización en el tiempo por bit para la transmisión/recepción con relación a los valores ideales es mínima y en algunos casos se puede considerar despreciable. El usar *VHDL* 93 estándar permitió sintetizar la descripción en los *FPGAs* sin la necesidad de realizar cambios en la descripción (código) y, en consecuencia, garantizar la implementación en *FPGAs* de bajo consumo de energía.

Como trabajo futuro se pretende realizar una etapa donde el dato de transmisión pueda ser administrado por medio de un dispositivo de comunicación universal que controla puertos y dispositivos, y con el mismo conocer el dato de recepción.

## 6. Recomendaciones

Para continuar con el desarrollo del nodo inalámbrico, es necesario abordar temas de optimización de recursos de hardware, con el objetivo de disminuir la cantidad de hardware utilizado y con ello el consumo de energía eléctrica que tienen los nodos. Como se ha plasmado, también es necesario un sistema de administración para la arquitectura de referencia implementada en este trabajo, el cual puede ser a través de dispositivo de comunicación universal controlado por un programa para conexión de dispositivos, por lo que es necesario conocimientos en temas de comunicación y luminarias. Un punto clave para realizar el prototipo del nodo es la codificación del *conjunto* de instrucciones del protocolo *DALI* para el control de luminarias.

## Referencias


1. Bellido-Outeiriño, F.J., Quiles-Latorre, F.J., Moreno-Moreno, C.D., Flores-Arias, J.M., Moreno-García, I., Ortiz-López, M.: Streetlight Control System Based on Wireless Communication over DALI Protocol. Sensors 16, 597 (2016)
2. Quintero, J. R., Prieto, L. F., Sistemas inteligentes de transporte y nuevas tecnologías en el control y administración, pp. 53–62 (2015)
3. Domínguez, L. I., Ordaz, O. O., Arceo, J. G., Hernández, V. M., Solís, R.: Prototipo de un sistema para el control de tráfico vehicular y peatonal implementado en un FPGA. IEEE Sec Morelos. XIV CIINDET 2018, pp. 130–134 (2018)
4. DALI Standard. International Electrotechnical Commission IEC 62386.
5. Wisniewski, R.: Synthesis of compositional microprogram control units for programmable devices. Zielona Góra: University of Zielona Góra. p. 153 (2009)
6. Department of Defense: Military Standard, Standard general requirements for electronic equipment. (1992)
7. Rosello, V., Portilla, J., Riesgo, T.: Arquitectura de Radios Wake-up para redes de sensores inalámbricas basada en FPGA. SAAEI 2013, Madrid, Spain (2013)
8. Cárdenas, J.: Protocolos de Enrutamiento basados en la Estructura de Red Jerárquica hacia la Eficiencia Energética en Redes de Sensores Inalámbricos. Ciencia, Innovación y Tecnología 20162 (1), pp 37–54
9. Rodríguez Valido, M., Gutiérrez Castañeda, M., Cardell Bilbao, A., Ayala Alfonso, A., Díaz Gopar, J. J., Sobota Rodriguez, C., Magdaleno Castelló, E.: Web server empotrado en FPGA para monitorización de una red de sensores inalámbricos. En: IX Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica (2010)
10. Solis, R., et al.: Beneficios y Desafíos en el uso de una Red Inalámbrica de Sensores para el Monitoreo de una Red de Distribución de Agua Potable. En: Memorias del CONCYE 2011, pp. 21–23 (2011)
11. García Duran, A., Ordaz García, O., Arceo Olague, J. G., Hernández, C.: Adquisición de datos de sensores de una red de distribución de agua potable. C. Internacional de I Academia Journals, V. 7, No. 3, pp. 255–259 (2015)
12. García Durán, A., Hernández Dávila, V. M., Vega Carrillo, H. R., Ordaz García, O. O., Bravo Muñoz, I.: Analizador Multicanal embebido en FPGA. En: XVII International Symposium on Solid State Dosimetry ISSSD 2017 Dominican Republic (2017)
13. Ordaz, O. O., Hernández, M., Benavides, J. I., Arceo, J. G.: Desarrollo del CORE de un Procesador de Imágenes de tipo SIMD embebido en un FPGA. IEEE Internacional Sección Centro Occidente ROPEC, pp. 463–468 (2012)
14. Ordaz, O. O., Rico Sabag, A. A., Arceo, J G., González Carrillo, L. J.: Implementación de un Procesador Elemental en un FPGA. En: ENINVIE 2010, Zacatecas, Zac., pp. 76–83 (2010)
15. Ordaz, O. O., Hernández, M., Benavides, J. I., Arceo, J. G.: Eficiencia del uso de recursos en un FPGA para la descripción de un Procesador Elemental. En: XII Reunión Internacional de Otoño de Potencia, Electrónica y Computación 10 (2010)
16. Nava, D., Ordaz, O., Hernández, M.: Desarrollo de ensamblador para procesador de imágenes tipo SIMD. En: C. Internacional de I. - Academia Journals, V. 5, No. 3, T. 16, pp. 2364–2369 (2013)
17. Ordaz, O. O., Hernández, M., Benavides, J. I., Arceo, J. G.: Diseño de la Unidad Elemental de un Microprocesador en un FPGA. CONCyE 2011, pp 57–63. Zacatecas, México (2011)
18. Husain, S.: Digitally Addressable Lighting Interface (DALI) Communication. AN1465. Microchip Technology Inc. (2012)
19. Xilinx: Spartan-3 FPGA Data Sheet.

20. Digilent: BASYS 2 Reference Manual (2010)
21. User's Guide iCEblink40-HX1K Evaluation Kit Lattice Semiconductor.
22. User's Guide iCEblink40-LP1K Evaluation Kit Lattice Semiconductor.

ISSN 1870-4069

# Módulos embebidos en micro-tecnología FPGA de modelo estocástico de primer orden

Karen Alicia Aguilar Cruz, Romeo Urbieta Parrazales,
José Antonio Flores Escobar, Midory Esmeralda Vigueras Velázquez,
José de Jesús Medel Juárez

Instituto Politécnico Nacional, Centro de Investigación en Computación,
Ciudad de México, México

Karen_ali320@hotmail.com,
rurbieta700@gmail.com,jafloresescobar@hotmail.com,
midory.vigueras@gmail.com, jjmedelj@gmail.com

**Resumen.** En este artículo se describe un filtro digital diseñado en etapas y embebido en arreglo de compuertas de campo (FPGA) basado en el segundo momento de probabilidad y en el modelo autorregresivo de medias móviles (ARMA (1,1)) de primer orden; ambos en sus formas recursivas y no recursivas. Los modelos se describen como módulos a bloques para representar las variables de estimación e identificación, así como las variables de medición del proceso a la entrada y salida del sistema tipo caja negra, el cual representa a un proceso de energía eólica. El filtro estocástico es programado en lenguaje de descripción de hardware Verilog®, brindando una implementación en estimación digital. Para la verificación de su estructura se obtuvo el acoplamiento de los módulos considerados en el filtrado estocástico de primer orden.

**Palabras clave:** filtro digital estocástico, estimación, identificación, ARMA.

# Embedded Modules in Micro-technology FPGA of the First Order Stochastic Model

**Abstract**. This article describes a digital filter designed in stages and embedded in field gate arrays (FPGA) based on the second probability moment and in the autoregressive model of moving averages (ARMA (1,1)) of first order; both in their recursive and non-recursive forms. The models are described as block modules to represent the estimation and identification variables, as well as the process measurement variables at the input and output of the black box system, which represents a wind energy process. The stochastic filter is programmed in Verilog® hardware description language, providing an implementation in digital estimation. For the verification of its structure, the coupling of the modules considered in the first-order stochastic filtering was obtained.

**Keywords**: stochastic digital filter, estimation, identification, ARMA.

## 1. Introducción

La problemática que se encuentra en el diseño de estimadores digitales es encontrar el parámetro de innovación dados los datos medidos de un sistema de caja negra (BB, por sus siglas en inglés Black Box) de primer orden para determinar los estados estimados mediante algún criterio [1]. Lo anterior requiere la descripción de tres modelos estocásticos, descritos a continuación.

Primero, el modelo que representa al proceso de estudio a través de un modelo lineal no variante en el tiempo (LTI, por sus siglas en inglés Linear Time-Invariant) de primer orden, conteniendo la descripción de los estados internos a priori del proceso mediante el modelo lineal $\tilde{x}_{k+1} = a\tilde{x}_k + b\tilde{w}_k$ , donde, $k = 0,1,2,\dots,n$ representa los $n$ muestreos, $a$ y $b$ son constantes tales que $a$ y $b \in \Re$, y $\tilde{w}_k$ es ruido blanco con media y variancia definidas, $\tilde{w}_k \in N(\mu \approx 0, \sigma^2 < \infty)$ [2]. En esta notación, la tilde sobre las variables significa que son aleatorias. Si al modelo LTI se le hace pasar por un retardo se logra obtener el estado lineal $\tilde{x}_k$.

Segundo, el modelo del sensor lineal, que viene ubicado a la salida del modelo del proceso lineal para leer los estados internos $\tilde{x}_k$, caracterizado por una serie de mediciones o datos digitales de salida, representadas por $\tilde{y}_k = c\tilde{x}_k + d\tilde{v}_k$, donde $k = 0,1,2,\dots.n$, y $\tilde{v}_k$ , ruido de salida con características de media y varianza definidas.

*Tercero*, el modelo del estimador estocástico lineal, el cual tiene como entrada las mediciones efectuadas por el sensor $\tilde{y}_k$, y como función el criterio de estimación empleado para determinar el parámetro estimado, o parámetro de innovación, $\hat{a}_k$ y los estados estimados deseados $\hat{x}_k$. La notación ( $\widehat{\ }$ ) significa que los parámetros y variables son estimados.

De los criterios existentes en la teoría estocástica [3] para la estimación de parámetros, se considera el segundo momento de probabilidad estocástica. Este criterio se basa en los espacios de Hilbert $<\tilde{y}_k, \tilde{y}_{k-1}>$ usando los operadores de la probabilidad estocástica: media ($E\{\tilde{y}_k\}=0$), varianza ($\text{var}\{\tilde{y}_k\} = \sigma^2$) y covarianza ($\text{cov}\{\tilde{y}_k, \tilde{y}_{k+t}\} = 0$).

Por otra parte, para la elección de los sistemas digitales embebidos, ya sea en una sola pastilla o chip, para aplicaciones en filtros, control automático o procesamiento de señales, el análisis de las características eléctricas, mecánicas, y de programación juega un papel importante en la búsqueda de sistemas que sean eficientes, confiables e incluso de innovación para el diseño. Por ejemplo, dentro de las características eléctricas se encuentran la capacidad de memoria, la velocidad de procesamiento, el número de puertos de entrada y salida (E/S) y el número de bloques de programación; mientras que en las características mecánicas se encuentran las dimensiones, la forma, el material empleado, el acabado, etc. Para el caso de la programación, se verifica que cuenten con un conjunto de instrucciones acorde a la solución del problema, así como facilidad para borrar y grabar información.

En este trabajo, para la implementación del filtro, se seleccionó el sistema modular digital de Arreglos de Compuertas Programables en Campo (FPGA, por sus siglas en inglés Field-Programmable Gate Array), cuyas características eléctricas incluyen una alta velocidad de procesamiento, la cantidad de puertos de E/S y de bloques funcionales de programación, así como memoria y robustez a perturbaciones del ambiente [4-6].

# 2. Método

## 2.1. Modelo digital ARMA estocástico

El modelo digital ARMA estocástico se expresa mediante dos ecuaciones lineales definidas en (1) y (2). La ecuación (1) de incremento de estados $x_{k+1}$, incluye una suma de productos que representan al parámetro $a$ que se desea conocer, multiplicado por el estado interno $\tilde{x}_k$ y, al sistema de excitación $\widetilde{w}_k$ multiplicado por un factor constante $b$. La variable medida por el sensor $\tilde{y}_k$, definida en (2) corresponde al producto del parámetro $c$ y el estado interno correspondiente $\tilde{x}_k$, más una perturbación aleatoria $\tilde{v}_k$ de salida del sistema, por una constante $d$. En este caso, el sistema de estudio es considerado como un modelo BB caracterizado por los dos ruidos $\widetilde{w}_k$ y $\tilde{v}_k$:

$$\tilde{x}_{k+1} = a.* \tilde{x}_k + b.* \widetilde{w}_k, \tag{1}$$

$$\tilde{y}_k = c.* \tilde{x}_k + d.* \tilde{v}_k\,. \tag{2}$$

Observando las ecuaciones (1) y (2) se ven cuatro operaciones de multiplicación y dos de sumas. Usando estas dos expresiones matemáticas y simbología de bloques de circuitos digitales, se puede construir el modelo digital ARMA mostrado en la Fig. 1, seguido de su desarrollo en modelo digital en lenguaje de descripción de hardware Verilog®, Fig. 2. En este modelo, se observan los parámetros de medición y las variables de estados, considerando que los ruidos E/S cuentan con una distribución gaussiana.

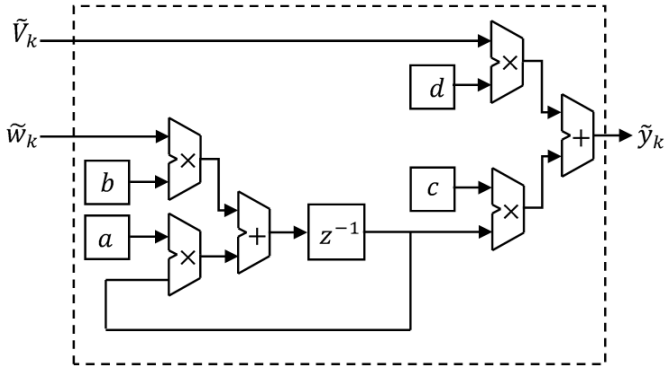



**Fig. 1.** Módulo en bloques del modelo digital ARMA estocástico.

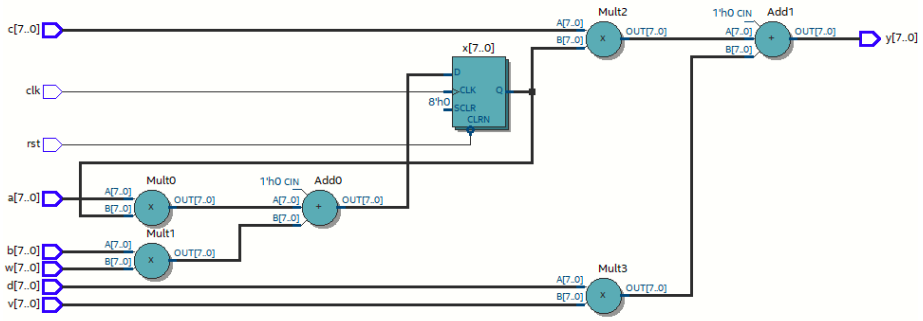



**Fig. 2.** Representación del modelo ARMA estocástico en un módulo digital Verilog®.

## 2.2. Modelo digital recursivo estocástico

La estimación recursiva consiste en la estimación secuencial del modelo ARMA para distintos tamaños muestrales. Dependiendo del valor de la muestra "$k$" que se emplee, es posible determinar la convergencia o estabilidad del modelo. Si el valor de la muestra es grande entonces hay cambio radical en la estimación del parámetro y por consecuencia los estados se pueden salir de entre los valores *cero* y *uno*, produciendo de esta manera una mala convergencia e inestabilidad. La idea es entonces que si no hay cambio estructural se espera que las estimaciones de los parámetros se mantengan constantes al ir aumentando la muestra en forma secuencial. A continuación, en las ecuaciones (3) y (4) se presenta el modelo recursivo estocástico a partir del modelo ARMA, incluyendo la definición de ruidos, y su representación digital:

$$\tilde{\tilde{y}}_k = a.* \tilde{y}_{k-1} + \tilde{\tilde{V}}_k, \tag{3}$$

$$\tilde{V}_k = -a.* d.* \tilde{v}_{k-1} + c.* b.* \tilde{w}_{k-1} + d * \tilde{v}_k, \tag{4}$$

donde $\tilde{\tilde{y}}_k$ es salida recursiva y $\tilde{\tilde{V}}_k$ es la mezcla de *ruidos blancos*. En la Fig. 3 se muestra el módulo digital de estas variables y su módulo digital de descripción de hardware en la Fig. 4.

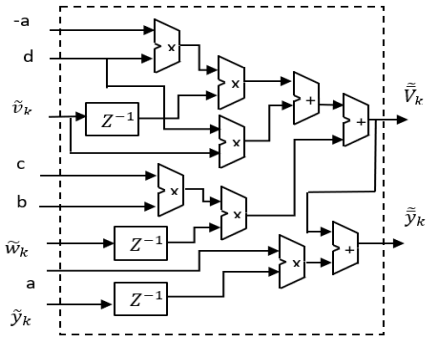



**Fig. 3.** Módulo en bloques del modelo digital recursivo estocástico.

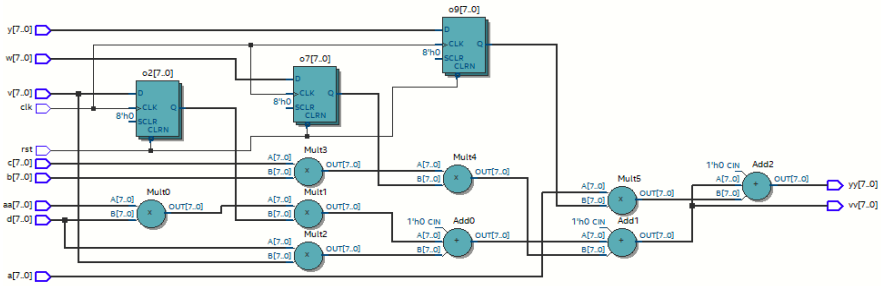



**Fig. 4.** Módulo digital Verilog® del modelo digital recursivo estocástico $\tilde{\tilde{y}}_k$.

## 2.3. Modelo digital recursivo de la varianza $\tilde{p}_k$

La varianza [7] $\tilde{p}_k$ del modelo estocástico, descrita en la ecuación (5), se presenta como una función de los muestreos digitales k, y dos sumandos. El primer sumando es una multiplicación de salidas $\tilde{\tilde{y}}_k \tilde{y}_{k-1}$ y el segundo una convolución entre los k muestreos y la propia función $\tilde{p}_{k-1}$ retardadas por un periodo de muestreo:

$$\tilde{p}_k = \frac{1}{k} \sum_1^n \tilde{\tilde{y}}_k \tilde{y}_{k-1} = \frac{1}{k} \left( \tilde{\tilde{y}}_k \tilde{y}_{k-1} + (k-1)\tilde{p}_{k-1} \right). \quad (5)$$

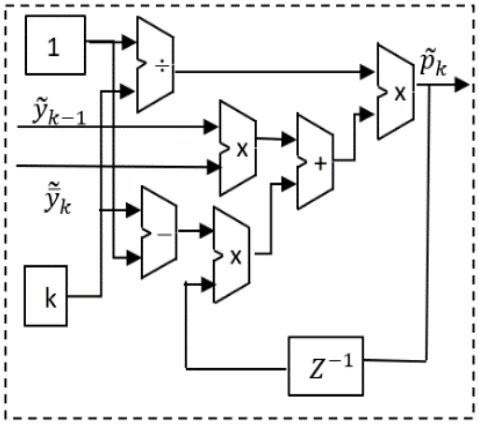



**Fig. 5.** Material del Modelo digital de la Varianza estocástica $\tilde{p}_k$.

En las Figs. 5 y 6 se muestran tanto el sistema digital en bloques que puede determinar la varianza $\tilde{p}_k$, así como el diagrama de implementación en Verilog®, tal como se efectuó en el primer módulo.

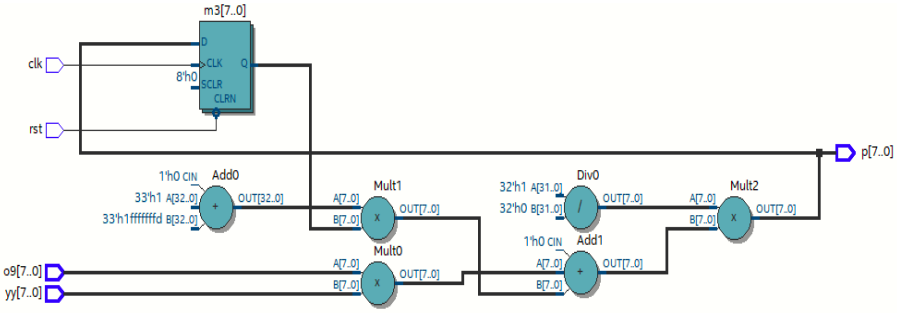



**Fig. 6.** Material del Módulo digital Verilog5® del modelo digital estocástico $\tilde{p}_k$.

## 2.4. Modelo digital de la covarianza $\tilde{q}_k$

La ecuación (6) muestra el término que permite obtener la covarianza [7] $\tilde{q}_k$ expresada en dos sumatorias que representan dos secuencias. La primera con la salida misma y la segunda con la respuesta y ruido de salida con un retardo.

$$\tilde{q}_k = \frac{1}{k}\left(\sum_1^n \tilde{y}_{i-1}^2 - d\sum_1^n \tilde{v}_{i-1}\tilde{y}_{i-1}\right). \tag{6}$$

Desarrollando las sumatorias se llega a (7):

$$\tilde{q}_k = \frac{1}{k}\left(\tilde{y}_{k-1}^2 - d * \tilde{v}_{k-1}\tilde{y}_{k-1} + (k-1)\tilde{q}_{k-1}\right). \tag{7}$$

La representación de la obtención de la covarianza se muestra en diagrama a bloques en la Fig. 7 y de forma implementada en Verilog®, en la Fig. 8.

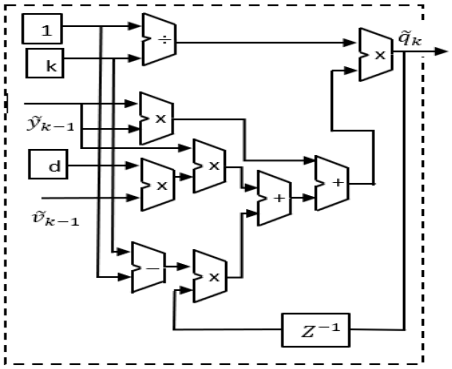



**Fig. 7.** Modelo digital en bloques de la covariancia $\tilde{q}_k$.

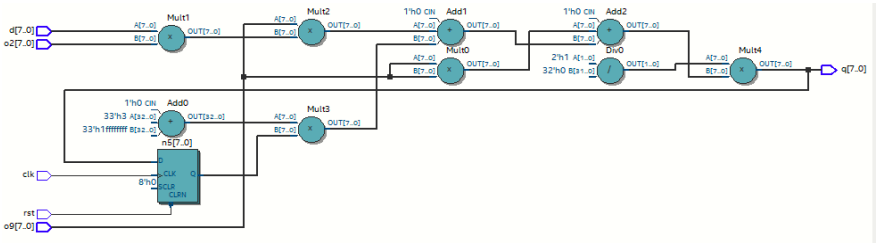



**Fig. 8.** Módulo en Verilog® del modelo de la Covarianza $\tilde{q}_k$.

## 3. Resultados

### 3.1. Modelo digital de estimación $\tilde{a}_k$

El modelo de estimación estocástica viene representado por el segundo momento de probabilidad estocástica, donde la ecuación (3) se convoluciona con $\tilde{y}_{k-1}$, i.e. $\xi\{\tilde{\tilde{y}}_k\tilde{y}_{k-1}\}$, obteniendo el parámetro estimado $\tilde{a}_k$ como en la ecuación (8).

$$\tilde{a}_k = \frac{\xi\{\tilde{\tilde{y}}_k\tilde{y}_{k-1}\}}{\xi\{\tilde{y}_{k-1}^2 - d*\tilde{v}_{k-1}\tilde{y}_{k-1}\}}. \tag{8}$$

Desarrollando (8) se obtiene el parámetro estimado (9), descrito como el cociente entre la varianza y la covarianza.

$$\tilde{a}_k = \frac{\tilde{p}_k}{\tilde{q}_k} = \frac{\left(\tilde{\tilde{y}}_k\tilde{y}_{k-1} + (k-1)\tilde{p}_{k-1}\right)}{\tilde{y}_{k-1}^2 - d*\tilde{v}_{k-1}\tilde{y}_{k-1} + (k-1)\tilde{q}_{k-1}}, \tag{9}$$

donde $\tilde{a}_k \in \mathfrak{R}\{0,1\}$ es el parámetro digital estimado en función de k muestreos {1,2, …, n}, y $\tilde{p}_k$, $\tilde{q}_k \in \mathfrak{R}$ {0,1} son la varianza y covarianza del modelo estocástico bajo el método del segundo momento.

Finalmente, en la Fig. 9 se tiene la integración de los módulos descritos anteriormente y en la Fig. 10, se aprecia el sistema de filtro digital estocástico que se grabará en el sistema digital de desarrollo basado en tecnología FPGA.

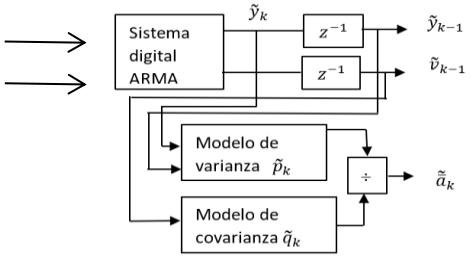



**Fig. 9.** Modelo digital de filtro digital de estimación e identificación basado en el segundo momento de probabilidad estocástica.

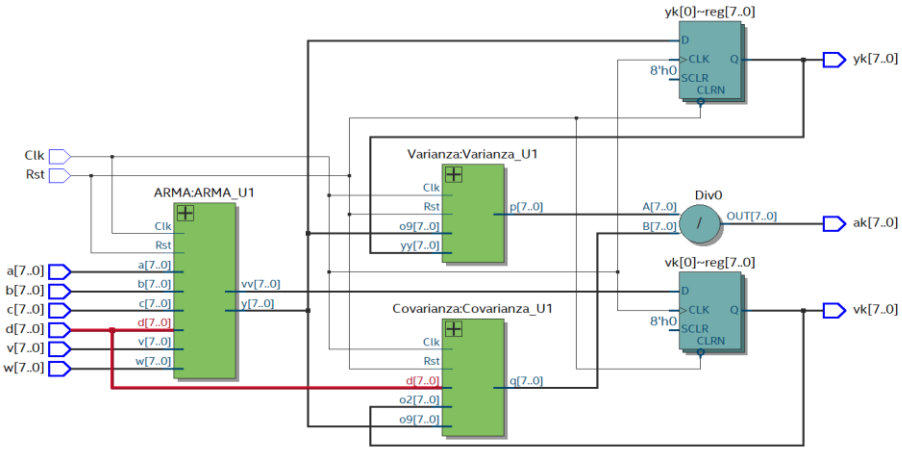



**Fig. 10.** Material del módulo Verilog® del modelo digital de estimación e identificación basado en el segundo momento de probabilidad estocástica.

## 4. Conclusiones

El presente trabajo presentó el método de estimación de parámetros e identificación de las variables de salida de un sistema, basado en el segundo momento de probabilidad estocástica, utilizando una notación discreta para el modelo digital estocástico que emula al modelo de primer orden de caja negra y a los parámetros de datos obtenidos en las mediciones de su entrada y salida.

El modelo estocástico para la caja negra se determinó a través de un modelo ARMA y, mezclando las ecuaciones de las variables que contienen los ruidos de $w_k, v_k$ de E/S se pudo llegar a un modelo ARMA recursivo de la variable de identificación. Esto con la idea de implementarlo en un sistema digital como es sistema de desarrollo FPGA de Altera. Mientras que para la descripción de la estimación se hizo uso de los conceptos de esperanza matemática, varianza y covarianza.

Como resultado, el parámetro de innovación $\tilde{\tilde{a}}_k$ estocástica resultó en un cociente entre la varianza y covarianza $P_k, Q_k,$ respectivamente. El resultado fue la implementación del sistema de filtrado estocástico e identificación en un sistema digital usando FPGA de Altera, brindando una herramienta que contiene los bloques de estimación basados en operadores estocásticos matemáticos y que es de gran utilidad en aplicaciones prácticas de filtrado lineal estocástico.

**Agradecimientos**. Los autores agradecen al Centro de Investigación en Computación del Instituto Politécnico Nacional (IPN) y al Consejo Nacional de Ciencia y Tecnología (CONACyT), por su apoyo en el desarrollo de este trabajo, a través de los proyectos SIP20180762 y SIP20181910.

## Referencias

1. Mendel, J. M.: Lessons in estimation theory for signal processing, communications, and control. Pearson Education (1995)
2. Medel Juárez, J. J., García Infante, J. C., Urbieta Parrazales, R.: Identificador con comparación entre dos estimadores. Revista Mexicana de Física, 57(5), pp. 414–420 (2011)
3. Kumar, P. R., Varaiya, P.: Stochastic systems: Estimation, identification, and adaptive control (Vol. 75). SIAM (2015)
4. Xia, D. Z., Hu, Y. W., Kong, L.: Adaptive Kalman filtering based on higher-order statistical analysis for digitalized silicon microgyroscope. Measurement, 75, pp. 244–254 (2015)
5. Sass, R., Schmidt, A. G.: Embedded systems design with platform FPGAs: principles and practices. Morgan Kaufmann (2010)
6. Gazzano, J. D. D., Crespo, M. L., Cicuttin, A., Calle, F. R.: Field-Programmable Gate Array (FPGA) Technologies for High Performance Instrumentation. IGI Global (2016)
7. Evans, M. J., Rosenthal, J. S.: Probabilidad y estadística. Reverté (2005)

ISSN 1870-4069

# Construcción de nano-dosímetro para el control automático de la dosificación

Midory Esmeralda Vigueras Velázquez, Romeo Urbieta Parrazales, Karen Alicia Aguilar Cruz, José de Jesús Medel Juárez

Instituto Politécnico Nacional, Centro de Investigación en Computación, Ciudad de México, México
midory.vigueras@gmail.com, rurbieta700@gmail.com, karen_ali320@hotmail.com, jjmedelj@gmail

**Resumen.** Un aspecto clave en la optimización del diseño de un sistema es la selección de los materiales, sensores y métodos que mejor satisfacen las necesidades de diseño, asegurando el máximo rendimiento y el mínimo costo. El enfoque del control automático para la dosificación de líquidos presentado en el este trabajo, consiste de dos fases de desarrollo: a) el diseño y simulación del sistema de control inteligente y, b) la implementación del prototipo del sistema de control inteligente; este último se llevó a cabo con flujo de líquidos de baja viscosidad contenido desde un recipiente cisterna (60 ml) a otro de menor capacidad (20 ml) usando componentes de micro tecnologías avanzadas como son los Sistemas-Micro-Electro-Mecánicos (MEM). El artículo servirá como una guía de referencia y recurso útil para quienes participan en el diseño y fabricación de microsistemas para la dosificación de líquidos mediante actuadores y dispositivos MEM. Cabe mencionar que el diseño en dosificadores de líquidos de baja viscosidad tiene diversas aplicaciones, tanto en la industria automotriz, en dispositivos de electrónica de consumo, en medicina y tecnologías relacionadas con la salud, etc., donde la calidad, precisión, confiabilidad, tamaño, capacidades micrométricas de la mezcla son de suma importancia para el consumidor.

**Palabras clave:** control inteligente de líquidos, MEM, Arduino, Raspberry Pi 3.

# Construction of Nano-dosimeter for the Automatic Control of Dosing

**Abstract**. A key aspect in optimizing the design of a system is the selection of materials, sensors and methods that best meet design needs, ensuring maximum performance and minimum cost. The automatic control approach for the dosing of liquids presented in this paper, consists of two phases of development: a) the design and simulation of the intelligent control system and, b) the implementation of the prototype of the intelligent control system; The latter was carried out with a flow of low viscosity liquids from a cistern (60 ml) to a smaller one (20 ml) using advanced micro-technology components such as Micro-Electro-

Mechanical Systems (MEM). The article will serve as a reference guide and useful resource for those involved in the design and manufacture of microsystems for the dosing of liquids by means of actuators and MEM devices. It should be mentioned that the design of low viscosity liquid dispensers has various applications, both in the automotive industry, in consumer electronics devices, in medicine and health-related technologies, etc., where quality, precision, reliability, size, micrometric capabilities of the mix are of utmost importance to the consumer.

**Keywords**: intelligent liquid control, MEM, Arduino, Raspberry Pi 3.

## 1. Introducción

Hoy en día el controlador de fluidos de baja viscosidad tiene gran influencia y aplicación en diversos campos industriales, sobre todo en la industria militar. La física de fluidos interactúa directa con el aire o con el agua, e indirectamente a través de muchos sistemas fluídicos que ellos incorporan. El objetivo del trabajo es mejorar la eficiencia o bien el desempeño a través de calcular los parámetros del sistema y verificarlos en la simulación, así como embeber el sistema de Control Inteligente en un dispositivo digital, con la intención de controlar el flujo, para atrasar o avanzar la transición, suprimir, mejorar la turbulencia, prevenir, o promover la separación. Como resultado se observa la reducción de resistencia, mejora la elevación, aumento de mezcla, mejora la transferencia de calor y supresión de los ruidos inducidos, etc. [1,2,3].

En la mayoría de los casos de fluidos se trabaja en la capa delgada del fluido, de escasos milímetros de grosor, conocida como capa límite, que se forma entre los componentes del vehículo y el fluido circundante (véase la Fig. 1). La posibilidad de crear tecnología a estas dimensiones ha llevado a los pioneros de MEM a desarrollar componentes que se usan en los sistemas de medición y control automático, como son los actuadores y sensores. Cabe mencionar que los objetivos del protocolo del proyecto se organizan de la siguiente forma: diseño, simulación, e implementación para diversas aplicaciones en el campo industrial, habitacional, aérea, médica, etc., usando nuevas tendencias en MEM, que existen en el mercado mundial. Para ello, a continuación, se discuten los diferentes conocimientos teóricos y experimentales [4,5].

*Flujo de capa límite*. A continuación, se presenta en la Fig. 1 el flujo de fluidos y sus diferentes capas, donde se puede apreciar que una ***capa límite*** consta de tres subcapas importantes: una subcapa *viscosa o laminar*, una subcapa de *transición*, y una subcapa *turbulenta*.

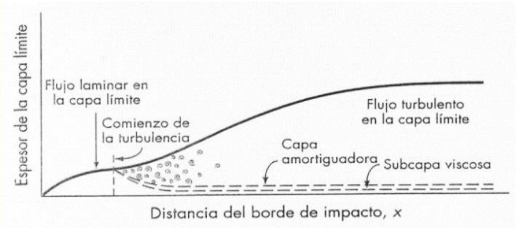



**Fig. 1.** Espesor de la capa límite en función de la distancia del borde de impacto, las diferentes subcapas.

Debido a cuerpos que están inmersos en un fluido, las capas límites toman diferentes formas, tal como puede observarse en la Fig. 2.

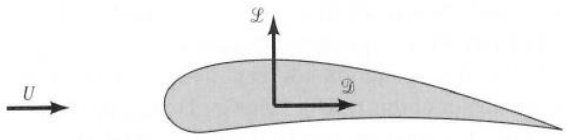
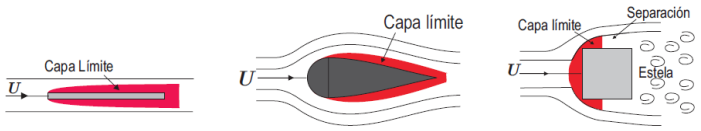



**Fig. 2.** Resultante de fuerzas horizontal y vertical sobre un perfil alar de un cuerpo dentro de un fluido.




(a) Placa plana delgada. (b) Cuerpo aerodinámico. (c) Cuerpo obstructor.

**Fig. 3.** Capas límites de cuerpos (color rojo) inmersos en fluido. a) Placa plana delgada, b) Cuerpo aerodinámico, c) Cuerpo obstructor.

La magnitud de la capa límite dependerá de la forma que tome el flujo alrededor del cuerpo. La Fig. 3 muestra tres tipos de cuerpos dentro de un fluido, y sus respectivas capas límite están pintadas de rojo.

El cálculo de la capa límite tiene que ver con dos parámetros importantes: el coeficiente de fricción (viscosidad), y su espesor. Estos parámetros difieren fuertemente dependiendo de las capas antes mencionadas [6].

*Número de Reynolds.* En 1851 George Gabriel Stokes introdujo el concepto de número de Reynolds y fue nombrado por Osborne Reynolds en 1883. El número de Reynolds (véase ecuación 1) es un número adimensional utilizado en mecánica de fluidos, diseño de reactores y fenómenos de transporte para caracterizar el movimiento de un fluido. Su valor indica si el flujo sigue un modelo laminar o turbulento. El número se puede definir como las relaciones entre fuerzas inerciales y las fuerzas viscosas presentes en un fluido. Se puede decir que relaciona la densidad ($\delta$) y la velocidad (Vs), dimensiones que aparece en muchos cálculos de la dinámica de fluidos, sobre todo en las ecuaciones de Navier Stokes que gobiernan los movimientos de los fluidos. Por ejemplo, si el número vale menos 2100, el flujo es laminar; si va más allá de los 4000, es flujo turbulento. Así de sencillo:

$$Re_x = \left(\frac{\rho}{\mu} U\right) X, \tag{1}$$

donde $\rho$=densidad del fluido, $\mu$=viscocidad dinámica, $U = velocidad\ de\ corriente\ libre$, $X = distancia\ al\ borde\ del\ ataque$.

*Coeficientes de fricción*. El coeficiente de fricción en un líquido o un gas se le denomina viscosidad y se mide mediante un orificio por donde sale el fluido. La

velocidad con que este sale es la propiedad de viscosidad de los fluidos (véase la Fig. 4). Como se puede ver, este depende indirectamente de los números de "Reynolds" [7].

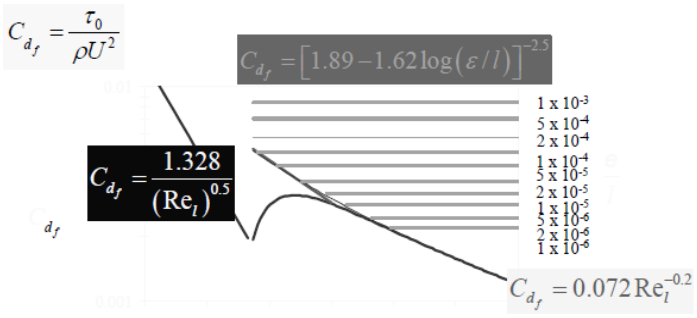



**Fig. 4.** Coeficientes de fricción en una placa plana usando los números de Reynolds.

*Efecto Piezoeléctrico y piezoeléctrico inverso*. En 1880, Jacques y Pierre Curie descubrieron que al aplicar presión a un cristal de cuarzo se generaban cargas eléctricas en el cuarzo, a lo que se le llamó "el efecto piezoeléctrico". Más tarde, ellos verificaron que si ahora aplicaban un campo eléctrico al cristal de cuarzo proporcionaba una deformación al material. Este efecto lo llamaron "piezo inverso". De forma tal que se tiene que los materiales piezoeléctricos convierten energía mecánica en eléctrica y viceversa. La palabra piezoeléctrico es una palabra compuesta por piezo y eléctrica, donde "piezo" se deriva de la palabra griega πιεζω, que significa estrechar, apretar u oprimir.

Las propiedades de los materiales piezoeléctricos son de dos orígenes: naturales y no naturales. Dentro de los naturales está el cuarzo, la turmalina, la sal de Rochelle, etc. A partir de los materiales naturales se han desarrollado los no naturales, mediante una combinación de materiales cerámicos ferro eléctrico policristalino, como el $BaTiO_3$, y el Zirconato Titanato de Plomo (PZT). Este último material es muy empleado en aplicaciones como actuadores o sensores. Generalmente los campos eléctricos para efectuar un movimiento mecánico en un material PZT cerámico andan arriba de los 2000 V/mm.

## 2. Métodos

El diseño y la simulación de control inteligente se llevó a cabo tomando en consideración las leyes de la física de los fluidos y expresados con la teoría de control inteligente, basados la teoría de los conjuntos y la lógica difusa e implementados en bloques usando Simulink-Matlab® para su experimentación mediante gráficos en dos dimensiones. Los cálculos de los parámetros del prototipo de control inteligente se lograron en lazo cerrado (control retroalimentado), cuya variable a controlar fue el flujo de líquido de baja viscosidad entre un recipiente cisterna a un recipiente de producción. La estructura del controlador inteligente está basada en el modelo de Mamdani, el cual contiene tres submodelos importantes: fuzzificador, sistema de inferencia difusa y el defuzzificador. El diseño de los submodelos de Mamdani son como sigue: el submodelo

de fuzzificador se diseñó con tres funciones de membresía triangulares, el submodelo es parametrizado con inferencia de Mamdani que lleva tres reglas con los conectores lógicos Si, Entonces; el defuzzicador fue hecho también con tres funciones triangulares [9,10].

En la etapa del diseño solamente se requieren los rangos de operación del proceso de microbombas MEM, así como el rango de medición de actuador piezoeléctrico MEM para obtener la simulación del controlador inteligente. Las premisas consideradas a priori son que se considerará un flujo laminar y de baja viscosidad para efectos de diseño y simulación.

El prototipo de implementación consiste en dos modelos: el modelo de Proceso de Fluidos de baja viscosidad, el modelo de Control Inteligente embebido en el modelo electrónico digital, en donde el modelo de Control Inteligente cuenta con Sensor/Actuador basado en tecnología electrónica MEM y va acoplado con "drivers" electrónicos de voltaje medible, lo que permite el monitoreo basado con tecnología Raspberry Pi 3 y pantalla "touch screen" [10].

### 2.1. Método de sistema de control inteligente (SCI). Prototipo de simulación gráfica

La construcción de este sistema se detalla a continuación, mediante las siguientes etapas:

1. Medición de señales: Se utilizaron distintos tipos de dispositivos electrónicos, entre los cuales están: excitador de voltaje tipo Boots, bomba piezoeléctrica MEM, sensor piezo eléctrico MEM y rasberry Pi 3.
2. Adquisición de datos y procesamiento: Las siguientes fases corresponden a la conversión analógico-digital y al procesamiento de la información. En este caso se empleó un microcontrolador Arduino nano, que incluye el control inteligente debido a su capacidad de procesamiento y amplio conjunto de interfaces de comunicación, como son puertos paralelos, números de convertidores analógico-digitales e interfaces de transmisión síncrona o asíncrona.
3. Interfaz: El desplegado de la información se realiza mediante un LCD de 2 líneas por 16 columnas, en donde los valores son mostrados.

### 2.2. Modelo de análisis

El método empleado para la implementación del diseño de Sistema de Control Inteligente fue el de Mamdani, que consiste en tres principales módulos son: Fuzzificador, Base del Conocimiento, y Defuzzificador.

La variable de entrada presenta valores y rangos diferentes, por lo que es necesario transformarlos a un espacio en donde todas tengan un mismo rango de medición. Esto se realiza mediante la aplicación de una función de pertenencia (μ), la cual estandariza a cada una de las mediciones de cada parámetro en valores entre 0 y 1. No existe alguna regla que defina cómo construir una función de pertenencia; sin embargo, una función

lineal simplifica el proceso. Derivado de esto, se diseñaron funciones triangulares, mismas que se pueden expresar mediante la ecuación 2.

$$\mu(x,a,b,c,d) = max\left\{0, min\left[\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right]\right\}, \quad (2)$$

donde x representa la variable de entrada; a, b, c y d son los parámetros que definen a la función de entrada, en caso de una función triangular b=c.

En el método de fuzzificación se plantea con los mínimos recursos, tres conjuntos para la variable de entrada de Error de Flujo. Mientras que para la Base de Reglas fueron solamente tres reglas Si-Entonces. Finalmente, el defuzzificador solamente se diseña con tres funciones de membresía (FEM). Como se puede observar en la Fig.5.

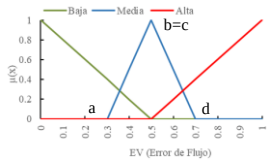
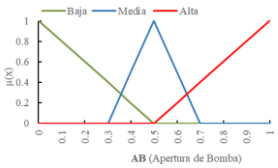
| **Fuzzificador** | **Base del Conocimiento** | **Defuzzificador** |
|---|---|---|
|  | R1. Si EF es B, Entonces AB es A<br>R2. Si EF es M, Entonces AB es M<br>R3. Si EF es A, Entonces AB es B |  |

**Fig. 5.** Método de Mamdani empleado, funciones de Membresía de Entrada/Salida (Error de flujo/Apertura t de Bomba).

## 2.3. Prototipo de simulación grafica a través del método de SCI

En la Fig.6. se presenta la descripción breve de la simulación de las diferentes técnicas de control empleadas para llevar a cabo la comparación de respuestas que se esperan, para determinar la estabilidad de cada uno de ellos y observar de una manera visual las trayectorias [12].

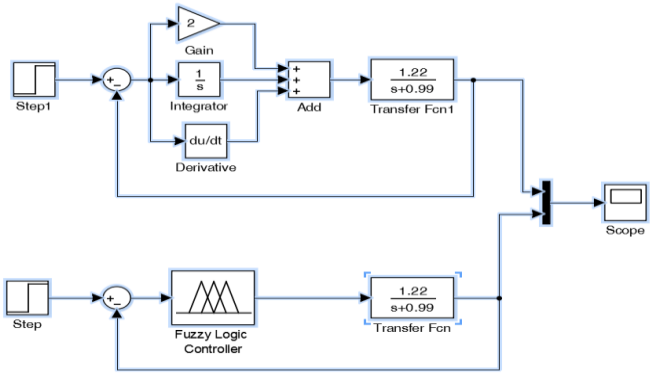



**Fig. 6.** Diagrama a Bloques de Controlador Clásico PID (a), y controlador inteligente Lazo (b) usando el método de Mamdani (b).

### 2.4. Método de control inteligente. Prototipo de hardware

Los parámetros encontrados con el método de control inteligente, vía la simulación del apartado anterior, determina el Prototipo de hardware siguiente [13], véase la Fig. 7.

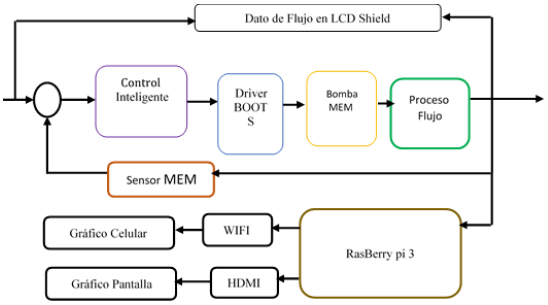



**Fig. 7.** Diagrama a Bloques de Método de SCI de Flujo de baja Viscosidad.

## 3. Resultados

En el método diseñado empleado para determinar la función de transferencia del proceso de flujo de baja viscosidad, las respuestas de la bomba piezoeléctrica MEM se presenta en la Fig. 8, y la respuesta del sensor de flujo tipo piezoeléctrico MEM está caracterizado por la curva de la Fig. 9.

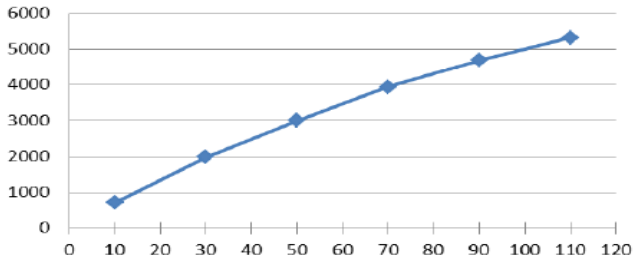



**Fig. 8.** Gráfico de la Salida de La Bomba MEM (micro Lt/min Vs Frecuencia ciclos/min).

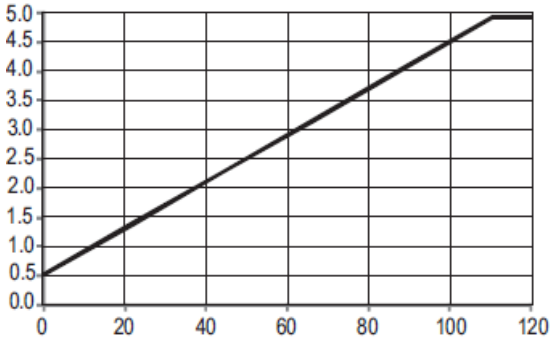



**Fig. 9.** Gráfico del Sensor de Flujo x = razón de Flujo de Masa, y = voltaje de Salida.

Ahora seleccionando un punto de operación a f = 100 Hz se tiene valores de 5 ml/min de flujo, correspondiendo a un voltaje de 4.5 voltios. Con ello, el flujo nominal ocupa un tiempo de 0.01 s. Con este valor constante de tiempo del proceso de flujo del fluido a baja viscosidad se tiene una función de transferencia como la que se ve en la Fig. 9. Usando Simulink- Matlab® se diseñó primero para un sistema de control clásico. El control clásico (PID) arroja los rangos de operación de entrada/salida (E/S) del controlador (a). Mismos que servirán para determinar los rangos del Sistema de control Inteligente (b) [11].

### 3.1. Resultado de prototipo de simulación usando los métodos de control clásico e inteligente

La Fig. 10 muestra los resultados del prototipo de Simulación de los Métodos del SCI aplicado a Proceso de Flujo de fluido de baja viscosidad, corriendo a una consigna de 1 ml/minuto. En la figura se puede apreciar la convergencia o estabilidad de ambos métodos de Control de lazo cerrado aproximadamente a 20 s.

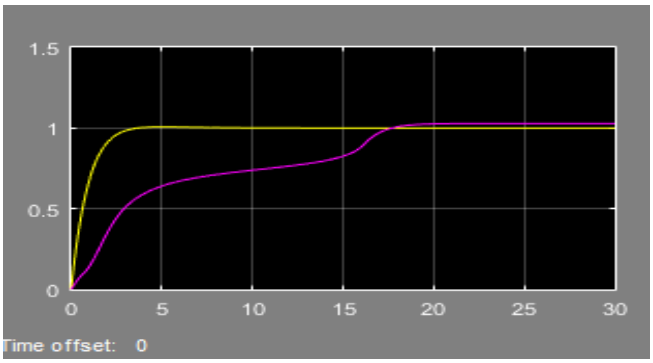



**Fig. 10.** Respuestas a los sistemas de control Clásico (A)/Inteligente (M). Respuestas a los SCC (amarillo) y SCI(Magenta). Set point Normalizado (2m/min).

El resultado del Prototipo de hardware usando el método de Control Inteligente, se muestra en la Fig. 11.

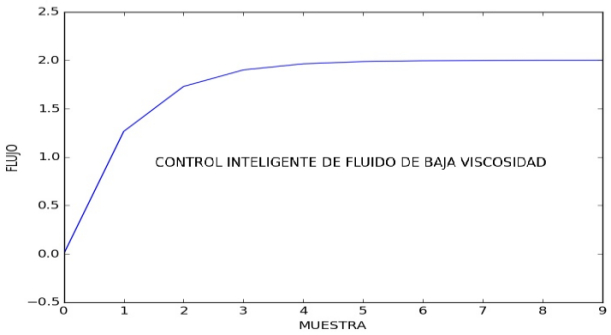



**Fig. 11.** Respuesta del Método de SCI Aplicado a 2 ml/min en Pantalla gráfica Rasberry.

## 4. Conclusiones

El presente trabajo desarrolló una herramienta especializada en la dosificación de líquidos. Su principal ventaja es el control de flujos de baja viscosidad. En comparación con los sistemas existentes en el mercado, este es más accesible ya que su producción involucra un menor costo.

En este trabajo presentamos un nuevo modelo exitoso para controlar la apertura de la bomba MEM para la dosificación de fluidos mediante control inteligente. Las reglas del sistema de control se basaron en el error de flujo, para garantizar la apertura óptima y correcta de las micro bombas MEM. Se definieron funciones de membresía triangulares para normalizar las evaluaciones particulares en un rango de [0 - 1]. De acuerdo con los resultados, el índice propuesto en este artículo demuestra una mejor eficiencia y precisión en comparación con el otro método típico PID. La eficacia y la precisión son muy importantes en la administración de fluidos.

Como trabajo futuro, es importante estudiar las características de otros líquidos para introducir otros tipos de parámetros que deben considerarse. Este trabajo tiene un impacto académico, social y económico, puesto que permite ver la integración de diferentes materias y conocimientos científicos con un fin común y con tecnología de punta a la altura de los países industrializados, y reduce los costos de instalación, mantenimiento, operación y de bajo consumo por usar tecnologías tipo MEM.

**Agradecimientos**. Los autores agradecen al Centro de Investigación en Computación del Instituto Politécnico Nacional (IPN) y al Consejo Nacional de Ciencia y Tecnología (CONACyT), por su apoyo en el desarrollo de este trabajo, a través de los proyectos SIP20180762 y SIP20181910.

## Referencias

1. Prieto, P.: La innovación, clave de los sistemas de mando y control de Defensa. Homeland Security and Defense (2012)
2. Clyde Warsop: AEROMENS II. Advanced Flow Control Using MEMS Results and Lessons BAE SYSTEM. Learned (2006)
3. Jiang Chengyu, Deng Jinjun, Ma Bringhe, Yuan Weizheng. Advanced flow measurement and active flow control of aircraft with MEMs (2012)
4. Orozco-Lozano, W. A.: Diseño y simulación de las fuerzas de arrastre y sustentación en los autos. Universidad Autónoma del Caribe. Vol. 4. Jul-Dic, pp. 26–33 (2006)
5. Mercado, J.R., Guido, P., Sánchez-Sesma, J., Íñiguez, M.: Fórmulas para el coeficiente de arrastre y la ecuación Navier-Stokes fraccional. Tecnología y Ciencias del Agua. Vol. V, núm. 2, pp. 149–160 (2014)
6. Gherardelli, C.: Mecánica de Fluidos. Cap. 10 (2014)
7. Manuel, F., Mejía de Alba, García-Fernández, L. E., Gutiérrez-Almonacid, M. A.: Metodología de obtención de los coeficientes de sustentación y arrastre para un rango amplio de números de Reynolds y ángulos de ataque para aplicaciones en turbinas eólicas. Universidad de América (2011)
8. Cupich-Rodríguez, M., Elizondo Garza, F. J.: Actuadores piezoeléctricos.

9. León-Sandoval, J. J.: Componentes de Sistemas de Control de Flujo y Temperatura para aplicaciones en la Industria de la Fundición. Monterrey N.L. Universidad Autónoma de Nuevo León. Tesis Maestría (1989)
10. García, R. J., Pinto, A. D., Rengel, J. E., Torres, J. M., González, J. A., Pérez, N. A.: Diseño de una estrategia de control difuso aplicada al proceso de ultracongelación de alimentos.
11. Escaler, X., Baliu-Henne, A.: Simulación de la respuesta fluido- dinámica del sistema de refrigeración del sincrotrón ALBA. Detección de aire en tuberías (2016)
12. González-Fontanet, J. G., Haber Guerra, R. E., Matía, F., Novo, M.: Diseño de sistemas de control en cascada clásico y borroso para el seguimiento de trayectorias. Jornada de Automática 2017. Cuba (2017)
13. Servo Flow Corporation: Microcomponents. Application Notes.

ISSN 1870-4069

# Metodología para la representación de hologramas tridimensionales en alta definición

Jesús Jaime Moreno Escobar, Oswaldo Morales Matamoros, Ricardo Tejeida Padilla

Instituto Politécnico Nacional, ESIME-Zacatenco, México
jemoreno@esimez.mx

**Resumen.** El presente trabajo consiste en una metodología para la Captura y Reproducción de Hologramas tridimensionales en alta definición. Esta propuesta hace uso de herramientas de Visión por Computadora e Inteligencia Artificial, la cual está dividida en cinco fases. En la primera fase se plantean los problemas de las distintas técnicas de visualización 3D y cómo este proyecto los soluciona, como es el uso de dispositivos 3D. En tanto que la fase dos tiene como finalidad explicar los antecedentes de la holografía y los hologramas; además, explica cómo con ayuda de la visión estereoscópica se han creado dispositivos, métodos y técnicas que permiten la visualización de objetos en 3D. Es por ello que en la siguiente fase da las bases teóricas de los elementos usados en la construcción de este proyecto, el funcionamiento del ojo humano, lo espacios de color y los principios para poder extraer un color cromáticamente puro (ChromaKey Verde) como fondo de un escenario. La metodología matemática general se propone hasta la cuarta fase donde se consideran todas técnicas que se utilizan para la construcción y diseño de un módulo de captura, que se utiliza para realizar la codificación del holograma final. Además, se exponen las consideraciones para la construcción de un módulo para la representación del holograma y poder visualizarlo en una pirámide holográfica. Finalmente se plantean y explican los principales experimentos, con la pirámide holográfica a dos ángulos de inclinación diferentes, la iluminación del módulo de captura, calibración de la mesa giratoria y proceso de codificación y representación holográfica.

**Palabras clave:** holograma, visión por computadora, sistemas en tiempo real, ChromaKey.

## Methodology For Representing Real Time 3D-Holograms

**Abstract.** The present work consists of a methodology for the Capture and Representation of three-dimensional holograms in high definition. This proposal makes use of computational tools, such as Computer Vision and Artificial Intelligence, this methodology is divided into five phases or steps. In the first step the problems of the different techniques of 3D visualization are presented and how this project solves them, as is the use

of 3D devices. While phase two is intended to explain the background of holography and holograms, it also explains how, making use of stereoscopic vision, devices, methods and techniques have been created that allow the visualization of 3D objects. That is why in the next phase gives the theoretical basis of the elements used in the construction of this project. Also, we explain how human eye obtains natural images, color spaces and principles to extract a chromatically pure color (ChromaKey green) as a scene background. The mathematical methodology is proposed in the fourth phase where all techniques are considered that are used for the construction and design of a capture module, which is used to perform the coding of the final hologram. In addition, the considerations for the construction of a module for the representation of the hologram and to visualize it in a holographic pyramid are exposed. Finally, the main experiments are presented and explained, with the holographic pyramid at two different angles of inclination, also we perform a test of illumination of the capture module, in addition we calibrate the turntable and process of coding and holographic representation.

**Keywords:** hologram, computer vision, real time systems, ChromaKey.

## 1. Introducción

En la actualidad existen técnicas para crear imágenes tridimensionales o dar la ilusión de profundidad en la imagen. La estereoscopía, que consiste en recoger información visual tridimensional de una imagen, la cual los ojos (derecho e izquierdo) debido a su separación, obtienen dos imágenes con diferencias entre ellas y el cerebro procesa las diferencias de ambas imágenes, dando una sensación de profundidad de los objetos. El problema de utilizar este método es que no se logra la apreciación tridimensional, sino que se obtiene la sensación de profundidad de una imagen, así se logra una ilusión tridimensional[2].

El proyecto propuesto crea imágenes tridimensionales holográficas dando un volumen a objetos determinados, así se facilita su estudio y manipulación, como un holograma. Este da una apreciación de profundidad tridimensional que se observa desde diferentes ángulos sin perder la forma. Estas crean imágenes bidimensionales dando la sensación de profundidad mediante imágenes planas reflejadas en las caras de una pirámide holográfica[1]. El sistema de generación de hologramas constará de tres fases: captura, codificación y representación.

La fase de captura, se obtienen los videos del objeto con un fondo cromáticamente verde. Así se permite la representación del mismo en tres dimensiones en la pirámide holográfica. La fase de codificación, se obtiene una serie de imágenes o fotogramas de los videos capturados en la primera fase. Con el objeto con el fondo Chroma verde sustituido por un fondo negro absoluto. La última fase, permitirá que se observe un holograma en el centro de una pirámide holográfica, esto debido a la refracción y reflexión de la luz en las caras de la misma.

El módulo de captura obtiene de sus cuatro sensores, imágenes de cuatro perfiles del objeto (frente, atrás, perfil izquierdo y perfil derecho). Así, los sensores a utilizar son cuatro, las cuales se activan por separado ya que el PC utilizado

sólo permite utilizar un sensor a la vez. Este sistema crea hologramas con un tamaño a escala 2:1 del objeto original, es decir, el holograma estará reducido a la mitad de su resolución respecto a la original. La pirámide holográfica muestra en su centro, por la refracción de luz en sus caras, el holograma de un objeto de la mitad de su tamaño real, sin mostrar pérdida de sus características o colores.

## 2. Antecedentes

La estereoscopía es una técnica para obtener imágenes que generen la sensación de tres dimensiones. La palabra estéreo proviene del griego que significa relativo al espacio[6,7]. En el año de 1838, la estereoscopía fue definida oficialmente por Sir Charles Wheatstone por su explicación de visión binocular y fue el primero en idear un aparato para proporcionar visión en relieve o en tres dimensiones, el estereoscopio.

El estereoscopio permitía la visión de dos imágenes y cada una correspondiente a los 65mm de disparidad en los ojos. En 1849, Sir David Brewster creó y construyó una cámara binocular que sacaba dos imágenes y las sincrónicamente las cuales permitían realizar retratos estereoscópicos. La técnica estereoscópica evolucionó en la segunda mitad del siglo, se adaptó a mejoras de procedimientos como desde el estéreo daguerrotipo al veráscopo de Richard. Tuvo gran aceptación a principios del siglo XX. Muchos fotógrafos del siglo XIX y principios del XX realizaron tomas estereoscópicas[3]. En el año de 1600 Giovanni Battista Della Porta presentó la técnica de dibujo estereoscópico, que consistía en dibujar dos imágenes de un objeto visto con un ligero corrimiento horizontal para dar la sensación de profundidad. Esta percepción provoca en el observador una sensación de inmersión en la escena que se encuentra frente a él. La visualización 3D en sus inicios utilizaba lentes especiales llamados anáglifos, debido a que era la única manera de generar una sensación de profundidad en el cerebro del espectador[4]. Con el avance tecnológico se crearon nuevos artefactos para la visualización 3D tales como:

- *Lentes anáglifos.*
- *Visualización por Polarización.*
- *Visualización por secuencia Estero Activa o alternativa.*
- *Head Mounted Display (HMD).*
- *Monitores autoestéreo.*

Existen cuatro técnicas para crear sensación de profundidad, las cuales han evolucionado pero la base de todas ellas es principalmente la estereoscopia:

1. *Efecto Pulfrich.*
2. Chromadepth.
3. *Foto escultura.*
4. *Reliefografía.*

# 3. Propuesta metodológica del sistema holográfico

## 3.1. Funcionamiento general del sistema

La Figura 1 muestra el sistema de Captura y proyección holográfica está constituido por tres fases en su diseño:

1. Captura,
2. Procesamiento y
3. Visualización.

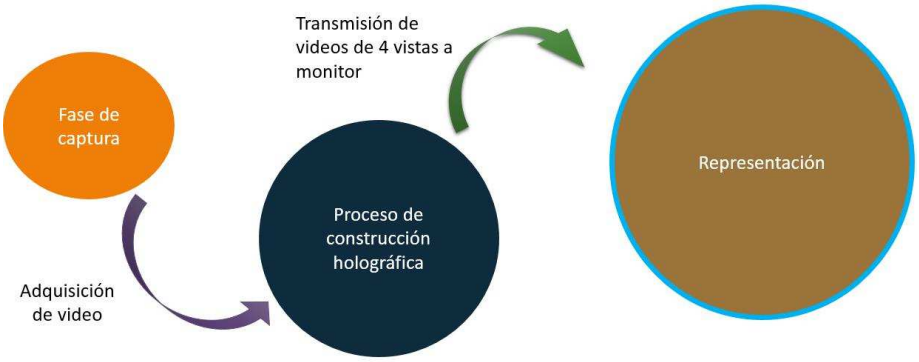



**Fig. 1.** Representación del funcionamiento general de un sistema de captura y reproducción holográfico.

## 3.2. Diseño del módulo de captura

La fase de Captura está formada por un módulo (caja) con dimensiones de 100 cm de largo, 100 cm de ancho y 20 cm de altura, cuatro cámaras Web, una base giratoria e iluminación uniforme. Esta fase esta subdividida en cinco subfases, las cuales son:

a) La Figura 2 muestra las consideración en las Dimensiones del módulo. Así, las dimensiones son determinadas de acuerdo con los objetos de captura y las cámaras web usadas. Los objetos tienen que tener una medida máxima de 10 cm de altura y 7 cm de anchura, para que los fotogramas sean correctamente tomados. Las dimensiones finales del módulo se muestran en la Figura 3.
b) Selección y Posición del sensor USB: Este tipo de sensor se asemeja a una pequeña cámara digital que se conecta a una computadora, para que esta capture y transmita imágenes a través de Internet, por ejemplo. Los sensores USB utilizados no transmiten imágenes a través de Internet, sino a través del puerto USB de la computadora al programa MATLAB mediante un *socket* de Video, Figura 4.

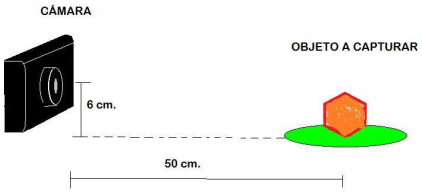



**Fig. 2.** Cámara Web a la distancia correcta para una buena captura.

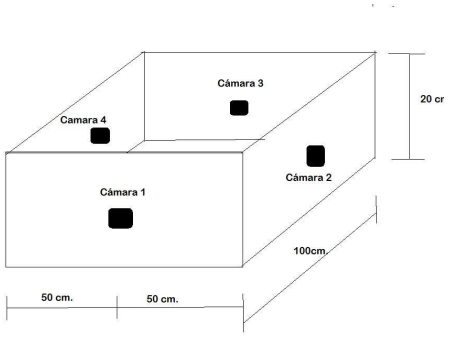



**Fig. 3.** Modulo de captura con dimensiones finales.

c) Elección de Pintura Cromáticamente Verde: En el interior del módulo de captura está uniformemente pintado de un color verde o Chroma Key, Figura 5. Esta llave de color ya sea verde o azul es necesaria ya que, es menos costoso computacionalmente segmentar las cosas que no sean este color.
d) Control de Giro de Base: La base giratoria es una base circular puesta en el centro del módulo de captura, la función es girar a determinadas revoluciones por minuto, el objeto deseado. Las revoluciones por minuto se determinan de acuerdo con el número de fotogramas requeridos de cada vista del objeto, Figura 6.

## Algoritmo para la captura holográfica

1. Asignación de nombre y número de pines de la placa Arduino a los de pines de motor, Pin 12 ($\alpha$), pin 11 ($\beta$), pin 10 ($\gamma$) y pin 9 ($\delta$).
2. Velocidad de giro $\upsilon = 2milisegundos$.
3. Definición del contador $\kappa$ de tipo entero.
4. Declaracion de variable de tipo entero para número de vueltas, $\lambda$.
5. Asignación de salidas digitales en el puerto de Arduino, Pines $\alpha$, $\beta$, $\gamma$ y $\delta$.
6. Mientras no se den $\phi = 4560$ pasos en el Motor $\Delta$ no completará una vuelta completa en la base giratoria. Así, $\lambda = \lambda + 1$.

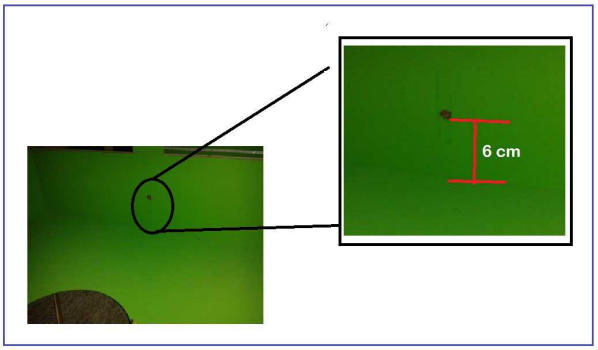



**Fig. 4.** Posición del sensor USB en las paredes del módulo de captura.

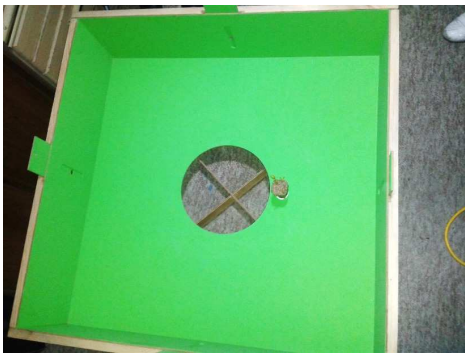


**Fig. 5.** Pintura del módulo de captura sin tapa.

7. Habilitar secuencialmente las salidas digitales para activar las bobinas del Motor $\Delta$ a pasos para girarlo.
8. Se incrementa $\kappa$, es decir, $\kappa = \kappa + 1$.
9. Si $\kappa$ es menor que 4 entonces se repiten los pasos del 2 al 8, de lo contrario el algoritmo se termina.

**Acondicionamiento lumínico** La iluminación instalada fue de luz amarilla ya que la luz blanca refleja el color verde en los objetos, Figura 7. La iluminación es uniforme ya que se necesita evitar reflejos en las paredes y lentes de los sensores o exceso de brillo en los objetos, las paredes del módulo o la base giratoria.

### 3.3. Fase de codificación de imágenes

La fase de procesamiento de imágenes, es un subsistema que complementa el funcionamiento general del prototipo. El subsistema funciona herramientas o *toolboxes* en MATLAB, que detecta los pixeles verdes lo más cercanos al Chroma Key, y así sustituirlos con pixeles de color negro.

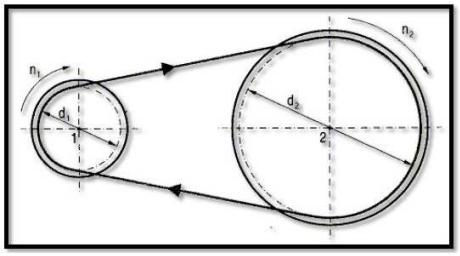



**Fig. 6.** Sistema de poleas, relación diámetros y velocidades.

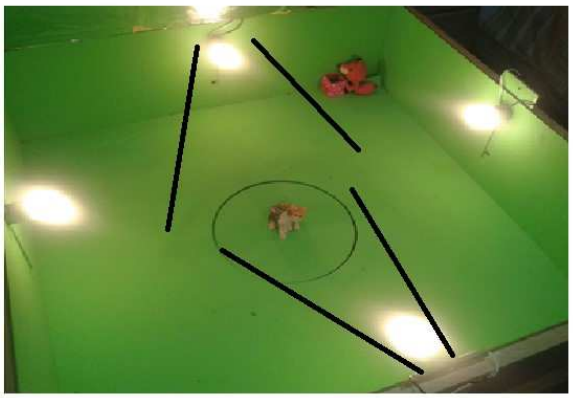


**Fig. 7.** Foco y la incidencia de luz al objeto.

Como cualquier sistema o subsistema el procesamiento de hologramas se lleva a cabo en tres fases, como se observa en la Figura 8, las cuales son:

1. Lectura
2. Codificación
3. Representación

### Algoritmo general de la codificación

1) Leer la imagen que contiene el fondo negro.
2) Declarar una variable de tipo videoinput. Esta variable capturará imágenes del sensor USB (1, 2, 3 ó 4) con las características del mismo.
3) Se crea un archivo de video en formato avi y se le asigna un nombre.
4) Se inicializa el video creado, para posteriormente ingresar las imágenes y al mismo tiempo se ejecuta el comando de inicialización de una vuelta para la mesa giratoria.
5) Se inicia un ciclo para la adquisición de las imágenes, este ciclo está determinado por el número de fotogramas a capturar, en este caso mil fotogramas.


**Fig. 8.** Diagrama cibernético de primer orden, es decir, entrada, proceso, salida del subsistema de procesamiento de imágenes.

6) Dentro del ciclo se crea una variable que guarda la imagen tomada por sensor USB o $I$.
7) Se asigna el valor del umbral obtenido de la experimentación $\mu$.
8) Se obtienen las dimensiones de la imagen y se guardan en dos variables $Col_I$ y $Rows_I$.
9) Se redimensiona la imagen de fondo negro con las dimensiones $Col_I$ y $Rows_I$, al tamaño de la imagen capturada $I$.
10) Se transforma del espacio de color de RGB a YCbCr, de la imagen capturada $I$.
11) Se elimina el fondo verde y se sustituye por la imagen con fondo negro, obteniendo una nueva imagen $I_{chroma}$.
12) Se convierte la imagen $I_{chroma}$ al espacio de color RGB.
13) Dependiendo del número asignado al sensor USB se rota la imagen un ángulo en específico.
14) Se agrega la imagen $I_{chroma}$ al archivo de video en el fotograma correspondiente.

**Algoritmos particulares de la codificación** El algoritmo general está formado por tres subalgoritmos para codificar las imágenes:

a) Adquisición de Fotogramas,
b) Extracción de fondo verde cromático y
c) Creación de Video.

a) Adquisición de fotogramas.
   1) Se reconoce el sensor correspondiente.
   2) Activación del sensor.
   3) Seleccionar resolución de los fotogramas a obtener.
   4) Seleccionar espacio de color YCbCr de los fotogramas.
   5) Seleccionar límite de fotogramas a adquirir.
   6) Iniciar ciclo de captura de los fotogramas que componen al video.
b) Extracción de fondo verde cromático.
   1) Guardar la imagen de fondo negro absoluto.
   2) Obtener la imagen con fondo verde.

3) Seleccionar el valor de umbral a utilizar.
4) Obtener las dimensiones de la imagen.
5) Redimensionar la imagen de fondo negro absoluto a las dimensiones de la imagen con fondo verde.
6) Realizar una trasformación de componentes a la imagen con fondo verde.
7) Convertir la imagen obtenida a escala de grises.
8) Separar el fondo verde y el objeto de la imagen.
9) Extraer el fondo verde y lo sustituimos por la imagen de fondo negro absoluto.
10) Agrupar la imagen de fondo verde con la del objeto.
11) Recuperar el espacio de color RGB de la imagen.
12) Dependiendo de la cámara activada se rotará la imagen final 0°, 90°, 180° y 270°, respectivamente.

c) Creación de video.
1) Se crea un objeto de tipo video.
2) Al objeto de video se le asigna un nombre.
3) Se crea el archivo de video con el nombre deseado con extensión avi.
4) Se inicia el objeto de video.
5) Dentro del ciclo de fotogramas, se guarda cada fotograma respectivamente en el objeto de video.
6) Se cierra el objeto de video.
7) Finalmente se obtiene un archivo de video a partir de los fotogramas adquiridos.

### 3.4. Fase de representación

La fase de representación consta de dos elementos principales, para la correcta visualización. El primer elemento es el monitor de proyección el cual proyectará un video con las cuatro vistas del objeto (frente, atrás, perfil derecho, perfil izquierdo), con fondo negro, en la Pirámide holográfica. El segundo elemento es la pirámide holográfica la cual nos permitirá la visualización del objeto deseado en tres dimensiones.

a) Monitor de proyección: Para saber qué pirámide utilizar se debe considerar la forma de proyección del holograma. Como se obtienen cuatro videos, cada uno de una vista del objeto a modelar.
b) Pirámide holográfica: Una vez determinado el modo de proyección, se construye la pirámide holográfica a utilizar. Dado que la proyección es de cuatro vistas del objeto, la proyección debe ser en cuatro superficies, así, la pirámide deberá tener cuatro caras en las que la luz de la proyección se refractará a 45 grados.

Así para las Consideraciones Técnicas para la construcción de la Piramide Holográfican, por un lado se considera primeramente el ángulo en el cual el espectador la observa. Así que se tomó un ángulo de visualización recto para observar el holograma en una posición cómoda y para que este se visualice en el

centro de la pirámide. Por otro lado, se considera cuáles son las dimensiones del monitor a utilizar.

Para calcular las dimesiones en el Diseño de la Pirámide Holográfica se estima la altura máxima para una pirámide de 45°. En una pirámide la altura máxima es igual a la distancia de la punta de la pirámide hacia el centro de su base, Figura 9.

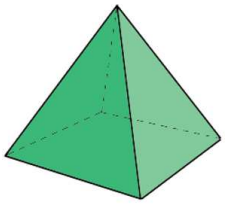


**Fig. 9.** Diseño de la Pirámide Holográfica.

## 4. Experimentos y resultados

Como prueba de que los hologramas cumplen con la función de dar volumen a un objeto en una pirámide holográfica se realiza el siguiente experimento:

Se invita a 50 personas a observar los hologramas de *Gatitos* (Figura 10) y *Wall-E* [5] (Figura 11) para que den su calificación de calidad a los mismos. La calificación es de 1 a 10. Donde 1 no es considerado un holograma, es decir, sin volumen y sin calidad y donde 10 se considera un holograma de buena calidad. Para conocer mejor la calidad que tienen los hologramas se realiza el estudio de los datos obtenidos mediante el cálculo de la media, moda, varianza y desviación estándar.

- Moda para Holograma Gatitos = 9.09270683
- Media para Holograma Gatitos = 9.3
- Varianza para Holograma Gatitos = 0.29089184
- Desviación estándar para Holograma Gatitos = 0.544819611

- Moda para Holograma Wall-E= 9.059231373
- Media para Holograma Wall-E= 9
- Varianza para Holograma Wall-E= 0.27170484
- Desviación estándar para Holograma Wall-E= 0.52654519

Ahora, se toma como medida importante la desviación estándar o típica, dado que esta determina el promedio de la fluctuación o dispersión de los datos respecto a su punto central o media. La fluctuación o datos que se encuentren fuera de esta desviación determinan las deficiencias de los hologramas.

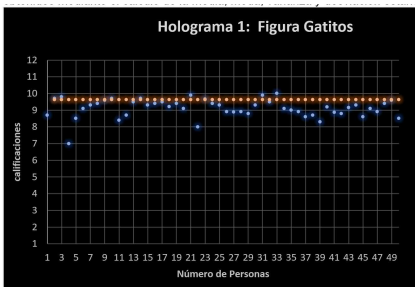



**Fig. 10.** Gráfica de dispersión del nivel de satisfacción del holograma de gatitos.

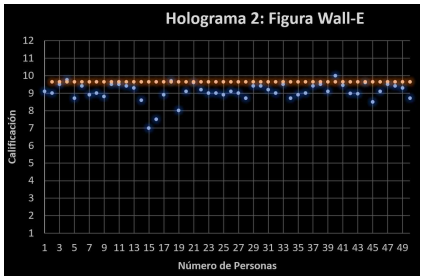



**Fig. 11.** Gráfica de dispersión del nivel de satisfacción del holograma de Wall-E.

Con las gráficas se concluye que el observador tiene una buena impresión y expereiencias de los hologramas finales pues las fluctuaciones que se muestran en las gráficas (puntos fuera de las curvas de desviación estándar) son pocas y sus resultados en promedio son mayores que 9.

## 5. Conclusiones

Es posible concluir que este sistema no depende de ningún dispositivo especial de proyección 3D para dar volumen al holograma de alta definicón, ni depende de un dispositivo de visualización 3D para observar el holograma. Los objetos trasparentes, de color verde, negro o colores muy brillosos dan resultados no deseados. Con una iluminación uniforme en el módulo de captura se logran buenos resultados al extraer el fondo verde de la imagen, obteniendo una imagen HD. Se construyó un módulo de captura capaz de recibir información específica de las características de un objeto por medio de sus cámaras. Además, se construyó una pirámide holográfica en la que se visualizan hologramas a escala 2:1 respecto al objeto original. El sistema diseñado y construido facilita la creación de hologramas de casi cualquier objeto en poco tiempo, a diferencia de otros sistemas, lo que optimiza la aplicación de sistema en distintas ramas.

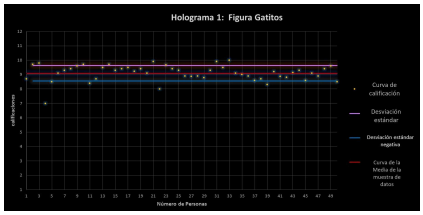



**Fig. 12.** Gráfica de dispersión con desviación estándar de la muestra de datos para el holograma Gatitos.

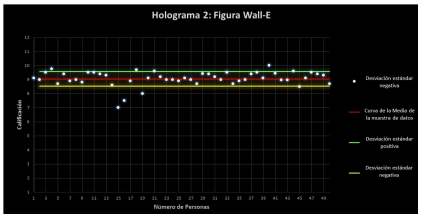



**Fig. 13.** Gráfica de dispersión con desviación estándar de la muestra de datos para el holograma Wall-E.

**Agradecimientos.** Este trabajo es desarrollado con recursos e instalaciones del Instituto Politécnico Nacional, México por medio del Proyecto SIP 20180514 y la Comisión de Operación y Fomento de Actividades Académicas (COFAA). Cabe resaltar que los resultados de este trabajo fueron realizados por los tesistas de Nivel Licenciatura Leslie Marie Ramírez Álvarez y Luis Omar Hernández Vilchis. También, se le agradece por un lado al Ing. Daniel Hazet Aguilar Sánchez por el apoyo soporte lógistico y técnico; y por otro a los revisores que aportaron sus valiosos conocimientos para mejora del presente artículo.

## Referencias

1. Jiao, S., Tsang, P.W.M., Poon, T.C., Liu, J.P., Zou, W., Li, X.: Enhanced autofocusing in optical scanning holography based on hologram decomposition. IEEE Transactions on Industrial Informatics PP(99), 1–1 (2017)
2. Kotgire, P.P., Mori, J.M., Nahar, A.B.: Hardware co-simulation for Chroma-keying in real time. In: International Conference on Computing Communication Control and Automation. pp. 863–867 (Feb 2015)
3. Lee, H.M., Ryu, N.H., Kim, E.K.: Depth map based real time 3d virtual image composition. In: 17th International Conference on Advanced Communication Technology (ICACT). pp. 217–220 (July 2015)
4. Nishitsuji, T., Shimobaba, T., Kakue, T., Ito, T.: Review of fast calculation techniques for computer-generated holograms with the point light source-based model. IEEE Transactions on Industrial Informatics PP(99), 1–1 (2017)
5. Stanton, A.: Wall-E. Walt Disney Pictures - Pixar Animation Studios (2008)

6. Su, P., Cao, W., Ma, J., Cheng, B., Liang, X., Cao, L., Jin, G.: Fast computer-generated hologram generation method for three-dimensional point cloud model. Journal of Display Technology 12(12), 1688–1694 (Dec 2016)
7. Wang, J., Zheng, H.D., Yu, Y.J.: Achromatization in optical reconstruction of computer generated color holograms. Journal of Display Technology 12(4), 390–396 (April 2016)

ISSN 1870-4069

# Historial y reversibilidad en el sublenguaje clásico QML

Nely Plata César, José Raymundo Marcial Romero

Universidad Autónoma del Estado de México, México

**Resumen.** La presente investigación incorpora un historial de cálculos, ajusta el modelo operacional y define las reglas para aplicar reversibilidad en el lenguaje de programación cuántico QML. Sólo se abordarán las instrucciones clásicas del lenguaje mencionado. Esto será el preámbulo para incorporar historial y reversibilidad contemplando datos y control cuántico.

**Palabras clave:** lenguaje de programación cuántico, QML, reversibilidad, historial.

## History and Reversibility in the Classic Sublenguage QML

**Abstract.** The present investigation incorporates a history of calculations, adjusts the operational model and of ne the rules to apply reversibility in the quantum programming language QML. Only the classic instructions of the aforementioned language will be addressed. This will be the preamble to incorporate history and reversibility contemplating data and quantum control.

**Keywords:** quantum programming language, QML, reversibility, history.

## 1. Introducción

El cómputo cuántico se enfoca en aplicar los principios de la mecánica cuántica, de tal manera que sus propiedades se hereden a la computación y al desarrollo de la misma. Las aplicaciones y cálculos que se pueden realizar bajo estos sistemas son múltiples; por ejemplo, el estudio y desarrollo de lenguajes de programación cuánticos. Actualmente, debido a que no se cuenta con una computadora cuántica, estos lenguajes están siendo implementados en máquinas clásicas [14,7,18,3,20]. Existen cuatro principios o postulados que rigen el comportamiento del cómputo cuántico:

El primero, son estados puros; es decir, utilizar vectores unitarios, los cuales se encargarán de almacenar la información en memoria. La evolución del sistema indica cómo cambiar de un estado a otro, para lo cual se aplican matrices unitarias, éstas tienen implícitas las instrucciones que se desean que efectúe la

computadora y serán aplicadas al estado que se desea evolucionar. El tercero es el axioma de medición u observación, se centra en que existe un observador que al ver el estado del sistema éste se colapsa a ese valor y se determina con qué probabilidad sucede. Por último, el cuarto principio es el de sistemas compuestos, encargado de definir estados conformados por distintos sistemas y el cómo interactúan entre ellos [13,15].

Una propiedad que debe tener un sistema cuántico, es la reversibilidad, es decir, si se está en un punto de algún cálculo, se puede regresar al paso anterior y así sucesivamente (sin mediciones intermedias). Esto se logra debido a que las operaciones están representadas por matrices unitarias y éstas tienen implícita tal propiedad; entonces, al aplicar dos veces la misma matriz que llevó a un estado actual de la computadora, se puede regresar al paso anterior.

Hay investigaciones que estudian cómo convertir cómputos clásicos o cuánticos en reversibles, así como técnicas para su construcción [16,1,10,8,22]. El definir explícitamente la reversibilidad permitirá reconstruir un programa o saber que sucedió antes de cualquier cálculo.

Una de las posibles ideas para poder aplicar explícitamente reversibilidad, es almacenar ordenadamente las operaciones ejecutadas durante las derivaciones de un programa, al resultado de lo que se guardó se le puede llamar historial [21]. Partiendo de lo anterior, el vínculo entre la reversibilidad e historial de cálculos es el área que se abordará, esto enfocado en el lenguaje de programación cuántico QML.

De manera general, en la literatura existen lenguajes de programación cuánticos basados en los paradigmas imperativo y funcional [14]. Dentro de los funcionales, se encuentran algunos como QPL, QML, cálculo lambda cuántico, nQML, entre otros [4,11,21,18,19].

El lenguaje QML, es propuesto por Grattage y Altenkirch, quienes desarrollan una semántica denotacional y operacional, aplican propiedades cuánticas, teoría de categorías y lógica [4,9]. Para abordar el cómo modelar la reversibilidad en QML, algunos autores han propuesto alternativas que no estrictamente van enfocadas a este lenguaje, pero resultan funcionales para nuestro propósito.

El artículo se estructura de la siguiente forma: En la sección 2 están las investigaciones de interés acerca de reversibilidad e historial, también se encontrará la definición del lenguaje de programación cuántico QML, puntualizando en el sublenguaje clásico del mismo. La sección 3, presenta nuestra aportación respecto al historial de cálculos y reversibilidad en QML, definiendo sus reglas de funcionamiento. En el bloque 4, está la propuesta del modelo operacional y la incorporación de las funciones inversas relacionadas al mismo, y, finalmente, la sección 5, presenta las conclusiones y trabajo a futuro.

## 2. Trabajo relacionado

En esta sección mencionaremos la investigación base para trabajar una pila de historial y con ésta aplicar reversibilidad, así como lo relacionado con QML, esto es, su sintaxis, tipos, reglas, etc.

### 2.1. Historial y reversibilidad

Uno de los pioneros en abordar la reversibilidad es Landauer [12], quien aporta la idea de que una computadora puede efectuar la reversibilidad si se guarda su historial de cómputos [17]. El historial, expresa o plasma los antecedentes de determinados procesos computacionales.

Otros investigadores sobresalientes son Abramsky y Bennett, quienes mostraron que el cómputo clásico puede ser transformado en cómputo reversible [2,6]. Partiendo de tales autores, Van Tonder [21] retoma sus ideas, en particular de Bennett [6], esto para incorporar el historial de operaciones en el lenguaje cálculo lambda cuántico. De esto, se define que la investigación de Van Tonder se considerará base para nuestra propuesta.

Posteriormente, Grattage, Altenkirch, Vizzotto y Sabry, retoman QML y desarrollan un modelo operacional y teoría ecuacional (omitiendo mediciones) abordando primero su parte clásica y los resultados extrapolados a lo cuántico [5]. Esta investigación es sobresaliente, porque determina una forma de trabajar un lenguaje cuántico; es decir, primero modelar cualquier propiedad en el entorno clásico, y una vez concluido, extenderlo a lo cuántico.

Descrito lo anterior, se establece que este artículo se enfocará en las dos investigaciones previas; es decir, Van Tonder y Grattage, Altenkirch, Vizzotto y Sabry, fusionándose para implementar una pila de historial en el sublenguaje clásico de QML, lo cual, será nuestro aporte central.

### 2.2. Sublenguaje clásico QML

Lenguaje que modela operaciones reversibles e irreversibles (a través de circuitos), utiliza datos cuánticos y control clásico, emplea teoría de categorías para modelar circuitos cuánticos, define una semántica operacional y denotacional, y desarrolla un simulador elemental en Haskell [9,4,20].

Para este caso se considera sólo el subconjunto de términos que excluyen elementos y superposiciones cuánticas. Este sublenguaje de forma precisa tiene su sintaxis, reglas para términos bien formados y su modelo operacional [5].

Su sintaxis se observa en la Tabla 1, los programas que se pueden formar son los convencionales como let, if, valores true, false, variables y combinación de estos. Los elementos de esta sintaxis tienen asociado un tipo, y los tipos están dados por la siguiente gramática $\sigma : \mathcal{Q}_1 | \mathcal{Q}_2 | \sigma \otimes \tau$. $\mathcal{Q}_2$ corresponde al qubit, es decir, el conjunto de 0 y 1, mientras $\mathcal{Q}_1$ no acarrea información y es el valor unitario denotado como $()$. Los tipos son finitos y no recursivos.

Los contextos de tipos $(\Gamma, \Delta)$ están dados por $\Gamma, x : \sigma = \bullet \mid \Gamma, x : \sigma$, donde, $\bullet$ es el conjunto vacío. Tales contextos se conciben como funciones que dado un conjunto finito de variables devuelven tipos; dentro de un programa se asume que cada variable definida aparece al menos una vez. Para mapear pares de contextos a contextos, se agrega el operador $\otimes$, realizando lo que se presenta en la Tabla 2.

Se define una expresión de la forma $\Gamma \vdash t : \sigma$, esto es, que dado $\Gamma$ se deriva el término $t$ del tipo $\sigma$. Ésta es interpretada por la función: $[\![\Gamma \vdash t : \sigma]\!] \in$

**Tabla 1.** Sintaxis del sublenguaje clásico QML.

| | | |
|---|---|---|
| (Variables) | $x, y, \ldots \in$ | $Vars$ |
| (Amplitudes de prob.) | $\kappa, \iota, \ldots \in$ | $\mathbb{C}$ |
| (Patrones) | $p, q ::=$ | $x \mid (x, y)$ |
| (Términos) | $t, u ::=$ | $x$ |
| | | $\mid () \mid (t, u)$ |
| | | $\mid$ **let** $p = t$ **in** $u$ |
| | | $\mid$ **false** $\mid$ **true** $\mid$ **0** |

**Tabla 2.** Mapeo de contextos a contextos.

| | |
|---|---|
| $\Gamma, x : \sigma \otimes \Delta, x : \sigma$ | $= (\Gamma \otimes \Delta), x : \sigma$ |
| $\Gamma, x : \sigma \otimes \Delta$ | $= (\Gamma \otimes \Delta), x : \sigma$ si $x \notin dom\Delta$ |
| $\bullet \otimes \Delta$ | $= \Delta$ |

$[\![\Gamma]\!] \to [\![\sigma]\!]$, tal que, un contexto devuelve un tipo de la coleccion de qubit. Y semánticamente los tipos se definen como:

$$[\![Q_1]\!] = \{0\},$$
$$[\![Q_2]\!] = \{0, 1\},$$
$$[\![\sigma \otimes \tau]\!] = [\![\sigma]\!] \times [\![\tau\ - NoValue -]\!],$$

donde el 0 se asocia con falso y 1 con verdadero. Una vez que se tiene la sintaxis y tipos, se pueden generar programas; sin embargo, hay reglas para definirlos adecuadamente y éstas se observan en la Figura 1, página 29 en [5].

### 2.3. Modelo operacional

Se mencionará el modelo operacional del sublenguaje clásico QML propuesto por los autores respectivos, éste describe las reglas de derivación para cada unos de los términos, está basado en composiciones y funciones auxiliares (página 30 en [5]) que permitirán ir evolucionando un programa hasta llegar a un resultado. Se presenta en la Tabla 3.

Considerar que a este modelo incorporaremos el historial de operaciones, implicando ajustar las funciones relacionadas con el mismo. Para observar su funcionamiento original se presenta el siguiente ejemplo (1) relacionado a la operación clásica *not*.

**Ejemplo 1** $not\ false = \textbf{\textit{if}}^{\circ}\ false\ \textbf{\textit{then}}\ false\ \textbf{\textit{else}}\ true$
$[\![\bullet \otimes \bullet \vdash \textbf{\textit{if}}^{\circ} false\ \textbf{\textit{then}}\ false\ \textbf{\textit{else}}\ true : Q_2]\!] = (g|h) \circ (f \times id) \circ \delta_{\Gamma,\Delta}$

*Esto es:*

$$\begin{aligned}
[\![\bullet \otimes \bullet \vdash not\ false : Q_2 ]\!] &= \big(const\ 0 \mid const\ 1\big) \circ \big(const\ 0 \times id\big) \circ \big(id^*\big) \\
&= \big(const\ 0 \mid const\ 1\big) \circ \big(const\ 0 \times id\big) \circ id^*(0) \\
&= \big(const\ 0 \mid const\ 1\big) \circ \big(const\ 0 \times id\big)(0,0) \\
&= \big(const\ 0 \mid const\ 1\big) \circ \big(const\ 0\ (0),\ id(0)\big) \\
&= \big(const\ 0 \mid const\ 1\big)(0,0) \\
&= const\ 1\ 0 \\
&= 1
\end{aligned}$$

*Devuelve el valor 1 (true) e invierte exitosamente el argumento inicial 0 (false).*

Partiendo de lo anterior, procedemos con nuestra investigación, la cual consistirá en: agregar una pila de historial que se obtendrá a partir de modificar el modelo operacional, las reglas para aplicar reversibilidad y la incorporación de funciones inversas.

**Tabla 3.** Modelo operacional.

$$\begin{aligned}
[\![\bullet \vdash () : Q_1 ]\!] &= const\ 0 \\
[\![\bullet \vdash false : Q_2 ]\!] &= const\ 0 \\
[\![\bullet \vdash true : Q_2 ]\!] &= const\ 1 \\
[\![x : \sigma \vdash x : \sigma ]\!] &= id_* \\
[\![\Gamma \otimes \Delta \vdash \mathbf{let}\ x = t\ \mathbf{in}\ u : \sigma ]\!] &= g \circ (f \times id) \circ \delta_{\Gamma,\Delta} \\
&\text{donde:} \\
&\quad f = [\![\Gamma \vdash t : \sigma ]\!] \\
&\quad g = [\![\Delta,\ x : \sigma \vdash u : \tau ]\!] \\
[\![\Gamma \otimes \Delta \vdash (t,u) : \sigma \otimes \tau ]\!] &= (f \times g) \circ \delta_{\Gamma,\Delta} s \\
&\text{donde:} \\
&\quad f = [\![\Gamma \vdash t : \sigma ]\!] \\
&\quad g = [\![\Delta \vdash u : \tau ]\!] \\
[\![\Gamma \otimes \Delta \vdash \mathbf{let}\ (x,y) = t\ \mathbf{in}\ u : \rho ]\!] &= g \circ (f \times id) \circ \delta_{\Gamma,\Delta} \\
&\text{donde:} \\
&\quad f = [\![\Gamma \vdash t : \sigma \otimes \sigma ]\!] \\
&\quad g = [\![\Delta,\ x : \sigma,\ y : \tau \vdash u : \rho ]\!] \\
[\![\Gamma \otimes \Delta \vdash \mathbf{if}^\circ\ c\ \mathbf{then}\ t\ \mathbf{else}\ u : \sigma ]\!] &= (g|h) \circ (f \times id) \circ \delta_{\Gamma,\Delta} \\
&\text{donde:} \\
&\quad f = [\![\Gamma \vdash c : Q_2 ]\!] \\
&\quad g = [\![\Delta \vdash t : \sigma ]\!] \\
&\quad h = [\![\Delta \vdash u : \sigma ]\!] \\
[\![\Gamma \vdash t : \sigma ]\!] &= f \times id^* \\
&\text{donde:} \\
&\quad f = [\![\Gamma,\ x : Q_1 \vdash t : \sigma ]\!]
\end{aligned}$$

## 3. Historial y reversibilidad

En esta sección se encuentra una de nuestras aportaciones al implementar una pila de historial en el sublenguaje clásico QML.

Considerando que cada derivación está dada a partir de funciones auxiliares que se encargan de realizar ciertas operaciones, entonces, sus funciones inversas serán almacenadas en un historial. Esto se irá formalizando a continuación.

**Definición 1 (Estado computacional)** *Un* estado computacional *es una secuencia de la forma:*

$$\mathbf{h_{t_n}\_}; \cdots ; \mathbf{h_{t_2}\_}; \mathbf{h_{t_1}\_}; t_1 \circ t_2 \circ \cdots \circ t_n$$

*donde,* $\mathbf{h_{t_n}\_}; \cdots ; \mathbf{h_{t_2}\_}; \mathbf{h_{t_1}\_}$ *es la* pila del historial *y* $t_1 \circ t_2; \circ \cdots \circ t_n$ *el* registro computacional.

**Definición 2 (Factorización)** *Dada una expresión de la forma:*

$$\mathbf{h_{t_1}\_}; t_1 \circ \mathbf{h_{t_2}\_}; t_2 \circ \cdots \circ \mathbf{h_{t_n}\_}; t_n$$

*el símbolo* ***;*** *denota concatenación de expresiones. Las expresiones* $h_{t_i}$*, son historiales correspondientes a cada término* $t_i$*, respectivamente. Estos deben ser factorizados cómo:*

$$\mathbf{h_{t_1}\_}; t_1 \circ \mathbf{h_{t_2}\_}; t_2 \circ \cdots \circ \mathbf{h_{t_n}\_}; t_n = \mathbf{h_{t_n}\_}; \cdots ; \mathbf{h_{t_2}\_}; \mathbf{h_{t_1}\_}; t_1 \circ t_2 \circ \cdots \circ t_n$$

Es decir, los historiales $h_{t_i}$ se colocan del lado izquierdo de la expresión, de manera inversa y consecutiva respecto al orden de aparición. Y respecto a $h_{t_i}\_; t_i$, la parte $h_{t_i}$ representa y almacena la operación inversa que ejecuta $t_i$.

**Propiedad 1** *Sea un estado computacional (resultado de derivaciones) con la forma:*

$$\mathbf{h_{t_n}\_}; \cdots ; \mathbf{h_{t_2}\_}; \mathbf{h_{t_1}\_}; q,$$

*donde* $q$ *es un término irreducible (por el modelo operacional).*

A una expresión con esta forma, se puede aplicar *reversibilidad* a partir de la pila del historial $(h_{t_1}\_; h_{t_2}\_; \cdots ; h_{t_n}\_)$. Considerando los casos:

I) $h_{t_i}\_$. Pasar como argumento $q$ al historial inmediato anterior, remplazandolo por el símbolo $\_$, y aplicando la función respectiva. Es decir:

$$h_{t_i}\_; q = h_{t_i}(q)$$

II) $(h_{t_i}\_ \times h_{t_j}\_)$. El argumento $q$ debe ser una tupla $(q_1, q_2)$, y se aplica al historial en la forma:

$$(h_{t_i}\_ \times h_{t_j}\_)(q_1, q_2) = (h_{t_i}\ q_1, h_{t_j}\ q_2)$$

Esto se ejemplifica de forma general a continuación.

**Ejemplo 2** *Sea un estado computacional con el argumento inicial (p); posterior a la ejecución de todas las funciones $t_i$ se tiene:*

$$\begin{aligned}\mathbf{h_{t_n}}-;\cdots;\mathbf{h_{t_2}}-;\mathbf{h_{t_1}}-;t_1\circ t_2\circ\cdots\circ t_n(p) &= \mathbf{h_{t_n}}-;\cdots;\mathbf{h_{t_2}}-;\mathbf{h_{t_1}}-;t_1\circ t_2;\circ\cdots\circ(p')\\ &= \vdots\\ &= \mathbf{h_{t_n}}-;\cdots;\mathbf{h_{t_2}}-;\mathbf{h_{t_1}}-;(p''')\end{aligned}$$

*A partir de la pila del historial se puede aplicar reversibilidad, como:*

$$\begin{aligned}\mathbf{h_{t_n}}-;\cdots;\mathbf{h_{t_2}}-;\mathbf{h_{t_1}}-;(p''') &= \mathbf{h_{t_n}}-;\cdots;\mathbf{h_{t_2}}-;\mathbf{h_{t_1}}(p''')\\ &= \mathbf{h_{t_n}}-;\cdots;\mathbf{h_{t_2}}-;(p'')\\ &= \mathbf{h_{t_n}}-;\cdots;\mathbf{h_{t_2}}(p'')\\ &= \mathbf{h_{t_n}}-;\cdots;(p')\\ &= \mathbf{h_{t_n}}-;(p')\\ &= \mathbf{h_{t_n}}(p')\\ &= (p)\end{aligned}$$

*Retornando al valor inicial dado.*

La siguiente etapa es implementar la pila del historial en el modelo operacional; para esto, se agregan las funciones inversas, esperando obtener el registro computacional. Esto de muestra a continuación.

## 4. Modelo operacional incorporando historial

Dado el modelo de la Tabla 3, las reglas de cada uno de los términos donde la función es explícita, se concatena su función inversa, por ejemplo $false$, $true$, $x$ y otros casos. Con esto, la combinación de las funciones y sus inversas darán pie a un estado computacional, donde el historial se distingue del término por el símbolo; y el tipo de letra negra (Tabla 4).

**Tabla 4.** Modelo operacional con historial.

$$
\begin{aligned}
[\![\bullet \vdash () : Q_1 ]\!] &= (\boldsymbol{const\ 0})^{-1}_{\_}; const\ 0 \\
[\![\bullet \vdash false : Q_2 ]\!] &= (\boldsymbol{const\ 0})^{-1}_{\_}; const\ 0 \\
[\![\bullet \vdash true : Q_2 ]\!] &= (\boldsymbol{const\ 1})^{-1}_{\_}; const\ 1 \\
[\![x : \sigma \vdash x : \sigma ]\!] &= \boldsymbol{id^*}_{\_}; id_* \\
[\![\Gamma \otimes \Delta \vdash \mathbf{let}\ x = t\ \mathbf{in}\ u : \sigma ]\!] &= g \circ (f \times \boldsymbol{id}^{-1}_{\_}; id) \circ ((\boldsymbol{\delta_{\Gamma,\Delta}})^{-1}_{\_}; \delta_{\Gamma,\Delta}) \\
&\text{donde:} \\
&\qquad f = [\![\Gamma \vdash t : \sigma ]\!] \\
&\qquad g = [\![\Delta,\ x : \sigma \vdash u : \tau ]\!] \\
[\![\Gamma \otimes \Delta \vdash (t,u) : \sigma \otimes \tau ]\!] &= (f \times g) \circ ((\boldsymbol{\delta_{\Gamma,\Delta}})^{-1}_{\_}; \delta_{\Gamma,\Delta}) \\
&\text{donde:} \\
&\qquad f = [\![\Gamma \vdash t : \sigma ]\!] \\
&\qquad g = [\![\Delta \vdash u : \tau ]\!] \\
[\![\Gamma \otimes \Delta \vdash \mathbf{let}\ (x,y) = t\ \mathbf{in}\ u : \rho ]\!] &= g \circ (f \times \boldsymbol{id}^{-1}_{\_}; id) \circ ((\boldsymbol{\delta_{\Gamma,\Delta}})^{-1}_{\_}; \delta_{\Gamma,\Delta}) \\
&\text{donde:} \\
&\qquad f = [\![\Gamma \vdash t : \sigma \otimes \sigma ]\!] \\
&\qquad g = [\![\Delta,\ x : \sigma,\ y : \tau \vdash u : \rho ]\!] \\
[\![\Gamma \otimes \Delta \vdash \mathbf{if}^{\circ}\ c\ \mathbf{then}\ t\ \mathbf{else}\ u : \sigma ]\!] &= (g|h) \circ (f \times \boldsymbol{id}^{-1}_{\_}; id) \circ ((\boldsymbol{\delta_{\Gamma,\Delta}})^{-1}_{\_}; \delta_{\Gamma,\Delta}) \\
&\text{donde:} \\
&\qquad f = [\![\Gamma \vdash c : Q_2 ]\!] \\
&\qquad g = [\![\Delta \vdash t : \sigma ]\!] \\
&\qquad h = [\![\Delta \vdash u : \sigma ]\!] \\
[\![\Gamma \vdash t : \sigma ]\!] &= f \times \boldsymbol{id_*}_{\_}; id^*) \\
&\text{donde:} \\
&\qquad f = [\![\Gamma,\ x : Q_1 \vdash t : \sigma ]\!]
\end{aligned}
$$

Para poder derivar adecuadamente un término, también se agregan las funciones inversas, esto con la idea de aplicar la operación que previamente se efectuó.

**Funciones auxiliares inversas** Sea $S = \{0, 1\}$, donde $a, a', b \in S$

- $\boldsymbol{id^{-1}} : S \to S$, por lo tanto, $id^{-1} = id$
- $\boldsymbol{(id^*)^{-1}} : S' \times S \to S$, donde $(id^*)^{-1} = id_*$

$$(id^*)^{-1}(a', a) = \begin{cases} (id^*)^{-1}(a', a) = a, \text{ si a'=0} \\ (id^*)^{-1}(a', a) = a', \text{ si a'=1} \end{cases}$$

- $\boldsymbol{(id_*)^{-1}} : S \to Q_1 \times S$, donde $(id_*)^{-1} = id^*$
- $\boldsymbol{(const\ 0)^{-1}} : S \to Q_1$, tal que

$$(const\ 1)^{-1}(a) = \begin{cases} 1, \text{ si a=0} \\ 0, \text{ si a=1} \end{cases}$$

- $\boldsymbol{\delta^{-1}} : (S, S) \to S$, donde $\delta^{-1}(a, a) = a$.
- $\boldsymbol{swap^{-1}} : T \times S \to S \times T$, donde $swap^{-1}(b, a) = (a.b)$.

- $\boldsymbol{(f \times g)^{-1}} : (T_1 \times T_2) \to (S_1 \times S_2)$,
  donde $(f \times g)^{-1} = (f^{-1} \times g^{-1})$, por lo tanto

$$(f \times g)^{-1}(a, b) = (f^{-1}\ a, g^{-1}\ b)$$

- $\boldsymbol{(\delta_{\Gamma,\Delta})^{-1}} : [\![\Gamma]\!] \times [\![\Delta]\!] \to [\![\Gamma \otimes \Delta]\!]$

$$(\delta_{\Gamma,\Delta})^{-1} = \begin{cases} (\delta_{\Gamma',\Delta'})^{-1} \times \delta^{-1} & \text{si } \Gamma = \Gamma', x : \sigma \text{ y } \Delta = \Delta', x : \sigma \\ (\delta_{\Gamma',\Delta})^{-1} \times id & \text{si } \Gamma = \Gamma', x : \sigma \text{ y } x \notin dom(\Delta) \\ id_* & \text{si } \Gamma = \bullet \end{cases}$$

- Condicional:
  $\boldsymbol{(f^{-1}|g^{-1})} : T \to ([\![Q_2]\!] \times S)$, donde:

$$(f^{-1}|g^{-1})(s) = \begin{cases} (a, f^{-1}\ a) & \text{si a=1} \\ (a, g^{-1}\ a) & \text{si a=0} \end{cases}$$

Con esto, se conluyen las definiciones respecto a historial y reversibilidad. Para poder observar su funcionamiento, se procede a desarrollar un programa con los resultados previamente descritos.

### 4.1. Aplicación

Sea el programa *not* con reversibilidad:

$$not\ false = if^{\circ}\ false\ \textbf{then}\ false\ \textbf{else}\ true$$

$$[\![\bullet \otimes \bullet \vdash \textbf{if}^{\circ} false\ \textbf{then}\ false\ \textbf{else}\ true : Q_2]\!] = (g|h) \circ (f \times \boldsymbol{id_{-}^{-1}}; id) \circ (\boldsymbol{\delta_{\Gamma,\Delta}^{-1}}_{-}; \delta_{\Gamma,\Delta})$$

Donde:

$$\begin{aligned} f &= [\![\bullet \vdash false : Q_2]\!] = \boldsymbol{const\ 0_{-}}; const\ 0 \\ g &= [\![\bullet \vdash false : Q_2]\!] = \boldsymbol{const\ 0_{-}}; const\ 0 \\ h &= [\![\bullet \vdash true : Q_2]\!] = \boldsymbol{const\ 1_{-}}; const\ 1 \\ \delta_{\Gamma,\Delta} &= id^* \end{aligned}$$

Por lo tanto:

$$[\![\bullet \otimes \bullet \vdash not\ false : Q_2]\!] = \Big((\boldsymbol{const\ 0_{-}}; const\ 0)|(\boldsymbol{const\ 1_{-}}; const\ 1)\Big) \circ \Big((\boldsymbol{const\ 0_{-}}; const\ 0) \times (\boldsymbol{id_{-}}; id)\Big) \circ \Big((\boldsymbol{id_{*-}}; id^*)\Big)$$

Factorizando:

$$
\begin{aligned}
&= \Big( (\boldsymbol{const\ 0\_} | \boldsymbol{const\ 1\_}); (const\ 0 | const\ 1) \Big) \circ \Big( (\boldsymbol{const\ 0\_} \times \boldsymbol{id\_}); \\
&\quad (const\ 0 \times id) \Big) \circ \Big( \boldsymbol{id_{*}\_}; id^{*} \Big) \\
&= \Big( \boldsymbol{id_{*}\_; (const\ 0\_} \times \boldsymbol{id\_); (const\ 0\_} | \boldsymbol{const\ 1\_) } \Big); \\
&\qquad \Big( (const\ 0 | const\ 1) \circ (const\ 0 \times id) \circ id^{*}(0) \Big) \text{ se pasa el argumento } 0 \\
&= \Big( \boldsymbol{id_{*}\_; (const\ 0\_} \times \boldsymbol{id\_); (const\ 0\_} | \boldsymbol{const\ 1\_) } \Big); \\
&\qquad \Big( (const\ 0 | const\ 1) \circ (const\ 0 \times id)(0,0) \Big) \\
&= \Big( \boldsymbol{id_{*}\_; (const\ 0\_} \times \boldsymbol{id\_); (const\ 0\_} | \boldsymbol{const\ 1\_) } \Big); \\
&\qquad \Big( (const\ 0 | const\ 1) \circ (const\ 0\ 0, \times id\ 0) \Big) \\
&= \Big( \boldsymbol{id_{*}\_; (const\ 0\_} \times \boldsymbol{id\_); (const\ 0\_} | \boldsymbol{const\ 1\_) } \Big); \\
&\quad \Big( (const\ 0 | const\ 1)(0,0) \Big) \\
&= \Big( \boldsymbol{id_{*}\_; (const\ c\_} \times \boldsymbol{id\_); (const\ 0\_} | \boldsymbol{const\ 1\_) } \Big); \\
&\quad \Big( (const\ 0 | const\ 1)(0,0) \Big) \\
&= \Big( \boldsymbol{id_{*}\_; (const\ c\_} \times \boldsymbol{id\_); (const\ 0\_} | \boldsymbol{const\ 1\_) } \Big); \Big( const\ 1\ (0) \Big) \\
&= \Big( \boldsymbol{id_{*}\_; (const\ c\_} \times \boldsymbol{id\_); (const\ 0\_} | \boldsymbol{const\ 1\_) } \Big); \Big( 1 \Big)
\end{aligned}
$$

Aplicando **reversibilidad**: Si el argumento inicial fue 0 y se obtuvo 1, entonces se espera regresar a 0.

$$
\begin{aligned}
(\mathbf{id}_{*}\_); (\mathbf{const\ 0}\_ \times \mathbf{id}\_); (\mathbf{const\ 0}\_ | \mathbf{const\ 1}\_); (1) = (\mathbf{id}_{*}\_); (\mathbf{const\ 0}\_ \times \mathbf{id}\_); \\
(\mathbf{const\ 0}\_ | \mathbf{const\ 1}\_)(1)
\end{aligned}
$$

$$
\begin{aligned}
&= (\mathbf{id}_{*\_}); (\mathbf{const}\ \mathbf{0}_{\_} \times \mathbf{id}_{\_}); \\
&\quad (1, const\ 0\ 1) \\
&= (\mathbf{id}_{*\_}); (const\ 0_{\_} \times id_{\_}) \\
&\quad (1, 0) \\
&= (\mathbf{id}_{*\_}); (const\ 0\ 1,\ id\ 0) \\
&= (\mathbf{id}_{*\_})(0, 0) \\
&= id_{*}(0, 0) \\
&= 0
\end{aligned}
$$

Hasta este punto, se tiene incorporado el historial para el sublenguaje clásico de QML, permitiendo aplicar reversibilidad exitosamente.

## 5. Conclusiones y trabajo a futuro

Se consideró el sublenguaje clásico del lenguaje de programación cuántico QML, del cual, se retoma su modelo operacional respectivo, al cual, se incorporó un historial de cálculos.

El modelo operacional con historial, se desarrolla incorporando las funciones inversas que fueron aplicadas al derivar un programa. Las operaciones ejecutadas generaron un pila de historial, ésta permite la aplicación de reversibilidad.

La reversibilidad se ejecuta al aplicar cada uno de los historiales respectivos de la pila mencionada, permitiendo regresar a un paso anterior, y así sucesivamente, hasta el inicial.

Como trabajo a futuro, se sugiere agregar el historial al lenguaje completo de QML, considerando datos y control cuántico. Y definir la forma de modelar la reversibilidad a partir de operaciones cuánticas.

## Referencias


1. Abdessaied, N., Amy, M., Drechsler, R., Soeken, M.: Complexity of reversible circuits and their quantum implementations. Theoretical Computer Science 618, 85–106 (2016)
2. Abramsky, S.: A structural approach to reversible computation. Theoretical Computer Science 347(3), 441–464 (2005)
3. Abramsky, S., Coecke, B.: Physical traces: Quantum vs. classical information processing. CoRR cs.CG/0207057 (2002)
4. Altenkirch, T., Grattage, J.: A functional quantum programming language. In: 20th Annual IEEE Symposium on Logic in Computer Science (LICS' 05). pp. 249–258 (2005)
5. Altenkirch, T., Grattage, J., Vizzotto, J.K., Sabry, A.: An algebra of pure quantum programming. Electronic Notes in Theoretical Computer Science 170, 23–47 (2007), proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005)

6. Bennett, C.H.: Logical reversibility of computation. IBM J. Res. Dev. 17(6), 525–532 (Nov 1973)
7. Gay, S.J.: Quantum programming languages: Survey and bibliography. Mathematical. Structures in Comp. Sci. 16(4), 581–600 (Aug 2006)
8. Glück, R., Kaarsgaard, R.: A categorical foundation for structured reversible flowchart languages. Electronic Notes in Theoretical Computer Science 336, 155–171 (2018)
9. Grattage, J.J.: A functional quantum programming language
10. Heunen, C., Karvonen, M.: Reversible monadic computing. Electronic Notes in Theoretical Computer Science 319, 217–237 (2015)
11. Lampis, M., Ginis, K.G., Papakyriakou, M.A., Papaspyrou, N.S.: Quantum data and control made easier. Electron. Notes Theor. Comput. Sci. 210, 85–105 (2008)
12. Landauer, R.: Irreversibility and heat generation in the computing process. IBM Journal of Research and Development 5(3), 183–191 (July 1961)
13. McMahon, D.: Quantum Computing Explained (2007)
14. Miszczak, J.A.: High-level structures for quantum computing. Synthesis Lectures on Quantum Computing 4 (05 2012)
15. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press (2010)
16. Nishida, N., Palacios, A., Vidal, G.: Reversible computation in term rewriting. Journal of Logical and Algebraic Methods in Programming 94, 128–149 (2018)
17. Nishida, N., Palacios, A., Vidal, G.: Reversible computation in term rewriting. Journal of Logical and Algebraic Methods in Programming 94, 128–149 (2018)
18. Selinger, P.: A brief survey of quantum programming languages. In: In Proceedings of the 7th International Symposium on Functional and Logic Programming. Springer (2004)
19. Selinger, P.: Towards a semantics for higher-order quantum computation (2004)
20. Sofge, D.A.: A survey of quantum programming languages: History, methods, and tools. In: Second International Conference on Quantum, Nano and Micro Technologies (ICQNM 2008). pp. 66–71 (Feb 2008)
21. van Tonder, A.: A lambda calculus for quantum computation. SIAM J. Comput. 33(5), 1109–1135 (May 2004)
22. Yokoyama, T.: Reversible computation and reversible programming languages. Electronic Notes in Theoretical Computer Science 253(6), 71–81 (2010)

ISSN 1870-4069

# Simulation of Newtonian Flows on Sudden Contraction Geometries: GPU Implementation

Rigo Alvarado[1], Juan J. Tapia[1], Hector D. Ceniceros[2]

[1] Instituto Politécnico Nacional-CITEDI, Tijuana, BC, Mexico
ralvarado@citedi.mx, jtapiaa@ipn.mx
[2] University of California Santa Barbara, UCSB, Santa Barbara, California, USA
hdc@math.ucsb.edu

**Abstract.** In this paper, a GPU-based simulation of a Newtonian fluid flow on sudden contraction geometries is presented. The fluid is modeled with the Navier-Stokes equations and solved by the projection method with first order in time and second order in space discretizations. A semi-implicit scheme of finite differences is used in the dicretization process. The solution of the resulting system of linear equations is considered as an optimization problem and is solved by the preconditioned biconjugate gradient stabilized method (BiCGSTAB) implemented on a graphic processor using the CUDA libraries cuSPARSE and cuBLAS.

**Keywords:** sudden contraction geometry, Navier-Stokes equations, GPU, CUDA.

## 1 Introduction

Sudden contraction geometries for different fluids are used in many areas of engineering and industrial processes such as heating pipes, polymer processing, tube capillary viscometers, biomedical instruments, thermoforming, injection molding, etc. Therefore, understanding and predicting the behavior of fluids in these geometries is of particular importance within fluid mechanics and is a problem that is continuously studied from different perspectives.

For example, in [6], a hybrid model reduction scheme to approximate the Navier-Stokes equations (NSE) with a low-dimensional model on a contraction geometry is presented. The use of a novel projection method based on midpoint rule for the solution of NSE is discussed in [13]. In [8], Guermond and Minev develop a high-order time approximation for the NSE as an alternative for the classical projection method.

However, the high computational cost involved with this type of studies and simulations is a problem that any investigation related to Computational Fluid Dynamics (CFD) and, therefore, researches related to flow on contraction geometries must overcome. This intensive computation is the product of many factors such as the size of the meshes necessary for the correct discretization, the physical domain that needs to be simulated and the number of variables that

must be solved. Therefore, the parallelization of CFD simulations is a topic that constantly receives attention and new proposals.

For example, in [17], Willis presents an MPI implmentation for fluid flow in pipelines. GPU/multicore-based solution to CFD simulations using the NSE are described in [4,10,12] and [16]. An OpenMP-based solution for the NSE on a cavity lid driven is developed in [1].

Therefore, we can state that the motivation for this work and its objective it is very clear: the GPU-based parallelization of the simulation of flow on sudden contraction geometries in order to decrease the computing time and improve the efficiency of the utilization of hardware.

## 2 Mathematical Model

The Navier-Stokes equation models the fluids movement. The non-dimensional vector form of this equation for incompressible fluids and its incompressibility constraint are defined [11] as:

$$\begin{aligned} \frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot (\mathbf{v} \otimes \mathbf{v}) = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{v}, \\ \nabla \cdot \mathbf{v} = 0, \end{aligned} \tag{1}$$

where $\mathbf{v}$ is the adimensional vectorial field of velocity and $p$ is the adimensional scalar field of pressure. In 2D, NSE involves three equations and three unknowns, $u$, $v$ and $p$.

Reynolds number is a dimensionless quantity defined as:

$$Re = \frac{\rho V_o L_o}{\mu}, \tag{2}$$

where $V_0$ is the reference velocity, $L_0$ is the characteristic length and $\rho$ is the fluid density. It represents the ratio between inertial and viscous forces. If it is a small value, the flow occurs in parallel lines and is called laminar flow; if the Reynolds number increases, the ordered structure loses its order, giving rise to a flow characterized by eddies and vortexes, called turbulent flow.

## 3 Numerical Solution

### 3.1 Projection Method

The main idea of the projection method is to temporarily solve the NSE (Eq. (1)) by omitting the pressure $p$ and then project the result into a vector field of solenoid velocity [2,3,14]. The process comprises three main steps:

– First, a temporary velocity field $\mathbf{v}^* = [u^*, v^*]$ is defined as:

$$\mathbf{v}^* = \mathbf{v}_{sol}^{t+1} + \Delta t \nabla p^{t+1}. \tag{3}$$

**Algorithm 1** Projection method for NSE for incompressible fluids

Initialization of variables, arrays and differentiation matrices
Initial condition for velocity field
**for** $iter = 1 : MAX$
Update of boundary conditions
Compute $u^*$ and $v^*$
Solve Poisson equation for pressure $p^{t+1}$
Calculate new velocities $u^*$ and $v^*$ with $p^{t+1}$
**end**

As pressure is neglected, the rectangular components of (1) are:

$$\frac{u^* - u^t}{\Delta t} + u^t \cdot \nabla u^t = \frac{1}{Re}\nabla u^*, \qquad \frac{v^* - v^t}{\Delta t} + v^t \cdot \nabla v^t = \frac{1}{Re}\nabla v^*, \quad (4)$$

for the horizontal and vertical components, respectively.

– Next, the divergence of (3) is calculated to obtain the Poisson equation:

$$\Delta t \nabla^2 p^{t+1} = \nabla \cdot \mathbf{v}^*. \quad (5)$$

The resulting pressure $p^{t+1}$ calculated with (5) is a scalar field that ensures that the final velocity $\mathbf{v}_{sol}^{t+1}$ will meet the incompressibility constraint.

– At last, the final velocity $\mathbf{v}_{sol}^{t+1}$ is computed as:

$$\mathbf{v}_{sol}^{t+1} = \mathbf{v}^* - \Delta t \nabla p^{t+1}. \quad (6)$$

Algorithm 1 shows the pseudo-code for the described projection method.

### 3.2 Problem Specification

We choose the 2:1 and 4:1 ratio contraction geometries (Fig. 1) because they are among the most used in investigations and practical applications [6,17]. Furthermore, the dimensions utilized were chosen to improve numerical accuracy at the contraction region. Also, it is important to mention that this 2D representation corresponds to a longitudinal cross-section of a pipe.

Boundary conditions for the numerical solution are: for the velocity field $\mathbf{v}$, no-slip conditions at the wall ($u = v = 0$), $u = u_j$ (parabolic profile) and $v = 0$ at the inlet (inflow) and $\frac{\partial \mathbf{v}}{\partial \mathbf{n}} = 0$ at the outlet (outflow). The parabolic profile of the horizontal component $u$ of velocity is defined as:

$$u_j = u_{max}\left[1 - \left(\frac{r^2}{R^2}\right)\right], \quad (7)$$

where $u_{max}$ is the maximum value of $u$, right in the middle of the inflow, $R$ is the inlet radius, $r$ is the radius of the $j$-th element of $u$ at the inlet and $j$ is the vertical index for each one of these elements. For the scalar field of pressure $p$, homogeneous Neumann boundary conditions are used ($\frac{\partial p}{\partial \mathbf{n}} = 0$) at all boundaries except at the outlet, where $p = 0$. The described boundary conditions are used for both contractions, i.e. geometries 4:1 and 2:1.

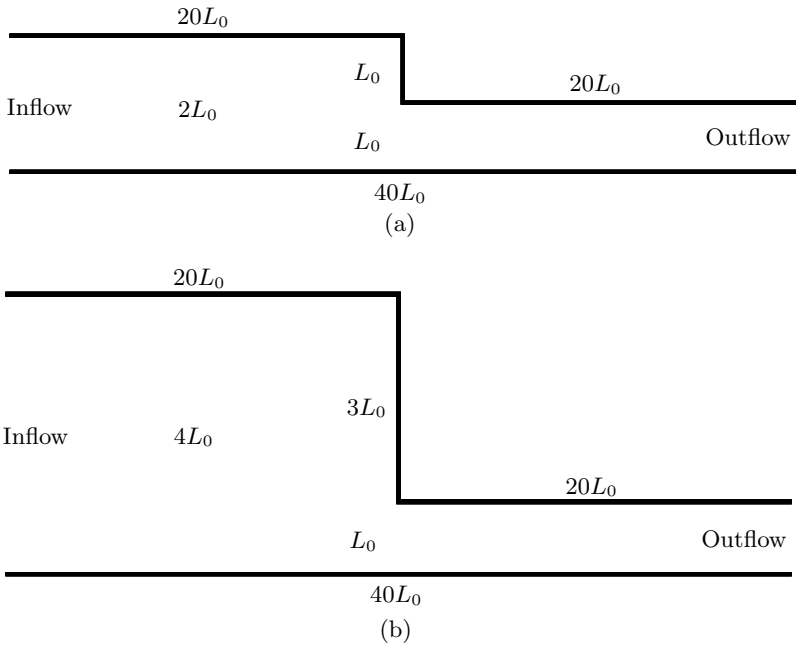



**Fig. 1.** Sudden contraction geometry (a) 2:1 and (b) 4:1

### 3.3 Spatial Discretization

Domain is discretized with a regular mesh and centered finite differences of order $O(h^2)$. Spatial step size for both components depends upon the geometry. For contraction 2:1 (Fig. 1 (a)), $\Delta x = \frac{20L_0}{\text{COLS}}$ and $\Delta y = \frac{2L_0}{\text{ROWS}}$ where $COLS = 20N$ represents the maximum number of cells in the horizontal direction ($x$ axis), $ROWS = 2N$ represents the maximum number of cells in the vertical direction ($y$ axis) and $L_0 = 1$ is the width of the outlet. For contraction 4:1 (Fig. 1 (b)), $\Delta x = \frac{20L_0}{\text{COLS}}$ and $\Delta y = \frac{4L_0}{\text{ROWS}}$ where $ROWS = 4N$ represents the maximum number of cells in the vertical direction. Constant $N$ is an integer number, multiple of two, that is utilized to keep the right proportion between the width and length of the domain. In all simulations presented in this work, $\Delta x = \Delta y$. Also, in order to avoid a checkerboard solution, a full-staggered mesh (Fig. 2) is used in all cases [9].

Discretization for $[u^*, v^*]$ (Eq. (4)) and pressure $p$ (Eq. 5) produces the linear systems of equations:

$$Au^* = RHS_{u^*}, \qquad Bv^* = RHS_{v^*}, \qquad Cp^{t+1} = RHS_p, \tag{8}$$

respectively, where $A$ and $B$ are sparse positive definite matrices, at least for $\Delta t$ reasonably small, and $C$ is a sparse semi-positive definite matrix. Also, $A$, $B$ and $C$ are non-symmetrical. For these reasons, the solution of these systems is con-

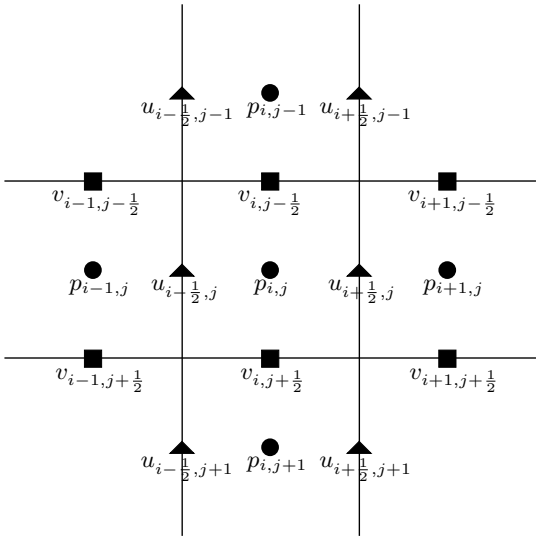



**Fig. 2.** Full-staggered mesh, $2D$

sidered as an optimization problem and hence, the preconditioned BiCGSTAB method is utilized to solve them.

To compute the final velocity, Eq. (6) is discretized as:

$$\begin{aligned} u^{t+1}_{i+1/2,j} &= u^{*}_{i+1/2,j} - \Delta t \frac{p^{t+1}_{i+1,j} - p^{t+1}_{i,j}}{\Delta x}, \\ v^{t+1}_{i,j+1/2} &= v^{*}_{i,j+1/2} - \Delta t \frac{p^{t+1}_{i,j+1} - p^{t+1}_{i,j}}{\Delta y}, \end{aligned} \tag{9}$$

for its horizontal and vertical components respectively.

## 4 General Purpose Computing on GPU: GPGPU

The use of GPUs to solve compute-intensive scientific and engineering applications is known as General-Purpose computing on Graphics Processing Units (GPGPU).

CUDA is a combination of a GPU hardware and a parallel programming model that allows the utilization of NVIDIA GPUs in GPGPU applications.

### 4.1 CUDA Programming Model

The heterogeneous CUDA programming model enables the use of a GPU as a co-processor of the CPU. In this context, GPU is called device and CPU is called host. A CUDA program is composed of serial code sections for the host (on some applications, as in this work, sections of the serial code can be parallelized with

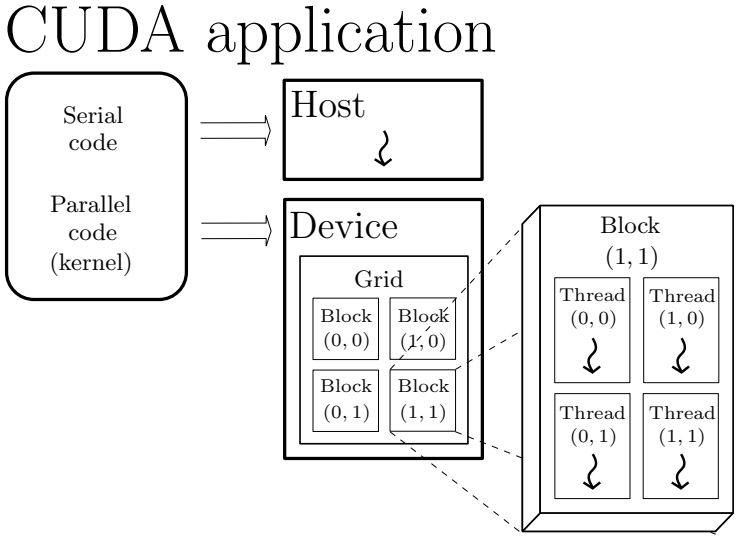



**Fig. 3.** CUDA programming model

OpenMP, MPI, pthreads, etc.) and parallel code sections for the device, called kernels. The serial code is executed by the main thread on the host; the kernels are executed in parallel within the device by a massive number of CUDA threads. This general structure is shown in Fig. 3.

### 4.2 cuSPARSE and cuBLAS Libraries

NVIDIA cuSPARSE library contains a set of basic linear algebra subroutines designed for sparse matrix operations that takes advantage of the CUDA parallel programming model as well as of the computational resources of the NVIDIA GPUs in order to perform efficiently its functions.

NVIDIA cuBLAS library is an implementation of BLAS (Basic Linear Algebra Subprograms) for dense matrices on top of the CUDA runtime. The library

**Algorithm 2** CUDA-based implementation overview

CUSPARSE-CUBLAS initialization
Compute $ilu(0)$ preconditioners for matrices $A$, $B$ and $C$ (cuSPARSE)
Initial condition for **v**
**for** $iter = 1 : MAX$
 Update BC
 Compute $RHS_{u^*}$ and $RHS_{v^*}$ (kernels)
 Solve $Au^* = RHS_{u^*}$ and $Bv^* = RHS_{v^*}$ (cuSPARSE-cuBLAS,BICGSTAB)
 Compute $RHS_p$ (kernel)
 Solve $Cp^{t+1} = RHS_p$ (cuSPARSE-cuBLAS,BICGSTAB)
 Update of $u^*$ and $v^*$ with $p^{t+1}$
**end**

includes matrix-vector and matrix-matrix products. The cuBLAS library also provides functions for writing and retrieving data from the GPU.

CUSPARSE and CUBLAS can be used with C and C++ programming languages. To use the functions of both libraries, they should be initialized, the data must be transferred from the host to the GPU memory and then it must be converted to the corresponding format.

## 5 Parallel Implementation

The systems of linear equations (8) are not suitable to be solved with direct methods due data dependency. Therefore, we choose to solved them as an optimization problem with the BICGSTAB method. Every step of the general BICGSTAB is parallelized with combinations of functions of cuSPARSE and cuBLAS libraries; the $ilu(0)$ preconditioner is also implemented with functions of the cuSPARSE library. In order to use the functions of these libraries, the arrays are copied to the GPU memory and then they are converted to the required formats of sparse matrices. Also, we utilized kernels to compute the RHS vectors for each of the aforementioned systems.

To check the GPU implementation and utilization we used the nvprof and nvvp profilers. The general structure of our implementation is shown in Algorithm 2.

## 6 Results

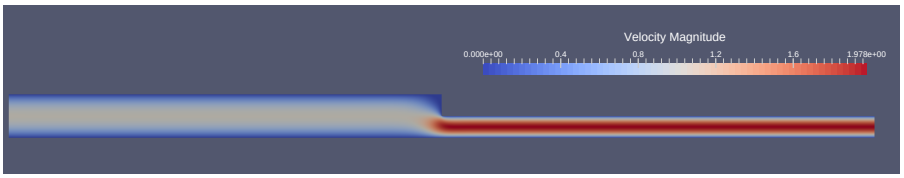
The simulations have been conducted under Ubuntu 14.04 *LTS* using C language on a computer with an *intel i7-4770* processor at 3.4 GHz and 16 GBytes of RAM memory. The CUDA implementation is made with a Maxwell *GeForce GTX* 970 GPU with compute capability 5.2 that has 4 GBytes of global memory. All the results presented are double precision floating point.

The results presented in Section 6.1 and 6.2 are computed with $Re = 1$, which means that there is a balance between inertial and viscous forces. This balance translates into laminar flows, such as the ones shown in those sections. On the other hand, some simulations of flows in which $Re >> 1$ are presented in the Section 6.3.

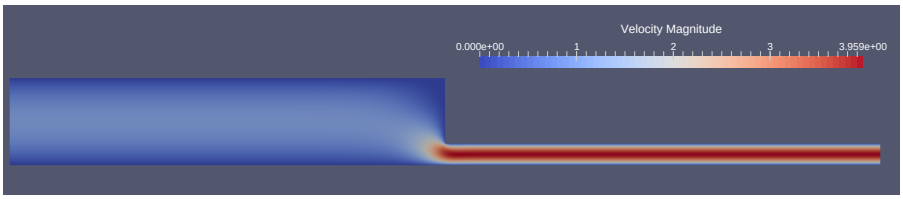
### 6.1 Contraction 2:1

Magnitude of velocity for a contraction 2:1 after 500 iterations is shown in Fig. 4 (a). Domain is discretized with $N = 128$. Furthermore, $u_{max} = 1$, $\Delta t = 0.003$ and $Re = 1$.

Simulation shows that the maximum velocity at the outflow is twice its value at the inflow, i.e. twice the value of $u_{max}$. This agrees with the Bernoulli equation, since the pressure in the outlet is smaller than the pressure in the inlet (Fig. 5 (c)) and, therefore, the velocity at the outlet is greater than the velocity at the

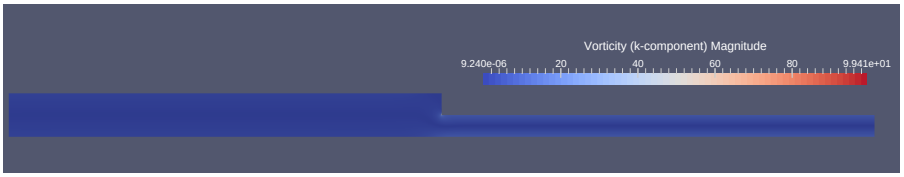



(a)

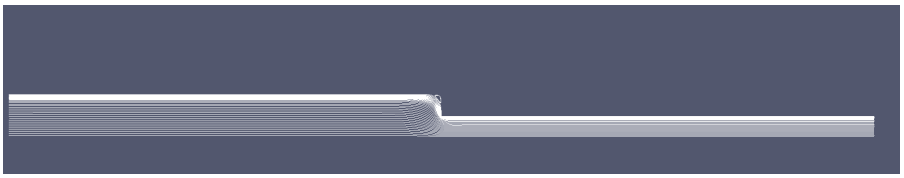



(b)

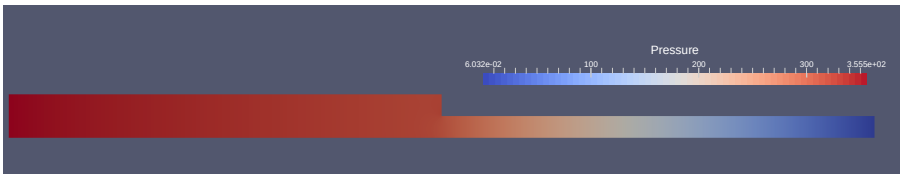
**Fig. 4.** Magnitude of velocity: (a) Contraction 2:1 and (b) Contraction 4:1.




(a)



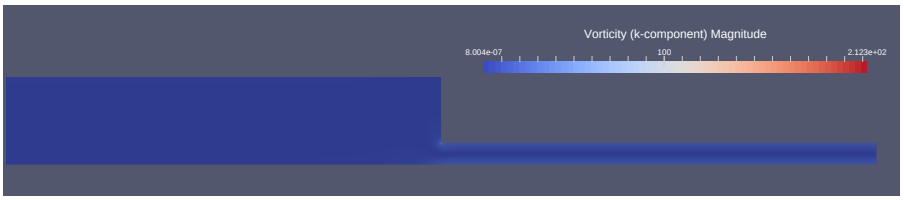
(b)




(c)

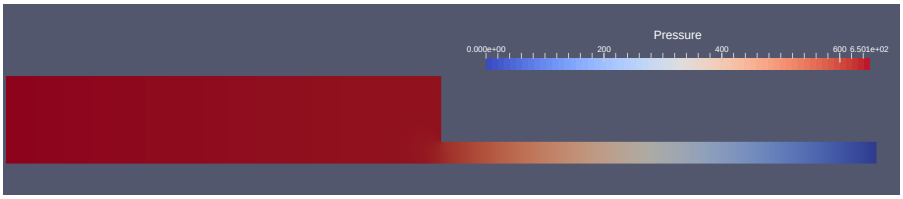
**Fig. 5.** Contraction 2:1. (a) Vorticity magnitude. (b) Streamlines. (c) Pressure

inlet to preserve the total energy of the flow. This behavior is known as the Bernoulli effect.




(a)

(b)




(c)

**Fig. 6.** Contraction 4:1. (a) Vorticity magnitude. (b) Streamlines. (c) Pressure

Magnitude of vorticity, defined as:

$$\omega = \nabla \times \mathbf{v}, \tag{10}$$

is shown in Fig. 5 (a).

It can be seen that the corner at the entrance to the narrow part of the geometry is the point where the vorticity has the largest value. Also, it can be noted that $\omega$ has a relatively high value close to the walls of the narrow section of the contraction. This is due to the fact that the fluid velocity is larger in this section and thus, the difference with the null velocity at the walls is greater here than everywhere else.

Streamlines, i.e. the paths or curves in which the function *stream* $\psi$, computed as:

$$-||\omega|| = \nabla^2 \psi, \tag{11}$$

has a constant value, are shown in Fig. 5 (b). These curves represent the path that would follow a massless particle in the flow and they are instantaneously tangent to the velocity vector.

### 6.2 Contration 4:1

The magnitude of velocity for a contraction 4:1 after 500 iterations is shown in Fig. 4 (b). Domain is discretized with $N = 128$. Furthermore, $u_{max} = 1$, $\Delta t = 0.003$ and $Re = 1$.

Similarly as in the simulation for contraction 2:1, in this case, the velocity and pressure values comply with the Bernoulli effect.

Vorticity magnitude, streamlines and pressure are shown in Fig. 6 (a), (b) and (c) respectively. It can be seen that the point for maximum vorticity, streamlines pattern and pressure behavior are similar to those obtained for contraction 2:1.

### 6.3 Simulations for High-Re Flows

As the value of $Re$ increases, inertial forces overpower viscosity forces and the flow loses its laminar structure and turns into a turbulent flow, which is characterized by eddies, vortexes and chaotic changes in the velocity and pressure.

Fig. 7 (a) and (b), show a close-up of the eddies that arise at the wall of the contraction 2:1 with $Re = 100$ and $Re = 1000$, respectively. Likewise, turbulences generated in the contraction 4:1 for fluids with $Re = 100$ and $Re = 1000$, are shown in Fig. 7 (c) and (d), respectively. For both geometries, $\Delta x = \Delta y = \frac{1}{128}$ and 500 iterations were computed. For the simulations with $Re = 100$, $\Delta t = 0.003$ and for the simulations with $Re = 1000$, $\Delta t = 0.0003$. The flow patterns obtained coincide with results presented in [7] and [15].

### 6.4 Convergence Analysis

Order $m$ of the implementation can be calculated with [5]:

$$m = \frac{\log\left(\frac{f_{\frac{h}{2}}(x,y) - f_h(x,y)}{f_{\frac{h}{4}}(x,y) - f_{\frac{h}{2}}(x,y)}\right)}{\log 2}, \tag{12}$$

if $h$ is sufficiently small and succesive meshes are related by a 2:1 proportion. Discretization error $e$ over the finest mesh is defined as [5]:

$$e_{\frac{h}{4}} \approx \frac{f_{\frac{h}{4}}(x,y) - f_{\frac{h}{2}}(x,y)}{2^m - 1}. \tag{13}$$

In this work, for both geometries, $m$ is computed with $N = 32$, $N = 64$ and $N = 128$. Time step $\Delta t$ is fixed as $(\frac{1}{128})^2$ because the implemented projection method is first order in time and second order in space. As a result, $\Delta t$ should be equal or smaller than the step size of the finest mesh, i.e. $\Delta t \leq \Delta x$. This is to observe second order in $\Delta x$ convergence.

Table 1 shows the results of the analysis for both velocity components of both geometries. The results denote a proper implementation.

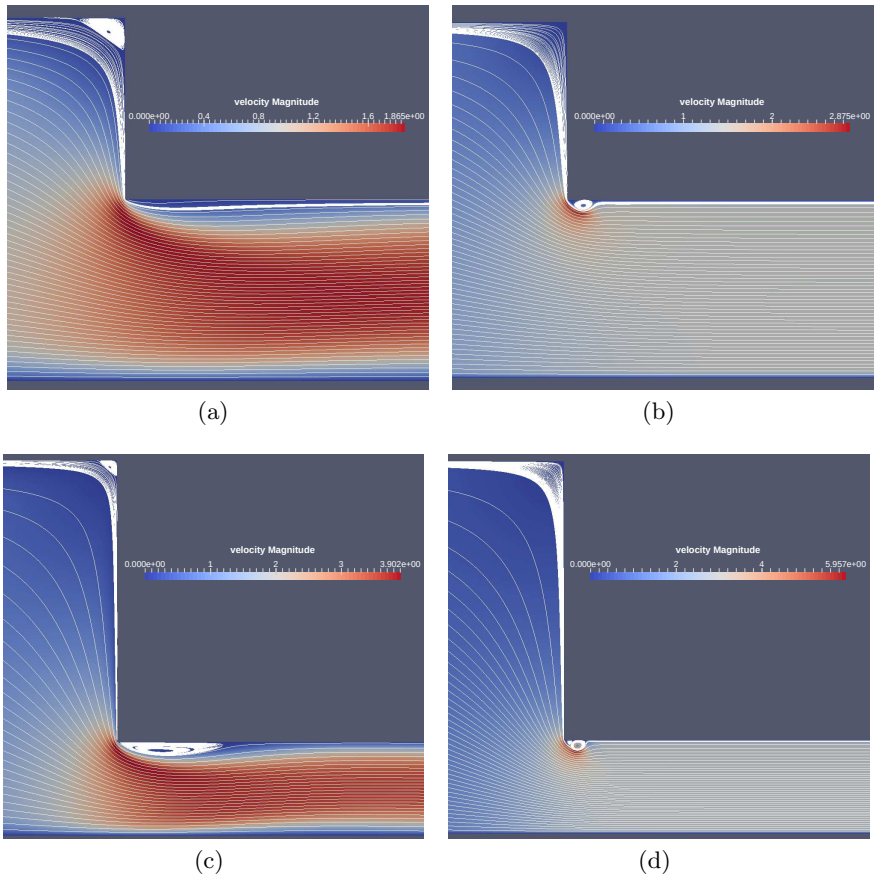



**Fig. 7.** Contraction 2:1. (a) and (b), zoom in of vortexes with $Re = 100$ and $Re = 1000$ respectively. Contraction 4:1. (c) and (d), zoom in of vortexes with $Re = 100$ and $Re = 1000$ respectively

**Table 1.** Order $m$ and error $e$ estimation after 500 iterations for $N = 128$

| **Component** | **Contraction 2:1** | | **Contraction 4:1** | |
|---|---|---|---|---|
| | $m$ | $e$ | $m$ | $e$ |
| $u$ | 2.016 | $1.969 \times 10^{-5}$ | 2.006 | $2.230 \times 10^{-5}$ |
| $v$ | 1.897 | $6.900 \times 10^{-7}$ | 1.959 | $3.8708 \times 10^{-12}$ |

Execution time for all the meshes utilized in the convergence analysis are shown in Table 2 and 3, for contraction 2:1 and 4:1, respectively. It is important to state that the pressure matrix C has a high value condition number and,

because of this characteristic, it converges slower than the other matrices. This issue contributes adversely on the computation time. However, we are currently working on the solution of this problem with the ADI method.

**Table 2.** Execution time for 500 iterations for different meshes, contraction 2:1

| **N** | $\Delta x$, $\Delta y$ | **Points** | **Cells** | **Execution time** ($min$) |
|---|---|---|---|---|
| 32 | 0.03125 | 62785 | 61440 | 247.2 |
| 64 | 0.015625 | 248449 | 245760 | 1117.2 |
| 128 | 0.0078125 | 988417 | 983040 | 5991.7 |

**Table 3.** Execution time for 500 iterations for different meshes, contraction 4:1

| **N** | $\Delta x$, $\Delta y$ | **Points** | **Cells** | **Execution time** ($min$) |
|---|---|---|---|---|
| 32 | 0.03125 | 103809 | 102400 | 306.2 |
| 64 | 0.015625 | 412417 | 409600 | 1347.5 |
| 128 | 0.0078125 | 1644033 | 1638400 | 8982.5 |

## 7 Conclusions and Future Work

The flow simulations on sudden contraction geometries described in this paper, use the functions of the CUSPARSE and CUBLAS libraries to implement on a GPU the preconditioned BiCGSTAB method. This method is used to solve the systems that generates the discretization of the Navier-Stokes equations and, therefore, its appropriate parallelization has a significant impact in the reduction of computation time, which is the main objective of this work.

The described method that numerically estimates the order and error of the discretization, can not only be applied for simulations of contraction geometries; is a general procedure that can be used for other CFD applications as well as for discretizations that arise from the numerical solution of differential equations related to other areas of science.

As future work, we will be working with simulations of non-Newtonian fluxes, e.g. Viscoelastic fluids in the sudden contraction geometries described herein; this is the final stage of our investigation. Meanwhile, we are going to tackle the problem of pressure matrix C with the ADI method and Thomas algorithm, a combination that appropriately fits a GPU application. Also, GPU clusters will be used to simulate larger physical domains.

**Aknowledgements.** This material is based upon work supported by a grant from the University of California Institute for Mexico and the United States (UC MEXUS) and the Consejo Nacional de Ciencia y Tecnología de México (CONACYT).

## References

1. Alamsyah, M.N.A., Simanjuntak, C.A., Bagustara, B.A.R.H., Pradana, W.A., Gunawan, P.H.: Openmp analysis for lid driven cavity simulation using lattice boltzmann method. In: ICoICT '17. pp. 1–6 (May 2017)
2. Chorin, A.J.: The numerical solution of the Navier-Stokes Equations for an incompressible fluid. Bull. Amer. Math. Soc 73(6), 928–931 (1967)
3. Chorin, A.J.: Numerical solution of the Navier-Stokes Equations. Math. Comp. 22(104), 745–762 (1968)
4. Deng, L., Fang, J., Wang, F., Bai, H.: Evaluating multi-core and many-core architectures through accelerating an alternating direction implicit cfd solver. In: ISPDC '16. pp. 1–10 (July 2016)
5. Ferziger, J.H., Peric, M.: Computational methods for fluid dynamics. Springer, Berlin, third edn. (2002)
6. Ge, X., Wen, J.T.: Hybrid model reduction for compressible flow controller design. In: IEEE CDC '11. pp. 6540–6545 (Dec 2011)
7. Griebel, M., Rüttgers, A.: Multiscale simulations of three-dimensional viscoelastic flows in a square-square contraction. J. Nonnewton Fluid Mech. 205, 41–63 (2014)
8. Guermond, J.L., Minev, P.: High-order time stepping for the incompressible Navier-Stokes equations. SISC 37(6), A2656–A2681 (2015)
9. Harlow, F.H., Welch, J.E.: Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Phys. Fluids 8(2), 2182–2189 (1965)
10. Hashimoto, T., Yasuda, T., Tanno, I., Tanaka, Y., Morinishi, K., Satofuka, N.: Multi-gpu parallel computation of unsteady incompressible flows using kinetically reduced local Navier-Stokes equations. In: Computers and Fluids. vol. 167, pp. 215–220 (2018)
11. Hoffmann, K.A., Chiang, S.T.: Computational Fluid Dynamics, vol. First. Engineering Education System, Kansas, fourth edn. (2000)
12. Huang, J., Lin, Z., Ma, C., Yuan, X.: Gpu speed-up for the implicit Navier-Stokes solver. In: Proceedings of the ASME Turbo Expo. vol. 6 (Jun 2013)
13. Lovrić, A., Dettmer, W.G., Kadapa, C., Perić, D.: A new family of projection schemes for the incompressible Navier-Stokes equations with control of high-frequency damping. Comput Methods Appl Mech Eng 339, 160–183 (2018)
14. Temam, R.M.: Sur l'approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires (ii). Arch. Rational Mech. Anal. 33(5), 377–385 (1969)
15. Trebotich, D.P., Colella, P., Miller, G.H.: A stable and convergent scheme for viscoelastic flow in contraction channels. J. Comput. Phys. 205(1), 315–342 (2005)
16. Wang, Y., Baboulin, M., Rupp, K., Le Maître, O., Fraigneau, Y.: Solving 3d incompressible Navier-Stokes equations on hybrid cpu/gpu systems. In: HPC '14. pp. 12:1–12:8. San Diego, CA, USA (2014)
17. Willis, A.P.: The openpipeflow Navier-Stokes solver. SoftwareX 6, 124–127 (2017)

ISSN 1870-4069

# Muestreo y reconstrucción de realizaciones de la suma de dos procesos Gaussianos

Vladimir Kazakov, Francisco Mendoza

Instituto Politecnico Nacional, ESIME Zacatenco,
Departamento de Telecomunicaciones,
Ciudad de México, México
vkaz41@hotmail.com, fcm2709@gmail.com

**Resumen.** Con base en la regla de la esperanza matemática condicional se investiga el procedimiento de muestreo-reconstrucción para la suma de dos procesos Gaussianos. También se estudian las funciones básicas y la función de error de reconstrucción, cuando el número de muestras y la longitud del intervalo de muestreo son arbitrarios.

**Palabras clave:** suma, procesos Gaussianos, muestreo, reconstrucción.

## Sampling and Reconstruction of Realizations of the Sum of Two Gaussian Processes

**Abstract.** Based on the rule of Conditional Mathematical expectation, the sampling-reconstruction procedure is investigated for the sum of two Gaussian processes. The basic functions and the reconstruction error function are also studied, when the number of samples and the length of the sampling interval are arbitrary.

**Keywords:** sum, Gaussian processes, sampling, reconstruction.

## 1. Introducción

La descripción del Procedimiento de Muestreo-Reconstrucción (PMR) de las realizaciones en un proceso aleatorio ha sido investigado durante muchos años (ver por ejemplo dos revisiones [1,2]). Según el teorema de Balakrisnhan (TB) [3] se puede reconstruir cualquier realización de un proceso aleatorio estacionario con un espectro de potencia restringido a una frecuencia $\omega_b$ usando la siguiente expresión:

$$x(t) = lim_{N\to\infty} \sum_{i=-N}^{N} (x(T_i)) \frac{sen\omega_b(t - iT_b)}{\omega_b(t - iT_b)}, \tag{1}$$

donde $x(T_i)$ la muestra de una realización en el instante $T_i$; $\Delta T$ es un intervalo de muestreo *periódico* entre las muestras vecinas:

$$T_b = T_i - T_{i-1} = \frac{\pi}{\omega_b}. \tag{2}$$

Enfatizamos que en TB el error es cero si y solo si el número de las muestras es *infinito* . En [5,6] fue probado que TB es válido solamente para procesos Gaussianos.

El algoritmo (1) se caracteriza por tener un error de reconstrucción igual a cero para todos los procesos independientemente de su función de densidad de probabilidad [4]. Introducimos la función básica $b_i(t)$.

Como se ve en (1) todas la muestras tienen la misma función básica del tipo $\frac{sen(x)}{x}$:

$$b_i(t) = \frac{sen\omega_b(t - iT_b)}{\omega_b(t - iT_b)}. \tag{3}$$

Notamos, que el algoritmo (1) y la función básica (3) ignoran tales características de gran importancia como: la función de densidad de probabilidad (*fdp* ), y la función de covarianza del proceso.

El modelo matemático de procesos aleatorios con espectro restringido es citado con mucha frecuencia en la literatura. Pero TB no da ninguna información acerca de la de la influencia para PMR cuando el número $N$ de muestras son arbitrarias. El artículo presente da la claridad dentro de dichos problemas, porque para cada variante se obtienen dos características más importantes para cada PMR: la función básica y del error de reconstrucción. Estos resultados pueden ser útiles para los investigadores de los sistemas de comunicación con mensajes gaussianos con espectros rectangulares.

La meta del artículo presente es investigar el problema PMR de las realizaciones de procesos Gaussianos con un espectro rectangular, centrado en el origen y al encontrarse desplazado, cuando $\Delta T$ y $N$son arbitrarios.El conocimiento de la forma del espectro nos da la posibilidad de conocer la función de covarianza del proceso dado. Para cumplir este análisis usamos la metodología de la regla de esperanza matemática condicional [7]. La aplicación de dicha regla para la descripción de PMR de realizaciones de procesos aleatorios de varios tipos, fue descrita en algunas publicaciones [5,6], por mencionar algunos. En la investigación de PMR hay dos características principales: 1) la función óptima de reconstrucción; 2) la función mínima del error de la reconstrucción.

## 2. La regla de la esperanza matemática condicional en PMR de realizaciones gaussianas

Cada proceso gaussiano no estacionario esta descrito con tres características principales: la esperanza matemática $m(t)$, la varianza $\sigma^2(t)$ y la función de covarianza $R(t_1, t_2)$. Elegimos una realización del proceso y fijamos una multitud de las muestras $X, T = \{X(T_1), X(T_2), ..., x(T_N)\}$. En esta multitud el número de muestras $N$ y la locación de la muestras $T = \{T_1, T_2, ..., T_N\}$ son arbitrarios. Usando $X, T$ las características a priori $m(t), \sigma^2(t), R(t_1, t_2)$ se puede obtener las características del proceso condcional $x(t)|X, T = \tilde{x}(t)$, es decir, la esperanza matemática condicional $< x(t)|X, T >= \tilde{m}(t)$, y la varianza condicional $\langle[\tilde{x}(t) - \tilde{m}(t)]^2|X, T\rangle = \tilde{\sigma}^2(t)$.

La función $\tilde{m}(t)$ es la función de reconstrución de la realización muestreada y la función $\tilde{\sigma^2}(t)$ es la función de error de reconstrucción. La función $\tilde{m}(t)$ es la mejor estimación de la realización en el instante de tiempo actual $t$ y la función $\tilde{\sigma}^2(t)$ es el error mínimo de reconstrucción de dicha realización en el mismo instante de tiempo $t$. Las características principales de PMR son $\tilde{m}(t)$ y $\tilde{\sigma}^2(t)$ pueden ser descritas según [8]:

Considerando el caso estacionario para $\tilde{m}(t)$ y $\tilde{\sigma}^2(t)$; $m(t) = m = 0$, $\sigma^2(t) = \sigma = 1$, $R(t_1, t_2) = R(t_1 - t_2)$, tienen las formulas siguientes:

$$\tilde{m}(t) = \sum_{i=1}^{N} \sum_{j=1}^{N} R(t - T_i) a_{ij} x(T_j), \tag{4}$$

$$\tilde{\sigma}^2(t) = 1 - \sum_{i=1}^{N} \sum_{j=1}^{N} R(t - T_i) a_{ij} R(T_j - t), \tag{5}$$

$$||R(T_i - T_j)|| = \begin{Vmatrix} R(T_1 - T_1) \dots R(T_1 - T_N) \\ \dots \\ R(T_N - T_1) ... R(T_N - T_N) \end{Vmatrix}, \tag{6}$$

$$\|a_{ij}\| = \|R(T_i - T_j)\|. \tag{7}$$

La expresión (4) se reescribe en forma simplificada

$$\tilde{m}(t) = \sum_{j=1}^{N} x(T_j) b_j(t), \tag{8}$$

aquí es una función básica determinada por la formula

$$b_j(t) = \sum_{j=1}^{N} R(t - T_i) a_{ij}. \tag{9}$$

Cada muestra tiene su propia función $b_j(t)$.

Considerese la suma de dos procesos aleatorios gaussianos estacionarios $z(t)$ en la salida $y_1(t)$ y $y_2(t)$ de dos transformaciones lineales, caracterizados por su respuesta al impulso $h_1(t) y h_2(t)$. Los cuales tienen a la entrada $x(t)$ ruido blanco. Por tanto, la suma $z(t)$ de procesos aleatorios será otro proceso aleatorio gaussiano. Donde su función de covarianza normalizada $r(\tau)$ esta determinada por (10):

$$r(t) = \frac{R_{y_1}(\tau) + R_{y_2}(\tau) + 2R_{12}(\tau)}{R_{y_1}(0) + R_{y_2}(0) + 2R_{12}(0)}, \tag{10}$$

donde $R_{y_1}(\tau)$ es la función de covarianza del proceso aleatorio $y_1(t)$, $R_{y_2}(\tau)$ es la función de covarianza del proceso aleatorio $y_2(t)$, y $R_{12}(\tau)$ es la función de covarianza mutua entre los procesos $y_1(t)$ y $y_2(t)$. Se utilizan las ecuaciones (4)-(9) para definir las caracteristicas condiciones en la descripción del PMR.

## 3. Resultados obtenidos del PMR de la suma de procesos aleatorios para diferentes filtros lineales

A continuación se muestran los resultados para diferentes casos: 1) la suma dos procesos aleatorios cuando ambos procesos aletorios son transformados por un filtro RC de una etapa con diferentes constantes de tiempo; 2) la suma de un proceso aleatorio transformado por un filtro RC de dos etapas y otro proceso por un filtro RC de una etapa, con misma constante de tiempo; .

### 3.1. Suma de dos procesos aleatorios transformados por dos diferentes filtros RC de una etapa

De manera general, a la transformación lineal de primer orden de procesos gaussianos se obtienen procesos gaussiano markoviano, esto significa, que entre las variables aleatorias solo existira una dependencia inmedianta entre el vecino que represente el presente y el futuro. Por esta razon, la reconstrucción de un proceso aleatorio gaussiano markoviano es el mas fácil, para estudiar la influencia de las constantes de tiempo sobre el PMR.

Para usar la regla de la esperanza matemática condicional en el PMR es necesario conocer la función de covarianza. En nuestro caso la función de covarianza $r(\tau)$ de la suma de procesos aleatorios. Por la formula(10) se debe conocer la función de covarianza de cada proceso aleatorio a la salida de cada circuito RC de una etapa, es decir, $R_{y_1}(t)$ y $R_{y_2}(t)$, en (11) y (12):

$$R_{y_1}(t) = exp(-\alpha|\tau|), \tag{11}$$

$$R_{y_2}(t) = exp(-\beta|\tau|). \tag{12}$$

También, la función de covarianza mutua $R_{12}$ entre el proceso $y_1(t)$ y $y_2(t)$, que se define en (13):

$$R_{12}(\tau) = \frac{\alpha\beta}{\alpha+\beta} \begin{cases} exp(-\alpha|\tau|) \text{ si } \tau > 0, \\ exp(-\beta|\tau|) \text{ si } \tau < 0. \end{cases} \tag{13}$$

Realizando las sustituciones correspondientes de las formulas (11), (12) y (13) en (10). Se obtine la función de covarianza de la suma de los procesos, caracterizado por (11) y (12), a la salida de filtros RC de primera etapa. En Fig.1. se muestran las diferentes funciones de covarianza $R_{y_1}(\tau)$, $R_{y_2}(\tau)$, $R_{12}(\tau)$ y $r(\tau)$ con $\alpha = 1$ y $\beta = 0{,}1$.

A continuación se describe el PMR para la suma de dos procesos aleatorios, utilizando las formulas (4)-(9), estudiando la influencia de la constante de tiempo.

En Fig. 2. se muestran las funciones básicas cuando $\alpha = 1$ y $\beta = 0{,}1; 0{,}5; 0{,}9$ con un número de muestras $N = 2$ y un intervalo de muestreo $\delta T = 0{,}2$.

Enfatizamos algunas características generales de las funciones básicas: 1)cada muestra tiene su función básica; 2)en cada instante de muestreo, solamente una función tiene el valor de uno, mientras todas las demas funciones son cero en el

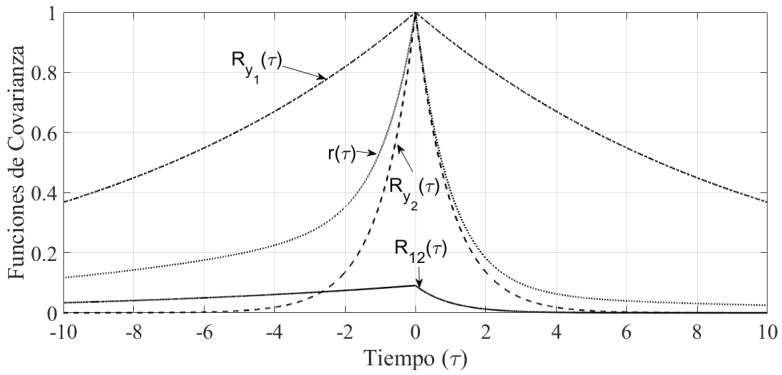



**Fig. 1.** Funciones de Covarianza $\alpha = 1$ y $\beta = 0{,}1$.

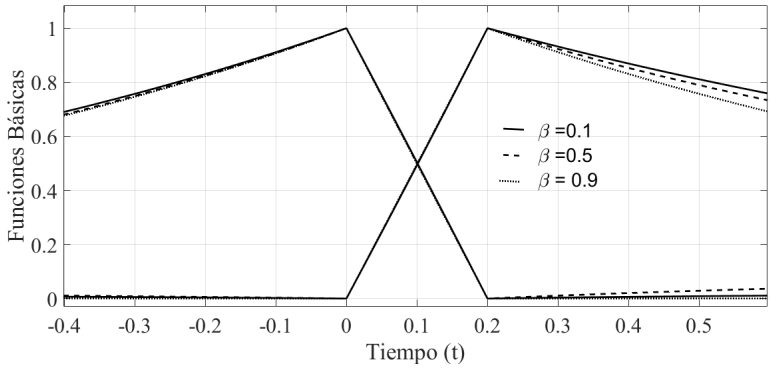



**Fig. 2.** Funciones básicas a diferentes anchos de banda

mismo intante de muestreo, es decir, cada función básica es orto-normal entre las demas. De manera particular, se observa: 1) en la región $T_1 < t < T_N$ (región de interpolación) el comportamiento de reconstrucción es igual a lineal, en los tres casos; 2) en las regiones de extrapolación ($t > T_N$) y retropolación ($t < T_1$) presentan diferentes pendientes con la variación de la constante de tiempo de uno de los filtro, fisicamente el cambio en la constante del tiempo no afecta directamente la eficiencia de reconstrucción.

Por último consideremos, la calidad de reconstrucción del PMR utilizando los mismos parametros, es decir, $\alpha = 1$ y $\beta = 0{,}1; 0{,}5; 0{,}9$ con un número de muestras $N = 2$ y un intervalo de muestreo $\delta T = 0{,}2$.

De manera general, la función de error de reconstrucción tiene las características: 1) a mayor longitud del intervalo de muestreo $\Delta T$ mayor el error de reconstrucción; 2) entre mayor cantidad de muestras menor será el error de reconstrucción. En Fig. 3. se observa que las funciones de error de reconstrucción

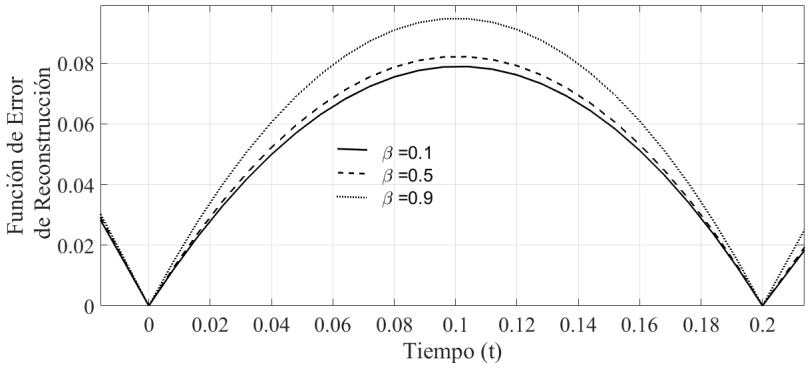



**Fig. 3.** Funciones básicas a diferentes anchos de banda

no varia significativamente, aun cuando se compara el caso $\beta = 0{,}1$ y $\beta = 0{,}9$ Esto quiere decir, que en el PMR no existe una influencia determinante por la variación las constantes de tiempo, sino del sistema al cual esta sometido la transformación de los procesos a sumar.

### 3.2. Suma de dos procesos aleatorios transformados por un filtro RC de dos etapas y una etapa

A diferencia del caso anterior que se estudiaba la influencia del ancho de banda en PMR de la suma de procesos aleatorios. Ahora se investigará la influencia de la estructura de los sistemas sobre el PMR, cuando el ancho de banda es igual para dos sistemas diferentes. Por tanto, se comenzará definiendo la funciones de covarianza en la salida de los filtros RC de primera y segunda etapa, cuando a su entrada existe ruido blanco, en (14) y (15) respectivamente:

$$R_{y_1}(\tau) = exp(-\alpha|\tau|), \tag{14}$$

$$R_{y_1}(\tau) = (1 + \alpha|\tau|)exp(-\alpha|\tau|). \tag{15}$$

Se obtiene la función de covarianza mutua entre los procesos $y_1(t)$ y $y_2(t)$, en (16):

$$R_{12}(\tau) = \begin{cases} \frac{(1+2\alpha\tau)exp(-\alpha|\tau|)}{\sqrt{2}} & \text{si } \tau > 0 \\ \frac{exp(-\beta|\tau|)}{\sqrt{2}} & \text{si } \tau < 0 \end{cases} \tag{16}$$

De (10) se obtine la función de covarianza de la suma de los procesos procesos aleatorios $y_1(t)$ y $y_2(t)$. En Fig.4 se muestran las diferentes funciones de covarianza definidas en (11)-(13).

Usando las ecuaciones (4)-(9) se ilustrará las caracteristicas condicionales para el estudio del PMR. En Fig. 5, se muestran las funciones básicas del PMR

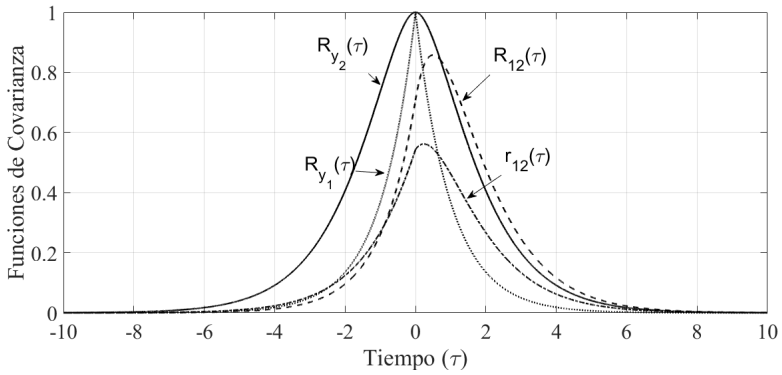



**Fig. 4.** Funciones de covarianza

cuando con una constante de tiempo $\alpha = 1; 0{,}5$ y un intervalo de muestreo $\Delta T = 0{,}2$.

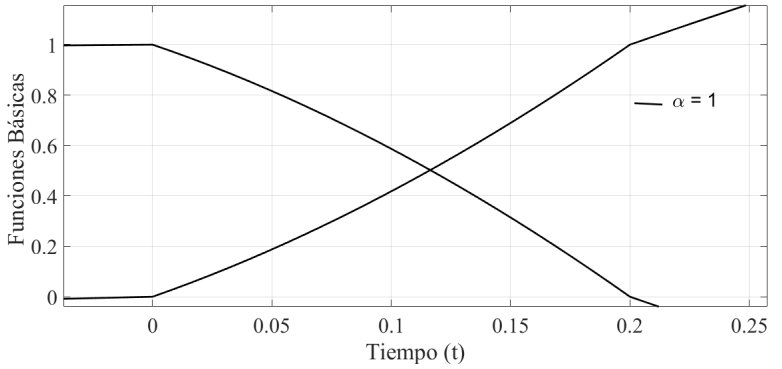



**Fig. 5.** Funciones básicas

En Fig. 5, se observa que la tendencia en la región de interpolación no es lineal, a diferencia de Fig.2. Esto es resultado de la interacción entre los filtros RC de una y dos etapas, dado que este último el proceso en su salida será de tipo no markoviano a diferencia de los casos anteriores. Por último consideremos la función de error de reconstrucción, con los mismos parametros al caso anterior.

En Fig. 6, se muestran la función de error de reconstrucción. Al compararse cuantitativamente con la Fig.3, se observa una mejora significativa en la calidad de la reconstrucción al presentar la influencia de un filtro de dos etapas, pues este último será un proceso no markoviano, es decir, que existe una relación de dependencia entre todas sus variables aleatorias. De manera particular, significa

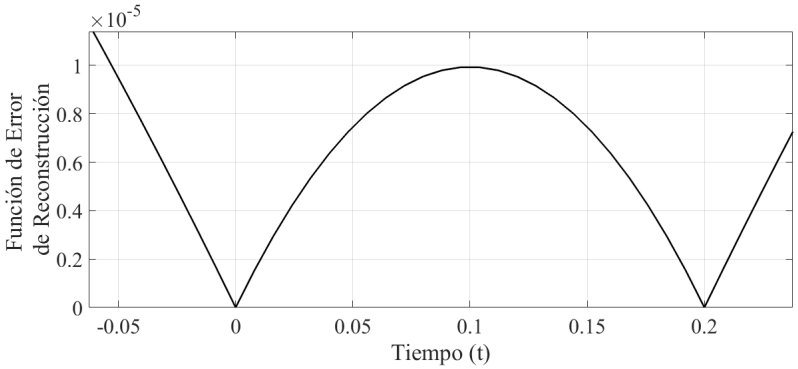



**Fig. 6.** Funciones básicas

que la dependencia estadística existente en un proceso juega un papel importante para reconstrucción de dicho proceso.

## 4. Conclusiones

Utilizando la regla de la esperanza matemática condicional, como metodología de la investigación, del cálculo de errores de reconstrucción cuando el número de muestras es limitado y cuando los intervalos de muestreo son arbitrarios, en un proceso Gaussiano como resultado de la suma de dos procesos gaussianos transformados linealmente, se observa que a mayor dependencia estadística entre las variables aleatorias (en los instantes de muestreo) que componen el conjunto de muestras la reconstrucción será de mayor calidad. Por tanto, el tipo de transformación lineal influyen en el PMR. También se observa que la constante de tiempo no afecta de manera significativa la calidad de la reconstrucción.

Es bien sabido que el área de procesamiento de señales de voz, el procedimiento de Muestreo-Reconstrucción es una etapa importante para el tratamiento de este tipo de señales. La aplicacion de la REMC en el PMR de señales de voz es una alternativa a investigar en la identificación de fonemas [8].

## Referencias

1. Jerry, A.: The Shannon sampling theorem: Its various extensions and applications. A tutorial review. Proc. IEEE, vol. 65, pp. 1565–1596 (1977)
2. Jerry, A.: Bibliographic List. In: Advanced Topics in Shannon sampling and interpolation theory. Springer, NY (1992)
3. Balakrishnan, A.: A Note on the Sampling Principle for Continuous Signals. IRE Trans. On information theory, Vol. IT-3, pp. 143–146 (1957)
4. Balakrishnan, A.: On the Problem of Time Jitter in Sampling. IRE Trans. On information theory, Vol. IT-3, pp. 226–236 (1962)

5. Kazakov, V.: The sampling-reconstruction procedure with a limited number of samples of stochastic processes and fields on the basis of the conditional mean rule. Electromagnetic waves and electronic systems, Vol.10, Num. 1-2, pp. 98–116 (2005)
6. Kazakov, V.: Sampling-Reconstruction procedures of Gaussian process realizations. In: Probability: interpretation, theory and applications. Y. Shmaliy (ed.), Nova science publisher inc., USA, NY, pp.269–297 (2012)
7. Pfeiffer, P.: Probability for applications. Springer Verlag (1990)
8. Proakis, J., Manolakis, D.: Digital signal processing: Principles, algorithms, and applications. Springer Verlag (1990)

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
diciembre de 2018
Printing 500 / Edición 500 ejemplares