SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

ISSN: 1870-4069

RCS

Research in Computing Science

Vol. 83

Advances in Computing Science

L. E. Berthet Dorantes
J. C. Chimal Eguía
E. A. Santos Camacho
E. Castillo Montiel

Advances in Computing Science

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 83,** noviembre 2014. 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 83**, Noviembre 2014. Printing 500.

Volume 83

Advances in Computing Science

Luis Enrique Berthet Dorantes,
Juan Carlos Chimal Eguía,
Evelia Araceli Santos Camacho,
Erandi Castillo Montier (eds.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2014

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 and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Preface

The purpose of this volume is to present the most recent advance in selected areas of Computer Science, The works included in this volume where carefully selected by the editors on the basis of the blind reviewing process and the main criteria for selection were originality and technical quality. This issue of Research in Computing Science will be useful for researches and students working in he different areas of Computer Science, as well as, for all reader interested you want to enrich your knowledge in this file.

In total, we received 72 paper that were submitted for evaluation; each submitted paper was reviewed by 2 independent members of the editorial board of the volume or additional reviewers. The acceptance rate is 38%. This volume contains revised version of 12 accepted papers, divided into six sections corresponding to the areas: Real Time & Automation, Intelligent Processing of Geospatial Information, Micro technologies & Embedded Systems, Natural Language Processing, Neural Networks & Unconventional Computation, Simulation & Modeling. The missing papers are published in the volume number 81 of Research in Computing Science.

We would like express our gratitude to all people who help to elaborate this volume. First to the authors of the papers for the technical excellence of their works, that guarantees the quality of this publications. We also want to thank the members of the editorial board for their hard work in evaluating and selecting the best's papers out of many submissions that we received. We express sincerely our gratitude to the Sociedad Mexicana de Inteligencia Artificial (SMIA) for its collaboration in elaborates this publication. Also we want to give special recognition to the Centro de Investigación en Computacion of the Instituto Politécnico Nacional (CIC-IPN) for facilities give in order to achieve the success in the publication of this volume. The submission, reviewing, and selection process was supported for free by the EasyChair system, www.EasyChair.org. Also we want to give special recognition to ORACLE, its participation as a sponsor.

Luis Enrique Berthet Dorantes
Juan Carlos Chimal Eguía
Evelia Araceli Santos Camacho
Erandi Castillo Montiel
November 2014

Table of Contents

Neural Networks & Unconventional Computation

Simulation & Modeling

Real Time & Automation

Controlador PID de una celda termoeléctrica: Sintonización por medio de algoritmos genéticos con codificación real

Juan Fernando García-Mejía[1], Allan Antonio Flores-Fuentes[2], Carlos Eduardo Torres-Reyes, José Arturo Pérez-Martínez[3]

Universidad Autónoma del Estado de México Centro Universitario UAEMex Atlacomulco
fgarciam@uaemex.mx

Abstract. Una celda termoeléctrica es un dispositivo semiconductor que se aplica en la refrigeración, dado que es un intercambiador eléctrico de temperatura la mejora a su desempeño puede realizarse por medio de un controlador PID, cuyo diseño en el presente trabajo se realizó en base a una técnica evolutiva denominada algoritmo genético, que utiliza como función de costo el inverso del error cuadrático medio. En este trabajo se realiza una comparación entre el desempeño del controlador sintonizado por medio de criterio de Zigler Nichols, y el ajustado por medio de un algoritmo genético, cuya codificación de parámetros se realizó con números reales, aplicando un operador de cruzamiento aritmético.

Keywords: Algoritmo genético, celda termoeléctrica, codificación real.

1 Introducción

Las celdas termoeléctricas (TEC, por sus siglas en inglés) son dispositivos semiconductores que permiten el intercambio de temperatura entre las superficies que lo forman en función de la aplicación de una determinada señal eléctrica. Mientras que una cara enfría, la otra calienta, esto se revierte mediante un cambio de polaridad del voltaje aplicado a la celda. Las aplicaciones de los TEC se encuentran en micro refrigeración y refrigeradores móviles [1].

La relación que existe entre la temperatura de enfriamiento y la corriente de alimentación de una celda termoeléctrica puede caracterizarse por medio de una función de transferencia en términos de la variable compleja s tal como se muestra a continuación en la ecuación 1 [2] donde $\tilde{I}(s)$ es la corriente de alimentación de la celda y $\tilde{T}_L(s)$ la temperatura de la celda.

$$\frac{\tilde{T}_L(s)}{\tilde{I}(s)} = G_I(s) = -6.4061\left(\frac{0.064s + 0.00854}{s^2 + 0.5964s + 0.00855}\right) \quad (1)$$

Una respuesta típica, mostrada en la Fig. 1 puede manipularse por medio de técnicas de ingeniera de control tales el controlador Proporcional Integral Derivativo (PID) que consiste de tres valores denominados k_p, k_i, k_d, la función de transferencia de este en su implementación paralela se observa en la ecuación 2.

Advances in Computer Science
Research in computing Science 83, 2014 pp. 3-13

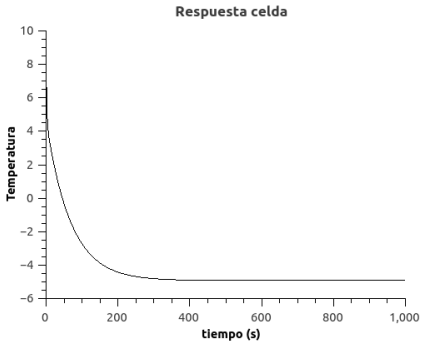


Fig. 1. Respuesta de la celda termoeléctrica.

$$PID = C(s) = \frac{U(s)}{E(s)} = k_p + \frac{k_i}{s} + k_d s \tag{2}$$

En base a las ecuaciones 1 y 2 es posible construir un esquema de control como se muestra en la figura 2.

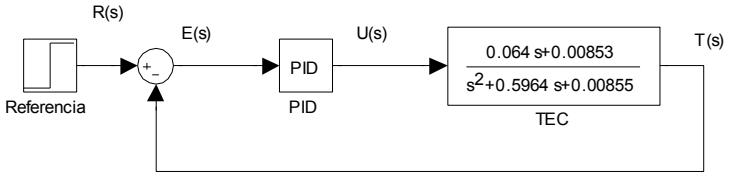


Fig. 2. Esquema de control.

El ajuste de los valores k_p, k_i, k_d se realiza de manera habitual por medio del criterio de Zigler-Nichols, una alternativa a este se encuentra en el uso de técnicas evolutivas, las cuales en términos generales son series de pasos definidos con aplicación en la optimización o de aprendizaje con la capacidad de evolucionar [3], una de ellas fue el algoritmo genético simple (GA, por sus siglas en inglés), desarrollado por John Holland en la década de los 60 en la Universidad de Michigan, en base a los principios de Charles Darwin y Gregory Mendel presentes en la naturaleza [4].

2 Algoritmos genéticos

En esta sección se documenta un breve estado del arte sobre el uso de algoritmos genéticos en la sintonización de controladores PID, posteriormente se describe el algoritmo que se emplea en el desarrollo de la propuesta que se documenta en el presente trabajo.

2.1 Estado del Arte

El uso de técnicas evolutivas y en especifico de algoritmos genéticos en la sintonización de controladores PID es documentado en la literatura especializada,

como muestra Mohd S *et al* [5] donde realizan una comparación entre evolución diferencial, algoritmos genéticos y criterio de Zigler-Nichols, demostrando que las técnicas evolutivas presentan ventajas en cuanto al desempeño con respecto a técnicas analíticas como Zigler-Nichols. Li J *et al* [6] realizaron un algoritmo genético con codificación real, el cual no demuestra diferencias significativas con respecto al uso de un enjambre de partículas. Reynoso G et al [7] muestra un procedimiento de sintonización automática de parámetros de un controlador PID en forma serie, el cual muestra ventajas sobre Zigler-Nichols al controlar la respuesta de un conjunto de plantas de prueba propuestas de manera teórica. Renato A *et al* [8] utiliza un algoritmo genético con codificación real para sintonizar un PID de dos grados de libertad, como planta de estudio se propone la función característica de un servomotor. Valarmathi *et al* [9] controla el nivel de liquido de un tanque modelado como un sistema no lineal utilizando un controlador PID. Yang M *et al* [Yang 2012] controlan la velocidad de rotación de un motor a través de un algoritmo genético cuyo criterio de paro es la convergencia del algoritmo.

En base a lo documentado en el estado del arte se puede aseverar que la sintonización de controladores tipo PID realizada por algoritmos genéticos ofrecen mayores ventajas que el criterio de Zigler-Nichols, [5], [7], [9], cabe destacar que la codificación empleada en los citados trabajos fue binara, esto puede presentar errores de truncamiento en el momento de representar las variables k_p, k_i, k_d; esto es solucionado por una codificación real en [6] y [8], por ultimo en [10] se documenta el paro de algoritmo genéticos por medio de la convergencia de la función objetivo.

Con lo anterior en mente se propone realizar la sintonización de un controlador PID aplicado a una celda termoeléctrica por medio de un algoritmo genético simple, con codificación real, pero haciendo énfasis en el estudio del operador de cruzamiento dado que no fueron documentados en [6] y [8] las variantes que presenta el cruzamiento aritmético empleado

2.2 Pseudocódigo del algoritmo propuesto

Un algoritmo genético simula algunos aspectos propios de la teoría de la evolución de las especies de Darwin. Los mejores individuos de una determinada población tienen mayores posibilidades de supervivencia y reproducción; las posibilidades disminuyen o son nulas para los débiles. Los pasos que caracterizan a un algoritmo genético simple son los siguientes [11] y [12].

i. definir una función de aptitud o función objetivo

ii. generar una serie de posibles soluciones de manera aleatoria (población)

iii. codificar la población

iv. evaluar con la población, iniciando así la i esima generación

v. seleccionar soluciones que se reproducirán

vi. aplicar una operación de cruzamiento

vii. mutar algunos elementos resultantes del cruzamiento

viii. reemplazar elementos de la población de la i esima generación con los mejores elementos de vi y vii

ix. detener, si se cumple criterio de paro, y en caso contrario saltar a iv

En términos generales los algoritmos evolutivos constituyen técnicas que se engloban bajo el concepto de soft computing, el cual es un enfoque que remarca la habilidad de la mente humana para razonar y aprender en un ambiente de incertidumbre e imprecisión. Este término fue acuñado por Lofti Zadeh en 1992 [13] Cuando existe una sinergia entre las técnicas antes mencionadas se denomina inteligencia computacional [14].

3 Metodología

En esta sección se muestra los pasos que se realizaron para optimizar el controlador PID, aplicado a la celda termoeléctrica. Donde el objetivo perseguido es la minimización del error medio cuadrático, el cual se explica como la diferencia que existe entre la respuesta del TEC y la referencia deseada.

3.1 Función objetivo

Como se menciono en secciones anteriores, un algoritmo genético tiene una función objetivo, para este caso la función objetivo se puede construir a partir de la función de transferencia en lazo cerrado del esquema propuesto en la figura 2 y las ecuaciones (1) y (2) obteniéndose la expresión matemática (3).

$$T(s) = \frac{G_I(s)C(s)}{1 + G_I(s)C(s)}R(s) \tag{3}$$

De (2) y (3) se puede observar que la salida $T(s)$ depende de los valores k_p, k_i, k_d , así mismo de la figura 2 se puede definir (4).

$$E(s) = R(s) - T(s) \tag{4}$$

Ahora bien, una función objetivo se puede definir como se muestra en (5) [15], a partir de esta información y del concepto de valor cuadrático medio es posible construir la función objetivo que se muestra en (6) donde *T* es el tiempo de simulación

$$\min(\max) f(x), x = [x_1, x_2, \dots x_n]^T \in \mathbb{R}^n \tag{5}$$

$$f_{obj}(k_p, k_i, k_d) = \max\left(\frac{1}{1 + \sqrt{\frac{1}{T}\int_0^T E(k_p, k_i, k_d)^2}}\right) \tag{6}$$

3.2 Población y Codificación

La colección de sujetos propuestos como posibles soluciones son generados de manera aleatoria (40 en total) con una distribución uniforme, codificando los cromosomas con números reales, de tal forma que el formato del cromosoma tiene la siguiente forma.

$$cromosoma = \left[k_p, k_i, k_d\right]^T$$

3.3 Selección

Los cromosomas que se seleccionaron para el cruzamiento en sucesivas generaciones fueron escogidos mediante una ruleta, donde los sujetos con mayor valor de afinidad, (mayor valor de $f_{obj}\left(k_p, k_i, k_d\right)$ se privilegian sobre los de menor afinidad. El operador de ruleta es el más estocástico de los métodos de selección, en relación con el torneo y el elitismo, es por eso que fue empleado en este trabajo.

3.4 Cruzamiento

El cruzamiento es determinado por el tipo de codificación. En este caso se emplea el operador de cruce denominado aritmético, cuyo procedimiento se muestra en esta sección.

Sean dos cromosomas $C_1 = \left[k_p^1, k_i^1, k_d^1\right]$ y $C_2 = \left[k_p^2, k_i^2, k_d^2\right]$ que fueron seleccionados mediante un procedimiento de ruleta, los descendientes de estos $H_k = \left[k_p^k, k_i^k, k_d^k\right]$ donde $k = 1{,}2$ son generados mediante, para $\alpha = [0{,}1]$

$$H_1 = \alpha(C_1 + ((1-\alpha) * C_2)$$

$$H_2 = \alpha(C_2 + ((1-\alpha) * C_1)$$

Se tienen 3 posibles situaciones con respecto al valor de α:

- Si $\alpha = 0.5$ se tiene un cruzamiento uniforme
- Si α varia en las generaciones se tiene un cruzamiento no uniforme, al igual que si varía en cada cruzamiento

En este trabajo se realizó el contraste entre las tres posibles situaciones presentes en el valor de α y el criterio de Zigler-Nichols, cabe destacar que los desarrollos documentados

3.5 Mutación

Este operador permite añadir variabilidad a un algoritmo genético, en este caso se mutan 2 individuos por cada generación a través del siguiente procedimiento:

A partir de un cromosoma C_i' se puede obtener un cromosoma transformado o mutado C_i'' a partir de la siguiente expresión donde el tamaño de paso de la mutación $\beta = [0,1]$ y la dirección de la misma se representa por d (7)

$$C_i'' = C_i' + \beta * d \quad (7)$$

3.5 Criterio de paro

Como se muestra en el pseudocódigo listado en la sección 2.2 el algoritmo se ejecutara hasta que se cumpla un determinado criterio, los cuales en términos generales son dos: un determinado número de ejecuciones (denominadas generaciones) o la convergencia del algoritmo, este último es el empleado en esta propuesta

4 Resultados

Para afirmar que la mejor elección de sintonización de un PID aplicado a un celda termoeléctrica es un algoritmo genético se desarrollaron una serie de simulaciones codificadas en Scilab con una temperatura de referencia o de *set point* de -5 grados centígrados, la primera se sintonizó el controlador por medio del ajuste de Zigler-Nichols, obteniéndose como respuesta la mostrada en la fig. 3, posteriormente se realizaron simulaciones ajustando los valores de k_p, k_i, k_d con un algoritmo genético, con los tres casos posibles del parámetro α que se especificaron en la sección 3.4. Por otra parte la fig. 6 y la 7 muestran los comparativos de la señal de salida de la celda termoeléctrica a distintos valores de α y la convergencia de los algoritmos respectivamente

Una síntesis de los resultados es mostrada en la tabla 1 donde se observa la comparación de resultados obtenidos con las técnicas expuestas en este documento

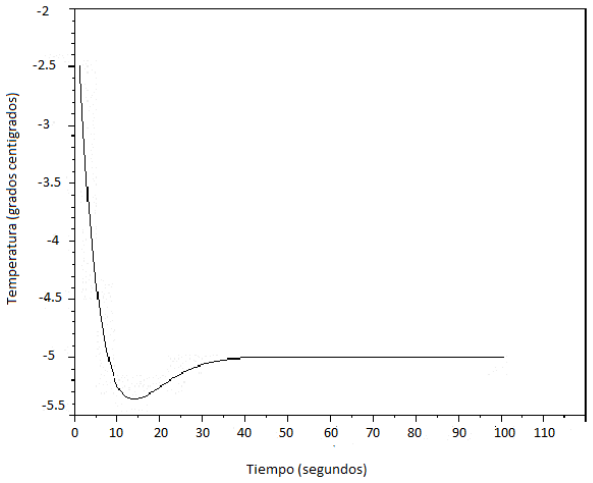


Fig. 3. Respuesta de la TEC con PID ajustado por Zigler-Nichols

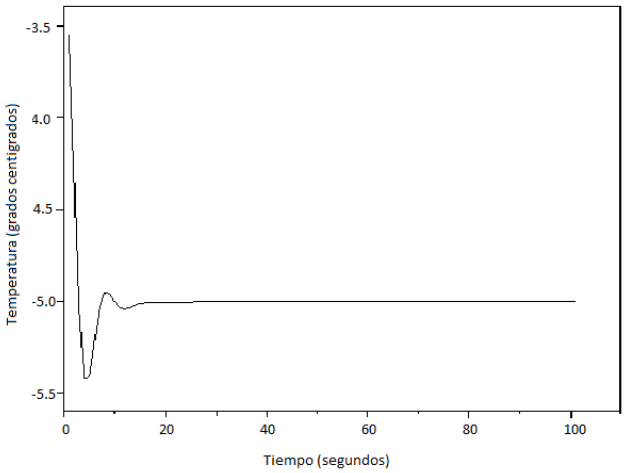


Fig. 4. Respuesta de la TEC con PID ajustado GA con cruzamiento aritmético no uniforme con α variando durante el algoritmo

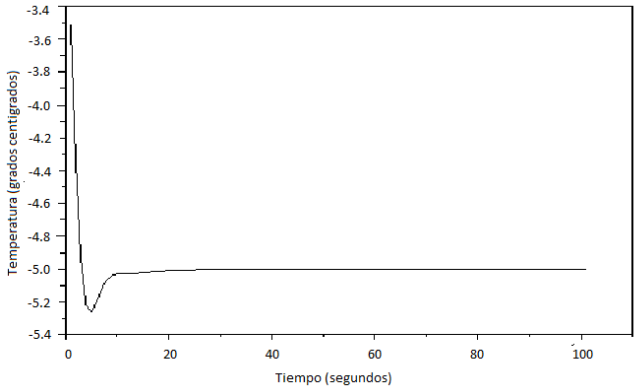


Fig. 5. Respuesta de la TEC con PID ajustado GA con cruzamiento aritmético no uniforme con α variando durante el cruzamiento

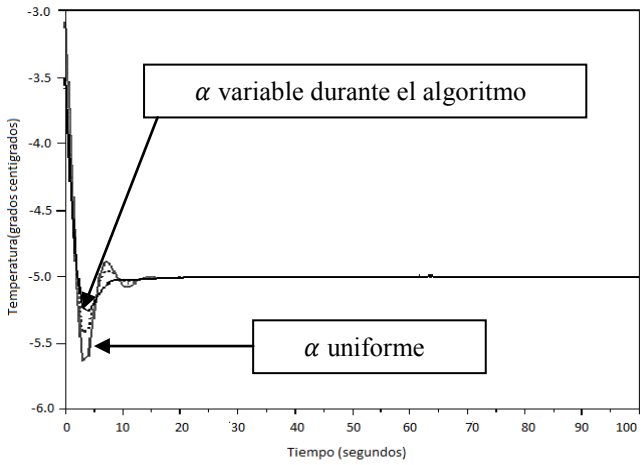


Fig. 6. Comparativo de la respuesta de la TEC para los distintos casos de α variando durante el algoritmo

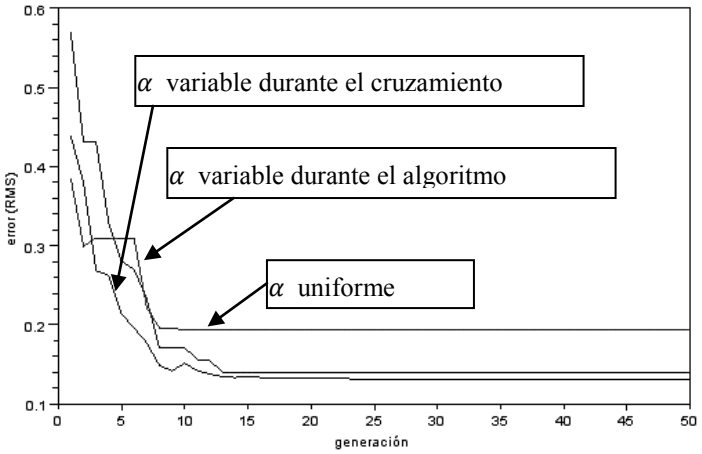


Fig. 7. Comparativo de la convergencia del GA para los distintos casos de α variando durante el algoritmo

Tabla 1. Resultados de los algoritmos empleados

Técnica	Error cuadrático medio	k_p k_i k_d	t_s	Valor del sobre impulso
Ajuste de Zigler-Nichols	0.33797	-2.4 -0.6 -2.3	40	-5.4
GA uniforme	0.14771	-3.23159 -5.4424 -3.88796	25	- 5.638
GA no uniforme generacional	0.13096	-5.27236 -6.97429 -5.9389	25	- 5.419
GA no uniforme en la cruza	0.11182	-7.21334 -5.57544 -5.7163567	30	- 5.263

5 Análisis de Resultados

De acuerdo a las figuras y la tabla producto de las simulaciones realizadas se observa que una técnica evolutiva es una buena opción para la sintonización de los parámetros de un controlador PID, esto concuerda con las referencias, [5], [7] y [9], el inconveniente de la codificación binaria es subsanado por la representación real, lo que involucra el uso de operadores específicos como el de cruce aritmético que a pesar de ser usado en [6] y [8] no muestran a detalle la elección del parámetro α.

De tal forma que los criterios de la comparación entre técnicas de sintonización fueron el tiempo de establecimiento, así como el error cuadrático medio y el valor del sobreimpulso los cuales son menores en un GA no uniforme en la cruza en relación con el GA no uniforme generacional, GA uniforme y ajuste de Zigler-Nichols.

6 Conclusiones

El uso de un algoritmo genético en el diseño de un controlador PID se justifica plenamente a partir de los resultados mostrados. También se plantea a futuro el uso de otros operadores de cruce tal como el BLX- α [16] así como otras técnicas evolutivas. Es destacable el uso de Scilab el cual es de software libre, en la escritura de los códigos empleados. Por otra parte la celda termoeléctrica es susceptible a ser controlada por una combinación de técnicas clásicas de control con inteligencia artificial

Agradecimientos. Los autores agradecen a la Secretaria de Investigación y Estudios Avanzados de la UAEMex por el apoyo brindado en la realización de este proyecto

Referencias

[1] Tarter R., Solid-state power conversion handbook.United State of America: John Wiley and Sons 1993
[2] Song Shaojing Temperature Control of Thermoelectric cooler Based on Adaptive NN-PID International Conference on Electrical and Control Engineering 2010
[3] Burger, C. Propeller performance analysis and multidisciplinary optimization using a genetic algorithm. Auburn University). ProQuest Dissertations and Theses 2007
[4] Yang X. S. Nature-Inspired Metaheuristic Algorithms United Kingdom : Luniver Press 2011
[5] M.S.Saad, H.Jamaluddin,I.Z.M. Darus "Implementation of PID Controller tuning using differential evolution and Genetic Algorithms," Int. J. of Innovative Computing, Information and Control (ICIC), Vol 8, num 11, pp 7761-7779, Nov 2012
[6] Li Junli,Mao Jianlin, Zhang Guanghui "Evolutionary algorithms based parameters tuning of PID controller", Control and Decision Conference, Mianyang, China, pp 416-420, 2011.
[7] G. Reynoso-Meza, J. Sanchis, J.M. Herrero, C. Ramos, "Evolutionary auto-tuning algorithm for PID controllers", IFAC Conf. on Advances in PID control PID'12, Brescia(Italy), March 28-30, 2012.FrB1.6.
[8] Renato A. Krohling and Joost P. Rey "Design of Optimal Disturbance Rejection PID Controllers Using Genetic Algorithms" IEEE Transactions on evolutionary computation, vol. 5, no. 1, february 2001
[9] R.Valarmathi, P.R.Theerthagiri, S.Rakeshkumar "Design and Analysis of Genetic Algorithm
Based Controllers for Non Linear Liquid Tank System" Advances in Engineering, Science and Management (ICAESM), pp 616 - 620 March 2012
[10] Ming Yang Lei Zhang ; Weiping Zhao ; Zhanshuang Hu "Optimization of dc motor rotary speed controller based on the genetic algorithm" World Automation Congress (WAC), 2012, pp 1-4 June 2012
[11] Mitchell, M. "An introduction to genetic algorithms. MIT Press" USA 1998.
[12] Mitsuo Gen, Runwei Cheng. "Genetic algorithms and engineering optimization". John Wiley and Sons, 2000
[13] Konar A. "Computational intellingence principles, techniques, and applications". Paises Bajos:Springer-Verlag 2005
[14] Yu X. Introduction to Evolutionary Algorithms. United States of America: Springer, 2010
[15] Ponstein J.P Approaches To The Theory Of Optimization Cambridge University Press 2004
[16] Ana Mª Sánchez "Algoritmos Genéticos para Codificación Real con Operador de Cruce Híbrido con Múltiples Descendientes: 2BLX0.5-2FR0.5-2PNX3-2SBX0.01" VI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados pp.411-418 Malago 2009

HeartDroid: Sistema de monitoreo cardiaco utilizando dispositivos Android

Cristian Eduardo Villegas-López, Sergio Sandoval-Reyes

Centro de Investigación en Computación. Instituto Politécnico Nacional. México City 07734, México
cristianevillegas@gmail.com; Sergio.Sandoval9@gmail.com

Resumen. Debido a los avances en la tecnología, la monitorización ambulatoria con instrumentos médicos y dispositivos comerciales se ha convertido en una herramienta cada vez más importante en el diagnóstico de algunas enfermedades y condiciones médicas, así como el seguimiento de las mismas en personas que han sido diagnosticadas con algún padecimiento. Este artículo describe a HeartDroid, una aplicación para dispositivos Android que permite la monitorización ambulatoria de personas que tengan un padecimiento cardiaco. La aplicación se encarga de monitorizar los signos vitales provenientes de la persona que porta un sensor Zephyr BioHarness 3 (BH3), con el cual se comunica por medio de Bluetooth al dispositivo Android. Estas bio-señales son analizadas en busca de anomalías y a su vez de forma paralela se almacenan de forma local para ser posteriormente revisadas a detalle. En caso de reconocer una anomalía en los signos de la persona, la aplicación de forma automática mostrará una alerta al usuario en su dispositivo Android y enviará una alerta a un médico/especialista y/o tutor para buscar asistencia, esto a través de mensajes SMS y en caso de que el médico o tutor cuenten con un dispositivo Android, ellos también pueden recibir una notificación e incluso ver la anomalía en pantalla.

Palabras clave: Android, monitoreo, arritmias, sensor, salud móvil, enfermedades cardiacas.

Abstract. Mobile health monitoring using medical instruments and commercial wearable devices has become increasingly important in the diagnostic of some cardiac diseases thus as the follow up and detection of emergent situations in the patients' health. This paper presents *HeartDroid* an Android mobile application to monitoring bioelectrical heart signals to detect cardiac arrhythmias using a Zephyr BioHarness 3 sensor, which captures these biosignals in real time and transmit them into a smartphone throughout a Bluetooth connection. These biosignals in turn are stored in the smartphone memory to be analyzed in detail in search of cardiac anomalies. When one is found automatically the app will display an alert and will send a sms alert message to his/her physician or tutor, in seek for help. The physician also receives a file which when it is processed on the phone, displays the graphic of the cardiac anomaly detected for diagnostic purposes.

Keyword: Android, monitoring, arrhythmias, sensor, mobile health, cardiac diseases.

Advances in Computer Science
Research in computing Science 83, 2014 pp. 13-23

1 Introducción

Debido a los avances en la tecnología, la monitorización ambulatoria con instrumentos médicos y dispositivos comerciales se ha convertido en una herramienta cada vez más importante en el diagnóstico de algunas enfermedades y condiciones médicas, así como el seguimiento de las mismas en personas que han sido diagnosticadas con algún padecimiento [1]. Tradicionalmente para la detección y monitorización de enfermedades cardiacas se les realiza a las personas un electrocardiograma (ECG), para de monitorear la actividad cardiaca del paciente y ver sus signos vitales en tiempo real [2]. Actualmente el ECG ofrece una gran variedad de posibilidades, no obstante la mayoría de los sistemas médicos e instrumentación necesitan conectar al paciente un conjunto de cables para que sus señales físicas puedan ser medidas y monitoreadas. Sin embargo no siempre es posible detectar todas las enfermedades en un solo estudio para ello existen alternativas como lo son el "Holter" o electrocardiograma (ECG) ambulatorio que es un sistema que permite grabar la actividad cardiaca eléctrica del paciente de manera ininterrumpida por un periodo de tiempo determinado para su posterior análisis. Lo anterior aunado a un registro manual de las actividades que realiza en el tiempo que porta el dispositivo da como resultado final un diagnóstico confiable [3]. No obstante estos sistemas tienen grandes limitaciones dado que para el diagnóstico tiene que haber transcurrido el estudio por completo, además de su costo que puede ir desde unos cientos de dólares en sus versiones básicas hasta rebasar los miles para equipos más avanzados.

Este artículo describe la aplicación HeartDroid, un sistema para dispositivos Android que permite la monitorización ambulatoria de personas que tengan un padecimiento cardiaco. El sistema se encarga de monitorear los signos vitales provenientes de la persona que porta un sensor Zephyr BioHarness 3 (BH3), el cual se comunica por medio de Bluetooth al dispositivo Android. Estas bio-señales son analizadas en busca de anomalías y de forma paralela se almacenan de forma local para ser posteriormente revisadas a detalle. En caso de reconocer una anomalía en los signos de la persona, el sistema de forma automática mostrará una alerta al usuario en su dispositivo Android y enviará una alerta a un médico/especialista y/o tutor para buscar asistencia. Esto a través de mensajes SMS y en caso de que el médico/especialista o tutor cuenten con un dispositivo Android ellos también pueden recibir una notificación e incluso ver detalles de la anomalía en pantalla.

El artículo está organizado de la siguiente manera: Sección 2 presenta un compendio de los trabajos relacionados con la monitorización ambulatoria en dispositivos Android. Sección 3 describe el diseño del sistema HeartDroid. Sección 4 muestra la implementación así como las pruebas realizadas y los resultados obtenidos. Las conclusiones y los trabajos a futuro se encuentran en la Sección 5.

2 Estado del arte

La Salud Móvil (mHealth) [4] es una de las tecnologías más innovadoras en cuidado de la salud. Su propósito principal es el de transmitir, almacenar, procesar y recuperar datos tanto en tiempo no-real como en tiempo real a los usuarios finales (pacien-

tes, doctores, familiares, etc.). Estos sistemas los podríamos clasificar dentro de tres grupos: 1) sistemas que recolectan las señales obtenidas de los sensores realizando el análisis y clasificación de los datos recolectados después de terminado un periodo de tiempo; 2) sistemas que realizan el análisis y clasificación en tiempo real de manera remota y 3) sistemas que proveen de análisis y clasificación en tiempo real de manera local. Algunos de los trabajos más importantes son descritos a continuación.

2.1 Análisis y clasificación después de la recolección de datos

Dentro de la primera clasificación de sistemas podemos hablar de sistemas basados en Holter (Figura 1), los cuales realizan la recolección de la señal ECG durante un periodo determinado de tiempo, siendo 24 horas lo más común. Sin embargo, no existe ningún tipo de procesamiento de la señal en tiempo real ya que el análisis se realiza después de que la recolección de datos llegó a su fin.

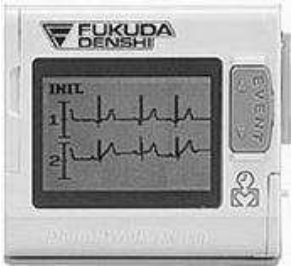


Fig. 1. Holter Fukuda Denshi, 48 horas máximo de grabación.

2.2 Análisis y clasificación de forma remota

Esta clasificación se caracteriza por realizar el procesamiento de la información recolectada de forma remota, esto implica que siempre se debe tener una conexión estable de datos. En [5] se describe el sistema Telecare (Figura 2) el cual presenta una pulsera que sensa el ritmo cardiaco, saturación de oxígeno en la sangre y agrega un nuevo sensor, un acelerómetro, el cual tiene como tarea monitorear la actividad motriz diaria. Todas las señales son enviadas mediante Bluetooth al teléfono inteligente donde posteriormente serán transmitidas a la Web para ser registrados en la base de datos. Una vez que los datos son sincronizados con la base de datos el personal médico realizará un análisis detallado de la información para proporcionar un diagnóstico y asistir al paciente que porta el dispositivo.

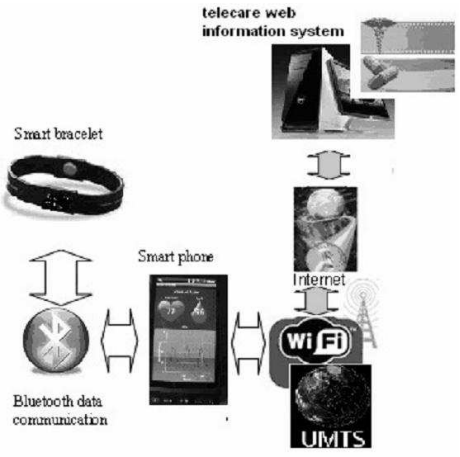


Fig. 2. Diagrama de bloques del sistema Telecare [5]

2.3 Análisis y clasificación en tiempo real de manera local

Este tipo de sistemas si bien no están aislados de una central que recolecta los datos y procesa la información enviada, sí realizan un análisis local de manera profunda a las señales que recolectan para generar un diagnóstico previo y ver si existe riesgo alguno. Esta característica permite que el paciente pueda saber en todo momento si corre algún riesgo incluso si el dispositivo pierde la conexión con la central, con lo que es una mejora a las dos clasificaciones anteriormente descritas.

El proyecto propuesto en [6] llamado HeartToGo (Figura 3) se enfoca principalmente en enfermedades cardiovasculares.

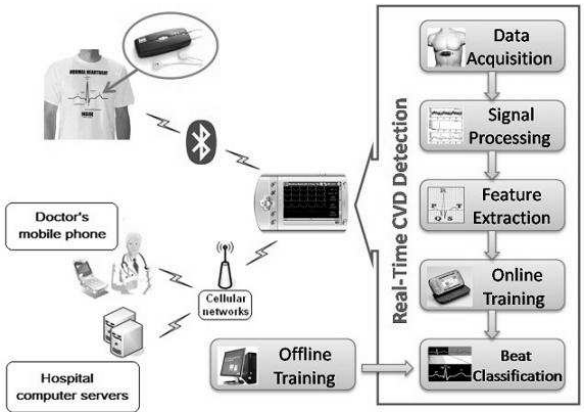


Fig. 3. Flujo de trabajo del procesado de la señal ECG en el sistema HeartToGo [6]

Este prototipo no solo permite colectar, almacenar, transmitir, y mostrar información de ECG en tiempo real, sino que además analiza la señal ECG adquirida y detec-

ta anomalías cardiacas, utilizando una red neuronal artificial capaz de clasificar 13 tipos de arritmias y el ritmo sinusal.

Como se ha mencionado en los trabajos anteriormente citados, el diagnóstico médico es uno de los temas más importantes en el cuidado de la salud. La precisión en el diagnóstico contribuye a que el paciente reciba un tratamiento adecuado, lo que conlleva a su recuperación y por ende a la cura o control de la enfermedad.

La aplicación HeartDroid que se describe a continuación queda dentro de la tercera clasificación de las antes presentadas, esto implica que cuenta con las características de registro de la información recolectada, además de poder realizar un análisis de estos datos para en base a reglas predefinidas dar un pre-diagnóstico del estado de salud del paciente y en caso de existir una anomalía informar al personal médico y tutores asignados. En la siguiente sección se describe el diseño de la aplicación así como su funcionamiento.

3 Diseño de HeartDroid

La aplicación de HeartDroid utiliza un sensor que proporciona una serie de datos que adquiere de la persona que lo porta. El sensor Zephyr BioHarness 3 utiliza el protocolo Bluetooth para la comunicación con el dispositivo Android. Sobre este sensor se crea una aplicación capaz de comunicarse con él y mostrar, almacenar e interpretar los datos adquiridos.

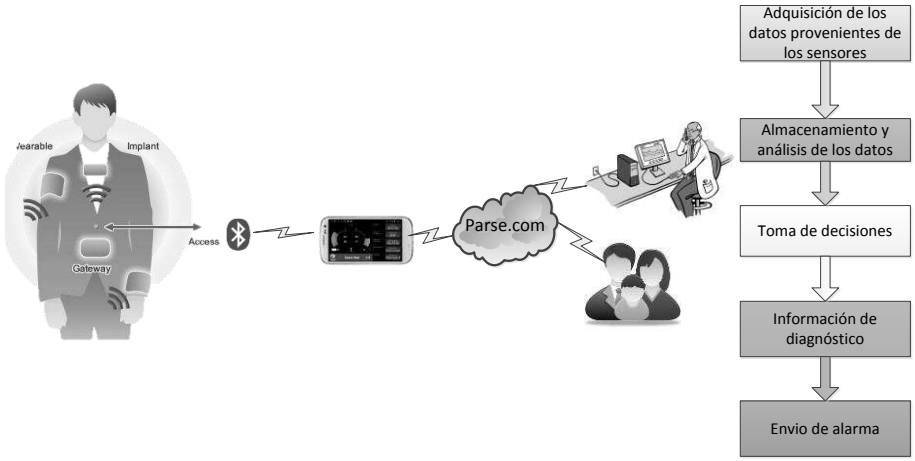


Fig. 4. Arquitectura de HeartDroid.

La principal característica de la aplicación son los mensajes de alerta que utiliza para comunicar tanto al portador del sensor como a un médico y/o tutor (que se hayan registrado para estar al pendiente) y proporcionar asistencia de ser necesario (Figura 4).

3.1 Servicio de monitorización

El núcleo de la aplicación es un Servicio, componente del sistema operativo que se ejecuta en el teléfono Android en segundo plano. Este servicio es el encargado de mantener la comunicación con el sensor BioHarness 3, realizar el análisis de los datos adquiridos y su almacenamiento, todo ello de forma automática sin que el usuario tenga que mantener la aplicación en pantalla, además, el servicio se encarga de la comunicación con la interfaz gráfica que muestra los signos vitales y envío de los mensajes de alarma.

El servicio puede convivir con todas las demás aplicaciones que estén trabajando en primer plano, sin embargo está configurado para reiniciarse automáticamente en caso de que el sistema requiera terminar su ejecución para liberar recursos, con lo que se trata de asegurar que siempre esté monitoreando al usuario (Figura 5).

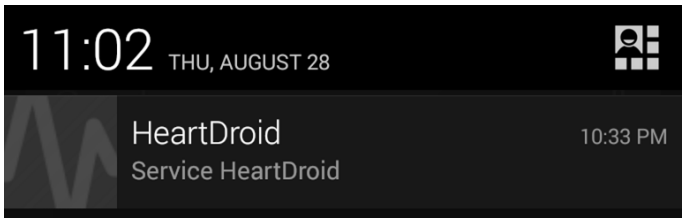


Fig. 5. Servicio HeartDroid corriendo en segundo plano

3.2 Alarmas

Las alarmas son de distintos tipos dependiendo de la situación. No todas las anomalías cardiacas (arritmias) tienen el mismo grado de peligrosidad, por lo cual la atención que requieren no es la misma. En la siguiente tabla se hace una clasificación de la peligrosidad de las arritmias que el sistema es capaz de reconocer.

Tabla 1. Clasificación de arritmias

Arritmia	Latidos por minuto (lpm)	Peligrosidad
Bradicardia peligrosa	< 40	Alta
Bradicardia (sinusal)	40 > lpm < 60	Baja
Ritmo Sinusal	60 > lpm < 100	Nula
Taquicardia lenta (sinusal)	100 > lpm < 120	Baja
Taquicardia rápida (sinusal)	120 > lpm < 180	Mediana
Taquicardia peligrosa	180 > lpm	Alta

Teniendo en cuenta la tabla anterior es necesario mencionar que se tienen que dar cumplir varios criterios dependiendo de la arritmia, esto con el fin de evitar lo más posible falsos positivos. Estos criterios son una combinación de datos ingresados en el historial médico y datos que se obtienen en tiempo real, entre ellos la actividad física que se realiza.

Si se sobrepasan los límites prestablecidos (en combinación con otros criterios) y la bradicardia o taquicardia continua por un periodo de tiempo mayor a un minuto, es indicio de una posible emergencia. En este caso se dispara una alarma audible y visual para la persona que porta el sensor y a su vez se envían notificaciones o mensajes SMS de alarma a una lista de médicos y tutores previamente ingresados en la aplicación. El médico al recibir la notificación puede ver con más detalle la situación y se le dan opciones para tomar acciones (Figura 6).

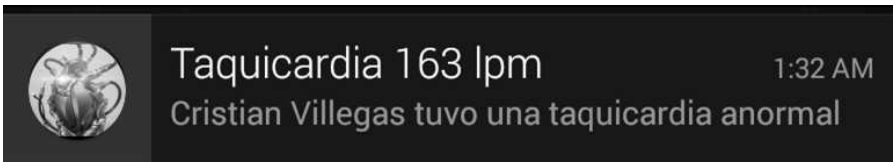


Fig. 6. Mensaje de notificación.

3.3 Funciones adicionales

La aplicación no solo permite lo antes mencionado, sino además tiene agregadas otras funcionalidades, tales como llevar una bitácora de lo que el paciente realiza durante su actividad diaria. Además es posible ver en tiempo real los datos adquiridos por el sensor de dos formas: una de manera gráfica mostrando el ECG, ritmo cardiaco, respiratorio y la actividad física, mientras la otra más sencilla solo muestra los valores en forma de texto. Como agregado la aplicación permite la gestión de una base de datos para pacientes, tutores, médicos. Desde esta base de datos se establecen tanto los especialistas y tutores que estarán relacionados con el monitoreo. Esta base de datos esta sincronizada con una base de datos de respaldo en la nube que proporciona el servicio de Parse.com (Figura 7). Este mismo servicio en la nube es el que facilita el envío y recepción de notificaciones de alarma entre dispositivos Android.

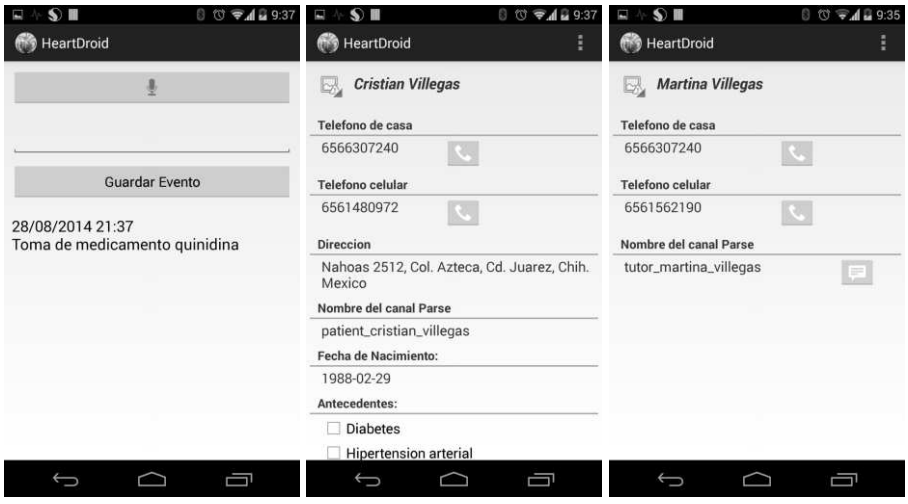


Fig. 7. Registro de eventos, información de la persona monitoreada, datos de un tutor.

4 Pruebas y Resultados

Con el fin de realizar pruebas se realizó una estancia en el hospital Cetro Médico de Especialidades (CME) de ciudad Juárez, Chihuahua del 10 de marzo al 10 de abril de 2014, bajo la tutela del cardiólogo Dr. Luis Rodolfo Flores Montaño jefe de Juárez Cardio-Vascular.

Para las pruebas se utilizó el sensor Zephyr BioHarness 3, un LG Nexus 4 con la versión de Android Kit-Kat (4.4.4) con conexión a internet vía 3G. Con el fin de establecer la validez de las lecturas proporcionadas por el sensor se realizó una comparativa entre el sensor BH 3 y el Holter (Philips DigiTrack-Plus) que utilizan para realizar las pruebas de ECG ambulatorio. La prueba consistió en que una persona saludable portara ambos dispositivos por 24 horas para el registro de sus signos vitales. El resultado de esta prueba permitió establecer que el sensor cuenta con la precisión necesaria para realizar un diagnóstico médico del estado de salud del paciente, quedando el personal médico conforme con el desempeño del sensor, permitiendo continuar con las pruebas de detección de arritmias bajo diferentes escenarios.

Las pruebas se condujeron de dos formas, la primera con el fin de asegurar que los tipos de alarma fueran activados correctamente se simularon diferentes condiciones, utilizando valores de ritmo cardiaco predefinidos al rango que se observa en la Tabla 1. Cabe mencionar que las alarmas no se activan en el momento de que el valor entra en el rango, sino que se configuró el prototipo para esperar un determinado tiempo en el que por ejemplo si una taquicardia se presenta, ésta debe sostenerse por al menos un minuto con valores de actividad física mínima para que pueda dispararse la alarma correspondiente. En todos los casos se disparó la alarma de forma adecuada ingresando los valores simulados correspondientes, con ello en mente se puso a prueba con personas portando el sensor (Figura 8).

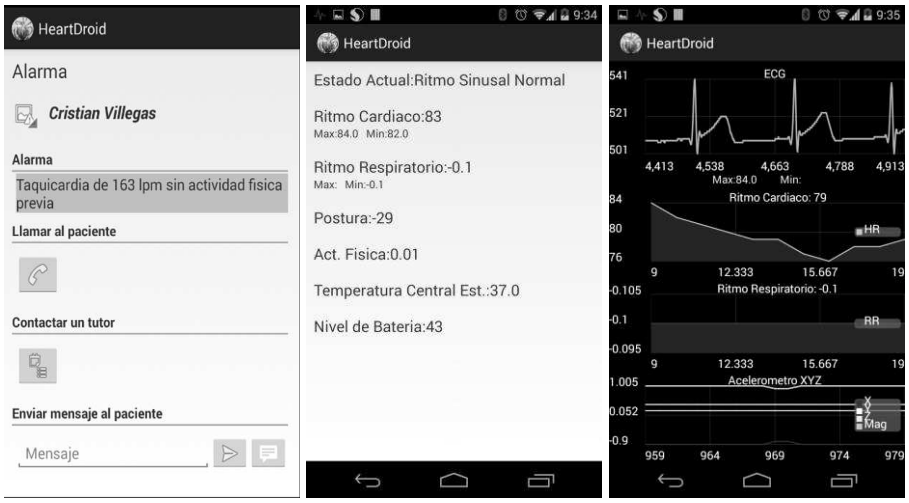


Fig. 8. Detalles el mensaje de alarma, visualización de signos vitales, ECG y actividad física.

Para la segunda prueba se crearon 3 escenarios donde un grupo de 5 personas participo: El primer escenario consistió en que cada persona debió mantenerse en reposo en una silla para lograr un ritmo cardiaco normal; la segunda prueba se realizó recostando a cada persona para bajar su ritmo cardiaco y llegar a una bradicardia y por último una situación de estrés, donde había que subir a una caminadora eléctrica y correr para provocar una taquicardia. Dado que provocar tanto una bradicardia por debajo de 40 lpm o una taquicardia arriba de 180 lpm es muy peligroso, estos valores solo fueron probados en las simulaciones (Figura 8).

Las pruebas tuvieron un tiempo variado dependiendo de la persona, sin embargo los resultados para todos fueron similares, en el caso de los dos primeros escenarios hubo un 100% de aciertos a la hora de disparar la alarma, sin embargo en el tercer escenario se encontró un contratiempo a la hora de que se realiza una actividad física intensa: el ruido se incrementó de forma considerable impidiendo lecturas correctas, oscilando desde los 0 latidos por minuto hasta los 200; el sudor y el contacto físico con la banda que se coloca en el pecho hace que los sensores no tengan un buen contacto.

Una vez detectado este problema se contactó a la empresa responsable del sensor la cual recomendó utilizar el BioModule Holder en lugar del Chest Strap que viene con el sensor por defecto (Figura 9).

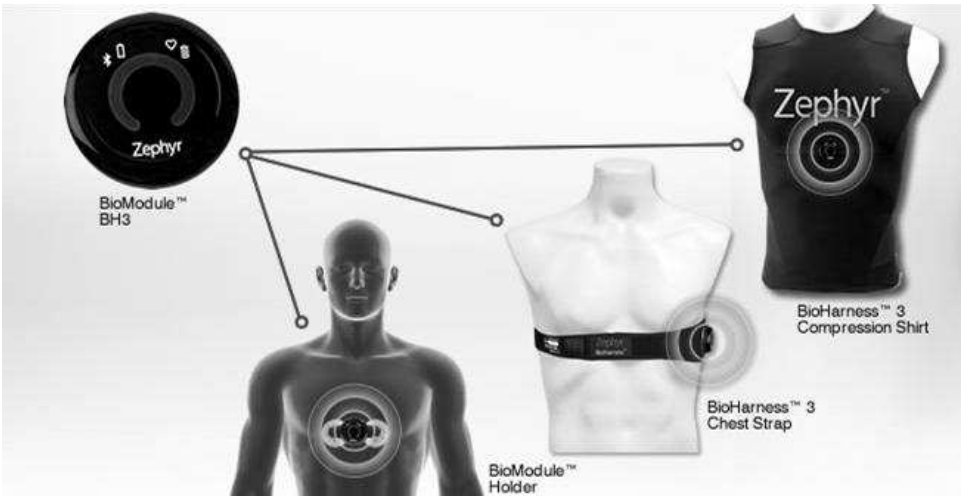


Fig. 9. Sensor BioHarness 3 [7]

Realizando una prueba posterior con este módulo alternativo que utiliza electrodos, se vio un descenso del ruido a la hora de hacer actividad física intensa, con lo que se pudo comprobar que la aplicación funciona de forma adecuada. Sin embargo existen diferencias entre ambos módulos, la principal y más significativa es que al usar el módulo de Holder se pierde la capacidad de adquirir el ritmo de respiración así como de estimar la temperatura corporal.

5 Conclusiones y trabajos a futuro

El monitoreo de signos cada vez se encuentra más presente en el día a día, utilizando tecnologías que nos permiten estar al tanto de nuestra salud, existiendo diversas propuestas como la de Apple y su sistema Health para el monitoreo de la salud, además del anuncio por parte de Google y su aplicación de Fit, ambos utilizando dispositivos diversos para la recolección de la frecuencia cardiaca. En este trabajo se diseñó la aplicación HeartDroid para monitoreo de enfermedades cardiacas utilizando el sensor Zephyr BioHarness 3 y un dispositivo Android. Este monitoreo de signos permite alertar tanto al usuario que lo porta así como a un médico y/o tutor de que ocurre una anomalía en tiempo real. Las pruebas realizadas y los resultados obtenidos muestran los inconvenientes de la presencia de ruido, sin embargo se superó este problema al colocar el sensor BH 3 en el Holder (Figura 9), que al tener mejor contacto con la piel por medio de electrodos adheribles, permite tener mayor precisión en las mediciones y un correcto aviso de la alarma correspondiente dada la situación. Este trabajo contribuye con un prototipo que puede utilizarse en ambientes reales, no con el fin de reemplazar a los dispositivos médicos actuales sino como un complemento para realizar monitoreo en tiempo real.

Como trabajo a futuro en la aplicación HeartDroid se pretende utilizar la información recolectada del ECG con el fin de detectar arritmias específicas que no son identificables solo utilizando el ritmo cardiaco, es decir, la identificación de cada latido individual, además que es importante mejorar el sistema de alarmas para incluir más información que ayude al correcto análisis y diagnóstico de la anomalía.

Agradecimientos

Este trabajo contó para su realización con el apoyo del Instituto Politécnico Nacional, a través de la "Beca Tesis Maestría" y el proyecto SIP 20140791.

Referencias

1. John G. Webster Encyclopedia of Medical Devices and Instrumentation Second Edition Volume 1pp. 12-13, 2006
2. Zhang Y.; Xiao H. "Bluetooth-Based Sensor Networks for Remotely Monitoring the Pshysical Signals of a Patient" IEEE Transactions on Information Technology in Biomedicine, Vol. 13, No. 6, November 2009.
3. Mikati I. "Holter Monitor", Verimed Healthcare Network, 2010.
4. mHealth New horizons for health through mobile technologies (2011), Global Observatory for eHealth series, Vol. 3, World Health Organization.
5. Postolache O.; Girao P.; Ribeiro M. "Enabling telecare assessment with pervasive sensing and Android OS smartphone" Medical Measurements and Applications Proceedings (MeMeA) IEEE International Workshop, 2011.
6. Jin Z.; Sun Y.; Cheng A. "Predicting Cardiovascular Disease from Real-Time Electrocardiographic Monitoring: An Adaptive Machine Learning Approach on a Cell Phone" 31st Annual International Conference of the IEEE EMBS, Minneapolis, Minnesota, USA, Septiembre 2-6 2009.
7. Zephyr BioHarness 3, http://zephyranywhere.com/products/bioharness-3

Intelligent Processing of Geospatial Information

An Improved Flexible Similarity Function for Clustering-Based Crime Analysis

Hiram Calvo, Salvador Godoy-Calderón, Marco A. Moreno-Armendáriz

Centro de Investigación en Computación, Instituto Politécnico Nacional,
Av. Juan de Dios Bátiz e/M.O. Mendizábal s/n, Nva. Ind. Vallejo, 07738
{hcalvo, sgodoyc, mam_armendariz}@cic.ipn.mx

Abstract. In this paper, a novel similarity function is used to identify hot-spots of criminal activity in large crime-datasets. This function considers the space and times when each crime was committed, as well as some elements of the perceived *modus operandi* of the perpetrator, in order to compare specific crime patterns and then cluster them using a density-based clustering algorithm. The clusters so formed are then graphically shown to the crime analyst using diverse GIS-tools, in order to provide him/her with high quality information about the current state of criminal activity. Several experiments performed, as well as a case-based comparison with previously published similar proposals, yield significant advantages of the proposed function over classical Euclidean-distance comparisons and other space-time similarity functions.

Keywords. Crime Analysis, Clustering, Spatio-Temporal Similarity Function.

1 Introduction

Recently, crime analysis has become one of the major concerns of public security departments around the world. Crime analysis is defined as a set of processes followed in order to gather, identify, organize, analyze and process data corresponding to the criminal activity in the area under study, to obtain useful information for preventing and fighting crime. Presently there are many analysis techniques (McCue, 2006; Osborne & Wernicke, 2003; Mena, 2003; Liu & Eck, 2008), and several commercial systems have been proposed in literature to address this problem. For more information, see (Levine, 1996, Chen *et al*., 2002) and BAIR's tools[1]. In practice, all of the commercially available crime-analysis systems are based on probability distribution analysis techniques, as well as on probabilistic forecast tools such as the Naïve-Bayes classifier. Only very few of these systems perform spatial and temporal analysis by using clustering techniques and time series analysis.

In Mexico, the legal system allows two types of police institutions: preventive and investigative. The task of the first one is focused on crime prevention; therefore, spatio-temporal analysis turns out to be of the utmost importance, since it allows the study of urban areas with high concentration of criminal activities. On the other hand, the main goal of the investigating police is to identify and capture individuals or groups responsible for committing specific crimes within a delimited jurisdiction. For this kind of police institution a careful analysis of each

[1] www.bairanalytics.com

Advances in Computer Science
Research in computing Science 83, 2014 pp. 25-34

crime's *modus operandi* (henceforth, MO) is crucial. The aforementioned differences among police institutions are relevant to this research because it is focused on managing both contexts. Both types of institutions operate on exactly the same datasets; the main difference is the weighting of the criminal features to be analyzed. If preventive analysis is our main concern, space-time features are the most important ones, followed by the type of crime and its features. If investigation analysis is our focus, some characteristics of crime contribute more information, followed by space-time features. Our proposed methodology may be adapted to the work of both: preventive and investigative police forces yielding, in both cases, a more suitable interpretation for the created clusters, in each of the analysis contexts.

1.1 Related work

Related to clustering-based crime-analysis, Nath (2006) presents a similar work to us. We proceed to describe it for comparison purposes. Nath uses a *k*-means algorithm is used for identifying crime patterns—a crime pattern is described as a specific group of criminal actions with similar MO characteristics. He calls a group or cluster of crimes, a pattern. Experiments are made on a small sample of data, shown in Figure 1.

Crime Type	Suspect's Race	Suspect's Sex	Suspect's Age gr	Victim's Age gr	Weapon
Robbery	B	M	Middle	Elderly	Knife
Robbery	W	M	Young	Middle	Bat
Robbery	B	M	?	Elderly	Knife
Robbery	B	F	Middle	Young	Piston

Figure 1. Example of criminal data (Nath, 2006)

By applying a *k*-means clustering algorithm to the datasets, the author groups crimes with similar MO. He explains that in the robbery sample (Figure 1) pattern behavior may be observed in rows 1 and 3, where the suspect's description matches, as well as the victim's profile. However, no explanation is given about the intra-class diversity of the crimes, and why the author used *k*-means as the clustering algorithm of choice. Figure 2 summarizes the results published by S.V. Nath.

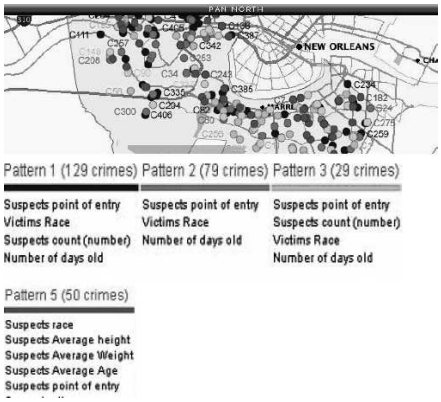


Figure 2. Experiments results (Nath, 2006)

In this paper we propose the use of a clustering technique based on pattern density, together with a space-time similarity function to identify areas with high concentration of crime (hot-spots). Then we compare the results obtained with our similarity function with those obtained by the proposed similarity function used in the original paper of the ST-DBSCAN algorithm (Birant and Kut, 1996). The comparison criteria used by the space-time similarity function and its specific use to cluster criminal activities are the main contributions of this paper.

2 Clustering-Based Crime Analysis

The purpose of using density-based clustering techniques in the context of crime-analysis is to achieve a non-statistical identification of the observed spatial and temporal trends in the commission of crimes, as well as to isolate exception cases that do not fit into those trends. This information is useful for the crime-analyst in order to develop specific strategies, both to fight and prevent delinquency, in the middle and long terms.

2.1 Pattern representation

Let us define a crime-pattern as the abstract representation of a single criminal phenomenon. This crime-pattern consists of three main components: crime specifics, space, and time; all three related to the perpetration of a specific crime. This allows the specialist to identify periods of high criminal activity and their geographic location. Therefore, the components of a criminal pattern are the following:

1) *Crime specifics*: It indicates the specific type of crime committed as well as many of its characteristics, such as the level of violence, number of persons implicated, types of weapons, *modus* operandi, etc.
2) *Space Location*: Geographic area where the crime was committed. This feature may be observed at different levels of detail: state level, patrolling sector, residential development or even street and number.
3) *Time Location*: Time when the crime was perpetrated. In this component, as well as in the one mentioned above, several levels of detail are possible: year, month, date or even the time of the day.

Each one of these components may consist of one or more variables that will be called pattern features that provide a higher level of detail to the pattern. A crime-pattern (D) has the following structure:

$$D = \left(\langle crime\ specifics \rangle, \langle space\ location \rangle, \langle time\ location \rangle \right)$$

The trend-identification process starts by analyzing a set of crime-patterns, each one with the same level of detail and within a limited geographic location, occurred within a given time interval. Table 1 contains a sample of burglary patterns obtained from the Cuautitlán Izcalli area, in the State of Mexico.

Table 1. Data sample (out of 80 patterns) of the *burglary* dataset.

Weapon	Location	Date
Firearm	Ensueños	22/08
Not specified	Cumbria	21/08
Sharp instrument	Arcos de la Hacienda	13/09
Banned weapon	San Isidro Labrador	01/03

In Figure 3 can be seen that such patterns are plotted in the map of the corresponding area divided into surveillance sectors.

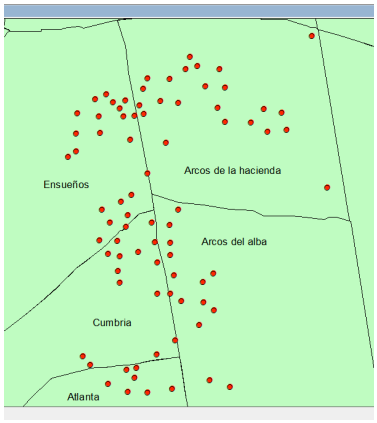


Figure 3. Criminal data referred to the geographic area where they were committed. Each red spot represents a crime incident of the "burglary" type.

In Table 2 we show another sample of the same dataset with a different level of detail from the one shown in the former sample. The differences between the two sets can be observed on the temporal and crime specific components of the patterns. This second set contains robbery-patterns and their location is another surveillance center within the same district of Cuautitlán Izcalli, Mexico.

Table 2. Data sample (out of 126 patterns) from the *robbery* data set.

Robbery type	Weapon	No. of Members	Location	Month
Break-in	Sharp instrument	2	El Rosario	DEC
Auto-parts theft	Without weapons	4	Adolfo López Mateos	FEB
Robbery to passer-by	Firearm	3	El Rosario	FEB
Robbery to passer-by	Sharp instrument	1	Cofradía-III	DEC

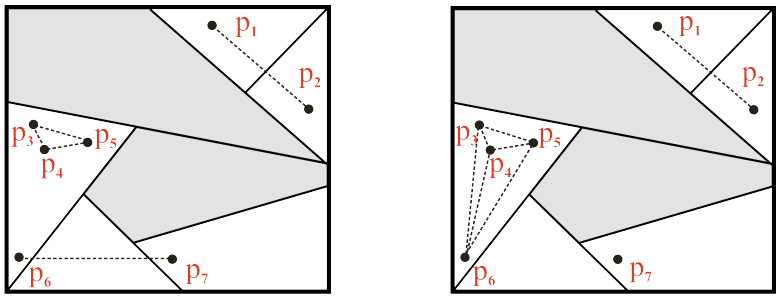
The complete first dataset (burglary) contains 80 patterns, while the second one (robbery) contains 126 patterns.

2.2 Space Comparison Criterion and the Similarity Function

The comparison between patterns is achieved by a similarity function formed by the weighted sum of a set of comparison criteria (Cc) normalized according to the number of features that form the pattern. Each of these comparison criteria

measures the similarity of each feature that makes-up the pattern. The way in which the similarity of the space component is measured is shown below.

Figure 4 shows an area divided into four surveillance sectors: A, B, C and D. A surveillance sector is a geographic zone made up by location (residential developments) and it is defined by the police department for the allocation of financial resources. According to the space comparison criterion based on Euclidean distance, patterns belonging to the same sector but separated by a long distance will not be similar (See figure 4a, patterns p4 and p6), while patterns which are geographically close to each other, even if they belong to different sectors, will be. See Figure 4a, patterns: p1 and p2.



4a) Euclidean comparison **4b)** Comparison by sector division

Figure 4. Surveillance Sectors

The spatial comparison criterion based on regions divided by surveillance sectors, strongly suggests that the maximum space-similarity should be achieved by patterns belonging to the same sector (See Figure 4b, patterns: p3 to p6), followed by patterns belonging to contiguous sectors (See Figure 4b, patterns: p1 and p2). This kind of clustering yields much more useful clusters because patrolling routines, as well as investigative teams usually schedule their operations by sector. Of course, different comparison criteria can be considered depending on the needs by each crime-analyst.

Our proposed similarity function is defined by the following equation:

$$f(o_i,o_j)=\frac{1}{r}\sum_{s=1}^{r}\left(\alpha_s Cc_s(o_i,o_j)\right) \quad (1)$$

Where:
- r is the number of features that make up the pattern.
- $o_i o_j$ are the patterns being compared.
- α_s is the weighting factor of feature s.
- $Cc_s(\,)$ is the space-time and attribute comparison criteria for feature s.

Experiments shown in Section 3 will show that this similarity function based on our space comparison criterion produce better results than the space comparison criterion based on Euclidean distance.

2.3 Density-Based Clustering

Density-based clustering algorithms can group patterns with a high spatial concentration within a delimited region isolating in the processes those patterns with a low degree of spatial similarity, which are regarded as non-related appearances of criminal activity. In this paper, we use the ST-DBSCAN algorithm, an extension of DBSCAN to work with space-time components (Birant and Kut, 2007). This algorithm was implemented as originally described with the exception of the similarity function. Our proposed spatial comparition criteria and similarity function (section 2.2, equation 1) were substituted in place of the original ones based on Euclidean distance and sector division.

2.3.1 The ST-DBSCAN algorithm.

Briant and Kut (2007) describe an extension of the DBSCAN algorithm for spatio-temporal clustering, called ST-DBSCAN. This extension proposes the separate calculation of space and time similarities between patterns.

The ST-DBSCAN algorithm requires, besides the dataset to be processed, the following two parameters: the minimum number of neighbors around an object to consider a high density situation, which will be called ***MinPts***; and the radius of the neighborhood which will be called ***Eps***. For more information about the ST-DBSCAN algorithm refer to (Briant and Kut, 2007).

3 Experimental Results

The values of ***Eps*** and ***MinPts*** in our implementation of ST-DBSCAN were calculated with the following equations:

$$Eps = 1 - \min(f(o_i, o_j)) \tag{2}$$

where

N is the number of patterns

$f(o_i, o_j)$ is the similarity function : $i = 1, 2, \ldots, n;\ j \neq i$

$$MinPts = |O| + 1 \tag{3}$$

where: $|O|$ is the cardinality of the pattern.

The first experiment was conducted over the *burglary* dataset. For this experiment the feature called *weapon* is the most relevant attribute, so the best result is deemed to be the one that groups criminal incidents committed with the same weapon.

A very important aspect to be considered is the clarification of the difference between the patterns identified as noise and the outliers in the dataset. Noise refers to those patterns that may be spatially similar to other patterns or groups, but do not share other similar characteristics, while an outlier is a pattern geometrically separated from other patterns or groups. This clarification is necessary because the ST-DBSCAN identifies noise.

The results generated by the Euclidean space similarity function vs. our proposed space competition criteria are shown in Figure 6. In both figures (5a and

5b) the patterns labeled by question signs represent noise patterns (elements that do not have characteristics similar to others).

Cluster ***A*** (see Figure 5a) represents burglary acts perpetrated with a firearm, while the ***n2*** noise pattern is another burglary crime perpetrated with a "non-specified" weapon. Despite this, not all of the patterns in cluster ***A*** were perpetrated with a "firearm". The four triangular patterns surrounding the ***n2*** pattern were committed "without weapons" (see Figure 5a). That is where the Euclidean space similarity function fails, because it groups them in the same cluster given their geographic similarity, although they are not closely related.

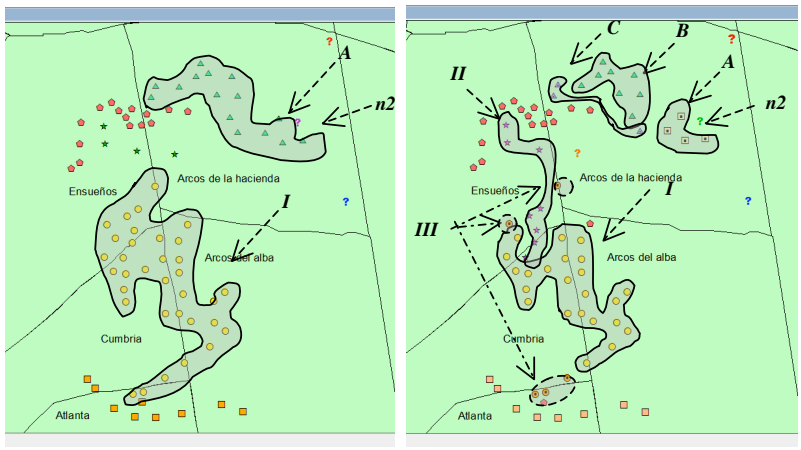


Figure 5. ST-DBSCAN results: 5(a) with Euclidean space similarity function (left), 5(b) with space similarity function based on sector division (right).

Figure 5b shows this difference, the noise pattern identified by ***n2*** is the same as the noise pattern identified in Figure 5a. Cluster ***A*** (see Figure 5b) contains the four criminal patterns perpetrated using a firearm, while the patterns that make-up Cluster ***B*** were committed without weapons. This result turns out to be very important because, following this path, crimes that were probably perpetrated by the same aggressors can be semantically identified.

In the second experiment we worked with patterns that have a higher level of detail, which means more descriptive features. Also, each component (set of related features) in the crime-patterns, were weighted as follows: 60% space-time, 30% crime-specifics and 10% crime features, due to the fact that some features are more important than others.

This experiment is more related with the work performed by the preventive police, due to the fact that the family of crimes related to robbery is the one under study. This crime-family is made up by: robbery to passerby, break-in, and auto-parts theft.

The weighting may be obtained through a criminology expert. The objective is to identify trends by taking advantage of the expert's knowledge in criminology. Figure 6 shows the results achieved.

Table 3 shows the results of our latest experiment. Of the two residential areas studied in the North areas, the one containing a higher amount of crimes from the robbery family is the Santa Barbara residential area, which belongs to the sector with the same name. Besides, we found that those months of the year with the

highest incidence of crime are July and September, so it is necessary to undertake programs and campaigns in such sector, and in that season of the year have prevention programs and campaigns to fight this type of crime.

Table 3. Results of clustering of the "robbery" type of crime data.

Cluster	Type of crime	Month	Year
●	Robbery to passer-by	July, August & September	2006, 2007 & 2008
■	Auto-parts theft	July & September	2007 & 2008
▲	Break-in	January & April	2008
⬟	Auto-parts theft	February & September	2006 & 2007
? *(noise)*	Break-in	December	2007

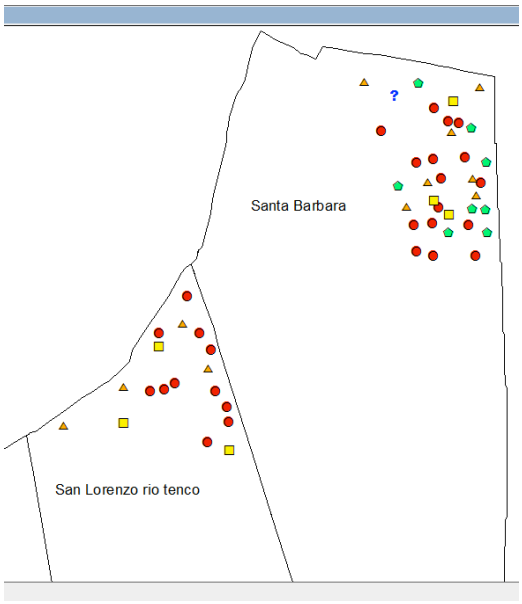


Figure 6. Clusters identified with ST-DBSCAN with a space similarity function based on sector division (Northern area).

4 Conclusions

This paper has shown that the ST-DBSCAN algorithm implemented with the space similarity function based on sector division generates better results than the one based on Euclidean distance, taking the following aspects into account: (1) The semantics adapt better to reality under the context of the type of analysis made and (2) Higher percentage of noise identification contributes to the reduction of elements for the analysis.

Acknowledgements: The authors thank the support of the Mexican Government (CONACYT, SNI, SIP-IPN, COFAA-IPN, and PIFI-IPN), and especially to the Cuautitlán Izcalli local government.

References:

1. Birant D., A. Kut,(2007) "ST-DBSCAN: An algorithm for clustering spatial-temporal data", Data & Knowledge Engineering 60: 208-221.
2. Chen, H., J. Schroeder, R. V. Hauck, L. Ridgeway, H. Atabakhsh, H. Gupta, C. Boarman, K. Rasmussen, A. W. Clements, (2002) "COPLINK Connect: information and knowledge management for law enforcement", Decisions Support Systems, 34: 271-285.
3. Ester M., H. P. Kriegel, J. Sander, X. Xu, (1996) "A density-based for discovering clusters in large spatial databases with noise", Proceedings of Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, 226- 231.
4. Levine, N. (1996) "Spatial statistics and GIS: Software tools to quantify spatial patterns", Journal of the American Planning Association, 62 (3): 381-392.
5. Liu, Lin and John Eck (2008), "Artificial Crime Analysis Systems. Using Computer Simulations and Geographic Information Systems", IGI Global.
6. McCue, C. (2007), "Data mining and predictive analysis: intelligence gathering and crime analysis", Butterworth-Heinemann ELSEVIER.
7. McCue, Colleen (2006), "Data Mining and Predictive Analysis. Intelligence gathering and Crime Analysis", Elsevier.
8. Mena, Jesus (2003), "Investigative Data Mining for Security and Criminal Detection", Butterworth Heinemann.
9. Nath, S. V. (2006), "Crime Pattern Detection Using Data Mining", Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology, 41-44.
10. Osborne, Deborah A. and Susan C. Wernicke (2003), "Introduction to Crime Analysis. Basic Resources for Criminal Justice Practice", The Haworth Press, Rutledge Taylor & Francis Group.

Micro technologies & Embedded Systems

Diseño de una Cola de Instrucciones para Acceso a Memoria (LSQ) Basado en la Política de Envejecimiento de Instrucciones

Eduardo Pacheco-González, César A. Hernández-Calderón, Mauricio Ontiveros-Rodríguez

Centro de Investigación en Computación, Instituto Politécnico Nacional D. F., México.

pachecoeg086@gmail.com, hdzces@outlook.com, ont.mauricio@gmail.com

Abstract. In this paper we present a Load/Store Queue (LSQ) design for Memory Access which operates under the aging instructions principle whit the aim of achieving memory disambiguation. This module has been developed considering its integration into a super-scalar processor´s pipeline, capable of issue two instructions per clock cycle. The LSQ has been described in Verilog on QuartusII program and the simulations were performed in ModelSim

Resumen. En el presente trabajo se expone el diseño de una Cola de Instrucciones para Acceso a Memoria (*Load/Store Queue* - LSQ) que opera bajo el principio de envejecimiento de instrucciones con el objetivo de conseguir la desambiguación de memoria. Este módulo ha sido desarrollado considerando su integración al *pipeline* de un procesador súper-escalar capaz de emitir dos instrucciones por ciclo de reloj. La LSQ ha sido descrita en Verilog sobre el programa QuartusII, y las simulaciones se han realizado en ModelSim.

Keywords: Diseño de Arquitectura de Computadoras, Acceso a Memoria, Colas FIFO, HDL, Verilog, FPGA.

1 Introducción

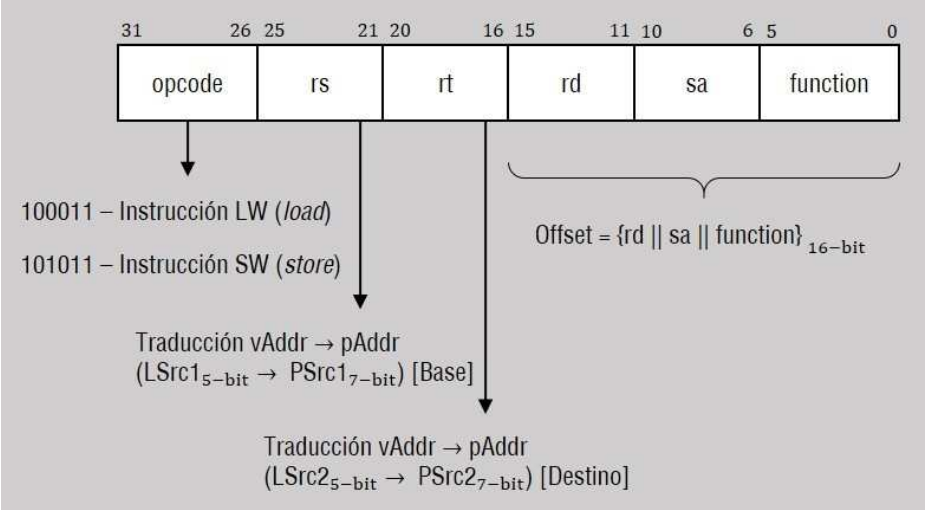
Hoy en día, los arquitectos de computadoras han mejorado el desempeño de los procesadores al implementar *pipelines* con mayor profundidad, al engrosar la emisión de instrucciones, y al ampliar la capacidad de la ventana de instrucciones con ejecución fuera de orden. En consecuencia, se han obtenido procesadores capaces de tener más de cien instrucciones en vuelo, incrementando la cantidad instrucciones de acceso a memoria. Para garantizar la ausencia de violaciones durante el acceso a memoria es necesario asegurar una desambiguación de memoria implementando una cola de ins-

Advances in Computer Science
Research in computing Science 83, 2014 pp. 37-48

trucciones (*Load/Store Queue* - LSQ) que determine, a partir de alguna política de ordenamiento, que instrucciones tienen mayor prioridad para acceder al contenido de la memoria [1][2][3].

Las instrucciones de acceso a memoria tipo MIPS se clasifican de forma general en dos grupos: *load* y *store*. Estas instrucciones, además de acceder a la Memoria de Datos, tienen acceso al Banco de Registros de Enteros (*Integer Register File* – IRF). Las instrucciones de tipo ***load*** son todas aquellas instrucciones con las que se desea actualizar algún valor del IRF, por lo que estas instrucciones requieren de una **lectura** de la Memoria de Datos para después escribir el dato en el IRF. Las instrucciones de tipo ***store*** son todas aquellas instrucciones con las que se desea almacenar un dato previamente procesado, en la Memoria de Datos, por lo que inicialmente se realiza la lectura del IRF, y posteriormente, la **escritura** en la Memoria de Datos.

Los procesadores que basan su diseño en un conjunto de instrucciones tipo MIPS trabajan con instrucciones con una longitud de 32-bits en las que se pueden definir seis campos principales al tratarse de instrucciones de CPU: *opcode*, *rs*, *rt*, *rd*, *sa* y *function* (Fig. 1)[1]. El campo *opcode* contiene el código principal para cualquier instrucción tipo MIPS, y es en donde se especifica la instrucción que se desea procesar. En el conjunto de instrucciones para MIPS64 se cuenta con veintiuna instrucciones de tipo *load* distintas, y con diecisiete instrucciones de tipo *store* [4].



[1] Dependiendo del tipo de instrucción con la que se trabaje (COP0-EH, COP1-FP, COP2, etc.) los campos en los que se divide la instrucción serán distintos, así como la operación a la que hacen referencia cada campo, sin que su longitud de 32-bits varíe. Cabe resaltar que el único campo que se encuentra presente para cualquier tipo de instrucción es el campo *opcode*.

Fig. 1. Formato de una instrucción de 32-bits tipo MIPS. El campo *opcode* permite identificar si se trata de una instrucción de acceso a memoria. Los campos *rs* y *rt* (5-bits cada uno) son direcciones virtuales (vAddr) que son traducidas a direcciones físicas (pAddr) para realizar la lectura/escritura del Banco de Registros. Los campos *rd*, *sa* y *function* son usados para definir el valor del *offset* (16-bit) usado para calcular la dirección de lectura/escritura de la Memoria de Datos, según sea el caso.

Los campos *rs* y *rt* son direcciones virtuales (vAdrr) de 5-bits que deben ser traducidas a direcciones físicas (pAddr) de 7-bits por una Unidad de Renombrado de Registros para realizar la lectura y escritura en el Banco de Registros, como se explicará más adelante. Por último, el valor del *offset* requerido para calcular la dirección de acceso a memoria estará conformado por los valores de los campos *rd*, *sa* y *function*[2].

El diseño que aquí se presenta ha sido desarrollado para integrarse en el procesador Lagarto II [5], un procesador súper-escalar con una arquitectura de 64-bits tipo RISC, y ejecución fuera de orden, capaz de emitir dos instrucciones tipo MIPS en cada ciclo de reloj.

2 Diseño y Operación de la LSQ

Para el diseño de la LSQ se debe considerar que ésta realiza un intercambio de datos con cuatro módulos del procesador: el Mapper, el Registro de Estados, el IRF y la Memoria de Datos (Fig. 2). Así mismo, recibe las señales de reloj y reset (clk y rst, respectivamente), además de un bit de control adicional denominado MemRdy, que determina si la memoria se encuentra disponible para ser escrita o leída. La LSQ ha sido descrita en Verilog sobre el programa QuartusII de la compañía Altera.

[2] Debido a que el contenido de los campos *rd*, *sa* y *function* no es considerado de forma independiente en las instrucciones de acceso a memoria, no es necesario el conocer la descripción del procesamiento de esta información durante su recorrido por el *pipeline* para nuestros fines.

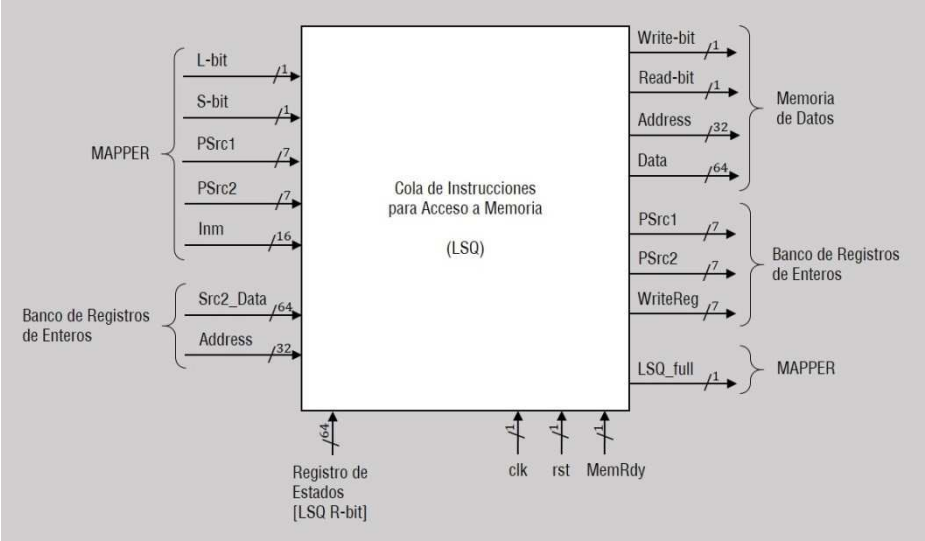


Fig. 2. En este esquema se muestran las señales de entrada y salida de la LSQ con las que se encuentra integrada procesador, siendo cuatro módulos con los que interactúa: Memoria de Datos, IRF, Mapper y el Registro de estados. Éste último, en combinación con el bit de control MemRdy, juegan un papel importante para el control de la operación del módulo.

En la Fig. 3 se presenta a detalle el diseño de la microarquitectura la LSQ, donde destacan los cinco módulos que forman la LSQ: Planificador de Instrucciones, Unidad de Extensión de Signo, Unidad de Cálculo de Direcciones (*Address Calculation Unit* – ACU), Buffer de Instrucciones y la Lógica de control.

2.1 Operación General de la LSQ

La información de las instrucciones proviene del Mapper, el cual organiza y emite dicha información hacia el Planificador de Instrucciones, que es el primer módulo que integra la LSQ, siempre y cuando el bit LSQ_full se lo permita. Este bit de control le hace saber al Mapper si el Planificador o el Buffer cuentan con espacio para poder almacenar los datos que son gestionados por la cola, este conteo es administrado por la Lógica Control.

El IRF se encuentra inmerso en el camino de datos de la LSQ. Sin importar el tipo de instrucción de la que se trate, se realiza la lectura del IRF en la dirección PSrc1 y la extensión de signo del *offset* en el momento en el que la información sale del Planificador. El dato resultante de la lectura del IRF es usado como operador en la ACU, en combinación con el valor del *offset* con signo extendido, para calcular la dirección con la que se tendrá acceso a la memoria (eq. 1). Si se trabaja con una instrucción de tipo *load*, se actualizará el registro WriteReg del IRF con el dato WriteData procedente de la memoria, mientras que al trabajar con una instrucción de tipo *store*, la dirección PSrc2 será almacenada de forma directa en el Buffer.

$$Address = base_{32} + \{signo\ extendido_{16} || offset_{16}\} \quad (1)$$

La Memoria de Datos recibe una dirección de 32-bits (Address) y los bits de control para escritura (Write-bit) o lectura (Read-bit). Al tratarse de una instrucción tipo *load*, la señal Read-bit deberá estar en alto, por lo que se indicará a la Memoria de Datos que se requiere leer la información de la dirección Address. Por otro lado, si se trata de una instrucción de tipo *store*, la señal Write-bit deberá estar en alto, y se escribirá en la dirección Address de la memoria un dato de 64-bits (Data) procedente del IRF.

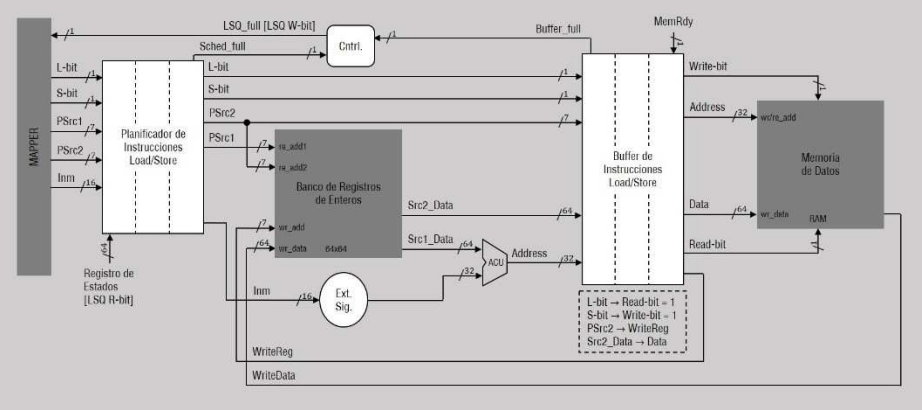


Fig. 3. El diseño de la LSQ comprende 5 módulos: Planificador de Instrucciones, Unidad de Extensión de Signo, Unidad de Cálculo de Direcciones (ACU), Buffer de Instrucciones y Lógica de Control. Los componentes ilustrados en gris son los elementos del procesador con los que interactúa la LSQ y no pertenecen propiamente al diseño de la misma.

2.2 Planificador de Instrucciones Load/Store

El Planificador de Instrucciones es el que marca la pauta a la LSQ sobre la información que será procesada ya que almacena y gestiona la emisión de los datos asociados a cuatro instrucciones de acceso a memoria (registros fuente, offset, bits de control, etc.) de forma simultánea. Estos datos procedentes del Mapper han sido previamente procesados por otros módulos incluidos en el *pipeline* del procesador.

Una vez almacenada la información, siempre la instrucción más vieja debe de ser emitida, para que lo anterior se cumpla, el vector de Registro de Estados es el encargado de indicar al Planificador que los registros fuente necesarios (PSrc1 y PSrc2) se encuentran listos, es decir, que el dato que almacena cada registro es válido para ser leído; Si se cumple finalmente, el Planificador genera la señal de control que permite la escritura de datos en el Buffer de Instrucciones, pero si lo anterior no llega a suceder el Planificador permitirá la entrada de datos hasta completar sus espacios disponibles de memoria, cuando se llegue a saturar el Planificador indicara al procesador que no alimente más instrucciones hasta tener espacios de memoria disponibles, este proceso es conocido como “Despertado de Instrucciones”.

2.3 Buffer de Instrucciones Load/Store

El Buffer de instrucciones, al igual que el Planificador, es de tipo *FIFO* (lo primero que entra es lo primero que sale), y está constituido por cinco ranuras para el almacenamiento de datos. Cada ranura está conformada por cinco campos para almacenar la información con la que se realizará el acceso a memoria: W-bit, R-bit, WriteReg, Address y Data.

Que la estructura sea de tipo FIFO nos permite mantener siempre el control sobre el orden en que nuestras instrucciones son emitidas, y así mantener la coherencia de las instrucciones contenidas en la LSQ.

Cuando la señal de control habilita la escritura en este módulo se almacenan los datos emitidos por el Planificador y los calculados en la LSQ. Este módulo representa el paso previo del acceso a memoria, por lo que la información que emita deberá de corresponder con el tipo de acceso que se desea realizar, ya sea lectura o escritura. Lo anterior es crítico, ya que de no tener almacenados los datos correctos en el Buffer la ejecución de instrucciones posteriores en el procesador arrojarán resultados erróneos, y en consecuencia, una mala ejecución del programa. Tomando en cuenta esto, se incluye una lógica que habilita las líneas de salida correspondientes al tipo de acceso a memoria, en las que se han almacenado datos válidos (Tabla 1).

Table 1. Campos válidos del Buffer de Instrucciones para el acceso a la Memria de Datos.

Instrucción	W-bit	R-bit	WriteReg	Address	Data
Load	0	1	✓	✓	-
Store	1	0	-	✓	✓

3 Validación

Para verificar el correcto funcionamiento del diseño y debido a que el procesador Lagarto II aún se encuentra la fase de diseño y con partes del pipeline aun en papel, la LSQ se ha integrado a la arquitectura de un Coprocesador vectorial [6] y se han realizado diversas simulaciones ejecutadas en ModelSim.

En la Fig. 4 y Fig. 5 se observa el resultado de la simulación del Coprocesador sin haber ejecutado ningún tipo de instrucción. En la Fig. 4 se muestra que el contenido inicial del IRF es de ceros, lo que indica que no hay ningún dato disponible para ser procesado por alguna instrucción. Por otra parte, en la Fig. 5, se muestra el contenido inicial de la Memoria de Datos, mismos que deben ser cargados al IRF mediante instrucciones de tipo *load* para trabajar con ellos.

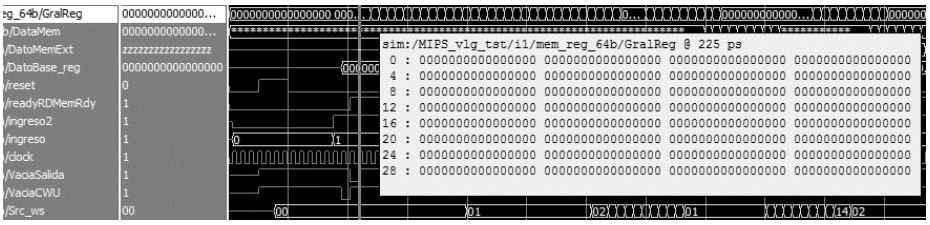
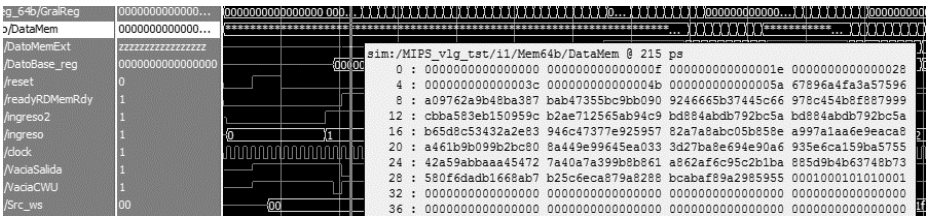


Fig. 4. Simulación del Coprocesador antes de ejecutar las instrucciones de tipo *load* para cargar los datos al IRF. En el recuadro de lado derecho se observa el contenido inicial del IRF.



a)

b)

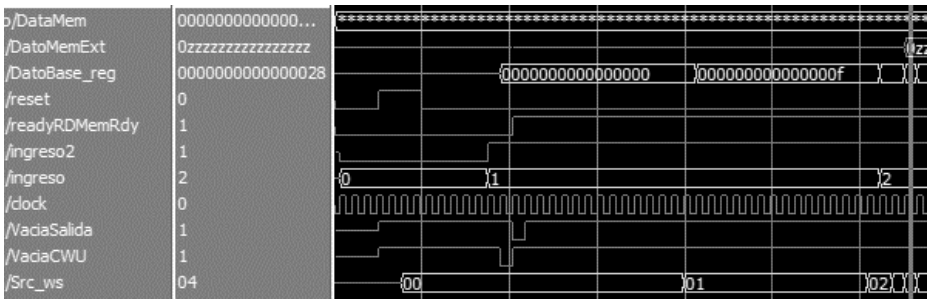
Fig. 5. a) Simulación del Coprocesador antes de ejecutar las instrucciones de tipo *load* para cargar los datos al IRF. b) En el recuadro se observa a detalle el contenido inicial de la Memoria de Datos.

Para las simulaciones posteriores se ha inicializado la Memoria de Instrucciones con un programa para ser procesado en el *pipeline* del Coprocesador. Las primeras diez instrucciones corresponden a la carga de información desde la Memoria de Datos hacia el IRF (Tabla 2).

En total se ejecutan treinta y dos instrucciones de tipo *load* con la finalidad de almacenar un nuevo dato en cada espacio disponible del IRF. La simulación resultante de la ejecución de esta secuencia de instrucciones tipo *load* se observa en la Fig. 6. Como se aprecia, el contenido del IRF ha dejado de ser cero, y en consecuencia, se tienen los datos necesarios para ejecutar instrucciones aritmético-lógicas en el Coprocesador.

Table 2. Primeras instrucciones almacenadas en la Memoria de Instrucciones

Registro	Inst.	rs	rt	rd	HEX
0	LD.df	$0	0x0	$0	4B000007
1	LD.df	$0	0x1	$0	4B010047
2	LD.df	$0	0x2	$0	4B020087
3	LD.df	$0	0x3	$0	4B0300C7
4	LD.df	$0	0x4	$0	4B040107
5	LD.df	$0	0x5	$0	4B050147
6	LD.df	$0	0x6	$0	4B060187
7	LD.df	$0	0x7	$0	4B0701C7
8	LD.df	$0	0x8	$0	4B080207
9	LD.df	$0	0x9	$0	4B090247



a)

```
sim:/MIPS_vlg_tst/i1/mem_reg_64b/GralReg @ 659 ps
 0 : 0000000000000000 000000000000000f 000000000000001e 0000000000000028
 4 : 000000000000003c 000000000000004b 000000000000005a 67896a4fa3a57596
 8 : a09762a9b48ba387 bab47355bc9bb090 9246665b37445c66 978c454b8f887999
12 : cbba583eb150959c b2ae712565ab94c9 bd884abdb792bc5a bd884abdb792bc5a
16 : b65d8c53432a2e83 946c47377e925957 82a7a8abc05b858e a997a1aa6e9eaca8
20 : a461b9b099b2bc80 8a449e99645ea033 3d27ba8e694e90a6 935e6ca159ba5755
24 : 42a59abbaaa45472 7a40a7a399b8b861 a862af6c95c2b1ba 885d9b4b63748b73
28 : 580f6dadb1668ab7 b25c6eca879a8288 bcabaf89a2985955 0001000101010001
```

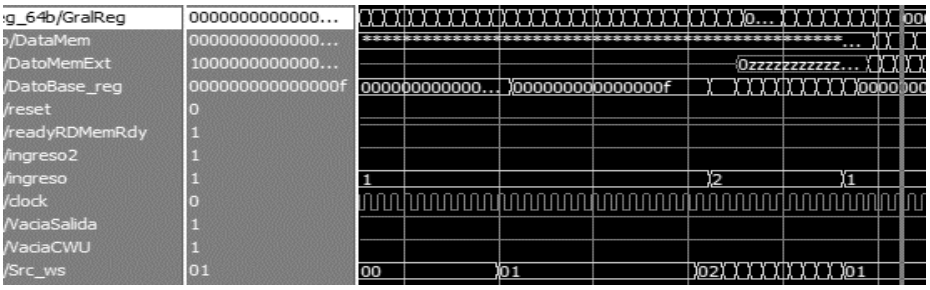
b)

Fig. 6. a) Simulación del Coprocesador después de ejecutar las instrucciones de tipo *load* para la carga de datos al IRF. b) En el recuadro se muestra a detalle el contenido del IRF después de la ejecución de las instrucciones de carga de datos.

Posteriormente, se ejecuta una serie de instrucciones de suma (addv) y resta (subv) para procesar los datos cargados anteriormente al IRF (Tabla 3). Con la ejecución de estas instrucciones se lleva a cabo una modificación en el contenido del IRF, ya que cada instrucción contiene un registro destino (*rt*) en el que será escrito el resultado.

Table 3. Instrucciones de suma y resta para el procesado de los datos cargados en el IRF

Inst.	Op.	rs	rt	rd	HEX
32	ADDV.df	$2	$1	$21	4801154E
33	ADDV.df	$3	$2	$22	4801154E
34	ADDV.df	$4	$3	$23	4801154E
35	ADDV.df	$5	$4	$24	4801154E
36	ADDV.df	$6	$5	$25	4801154E
37	SUBVdf	$7	$6	$26	4801154E
38	SUBVdf	$8	$7	$27	4801154E
39	SUBVdf	$9	$8	$28	4801154E
40	SUBVdf	$10	$9	$29	4801154E
41	SUBVdf	$11	$10	$30	4801154E



a)

```
sim:/MIPS_vlg_tst/i1/mem_reg_64b/GralReg @ 835 ps
 0 : 0000000000000000 000000000000000f 000000000000001e 0000000000000028
 4 : 000000000000003c 000000000000004b 000000000000005a 67896a4fa3a57596
 8 : a09762a9b48ba387 bab47355bc9bb090 9246665b37445c66 978c454b8f887999
12 : cbba583eb150959c b2ae712565ab94c9 bd884abdb792bc5a bd884abdb792bc5a
16 : b65d8c53432a2e83 946c47377e925957 82a7a8abc05b858e a997a1aa6e9eaca8
20 : a461b9b099b2bc80 000000000000002d 0000000000000046 0000000000000064
24 : 0000000000000087 00000000000000a5 997796b15d5b8bc4 c6f207a6ef1ad20f
28 : e5e2ef54f7eff2f7 286e0cfa8557542a faba210fa7bbe2cd 0001000101010001
```

b)

Fig. 7. a) Simulación del Coprocesador después de ejecutar las operaciones de suma y resta almacenadas en la Memoria de Instrucciones. b) En el recuadro se muestra el contenido del IRF con los resultados de cada operación ejecutada.

Una vez realizadas las operaciones requeridas, se realiza el respaldo de los datos obtenidos escribiéndolos en la Memoria de Datos mediante un conjunto de instrucciones de tipo *store*, con lo que se llevará la información del IRF a la Memoria de Datos. Este conjunto de instrucciones se muestra en la Tabla 4.

Table 4. Instrucciones de tipo *store* para realizar la actualización de la Memoria de Datos.

Inst.	Op.	rs	rt	rd	HEX
42	ST.df	$1	0x6	$21	4801154E
43	ST.df	$1	0x7	$22	4801154E
44	ST.df	$1	0x8	$23	4801154E
45	ST.df	$1	0x9	$24	4801154E
46	ST.df	$1	0xa	$25	4801154E
47	ST.df	$1	0xb	$26	4801154E
48	ST.df	$1	0xc	$27	4801154E
49	ST.df	$1	0xd	$28	4801154E
50	ST.df	$1	0xe	$29	4801154E
51	ST.df	$1	0xf	$30	4801154E

Los resultados de la simulación se aprecian en la Fig. 8.

a)

```
sim:/MIPS_vlg_tst/i1/Mem64b/DataMem @ 939 ps
    0 : 0000000000000000 000000000000000f 000000000000001e 0000000000000028
    4 : 000000000000003c 000000000000004b 000000000000005a 67896a4fa3a57596
    8 : a09762a9b48ba387 bab47355bc9bb090 9246665b37445c66 978c454b8f887999
   12 : cbba583eb150959c b2ae712565ab94c9 bd884abdb792bc5a bd884abdb792bc5a
   16 : b65d8c53432a2e83 946c47377e925957 82a7a8abc05b858e a997a1aa6e9eaca8
   20 : a461b9b099b2bc80 000000000000002d 0000000000000046 0000000000000064
   24 : 0000000000000087 00000000000000a5 997796b15d5b8bc4 c6f207a6ef1ad20f
   28 : e5e2ef54f7eff2f7 286e0cfa8557542a faba210fa7bbe2cd 0001000101010001
   32 : 0000000000000000 0000000000000000 0000000000000000 0000000000000000
   36 : 0000000000000000 0000000000000000 0000000000000000 0000000000000000
```

c)

Fig. 8. a) Simulación del Coprocesador después de ejecutar las instrucciones de tipo *store* para el almacenamiento de datos desde el IRF a la Memoria de Datos. b) En el recuadro se aprecia contenido de la Memoria de Datos después del respaldo de los datos resultantes de las operaciones de suma y resta.

4 Conclusiones

El diseño de una LSQ basado en el envejecimiento de instrucciones permite realizar el ordenamiento de las instrucciones de acceso a memoria con la finalidad de mantener la coherencia durante la ejecución de las instrucciones. En el desarrollo de este trabajo se describe a detalle el diseño de la cola, la cual requirió de estructuras de tipo FIFO y diversa lógica combinacional, en el diseño se contempla el intercambio de datos con otros módulos incluidos en el *pipeline* de un procesador súper-escalar. Así mismo, se presenta la implementación de la LSQ integrada al *pipeline* de un coprocesador vectorial, con la finalidad de verificar el correcto funcionamiento del módulo.

5 Referencias

1. Sethumadhavan, S., Desikan, R., Burger, D., Moore, C.R. y Keckler, S.W.: *Scalable Hardware Memory Disambiguation for High ILP Processors.* En: Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture. Washington, DC, USA. (2003)
2. Il-Park, Chong-Lian Ooi y Vijaykumar, T.N.: *Reducing Design Complexity of the Load/Store Queue.* En: Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture. Washington, DC, USA. (2003)
3. Baugh, L. y Zilles, C.: *Decomposing the load-store queue by function for power reduction and scalability*. IBM J. Res. Dev. 50, 2/3, pp. 287-297. (2006)
4. MIPS Technologies: MIPS64 Architecture for Programmers Volume II: The MIPS64 Instructions Set. Mountain View, CA, USA. (2005)
5. Reporte Técnico. Diseño de un procesador superescalar con ejecución fuera de orden usando lenguajes de descripción de hardware (VHDL). IPN-CIC Laboratorio de Micro-tecnología y Sistemas Embebidos. (2008)
6. Pacheco G. E. "Diseño de Procesadores Morfológicos tipo SIMD". Tesis de Maestría. México. CIC-IPN. (2014)
7. Ramírez-Salinas, M.A.: *Arquitectura del Sistema de Memoria*: http://www.microse.cic.ipn.mx
8. Dreslinski, R.: *Memory Speculation*: http://www.eecs.umich.edu/eecs/courses/eecs470/

Design of Transistor Level Circuit for Cosine Wave Mapping Function: Applied to Dynamics Models

J. Reyes-Rosales, V. González-Díaz, and J.F. Guerrero-Castellanos

Facultad de Ciencias de la Electrónica-BUAP. Av. San Claudio y 18 Sur, Col. San Manuel, C.P. 72570, Puebla, Pue. México
{jreyes, vrgdiaz,fguerrero}@ece.buap.mx

Abstract. This brief focuses on the design of analog circuits generating a trigonometric mapping function, such as the cosine function. The designed mapping circuit is to be used as a core component for a circuit emulating the non linear behavior such PVTOL (Planar Vertical Take-Off and Landing). With the analog solution for trigonometric synthesis functions the designer expects a continuous time response with no quantization error. In order to test the proposed approach, the vertical dynamics of a PVTOL system is emulated.

Keywords: Aircraft, Dynamic Model, Nonlinear Systems, Electronic Applications, Circuits Design, Nonlinear Equations, Continuous time systems.

1 Introduction

In the late 1970s and early 1980s "analog" was considered obsolete, anyone seriously contemplating the analog design was out of the modern age of electronic circuit design. This view changed abruptly with the ASIC revolution of the mid-1980s. Indeed, much analog did migrate into some digital form [1] and [2] .

Even today more than ever are becoming more functions and problems working with microcontrollers, microprocessors and useful devices for processing digital signals.

However the higher the amount of digital systems which are created for the data acquisition and process control interface circuits also more internally using analog circuits to operate at a continuous rate as in the case of high speed dividers are required, emulators for real-time control, analog processors focused on the area of the image, signal generators, etc... [3] , [4], [5], [6], [7]. To design embedded circuits, designers must understand the basic principles of analog design. This is to achieve a good performance at a reasonable cost [8].

This paper mainly focuses on the generation of cosine trigonometric function, noticing the problematic that has in certain analog solution analyzed mostly in the control area with non-linear models, such as VTOL (Vertical Take-off and Landing) and PVTOL (Planar Vertical Take-Off and Landing) nonlinear systems

Advances in Computer Science
Research in computing Science 83, 2014 pp. 49-59

whose mathematical models involve trigonometric functions [9], [10], [11] and [12] which are currently mostly simulated digitally. We will focus mainly on the PVTOL model presented in [9].

PVTOL Dynamic Model The following model is proposed for the simplified description of the dynamics of a PVTOL system.

$$\ddot{x} = -u_1 sin(\theta) + \epsilon u_2 cos(\theta) \tag{1}$$

$$\ddot{y} = u_1 cos(\theta) + \epsilon u_2 sin(\theta) - 1 \tag{2}$$

$$\ddot{\theta} = u_2 \tag{3}$$

Where "-1" is the gravitational acceleration and the variable ϵ is the (small) coefficient giving the coupling between the rolling moment and the lateral acceleration of the aircraft. Note that $\epsilon > 0$ means that applying a (positive) moment to roll left produces an acceleration to the right (positive x).

This simplified model is useful because particularized PVTOL system dynamics to a system of two controls that are taken as inputs which are u_1 and u_2 represent the thrust (upward direction) and angular acceleration (rolling moment or torque). As a result three outputs are considered: θ, x and y.

To model this simplified mathematical description on a digital system it is necessary to map the previously mentioned parameters θ, x, y, u_1 and u_2 into the digital domain. This task will add quantization noise to the system provided using a high sampling rate. Also, consider the phenomenon of latency to emulate digitally mapping system. On the other hand, a continuous-time model is more related to the mathematical description of the aforementioned model.

1.1 Behavioral Analysis of Non-linear Model

The nonlinear behavioral model described above is presented in blocks as shown in the Fig. 1, simulation results will be presented later, in order to show that the approximation of the cosine function proposed in this paper provides excellent results in almost any non-linear model in which a cosine function is required.

In this paper we present a continuous time representation related to analog circuits which provide a real time and no quantized mapping circuits useful to construct an analog PVTOL emulator. With the analog solution of trigonometric functions expected of this type of analog emulators there drawback of quantization noise is completely avoided. In addition a continuous time response is achieved at minimum expenses. Therefore, the main contribution of this paper is to offer an alternative to the solution of dynamic systems using the analog processing with a nonlinear circuit to transistor level.

2 Methods to Synthesize an Analog Cosine Mapping Function.

Realization of the cosine function is not supposed to be a simple task, because it was looking good accuracy and the mathematical calculation of the function is

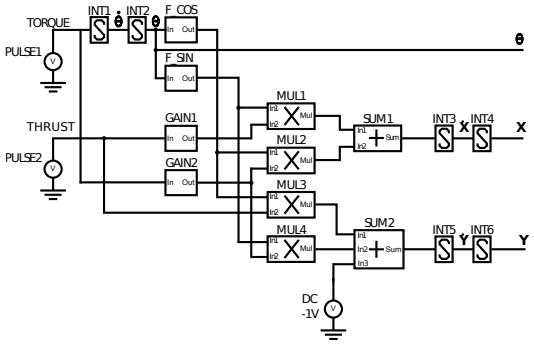


Fig. 1. Behavioral model blocks of PVTOL in Verilog-A

the least complex possible, this is as the main goal for analog circuit design is to obtain good accuracy with minimal resources . According to the literature studied were found at least 3 ways to do the cosine, one that is derived from series, the second is a proposed approximation very algebraically accurately analyzing the cosine function and the last method is a proposal that was implemented for the development of this work, appointed by approximation a nonlinear mapping.

In summary mode explained below each of these methods:

– Taylor:
The Taylor series is named in honor of the British mathematician Brook Taylor (1685-1731), is a representation of a function as an infinite sum of terms.
The approximation given an argument x is expressed as follows:

$$cos(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{2n!} ... \forall x \tag{4}$$

As the terms are increased it obtains a better approximation of the given function, cosine approximation with 4 terms can be seen in the Fig. 2. The measured error of the output waveform with respect to an ideal cosine function was $\pm 2.20\%$.

– Seevinck:
It is thus named in honor of the person who made this approximation implemented directly into electrical circuits using the translinear principle. Analyzing algebraically the cosine function, it obtain the equation 5.

$$cos(\pi x) = \frac{(1 - 4x^2)(2 - x^2)}{2 + x^2} \qquad -1 < x < 1 \tag{5}$$

The implementation of the cosine function using this approximation to transistor level is relatively simple, in addition, provides a very low $\pm 0.82\%$ error

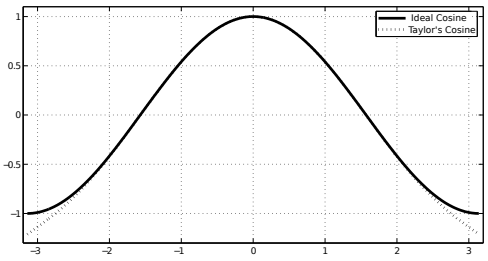


Fig. 2. Comparison of the approximation of cosine Taylor series and ideal cosine function

and is not restricted to rational functions. The author has implemented this approximation on a circuit designed for synthesizing cosine in [6].
An approximation of this equation is very good, as can be seen in the Fig. 3, that basically can not see any difference.

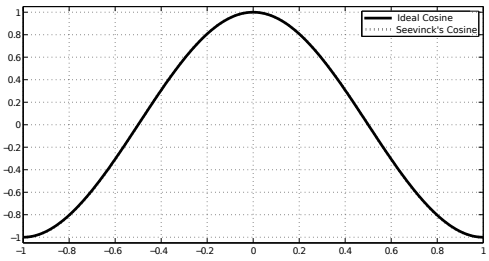


Fig. 3. Comparison of the approximation of cosine Seevinck and ideal cosine function

– The proposed non-linear mapping:
At difference of the techniques above mentioned, the technique proposed in the present work is based on the basic behavior of a MOSFET transistor and its response to certain inputs.
Thus, the implementation of this circuit is relatively simple and fully operational in voltage mode whit folding circuit. A detailed discussion is done in the next section.

The transistor level realization in analog form approximations by Taylor and Seevinck provide good results as shown in the previous figures, by Taylor approximation analog form in a circuit is a relatively difficult method because for a good approximation must be taken at least 4 terms and this mean that the design must be able to reach a sixth power, this requirement is difficult to achieve.

The approach Seevinck is not practical due to the resources required for its implementation, unlike the proposed method which requires few MOSFET transistors.

The goal of this paper is to find a new approach for the cosine function which is easy to implement to transistor level analyzing the behavior with the transfer function obtained at the output of a folding circuit without trying to implement some approximation formula directly, this novel design proposed called "Non-linear Mapping" will be explained more in detail in the next section ensuring relativity be a simple method for its implementation.

3 Implementation of the Non-linear Mapping Function to the Cosine.

3.1 Main Characteristics of the Cosine Function

To synthesize the cosine wave function it is important consider the periodicity, domain, the function is continuous throughout its domain, it is increasing function in the range [π, 2π] and decreasing in [0,π], that is even and the graph is symmetric with respect to the ordinate.

Among these features, it should be mentioned that the behavior in time to a period of -π to π of the cosine function is nonlinear, that is why we are working on the implementation as a nonlinear mapping for the approximation of the function.

At present the most circuits are performed in voltage mode, we can not forget this requirement, so another specification of our emulator circuit will be that works in voltage mode.

This is a advantage they face circuits in current mode, which usually require a voltage-current-voltage conversion so that, in terms of compatibility with the outside world, the technique of design in voltage mode is still present and quite important method [13].

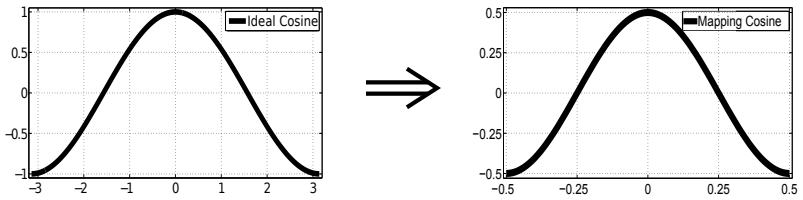
3.2 Circuit Implementation

The function of the mapping is to bring the values of the ideal cosine values as voltage at which the emulator will work, given that the technology implemented in this work is CMOS to 90nm, then the voltage values should range maximum of 1 volt on the Y axis or in amplitude, and instead of having a period of -π to π looking for a period of -0.5 to 0.5 as seen in the Fig. 4.

To perform this mapping, a technique called folding circuit will be used. This technique is based in the analysis of transfer function on that circuit.

A folding circuit implemented for the realization of a Gaussian is in [7]. the mistmach control circuit is shown in Fig. 5.

The topology with mismatch control [7] shown in the Fig. 5 will be used as part of the folding circuit proposed for this work, but in this case the number of transistors increases as the number of mismatch voltages in order to achieve



(a) Ideal cosine function (b) Mapping cosine function

Fig. 4. Objective mapping block cosine.

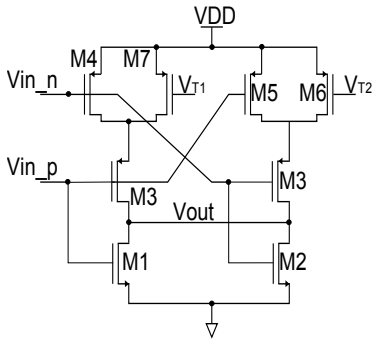


Fig. 5. Folding circuit with mismatch control.

a better fit of the signal and an optimal approximation of the desired function. Generalizing the cosine block that it will perform, this is divided internally into three sub-blocks, two main sub-block gain, and the folding circuit is performed can be seen in Fig. 6.

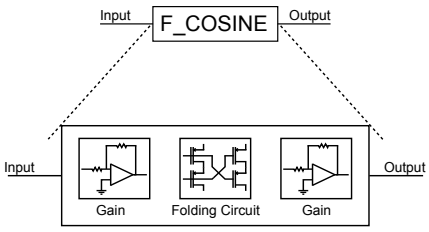


Fig. 6. General block cosine function.

In Fig. 7, we can see the folding circuit designed using standard 90 nm CMOS technology, having a pair of inputs fully differential, this means that when an entry is made low, the other input is set at a high level, both paths are switched off and the output voltage is zero. A true differential given level, the circuit begins

to conduct and output voltage starts to increase when the differential voltage is zero, the output reaches its maximum peak voltage, include the output (V_{out}) ends with a gain amplifier, this will help condition the signal amplitude required.

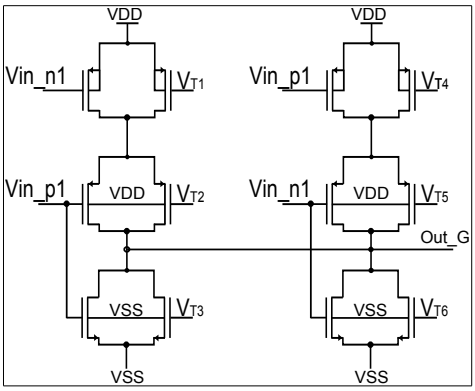
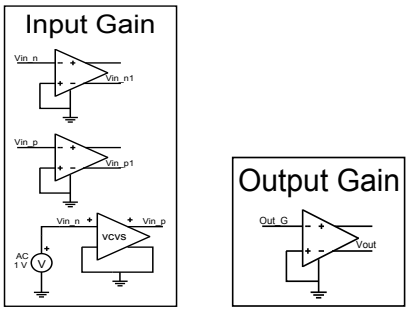


Fig. 7. Folding circuit for non-linear mapping.

To the input circuit (Vin_n1 y Vin_p1) also a pair of amplifiers are placed at certain gain as shown in Fig. 8, in order to adjust the output signal in the predetermined period.



(a) Gain to the input

(b) Gain to the output

Fig. 8. Gain block for fit end cosine function.

The final block of the cosine function by non-linear mapping approach is expressed in Fig. 9, where we can see the cosine mapping over a period of -0.5 to 0.5, as a dotted line and the approach to this mapping is emphasized by a solid line.

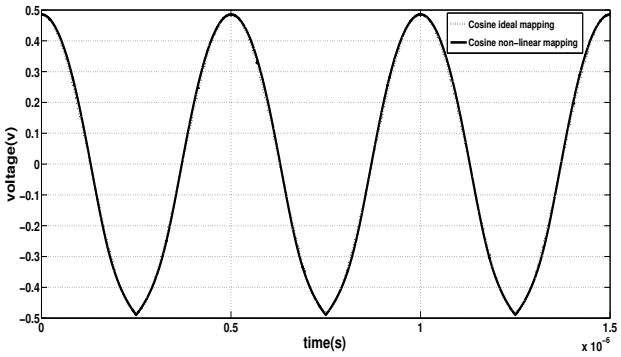


Fig. 9. Answer folding circuit.

This result (Fig. 9) demonstrates an alternative solution different to the digital process. It is important to mention what according the application will be the solution. Temperature variations can affect the circuit output for the moment this problem can be avoided with a temperature control, even so, the end is the real-time processing.

3.3 Electrical Parameters

Based in the file obtained for DC operating point simulation (Eldo-Spice), in the Tab. 1 shows current, voltage and consumption power values for the power supplies required for the MOSFETs of the folding circuit.

Table 1. General electrical parameters.

Source	**Current**	**Voltage**	**Power**
VSS	-3.5880mA	-500mV	-1.7940mW
VDD	-3.5880mA	500mV	-1.7940mW
V_{T1}	50nA	5.8mV	46.2066pW
V_{T2}	2.032nA	2mV	4.0640pW
V_{T3}	-856.9076pA	1.6mV	-1.3711pW
V_{T4}	6.1381nA	5.8mV	35.6010pW
V_{T5}	468.1643pA	2mV	936.3285pW
V_{T6}	-856.9076pA	1.6mV	-1.3711pW

4 Simulation Results.

In order to validate the results of the new approach of the cosine function mapped to transistor level, we will discuss only the vertical position of the system of

equations and the roll angle of the aircraft with the horizon, this means that in the system disappears the coupling between the rolling moment and the lateral acceleration of the aircraft ($\epsilon = 0$), therefore we obtain the following expressions:

$$\ddot{y} = u_1 cos(\theta) - 1 \tag{6}$$

$$\ddot{\theta} = u_2 \tag{7}$$

applying the system a control LQR, in Fig. 10 shows two responses in general . The solid lines (V(THETA)and V(Y)) are the responses on the system applying a cosine function mapping ideal, on the other hand the PVTOL system solutions represented by dotted lines (V(THETA_T)and V(Y_T)) are the responses obtained with the cosine block performed at transistor level and implemented within the model behavioral.

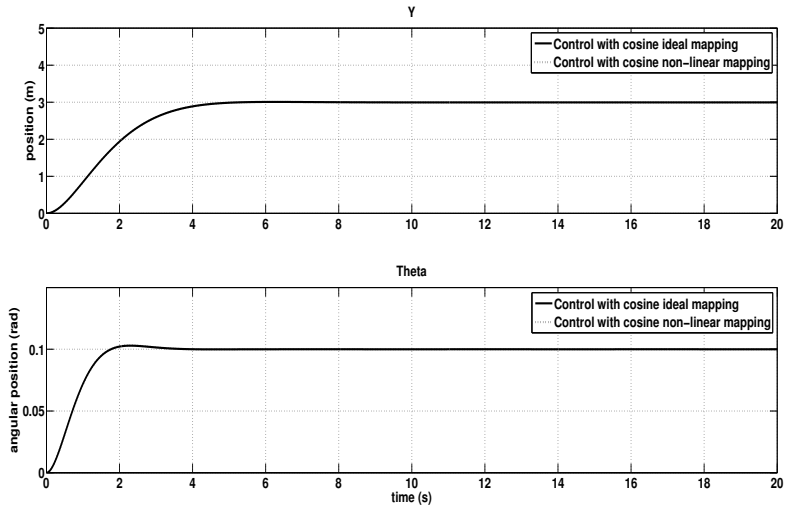


Fig. 10. Comparison of the original function mapping and transistor level function mapping

It is noteworthy that our new approximation will work correctly between the ranges of design to transistor level (-0.5 V to 0.5 V).

5 Conclusion

It is to be noted that this circuit produces very good results and a transistor level implementation has shown that it is relatively simple. All design was implemented using standard 90 nm CMOS technology and the measured error of the output waveform with respect to an ideal cosine function mapping was $\pm 1.14\%$.

As future work could implement a set of algebraic or trigonometric functions to transistor level with the technical folding proposed herein, emulating on a silicon chip presented any model of any system, better yet, making the the control on a chip, so that when the control is proved this can be done in continuous time but the model is very complex.

References

1. Rutembar,R.A.: Analog design automation: Where are we¿ where are we going¿. In: Custom Integrated Circuits Conference, Proceedings of the IEEE , pp. 13.1.1–13.1.7., (2009)
2. Lewyn, L.L., Ytterddal, T., Wulff, C., and Martin, K.: Analog Circuit Design in Nanoscale CMOS Technologies. In: Proceedings of the IEEE , pp. 1687–1714, San Diego, CA (1993)
3. Cruz, N.H: The Grid: Transformada Discreta del Coseno Realización Analgica. Masters thesis, Instituto Nacional de Astrofísica Óptica y Electrónica, Puebla, Mexico(2003).
4. Weihsing, L. and Shen-Iuan, L.: Low- voltage CMOS voltage-mode divider and its application. In: IEICE Trans. Fundamentals, E87-A(2), pp. 330–334, (2004)
5. Dualibe, C., Verleysen, M., and Jespers, P. : Two- quadrants CMOS analog divider, Electronic Letters, 34(12), 1164–1165 (1998)
6. Seevinck, E. : Simple, wide-range approximations to trigonometric and inverse trigonometric functions useful in real-time signal processing, IEEE Conference Electronic Circuits and Systems, 41–45 (1981)
7. Surano, A., Bonizzoni, E., and Maloberti, F. : On- Chip Sine Wave Frequency Multiplier for 40-GHz Signal Generator. In: Conference Publication, Research in Microelectronics and Electronics, pp. 284–287, University of Pavia, Pavia- ITALY(2009)
8. Coughlin, R.F. and Driscoll, F.F.: Amplificadores operacionales y circuitos integrados lineales. Pearson Prentice Hall., México (1999)
9. Hauser, J., Sastry, S., and Meyer, G.: Nonlinear control design for slightly nonminimum phase systems: Application to v/stol aircraft., Automatica, 28(4), 665–679 (1992)
10. Zavala, A., Fantoni, I., and Lozano, R.: Global stabilization of a PVTOL aircraft with bounded thrust. In: CConference on Decision and Control, pp. 4462–4467, Adelaide, Australia (2005)
11. Wood, R., Cazzolato, B., and Halim, D.: A Global Non-linear Control Design for a PVTOL Vehicle with Aerodynamics. In: 44th IEEE Conference on Decision and Control, pp. 7478–7483, Las Vegas, Nevada USA (2002)
12. Olfati-Saber, R.: Global Configuration Stabilization for the VTOL Aircraft With Strong Input Coupling, IEEE Transactions on Automatic Control, 47(11), 1949–1952 (November 2002)
13. Juan, M.B.: Diseño VLSI de un sistema difuso en modo corriente e interfaz de entrada en modo voltaje. Masters thesis, INSTITUTO POLITECNICO NACIONAL, México, México. (2007)

Natural Language Processing

A Set of Phonetic and Phonological Rules for Mexican Spanish Revisited, Updated, Enhanced and Implemented

Carlos-Daniel Hernández-Mena[a], Nancy-Norely Martínez-Gómez[b], and José-Abel Herrera-Camacho[a]

[a]Departamento de Procesamiento Digital de Señales
Universidad Nacional Autónoma de México (UNAM)
ca_hernandez@uxmcc2.iimas.unam.mx
abelherrerac1@gmail.com

[b]Facultad de Filosofía y Letras, UNAM
nancy_luthien@hotmail.com

Abstract. This paper revisits the phonetic and phonological rules of grapheme-to-phoneme conversion for Mexican Spanish released in 2004 for a particular computational phonetic alphabet called "Mexbet", which was specially designed for the Spanish spoken in Central Mexico. Mexbet has proved its value over the years but it is still not well known. Implementation of these rules into real programming code will be explained, and the *fonetica2* library that is an open-source tool which allow users to develop customized phonetic transcription algorithms, will be presented.

Keywords: grapheme-to-phoneme convertion, mexican spanish, mexbet, phonetic alphabet.

1 Introduction

In the field of phonetics, it is necessary to register the sounds of human languages, which is the reason for the phonetic alphabet. A phonetic alphabet is a set of symbols that represents the sounds of speech. The process to convert a word written in a regular alphabet, to a particular phonetic alphabet is called "transcription". Some examples of phonetic alphabets are the RFE alphabet that is exclusive for the Spanish language [1] or the IPA alphabet that was created by the International Phonetic Asociation [2]. It contains the speech sounds of all the languages over the world.

The use of these alphabets have disadvantages in most fields of computer science because of something as simple as the character codification. The IPA, the RFE and other "classic" phonetic alphabets utilize symbols that are impossible to deal with in programming codes, usually written in plain text with an ASCII or UTF-8 set of characters. The solution to these problems is the creation of computational phonetic alphabets that are nothing but the translation

Advances in Computer Science
Research in computing Science 83, 2014 pp. 61-71

of classic alphabets into a set of symbols easy to incorporate in programming codes. Normally, these kind of alphabets are written with ASCII symbols and that is why we can call them by the full name of "ASCII Computational Phonetic Alphabets" or ACPA for short. Some examples of these ACPA could be the SAMPA alphabet that was designed for many languages including Spanish [3], the WORLDBET alphabet that is an ASCII adaptation of the IPA alphabet [4] or the OGIBET alphabet that was created by the Oregon Institute of Science and Technology and then adapted for the Mexican Spanish by the Tlatoa Group[1] [5].

Mexbet is another example of an ACPA that we will discuss in the present document. Mexbet is an ACPA specially designed for the Mexican Spanish, mostly based on WORLDBET but also incorporating OGIBET. It first appaeared in [6] as a part of a master thesis. After that, Mexbet went through transformations and updates over the years and has been published more than once [7], [8]. The current version of Mexbet in which we are interested, appeared in 2004 on another master thesis written by Javier Cuétara [9]. In this thesis, Cuétara introduced a number of changes and corrections to the current version of that time. All the data that he utilized to carry out his analysis came from the DIMEx100 corpus [10] which is a phonetic and speech corpus for Mexican Spanish created at IIMAS-UNAM[2].

Some examples where Mexbet was successfully used are the VOXMEX corpus [11] which utilized an early version of it, and the DIMEx100 corpus that used a newer and extended one, nevertheless these papers do not expose the transcripcion rules of Mexbet with much detail, actually, the most complete information is in the Cuétara thesis but a basic knowledge in phonetics is necessary to understand it well, and not to mention that the thesis is written in Spanish, which means that Mexbet rules will be less understandable for non-Spanish speaking researchers and engineers.

As we can see, there is a real need to have unified standards and free access language resources for a healthy development of speech technologies in Mexico and that is why the CIEMPIESS-UNAM Project[3] was created. The CIEMPIESS-UNAM Project aims to develop and share free and open-source tools for speech processing in the Spanish language. We have recently released our 17 hours database for speech recognition called CIEMPIESS [12] that uses Mexbet and it is completely free. We think that this may define new conditions for the growth and development of the field of speech processing in Mexico.

In the next sections we will sum up the full specification of Mexbet (as anyone has done before) that has demostrated its usefullness in the field of automatic speech recognition.

[1] Tlatoa is (or was) a Mexican speech technologies research group from the "Universidad de las Américas de Puebla" founded in 1997.
[2] IMASS-UNAM website at `http://www.iimas.unam.mx/`
[3] See `http://odin.fi-b.unam.mx/CIEMPIESS-UNAM/`

2 Phonological Rules of Mexbet

Phonology is the branch of linguistics which studies the speech sounds (phonemes) as an abstract system [13]. This means that in phonology the phonemes are represented only in a prototypic way. The branch of linguistics which tries to represent the speech sounds in the most accurate way as possible is phonetics [14].

In the 2004 version of Mexbet a total of 17 consonants and 5 vocals for the Mexican Spanish are found. This phonological level of Mexbet is called "T22" although it was decided to include two more phonemes that come from the Náhuatl language (also known as "Aztec" [15]) in this revision of Mexbet, because they are common in the Mexican Spanish. These phonemes are /S/ like in the word "xoloescuincle" (in English this phoneme could sound as "sh" like in "shannon") and /tl/ that is always at word endings like in "popocatépetl" or "citlaltépetl".

Table 1 shows the Mexbet symbols for the Mexican Spanish phonemes which provide information about the manner and the point of articulation of each one. This information is useful in speech recognition because it helps to group similar acoustic models together as the HTK toolkit requires [16].

Table 1. Phonolical Level of Mexbet also known as "T22"

	Points of Articulation						
Manners of Articulation	**Consonants**	**Labial**	**Labiodental**	**Dental**	**Alveolar**	**Palatal**	**Velar**
	Voiceless Stop	p		t			k
	Voiced Stop	b		d			g
	Voiceless Affricate					tS	
	Voiceless Fricative		f		s	S	x
	Voiced Fricative					Z	
	Nasal	m			n	n~	
	Rhotic				r(/ r		
	Lateral				l	tl	
	Vowels				**Front**	**Central**	**Back**
	Close				i		u
	Mid				e		o
	Open					a	

In practice, a phonological transcription is nothing but a "transformation" of a word written in a conventional way (orthographic) into another that only includes the symbols of Table 1. Some of these transformations depends on the fact that many graphemes do not represent the speech sounds correctly, but they have to be written because of tradition or historical reasons, like the grapheme "h" that in spanish is usually mute, or the grapheme "z" that represents the phoneme /s/.

Table 2 shows the specific transformations of certain graphemes that were adopted in the *fonetica2* library that we will present in the following sections. In the case of the *fonetica2* library, it is assumed that every word is in Spanish, that is why transformations provided in Table 2 are sufficient and work pretty well. If you decide to do transcriptions of words in different languages like "cappuccino" or "ballet" it is probably that one will need to consider more transformations

or the use of stop word lists. To see how to deal with grapheme transformations for different languages see [17].

Table 2. Transformations adopted to do phonological transcriptions in Mexbet

No ASCII Symbol : Example	Transformation : Example	Ortographic Irregularity : Example	Phoneme Equivalence : Example	Ortographic Irregularity : Example	Phoneme Equivalence : Example
"á" : "cuál"	"cuAl"	"cc" : "accionar"	/ks/ : "aksionar"	"gui" : "guitarra"	/g/ : "gitaRa"
"é" : "café"	"cafE"	"ll" : "llamar"	/Z/ : "Zamar"	"que" : "queso"	/k/ : "keso"
"í" : "maría"	"marIa"	"rr" : "carro"	"R" : "caRo"	"qui" : "quizá"	/k/ : "kisA"
"ó" : "noción"	"nociOn"	"ps" : "psicología"	/s/ : "sicologIa"	"ce" : "cemento"	/s/ : "semento"
"ú" : "algún"	"algUn"	"ge" : "gelatina"	/x/ : "xelatina"	"ci" : "cimiento"	/s/ : "simiento"
"ü" : "güero"	"gWero"	"gi" : "gitano"	/x/ : "xitano"	"y" (end of word) : "buey"	/i/ : "buei"
"ñ" : "niño"	"niNo"	"gue" : "guerra"	/g/ : "geRa"	"h" (no sound) : "hola"	"ola"

Substitutions in Table 2 are the first step to make phonological transcriptions in Mexbet. For example, non ASCII symbols like "ñ" or "ü" are substituted for ASCII symbols like "N" and "W" (respectively) to prevent character codification issues. Also the graphic accents on vowels ("á", "é", "í", etc.) are substituted for the same vowel but in upper case. In general, at this point in the transcription process it is considered that vowels in upper case are tonic vowels even if they do not come from a vowel with a graphic accent. Other transformations of certain chains of graphemes (morphemes) that are shown in Table 2 are self-explanatory and the reason of choosing them are well discussed in [18].

There is a special problem with the grapheme "x" that in Mexican Spanish has 4 different sounds and there is no transcription rule for all of of them. Table 3 shows these four cases of the grapheme "x" and the substitution that has to be made in order to transcribe every case correctly.

Table 3. The four different sounds of the grapheme "x" in Mexican Spanish

Cases of "x"	Phoneme Equivalence : Example
"sexto","oxígeno"	/ks/ : "sexto","oksIgeno"
"xochimilco","xilófono"	/s/ : "sochimilco","silOfono"
"xolos","xicoténcatl"	/S/ : "Solos","SicotEncatl"
"ximena","xavier"	"j" : "jimena","javier"

If you want to make an automatic transcription of words containing the grapheme "x", it is possible one has to consider the use of stop word lists. Notice that in words like "ximena" and "xavier" the substitution of the grapheme "x" is the grapheme "j" and not a phoneme like in the other cases. That is because in Mexbet, the phoneme /x/ is represented by the grapheme "j". One has to be careful not to confuse the grapheme "x" with the phoneme /x/.

After applying transformations in Table 2 and Table 3, one may apply the last step of phonological transcription that is nothing but making the transformations

of the missing graphemes that has not been touched until now. Table 4 shows these last transformations of graphemes into phomemes.

Table 4. Final grapheme to phoneme transformations

Grapheme	Phoneme	Grapheme	Phoneme
"A"	/a_7/	"ch"	/tS/
"E"	/e_7/	"c"	/k/
"I"	/i_7/	"j"	/x/
"O"	/o_7/	"v"	/b/
"U"	/u_7/	"z"	/s/
"N"	/n∼/	"r"	/r(/
"y" not at word ending	/Z/	"R"	/r/
"W"	/u/		

Notice that in Table 4 the Mexbet symbol "_7" indicates the tonic vowel. Graphemes like "p" or "t" are not in Table 4 because their phoneme equivalence are represented with the same symbols (/p/ , /t/ respectively).

The next section shows how to transform a phonological transcription in Mexbet T22 into a phonetic transcription in Mexbet level T50.

3 Phonetic Rules of Mexbet

Phonetics is the branch of linguistics that studies the speech sounds from the point of view of their acoustic realization and it is very interested in the position of the organs that produces the speech sounds (articulation features) [14]. A main difference between phonology and phonetics is that the former deals with ideal sounds called phonemes and the latter investigates how these phonemes vary from person to person. The variants of the prototypic phonemes are also known as allophones.

Before applying the phonetic transcription rules it is necessary to have a good syllabification of the words that one wants to transcribe and for a good syllabification, it is usually needed to know where the tonic vowell is. Discussing the syllabification and the accentuation rules of the Spanish is out of the scope of the present article, but for more information about them, please see [19].

After having made a phonological transcription of a word, one can now transform it into a phonetic form. In the DIMEx100 corpus, it is evident that there is only one phonological level for Mexbet called T22, but there are two different levels of granularity in the phonetic version of Mexbet, the T44 and the T54[4]. Nevertheless, these levels does not contain all the Mexbet allophones that appeared in the 2004 version and they include symbols like "_c" for all the oclusive

[4] You can see all the versions of Mexbet used in the DIMEx100 corpus at `http://turing.iimas.unam.mx/~luis/DIME/CORPUS-DIMEX.html`

phonemes (e.g. /k_c/, /p_c/, /t_c/)[5]. In this revision, we want to introduce a new phonetic level of Mexbet called "T50" that contains all the phonemes and allophones of the version 2004 (except the phoneme /m_n/) plus the two Aztec phonemes /S/ and /tl/ and it does not contain symbols for closure moments.

The following list shows all the phonetic rules of Mexbet. They can allow users to perform accurate transcription algorithms. The Mexbet symbols in / / denote the phonemes and the symbols in [ ] denote the allophones. In this list we can find examples of some words transcribed in the T22 and the T50 level that illustrate well the corresponding rules. Transcriptions in / / are in T22 and transcriptions in [ ] are in T50. Words in " " are the words that we want to transcribe and we (manually) indicate the tonic vowel of each one with a capital letter (eg. "teclAdo", "ambulAncia", etc.).

1. **[a_j]: Followed by palatal consonant; In diphthong ending in /i/ e.g. "ayEr", "aIre" - / a . Z e_7 r(/ , / a_7 i . r(e / - [a_j . Z E_7 r(_\] , [a_j_7 i(. r(E] .**
2. **[a_2]: In diphthong ending in /u/; Followed by /o/; In closed syllable ending in /l/; Followed by /x/ e.g. "Aunque", "paOla", "altUra", "ajEno" - / a_7 u n . k e / , / p a . o_7 . l a / , / a l . t u_7 . r(a / , / a . x e_7 . n o / - [a_2_7 u(N . k_j e] , [p a_2 . o_7 . l a] , [a_2 l_[. t U_7 . r(a] , [a_2 . x e_7 . n o] .**
3. **/ a /: In every other context e.g. "cAsa" - / k a_7 . s a / - [k a_7 . s a] .**
4. / e /: In open syllable; In closed syllable ending in /m, n, s, d/ e.g. "pErro", "pensAr" - / p e_7 . r o / , / p e n . s a_7 r(/ - [p E_7 . r O] , [p e n . s a_7 r(_\] .
5. [E]: In closed syllable ending in a consonant (except /m, n, s, d/); in contact with /r/ or /r(/; In diphthong ending in /i/; Followed by /x/ e.g. "selvAtico", "mercAdo", "peinAr", "lejAno" - / s e l . b a_7 . t i . k o / , / m e r(. k a_7 . d o / , / p e i . n a_7 r(/ , / l e . x a_7 . n o / - [s E l . V a_7 . t i . k o] , [m E r(_\. k a_7 . D o] , [p E i(. n a_7 r(_\] , [l E . x a_7 . n o] .
6. **/ o /: In open syllable e.g. "comEr" - / k o . m e_7 r(/ - [k o . m E_7 r(_\] .**
7. **[O]: In closed syllable ending in any consonant; In contact with /r/ or /r(/; In diphthong ending in /i/; Followed by /x/ e.g. "montAr", "zOrra", "herOico", "mojAr" - / m o n . t a_7 r(/ , / s o_7 . r a / , / e . r(o_7 i . k o / , / m o . x a_7 r(/ - [m O n_[. t a_7 r(_\] , [s O_7 . r a] , [E . r(O_7 i(. k o] , [m O . x a_7 r(_\] .**
8. / i /: In open syllable e.g. "chiflAr" - / tS i . f l a_7 r(/ - [tS i . f l_0 a_7 r(_\] .
9. [I]: In closed syllable; In contact with /r/ or /r(/; Followed by /x/ e.g. "silbAr", "rIco", "quijAda" - / s i l . b a_7 r(/ , / r i_7 . k o / , / k i . x a_7 . d a / - [s I l . V a_7 r(_\] , [r I_7 . k o] , [k_j I . x a_7 . D a] .
10. [j]: At beginning of diphthong e.g. "diArio" - / d i a_7 . r(i o / - [d j a_7 . r(j o] .
11. [i( ]: At end of diphthong e.g. "sEis" - / s e_7 i s / - [s E_7 i(s] .
12. **/ u /: In open syllable e.g. "pulIr" - / p u . l i_7 r(/ - [p u . l I_7 r(_\] .**
13. **[U]: In closed syllable; In contact with /r/ or /r(/; Followed by /x/ e.g. "funcionAr", "puritAno", "sujEto" - / f u n . s i o . n a_7 r(/ , / p u . r(i . t a_7 . n o / , / s u . x e_7 . t o / - [f U n . s j o . n a_7 r(_\] , [p U . r(I . t a_7 . n o] , [s U . x e_7 . t o] .**
14. **[w]: At beginning of diphthong e.g. "cuEnto" - / k u e_7 n . t o / , [k w e_7 n_[. t o] .**
15. **[u( ]: At end of diphthong e.g. "reusAr" - / r e u . s a_7 r(/ - [r E u(. s a_7 r(_\] .**
16. / p /: In every context e.g. "repAso" - / r e . p a_7 . s o / - [r E . p a_7 . s o] .
17. **/ t /: In every context e.g. "tetEra" - / t e . t e_7 . r(a / - [t e . t E_7 . r(a] .**
18. [k_j]: Followed by front vowels e.g. "troquelAr" - / t r(o . k e . l a_7 r(/ - [t r(_0 O . k_j e . l a_7 r(_\] .
19. / k /: In every other context e.g. "cAza" - / k a_7 . s a / - [k a_7 . s a] .
20. **/ b /: At beginning of word; After nasal consonant e.g. "vAca", "sOmbra" - / b a_7 . k a / , / s o_7 m . b r(a / - [b a_7 . k a] , [s O_7 m . b r(a] .**
21. **[V]: In every other context e.g. "cabEza" - / k a . b e_7 . s a / - [k a . V e_7 . s a] .**
22. / d /: At beginning of word; After nasal consonant; After /l/ e.g. "dOna", "cOnde", "cElda" - / d o_7 . n a / , / k o_7 n . d e / , / s e_7 l . d a / - [d o_7 . n a] , [k O_7 n_[. d e] , [s E_7 l_[. d a] .
23. [D]: In every other context e.g. "ademAs" - / a . d e . m a_7 s / - [a . D e . m a_7 s] .

[5] This symbols represents the "closure moment" that is the time when the air is totally obstructed by the articulatory organs of the mouth when pronouncing an occlusive consonant.

24. **/ g /: At beginning of word; After nasal consonant e.g. “gAto”, “angOsto” - / g a_7 . t o / , / a n . g o_7 s . t o / - [g a_7 . t o] , [a N . g O_7 s_[. t o] .**
25. **[G]: In every other context e.g. “pegAr” - / p e . g a_7 r(/ - [p e . G a_7 r(_\] .**
26. / tS /: In every context e.g. “mochIla” - / m o . tS i_7 . l a / - [m o . tS i_7 . l a] .
27. **/ f /: In every context e.g. “refOrma” - / r e . f o_7 r(. m a / - [r E . f O_7 r(_\. m a] .**
28. [z]: Followed by voiced consonants e.g. “asbEsto” - / a s . b e_7 s . t o / - [a z . V e_7 s_[. t o] .
29. [s_[ ]: Followed by voiceless dental consonant e.g. “asterIsco” - / a s . t e . r(i_7 s . k o / - [a s_[. t E . r(I_7 s . k o] .
30. [z_[ ]: Followed by voiced dental consonant e.g. “dEsde” - / d e_7 s . d e / - [d e_7 z_[. D e] .
31. / s /: In every other context e.g. “recEta” - / r e . s e_7 . t a / - [r E . s e_7 . t a] .
32. **/ x /: In every context e.g. “mejOr” - / m e . x o_7 r(/ - [m E . x O_7 r(_\] .**
33. [dZ]: At beginning of word; After nasal consonant; After /l/ e.g. “yUnque”, “inyectAr”, “ulyAna” - / Z u_7 n . k e / , / i n . Z e k . t a_7 r(/ , / u l . Z a_7 . n a / - [dZ U_7 N . k_j e] , [I n_j . dZ E k . t a_7 r(_\] , [U l_j . dZ a_7 . n a] .
34. / Z /: In every other context e.g. “payAso” - / p a . Z a_7 . s o / - [p a_j . Z a_7 . s o] .
35. **/ m /: In every context e.g. “medievAl” - / m e . d i e . b a_7 l / , [m e . D j e . V a_2_7 l] .**
36. **[m_n]: After /n/. Not implemented because it modifies the syllabication e.g. “enmEdio” - “ e n . m é . d i o “ - / e n . m e_7 . d i o / - [e . m_n e_7 . D j o] .**
37. [m]: Followed by labial consonant e.g. “inviErno” - / i n . b i e_7 r(. n o / - [I m . b j E_7 r(_\. n o] .
38. [M]: Followed by labiodental consonant e.g. “infOrme” - / i n . f o_7 r(. m e / - [I M . f O_7 r(_\. m e] .
39. [n_[ ]: Followed by dental consonant e.g. “anteriOr” - / a n . t e . r(i o_7 r(/ - [a n_[. t E . r(j O_7 r(_\] .
40. / n /: At beginning of word or syllable e.g. “tIna” - / t i_7 . n a / - [t i_7 . n a] .
41. [n_j]: Followed by palatal consonant e.g. “inyecciOn” - / i n . Z e k . s i o_7 n / - [I n_j . dZ E k . s j O_7 n] .
42. [N]: Followed by velar consonant e.g. “hangAr” - / a n . g a_7 r(/ - [a N . g a_7 r(_\] .
43. **/ n~ /: In every context e.g. “nIño” - / n i_7 . n~ o / - [n i_7 . n~ o] .**
44. [l_[ ]: Followed by dental consonant e.g. “Alto” - / a_7 l . t o / - [a_2_7 l_[. t o] .
45. [l_j]: Followed by palatal consonant e.g. “salchIcha” - / s a l . tS i_7 . tS a / - [s a_2 l_j . tS i_7 . tS a] .
46. [l_0]: After /p, k, f/ e.g. “plAnta” - / p l a_7 n . t a / - [p l_0 a_7 n_[. t a] .
47. / l /: In every other context e.g. “lOco” - / l o_7 . k o / - [l o_7 . k o] .
48. **[r(_0]: After /p, t, k, f/ e.g. “crEma” - / k r(e_7 . m a / - [k r(_0 E_7 . m a] .**
49. **[r(_\]: At the end of word or syllable e.g. “mermAr” - / m e r(. m a_7 r(/ - [m E r(_\. m a_7 r(_\] .**
50. **/ r(/: In every other context e.g. “arEna” - / a . r(e_7 . n a / - [a . r(E_7 . n a] .**
51. / r /: At beginning of word or syllable; After /s, n, l/ e.g. “honrAdo” - / o n . r a_7 . d o / - [O n . r a_7 . D o] .
52. **/ S /: In every context e.g. “xicotEncatl” - / S i . k o . t e_7 n . k a . tl / - [S i . k o . t e_7 N . k a_j . tl] .**
53. / tl /: At end of word e.g. “popocatEpetl” - / p o . p o . k a . t e_7 . p e . tl / - [p o . p o . k a . t e_7 . p e . tl] .

4 Online Tools and Evaluation

As previously mentioned, the *fonetica2* library is an open-source software tool that contains functions to transcribe Spanish words into a phonetic and a phonological level[6]. The *fonetica2* library is coded in Python and includes a total of six functions that can be easily incorporated to the user code who can see them as black boxes. Each of these functions accept Spanish words in lowercase as arguments. These words can have the tonic vowel marked in uppercase or not (e.g. cAldo , aviOn , comida , etc.).The functions are:

– **vocal_tonica()**: Returns the same incoming word but with its tonic vowel in uppercase (e.g. cAsa, pErro, gAto, etc.).

[6] Download the *fonetica2* library at `http://www.ciempiess.org/downloads`

- **div_sil()**: Returns the syllabification of the incoming word.
- **TT()**: "TT" is the acronym for "Text Transformation". This function produces the text transformations in Table 2 over the incoming word. All of them are perfectly reversible.
- **TT_INV()**: Produces the reverse transformations made by the TT() function.
- **T22()**: Produces a phonological transcription in Mexbet T22 of the incoming word.
- **T50()**: Produces a phonetic transcription in Mexbet T50 of the incoming word.

In the website of the CIEMPIESS-UNAM Project you can also find four online tools to test the functions: vocal_tonica(), div_sil(), T22() and T50(). You only need to write a Spanish word in a textbox, press a button and see the result provided.

4.1 Evaluation of the vocal_tonica() function

For the evaluation of the vocal_tonica() function we used words extracted from the CIEMPIESS corpus. This database counts with 12155 tokens (or words with no repetitions). We took randomly 1539 words that is the 12.66% of the whole CIEMPIESS words. Then we eliminated the foreign words (87) and that is how we obtained a total of 1452 words to analyze. We manually checked if they were correctly accentuated. The result is that **90.35%** (1312 words) were correctly accented against 140 with a wrong position of their tonic vowels. Some of the reasons for these errors are that some words were conjugated verbs and names.

4.2 Evaluation of the T22() function

For the evaluation of the T22() function we counted with two different comparison elements. The first one is the pronouncing dictionary of the DIMEx100 corpus that counts with 11575 entries. The second one is the software called TRANSCRÍBEMEX that is mentioned with that name in the Cuétara thesis. This pronouncing dictionary was made by human transcribers of the DIMEx100 corpus aided by the TRANSCRÍBEMEX.

The TRANSCRÍBEMEX is a software tool with graphic interface coded in Perl that produces phonological transcriptions in Mexbet T22, phonetic transcriptions in Mexbet T54 and shows the syllabification of each word. It can be used to transcribe entire sentences and not just isolated words.

We decide to use only the transcriptions in T22 of the TRANSCRÍBEMEX to compare with our T22() function because the transcriptions in T54 are so different and even incomplete with respect to the transcriptions produced by our T50() function.

A problem with the TRANSCRÍBEMEX is that it produces a set of symbols called "archiphonemes" [20]: [-B], [-D], [-G], [-N], [-R]. An archiphoneme is a phonological symbol that groups several phonemes together. For example, [-D] is equivalent to any of the phonemes /d/ or /t/.

Another problem were the words with the grapheme "x" that, as previously mentioned, can have any of four different pronunciations depending on the sound of the "x" in the current word. The TRANSCRÍBEMEX only utilizes the sound /ks/ for the grapheme "x". For that reason, words with "x" in the analysis were eliminated.

The other element that was eliminated were the alternative pronunciations that in the DIMEx100 corpus appear with a digit in parentheses (e.g. DAÑADOS d a n∼ a d o s ; DAÑADOS(2) d a n∼ a o s). These alternate pronunciations were created by human transcribers based on the recordings that they had to transcribe.

Finally after all of these precautions, we compared the transcriptions of the TRANSCRÍBEMEX with the transcriptions of the T22() function. The result was that both tools are **99.2%** similar which means that our T22() function is reliable.

4.3 Indirect Evaluations

For the evaluation of the TT(), and the div_sil() functions we did not have elements of comparison available as with the T22() function, nevertheless, we did not need them.

As we have demonstrated, the T22() function is reliable with respect to the TRANSCRÍBEMEX that is also reliable. Then we assume that the TT() and the div_sil() functions works well because the T22() needs them to work, and if they were wrong, the transcriptions of the T22() would be wrong too.

We did not have comparison elements for the evaluation of the T50() function either, so we manually had to test hundreds of words in alpha versions of the *fonetica2* library. The result is that all the 74 example words in the phonetic rules listed in the previous section were generated with no errors using our T50() function.

5 Conclutions

We have presented a set of phonological and phonetic rules for Mexican Spanish in English, we have implemented them, we have evaluated them and we have demonstrated that they are reliable and we hope that this contribution improves the access to them for researchers and engineers all over the world.

We also have shown novel and open-source tools created by the CIEMPIESS-UNAM Project that can contribute for the development of speech technologies in Mexico and other countries.

Acknowledgements

We thank UNAM PAPIIT/DGAPA project IT102314, CEP-UNAM and CONACYT for finantial support. We thank Professor Javier Cuétara for his advice on the adaptation of these rules into real programming code, Ivan Vladimir Meza Ruiz for providing us the pronouncing dictionary of the DIMEx100 corpus, Alejandra Chavarría for her help on the test of an alpha version of the *fonetica2* library and Tara Hansen for her revision to the present manuscript.

References

1. Navarro-Tomás, T.: El alfabeto Fonético de la Revista de Filología Española. In: Anuario de Letras, vol. 6, pp. 5–10, Northamptom, Massachusetts (1966)
2. International Phonetic Association.: The principles of the International Phonetic Association. London, University College (1949/1971)
3. Llisterri, J., Mariño, J.B.: Spanish Adaptation of SAMPA and Automatic Phonetic Transcription. In: Espirit Project Technical Report, vol. 6819 (1993)
4. Hieronymus, J.L.: ASCII Phonetic Symbols for the World's Languages: Worldbet. J. of the Phonetic Association. 23 (1993)
5. Kirschning, I.: Research and Development of Speech Technology & Applications for Mexican Spanish at the Tlatoa Group. In: CHI'01 Extended Abstracts on Human Factors in Computing Systems, pp. 49–50, ACM (2001)
6. Uraga, E.: Modelado Fonético para un Sistema de Reconocimiento de Voz Continua en Español. Master's thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey-Campus Morelos, Mexico (1999)
7. Uraga, E., Pineda, L.A.: A Set of Phonological Rules for Mexican Spanish. IIMAS, UNAM (2000)
8. Uraga, E., Pineda, L. Automatic Generation of Pronunciation Lexicons for Spanish. In: CICLING, pp. 330-338. Springer, Heidelberg, Berlin (2002)
9. Cuétara-Priede, J: Fonética de la ciudad de México Aportaciones desde las Tecnologías del Habla. Master's thesis in Spanish Linguistics, UNAM (2004)
10. Pineda, L. A., Castellanos, H., Cuétara, J., Galescu, L., Juárez, J., Llisterri, J., Villaseñor, L.: The Corpus DIMEx100: transcription and evaluation. In: LREC, vol. 44, no. 4, pp. 347–370 (2010)
11. Uraga, E., Gamboa, C.: VOXMEX Speech Database: Design of a Phonetically Balanced Corpus. In: LREC (2004)
12. Hernández-Mena, C.D., Herrera-Camacho, J.A.: CIEMPIESS: A New Open-Sourced Mexican Spanish Radio Corpus. In: Calzolari, N., Choukri, K. , et al (eds.), LREC'14. pp. 371-375. ELRA, Reykjavik, Iceland (2014)
13. Salcedo, C.S.: The Phonological System of Spanish. In: Revista de Lingüística y Lenguas Aplicadas. Universitat Politecnica de Valencia (2010)
14. Odden, D.: What is Phonology? (1996)
15. Clavigero, F.S.: Rules of the Aztec Language: Classical Nahuatl Grammar. University of Utah Press (1973)
16. Hernndez-Mena, C.D., Herrera-Camacho, J.A.: Creating a Grammar-Based Speech Recognition Parser for Mexican Spanish Using HTK, Compatible with CMU Sphinx-III System. I. J. of Electronics and Electrical Engineering. vol. 3, no. 3, pp. 220–224. February (2015)
17. Daelemans, W. M., Van den Bosch, A. P.: Language-Independent Data-Oriented Grapheme-to-Phoneme Conversion. In: Progress in Speech Synthesis, pp. 77–89. Springer, New York (1997)
18. Ríos-Mestre, A.: La Información Lingüística en la Transcripción Fonética Automática del Español. In: Procesamiento del Lenguaje Natural, no. 13, pp. 381–387 (1993)
19. Quilis, A.: Tratado de Fonología y Fonética Españolas. Editorial Gredos (1993)
20. Pineda, L.A., Pineda, L.V., Cuétara, J., Castellanos, H., López, I.: DIMEx100: A New Phonetic and Speech Corpus for Mexican Spanish. In: IBERAMIA, pp. 974–983. Springer, Berlin, Heidelberg (2004)

A Semantic Approach to Develop Groupware

Anzures-García Mario[1], Sánchez-Gálvez Luz[1], Miguel J. Hornos[2] and Patricia Paderewski-Rodríguez[2],
[1] Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Avenida San Claudio y 14 Sur, Ciudad Universitaria. 72570 Puebla, México
{mario.anzures, sanchez.galvez} @correo.buap.mx
[2] Departamento de Lenguajes y Sistemas Informáticos, ETS Ingeniería Informática, Universidad de Granada, C/ Periodista Saucedo Aranda, s/n, 18071, Granada, Spain
{patricia, mhornos}@ugr.es

Abstract. This paper presents an architectural model-based semantic approach to develop groupware. This approach facilitates a structured and formal representation of a knowledge domain, particularly, the groupware domain. The representation is carried out by using a workflow ontology, which simplifies the development of groupware and its adaptation. The knowledge base of the workflow ontology is comprised for the architectural model elements. The architectural model is provided in order to support the static structure and dynamic nature of this type of applications. In such a way, this model may be able to supply the group organizational structure that supports the group dynamic behavior and, its interaction; as well as, the manner how this behavior will be displayed on the application. Finally, a case study to show the semantic approach usefulness is presented.

Palabras Clave: Semantic Approach, Groupware, Architectural Model, Workflow Ontology, Ontology.

1 Introduction

Nowadays, many organizations require structures that facilitate an efficient communication, collaboration and coordination between the members of the group with independence of their geographical location. These facts have increased the interest in collaborative applications, which generally are known as groupware. This term is defined as: a computer-based system that supports groups of people engaged in a common task (or goal) and provides an interface to a shared environment [1]. The development of groupware is a complex process, since these applications have to consider activities and protocols social of the organizations; shared workspace; interaction among users and between them with shared resources; group awareness; group memory; shared and individual view; and adaptation. So groupware can be adapted to different scenarios using a structured and formal representation of a knowledge domain.

Several tools or frameworks (such as Groupkit [2], ANTS [3], and SAGA [4]), architectures (e.g., Clock [5], and Clover [6]), and methodologies (AMENITIES -A

Advances in Computer Science
Research in computing Science 83, 2014 pp. 71-81

MEthodology for aNalysis and desIgn of cooperaTIve systEmS [7], ClAM (Collaborative Interactive Applications Methodology) [8], y TOUCHE [9].) have been developed in order to carry out construction processes of groupware. However, these proposals lack of the necessary adaptability to conform naturally to the inherent dynamics or the different scenarios that groupware often present. And also, these show the absence of a formal and structured representation to help eliminate ambiguity and redundancy in the development of groupware. Although, TOUCHE, also, proposes an ontology to CSCW Systems, this focuses on collaborative organizational structures, which corresponds to part of the work presented here; since it also adds elements to control the interaction, application itself and adaptation. The TOUCHE ontology is semi-informal. Moreover, our group organizational structures includes the terms of Rights/Obligations and status that are important to determine the tasks that a role can play in the organization.

In this paper, a formal ontology to develop groupware is proposed. The ontology is based on architectural model that supplies the vocabulary terms that are used as knowledge base for the development of groupware. The ontology is one of the strategies to provide a semantic approach to domain knowledge, since establish a formal vocabulary of terms, and some specification of their meaning; including definitions and of how concepts are related.

The rest of the paper is organized as follows. Section 2 describes briefly the architectural model used as knowledge base. Section 3 introduces to ontology. Section 4 explains semantic approach to develop groupware. Section 5 details a case study. Finally, Section 6 outlines the conclusions.

2 Architectural model

As collaborative applications become more and more complex, a critical aspect of their design is the overall structure of the application, and the ways in which that structure provides conceptual integrity for the same. This level of system design has come to be known as software architecture [10] defined as the fundamental organization of a system, embodied in its components, their relationships to each other, to the environment, and the principles governing its design and evolution.

The Architectures are described with informal (by diagrams and words), or formals (by architecture description languages –ADL), representations. In addition, standard general-purpose modeling notations, such as UML, Z [11], or B [12], have been used. There are different architectural styles [10], this paper focuses a layered style, where the architecture components are grouped in layers. A layer is a software technique for structuring the software architecture that can be used to reflect different abstraction levels in the architecture. The architectural model was developed through an extended literature review, which included studies and surveys from multiple venues, such as journals, conferences, and workshops. This model presents a layered style, which contains four layers (see Figure 1).

The Group Layer, facilities the elements that allow the interaction on the application. This layer supplies the **Group Organizational Structure** (GOS) that *is governed by* the **Session Management Policy** (SMP), which determines how the

users are organized. The SMP *defines* the **Roles** that **Users** can *play*. These Roles *establish* the set of **Rights/Obligations** and **Status** within the GOS, and **Tasks** that can be *carried out*. Each Task *is constituted* of **Activities** that *use* the prevailing shared **Resources**.

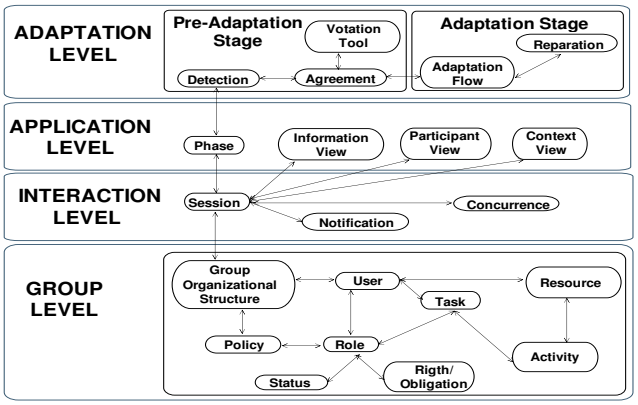


Figure 1. Layered architectural model for building groupware

The Interaction Layer, offers the elements to carry out the interaction among users and between them and application. So, it presents: **Session**, which establishes a shared workspace where a group will collaborate; **Notification**, informs to the users of every occurred change in the session, enabling the group awareness and group memory; and **Concurrency** ensures the consistency of the resources being shared, facilitating the manipulation of the users permissions, which are granted, in accordance with GOS and/or a lock mechanism established.

The Application Layer, allows controlling interaction by means of **phase** [13]—which in a coordination model is defined as each of the collaboration moments— and showing interaction through **views**. The views are user interfaces and there are three: **Information View** —IV, which displays the user information—; **Participant View** —PV, representing the shared view, displaying the changes in shared resources and therefore provide group awareness—; and **Context View** —CV, which displays the change history of shared resources thus providing group memory—.

The Adaptation Layer, facilitates application adaptation and presents two Stages [14] —Pre-Adaptation and Adaptation—. The former determines whether an Activity can adapt the application. For this, it contains a detection process that monitors the Activities in the Session; an agreement process —it is executed only if it is an adaptable process in a non-hierarchical organizational style— to reach a consensus on whether an adaptation process should be performed; and vote tool that allows several kinds of agreement, such as it based on the majority vote, the one based on a maximum or minimum value, etc. The latter performances the adaptation thus it presents the adaptation flow process and the one reparation, which returns each component to its previous state and notifies to users that adaptation cannot take place.

The elements of the first three layers constitute the knowledge base to develop groupware, along with the terms of change, adaptation, and application. These three terms are added, due to that a change in the resources requires to send notification, producing the views adaptation; so, the Application presents the latest changes.

3 Ontology

In recent years, the use of ontologies has extended in diverse areas as medicine [15], bioinformatics [16], groupware [9]; mainly, because they allow a formal explicit specification of a shared conceptualization of certain domain of interest. Conceptualization, refers to an abstract model of some knowledge in the world through the identification of relevant concepts of this. Explicit specification, means that the type of concepts used, and the constraints on their use are explicitly defined. Formal, reflects the fact that the ontology should be machine-readable. Shared, represents the notion that an ontology captures consensual knowledge that is not reserved to some individual, but it is accepted by a group. So, it is said that ontology establishes the vocabulary used to describe and represent domain knowledge to facilitate machine reasoning. The domain knowledge describes the main static information and the objects of knowledge. According to Gruber [17], domain knowledge in ontologies can be formalized using four kind of components: concepts, relations, axioms, and instances.

Ontologies can be highly informal, if they are expressed in natural language; semi-informal, if they are expressed in a restricted and structured form of natural language; formal, if they are expressed in an artificial and formally defined language (i.e. Ontolingua, OWL -Web Ontology Language); and rigorously formal, if they meticulously provide terms defined with formal semantics, theorems and proofs of properties, such as soundness and completeness. Ontologies require a logical and formal language to be expressed. The OWL representation facilities are directly based on Description Logics. This basis confers upon OWL a logical framework, including syntax and model-theoretical semantics, allowing a knowledge representation language capable of supporting a knowledge base, and a practical, effective reasoning. Protégé is used in order to develop ontologies with OWL to provide graphical interfaces that facilitate the knowledge representation and reasoning. The development of ontologies is a laborious and error-prone task, especially if done manually, so it is necessary to have tools that can automate some of this work and hide the features and formalisms for ontology specification languages. These tools provide graphical interfaces that facilitate the knowledge representation and reasoning. This paper uses Protégé, which in order to develop ontologies with OWL to provide graphical interfaces that facilitate the knowledge representation and reasoning. Protégé is an engineering tool open source ontology and a knowledge-based framework, which is widely used due to its scalability and extensibility with lots of plugins; to facilitate inference knowledge through reasoners, query languages, and rules. Therefore, it can be concluded that ontologies are an ideal solution for knowledge representation and reasoning, since it provides a set of symbols through a formal and structured vocabulary.

4 Semantic approach

In this work, a semantic approach is proposed, the approach is based on the workflow ontology. On the one hand, the development of groupware needs to carry out a set of

steps ordered, which can be made by means of a workflow. The term workflow typically refers to "automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another, for action, according to a set of procedural rules" [18]. However, workflows lack the expressive power to represent the domain knowledge and the sequence of operations. On the other hand, ontology describes knowledge domain through concepts, relations, axioms, and instances, although ontology does not specify how these entities should be used and combined.

Special attention has recently been paid to the development of workflow ontologies, for example, it presents a collaborative workflow for terminology extraction and collaborative modeling of formal ontologies using two tools Protege and OntoLancs [19]; it allows the development of cooperatives and distributed ontologies. based on dependencies management between ontologies modules [20]; it shows an ontology-based workflows for ontology collaborative development in Protégé [21], it presents the combination of workflows with ontologies to design way formal protocols for laboratories [22], it proposes a workflow ontology for the preservation digital material produced by an organization or a file system [23]. All these works focused on building workflow ontologies to represent collaborative work in different areas, however, this paper presents a workflow ontology to develop groupware using the architectural model elements as the ontology vocabulary.

The Table 1 shows the concepts, relations and axioms of the Workflow Ontology, which stablishes that: first, the **Application GOS** should be established; second, the **SMP** should be determined, it will govern the GOS; third, the **Roles** should be defined, these are configured **SMP,** and played by one or more users; fourth, the **Users** should be indicated, they are part of GOS; fifth, the **Right/Obligation** (**R/O**) and the **Status** should be designated, the former points out that can do a role, and the latter is role position in the GOS; sixth, the **Tasks** should be mentioned, which are carried out by role, showed in **View**, and part of **Phase**; seventh, the **Activities** should be defined, these are part of a **Task**, and can generate changes**;** eighth, ninth, the **Resources** should be established, these are used in activities, presented on the View, and locked through the Concurrency; the **Notification** should be specified, it is triggered by a change, reflected in the View, and it, also, activate the Adaptation; tenth, the **Concurrency** should be defined, it is triggered by a **change, and** it lock resources to guarantee mutually exclusive usage of same; eleventh, the **IV, PV**, and **CV** should be determined, the three compose a View; twelfth, the **Adaptation** should be showed, it is produced by Notification and displayed on the View; finally, the **Sessions** should be indicated, these are part of an Application, and are composed by **Views** and **Phases**.

5 Case study

This paper presents an Academic Virtual Space (AVS, EVA in Spanish) as case study, which provides the students with a shared workspace to simplify their access to the course material previously loaded by the professor. So, the AVS allows to

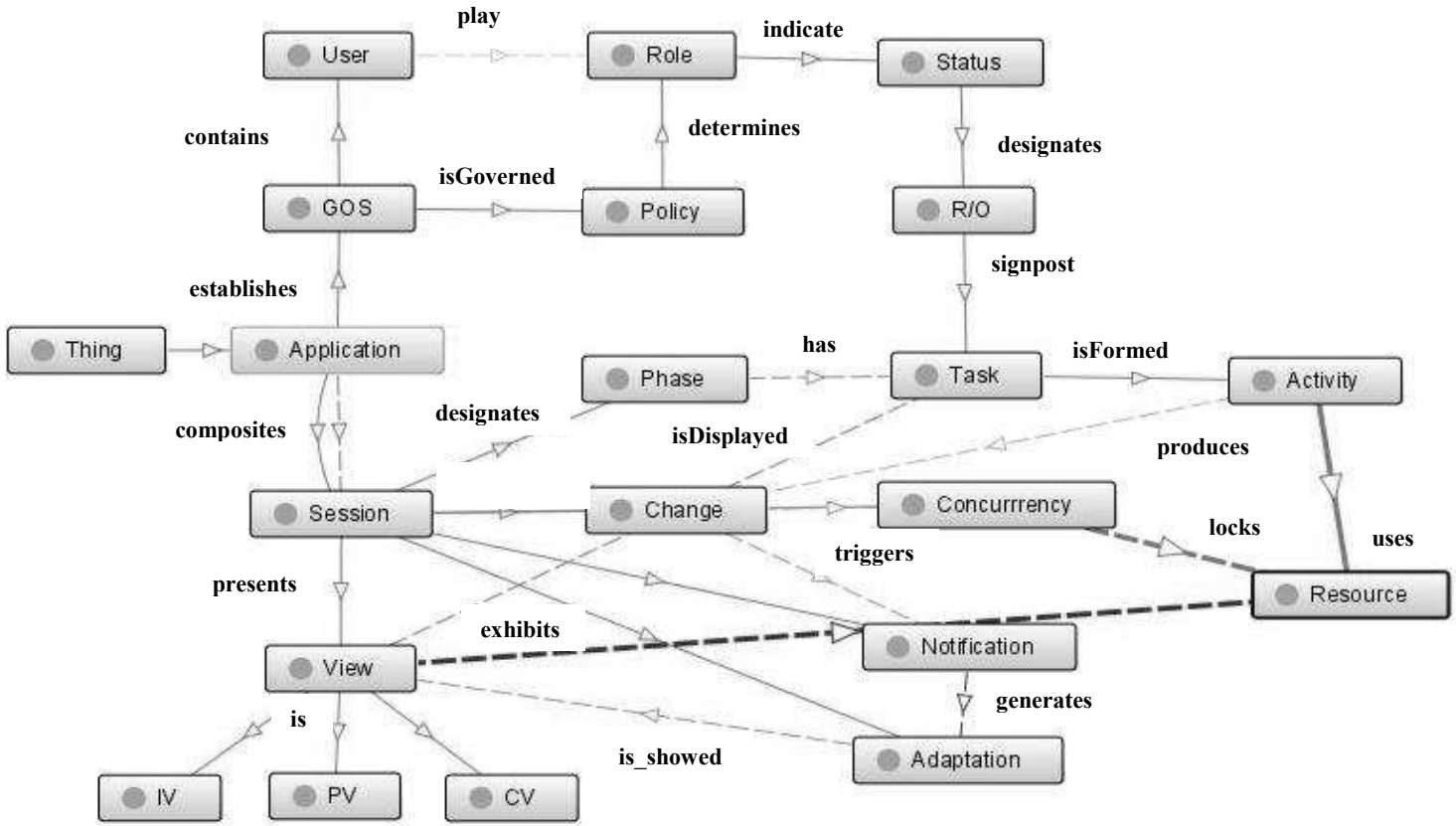


Figure 2. Workflow Ontology to develop groupware.

Table 1. Workflow Ontology components.

Relation	Domain (Concept)	Range (Concept)	Constrain
establishies inverse: isEstablished	Application	GOS	max cardinality=1
contains inverse: IsContained	GOS	User	min cardinality=2
isGoverned inverse: governs	GOS	Policy	max cardinality=1
determines inverse: isDetermined	Policy	Role	min cardinality=1
indicate inverse: isIndicated	Role	Status	max cardinality=1
designates inverse: isDesignated	Status	R/O	min cardinality=1
signpost inverse: isSignposted	R/O	Task	min cardinality=1
isFormed inverse: formed	Task	Activity	min cardinality=1
uses inverse: isUsed	Activity	Resource	min cardinality=1
has inverse: isHave	Phase	Task	min cardinality=1
isDisplayed inverse: display	Task	View	min cardinality=1
exhibits inverse: isExhibited	View	Resource	min cardinality=1
is	IV/PV	View	min cardinality=1
is	CV	View	min cardinality=1
produces inverse: is Produced	Activity	Change	min cardinality=0
triggers inverse: isTriggered	Change	Notification	min cardinality=0
triggers inverse: isTriggered	Change	Concurrency	min cardinality=0
locks inverse: isLocked	Concurrency	Resource	min cardinality=1
generates inverse: isGenerated	Notification	Adaptation	min cardinality=1
shows inverse: isShowed	View	Adaptation	min cardinality=1
is_part_of	View	Session	min cardinality=1
is_part_of	Phase	Session	min cardinality=1
composite	Application	Session	min cardinality=1

generate groups, where students and professors can share information, messages and files. According to workflow ontology carry out:

1. *GOS:* AVS-GOS.
2. *SMP:* AVS-P.
3. *Role:* It specifies two roles: Professor and Students.
4. Users: Professor Mario Anzures-García and Student Coral, Jesus, Ivan.
5. Professor with *status* 1. Student with *status* 2. The Professor and Student have the following RO in common: *accessing, and authenticating* the *Application, managing tasks and writing messages.* Furthermore, the Professor has the *generating group* R/O, and the Student of *Register in the Group.*
6. The Task are to *register user, to authenticate user, to manage user task, to send message, to generate group, and to enroll in the group. The two first Tasks compose the Access Phase, while the others comprise the Collaboration Phase.*

7, 8. Now, the Activities for each Task along with used Resources in they are defined:

   a) To *register user composites of the Activities* (see Figure 3):
      1. *Capturing login* using as Resources the Label, and Text Box

2. *Capturing password* using as Resources the Label, and Text Box
3. *Repeat password* using as Resources the Label, and Text Box.
4. *Capturing first name* with Resources the Label, and Text Box.
5. *Capturing last name* using as Resources the Label, and Text Box
6. *Capturing facebook* using as Resources the Label, and Text Box
7. *Capturing twitter* using as Resources the Label, and Text Box
8. *Capturing e-mail* using as Resources the Label, and Text Box.
9. *Uploading* Picture with Resources the Browse Button and File.
10. *Sending Data* using as Resources *the Register button.*

b) *To authenticate* user *composites of the Activities* (see Figure 3)*:*
   1. *Capturing login* using as Resources the Label, and Text Box
   2. *Capturing password* using as Resources the Label, and Text Box
   3. *Sending Data* using as Resources *the button send.*

c) *To manage user task composites of the Activities:*
   1. *It review file* using as Resources *the Label, Text Box, and file.*
   2. *Creating Delivery Date* using as Resource the Calendar.
   3. *Uploading Task* with Resources *the Label, Text Box, and file.*
   4. *Downloading Task* with Resources *the* Browse Button, *and file.*

d) *To send message composites of the Activities:*
   1. *Comment on the Wall* with Resources *Label, Text Box, and Button Publish.*

e) *To generate group composites of the Activities:*
   1. *Selecting Course* using as Resources *the Label, Text Box, Combo Box, and Button Send.*
   2. *Describing Course* with Resources *the Label, Text Box, and Button Send.*
   3. *Choosing Hours* using as Resources *the Label, Text Box, Combo Box, and Button Send.*
   4. *Creating Group Password* using as Resources *the Label, Text Box, and Button Send.*

f)To *enrol in the group with Activities:*
   1. *Select Teacher using the* Resources: *Teachers List, Groups List, Courses List, and Button Send.*
   2. *Select Group using the* Resources: *Teachers List, Groups List, Courses List, and Button Send.*
   3. *Select Course with Resources of Teachers List, Groups List, Courses List, and Button Send.*
   4. *Send Data with the Resource Button Send.*

9. The Notification is generated by the activities of *sending data, reviewing file, creating delivery date, uploading task, downloading task, comment on the wall, and creating group*.
10. The Concurrency is triggered by the activities of *sending data, reviewing file, creating delivery date, uploading task, downloading task, comment on the wall, and creating group*.
11. The AVS View presents IV, PV, and VC.

12. The Adaptation is activates by the activities of *sending data, reviewing file, creating delivery date, uploading task, downloading task, comment on the wall, and creating group*.
13. The session is AVS-Session.

Figure 3. AVS (EVA) developed with workflow ontology.

6 Conclusions and Future Work

This paper has presented a semantic approach to develop groupware based on architectural model. This approach supplies knowledge base, which facilitates and simplifies the development of groupware. The knowledge base supplies a vocabulary that easiness the representation, specification, analysis, and design of this kind of applications. The workflow ontology indicates how users are distributed and organized —i.e. how they communicate, coordinate, collaborate—; how the interaction and adaptation are controlled, as well as how it is presented the interaction in the Views. The future work is orientated to specify a methodology to develop groupware.

Referencias

1. Ellis, C.A., Gibas, S.J., and Rein, G.L. Groupware: some issues and experiences. Communications of the ACM, Vol. 34-1, pp. 39-58, (1991).
2. Roseman, M., and Greenberg, S. Building Real-time Groupware with GroupKit, a Groupware ToolKit. ACM Trans. Computer-Human-Interaction, Vol. 3, 66-106, (1996).
3. García, P., and Gómez, A. ANTS Framework for Cooperative Work Environments. IEEE Computer Society Press, Vol. 36, 3 Los Alamitos, CA, USA, pp. 56-62, (2003).
4. Fonseca, B., and Carrapatoso, E. SAGA: A Web Services Architecture for Groupware Applications, In Proc. of the CRIWG, LNCS 4154, Springer-Verlag, pp. 246-261, (2006).

5. Graham, T.C.N., and Urnes, T. Integrating Support for Temporal Media in to an Architecture for Graphical User Interfaces. Proc. of the International Conference on Software Engineering (ICSE'97), ACM Press, Boston, USA, pp. 172-182, (1997).
6. Laurillau, Y., and Nigay, L. Clover Architecture for Groupware. Proc. of the ACM Conference on CSCW. New Orleans, Louisiana, USA, pp. 236-245, (2002).
7. Gea, M., Gutierrez, F.L., Garrido, J. L., and Canas, J.J. AMENITIES: Metodología de Modelado de Sistemas Cooperativos presentado en COLINE02.Workshop de Investigaci6n sobre nuevos paradigmas de interacción en entornos colaborativos aplicados a la gestión y difusión del Patrimonio cultural. Granada, Spain, (2002).
8. Molina, A.I., Redondo, M.A., Ortega, M., and Hope, U. ClAM: A methodology for the development of groupware user interfaces. Journal of Universal Computer Science (2007).
9. Penichet, V.M.R., Lozano, M.D., and Gallud. J.A. An Ontology to Model Collaborative Organizational Structures in CSCW Systems. In Engineering the User Interface, Springer, pp.127-139, (2008).
10. Garlan, D., and Shaw, M. An introduction to software architecture, Advances in Software Engineering and Knowledge Engineering, 1, pp. 1-39, (1994).
11. Spivey, J.M. The Z notation: A reference manual. Prentice Hall, (1989).
12. Abrial. J.R. The B-book: Assigning Programs to Meanings, Cambridge University Press.
13. Anzures-García, M., Sánchez-Gálvez, L.A., Hornos, M.J., and Paderewski-Rodríguez, P. Service-based access control using stages for collaborative systems. Advances in Computer Science and Engineering. Research in Computing Science, Vol. 42, pp. 311-322, (2009)
14. Anzures-García, M., Sánchez-Gálvez, L.A., Hornos, M.J., and Paderewski, P. Security and adaptability to groupware applications using a set of SOA-based services. Advances in Computer Science and Engineering. RCS, Vol. 45, pp. 279-290, (2010)
15. Jovic, A., Prcela, M, and Gamberger, D. Ontologies in Medical Knowledge Representation. In Proceedings of the 29th Int. Conf. on Information Technology Interfaces, pp. 25-28, (2007).
16. Stevens, R., Goble, C.A., and Bechhofer, S. Ontology-based knowledge representation for bioinformatics. Henry Stewart Publications 1467 – 5463. Briefings in Bioinformatics, Vol. 1-4, pp. 398-414, (2000).
17. Gruber, R. A translation approach to portable ontology specification. Knowledge Acquisition. Vol. 5, pp. 199-220, (1993).
18. Allen, R. Workflow: An introduction. In Workflow Handbook, L. Fisher, Ed. Future Strategies, Lighthouse Point, FL, pp. 15–38, (2001).
19. Gacitua, R.; Arguello Casteleiro, M.; Sawyer, P.; Des, J.; Perez, R.; Fernandez-Prieto, M.J.; Paniagua, H., A collaborative workflow for building ontologies: A case study in the biomedical field. *Research Challenges in Information Science*, pp.121-128.
20. Kozaki, K., Sunagawa, E., Kitamura, Y. and Mizoguchi, R. A Framework for Cooperative Ontology Construction Based on Dependency Management of Modules. ESOE, Vol. 292 of CEUR Workshop Proceedings, pp. 33-44. (2007)
21. Sebastian, A., Noy, N.F., Tudorache, T., Musen, M.A. A Generic Ontology for Collaborative Ontology-Development Workflows. Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns, (2008) .
22. Maccagnan, A., Riva, M., Feltrin, E., Simionati, B., Vardanega, T., Valle, G., Cannata, N. Combining ontologies and workflows to design formal protocols for biological laboratorios, Automated Experimentation, Vol. 2-3, (2010).
23. Mikelakis, M., Papatheodorou, C. An ontology-based model for preservation workflows. In Proceedings of the 9th International Conference on Digital Preservation, (2012).

Neural Networks & Unconventional Computation

Artificial Neural Networks for Seasonal Time Series Applied to Tourism Demand Forecasting

Tomás Molinet Berenguer [1], Napoleón Conde Gaxiola [2],
José Antonio Molinet Berenguer [3]

[1] Centro de Estudios Multidisciplinarios del Turismo, Universidad de Camagüey, Cuba.
E-mail: tomas.molinet@reduc.edu.cu
[2] Escuela Superior de Turismo del Instituto Politécnico Nacional, México.
E-mail: napoleon_conde@yahoo.com.mx
[3] Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México.
E-mail: jmolinet@computacion.cs.cinvestav.mx

Abstract. In this paper, we propose a new artificial neural network (ANN) structure to predict seasonal time series and we study its application to tourism demand forecasting. Our proposal uses the seasons of tourist arrivals and values of months with similar behavior as input variables and achieves a forecast up to a year in advance. Our proposed approach is compared with respect to other forecasting techniques available in the literature, such as Autoregressive Moving Average (ARIMA) models and others ANN models. We study the validity and precision of the proposed model using two tourism demand time series. Our preliminary experimental results indicate that the proposed forecasting structure provides a longer period in advance and high forecasting accuracy than previously used ARIMA and ANN models.

Keywords: artificial neural networks, time series, tourism demand forecasting

1 Introduction

The use of artificial neural networks has received increasing attention in the modelling and forecasting of financial time series [6]. When modelling univariate time series using neural networks, the so-called time delay subset of the series is used as network input values. The values of this subset are previous to the value to be forecast, which constitutes the output of the neural network [3]. Several authors have rated that the best neural network structure is obtained when its input values correspond to the 12 months immediately prior to the month to be forecast [4,5],[9,10], which means that each time series value depends directly on the twelve previous values. Some modifications have been proposed to the structure that uses the twelve prior months. For example, in [12] the data are quarterly and use a range of 2 to 8 backward terms, *i.e.*, from 6 months to 24 months as inputs, and between 1 and 2 quarters as outputs of the network model. In [13] was proposed a new ANN methodology that uses a time index variable as the network input value, which may or may not be accompanied by the observations corresponding to the 12 previous months. They argue that thereby it is possible to capture the seasonal nature and tendency of the series in a better way. However, these ANN models only allow a

Advances in Computer Science
Research in computing Science 83, 2014 pp. 83-92

better forecast a short time in advance and by using values nearby in time for the forecast the results may be affected in non-consolidated destinations in which the behaviour of tourism demand can vary sharply from one year to the next.

In recent studies were proposed neural network models which incorporate other input values to improve the forecasting accuracy in seasonal and non-linear time series. In [1] was suggested a multiplicative seasonal ANN (MSANN) to forecast time series with both trend and seasonal patterns. The input layer of MSANN is composed of two parts. The first part consists of inputs for trend component and the other one includes inputs for seasonal component, the weighted sum of each component corresponds to one of two hidden neurons. On the other hand, in [15] was proposed a neural network model which considers that time series has both linear and nonlinear components. The input layer of this model is composed by two parts, one corresponding to the linear component and the other one contain the non-linear component. The hidden layer consists of two neurons, the output value of one is equal to the weighted sum of linear component and second neuron corresponds to weighted multiplication of the non-linear component.

In this study we propose an ANN structure which, by using time series, different seasons of arrivals and values of months with a similar behaviour, can obtain a tourism demand forecast up to one year in advance, both in mature destinations and non-consolidated ones, with a quality that can be considered of high and good accuracy, respectively.

2 Formulating the proposed ANN structure

One of the most widely used neural networks in time series forecasting is the Multilayer Perceptron (MLP) [7]. The neurons in the input layer represent the values of the variable in the past and the only neuron in the output layer represents the value to be forecast. Generally, the network input values are the 12 months prior to the month to be forecast. However, this limits the forecast to only a month in advance. This is why we propose a model which will allow us to obtain the value of tourism demand with a forecast of up to a year in advance and with a similar error to the models that forecast less time in advance.

The proposed model uses the values of tourism demand in previous years as input values for the ANN, thereby we obtain a forecast of up to a year in advance and by considering several periods of time, the particular behaviour of one of them will not affect the result to a great extent. The name of the month to be forecast and the season it belongs to also make up input values, as their use enables the ANN to better adjust the specific period to be forecast. Moreover, it is considered that by using the information afforded by the months nearest in time to the month to be forecast which correspond to the same season as the network input, this helps to improve the precision of the forecast, as the behaviour of tourism demand is similar in each of them.

Fig. 1 shows the network input values, which are: the name of the month to be forecast (name of month), the season it belongs to (season), the values of this month

in the previous *i*-th years (value in $year_{n-i}$) and the values of the two months nearest to the month to be forecast, which correspond to the same season for each of the previous *i*-th years ($neighbour_1$ in $year_{n-i}$ and $neighbor_2$ in $year_{n-i}$). With the aim of obtaining the forecast of the month in year *n* (value in $year_n$) as the output.

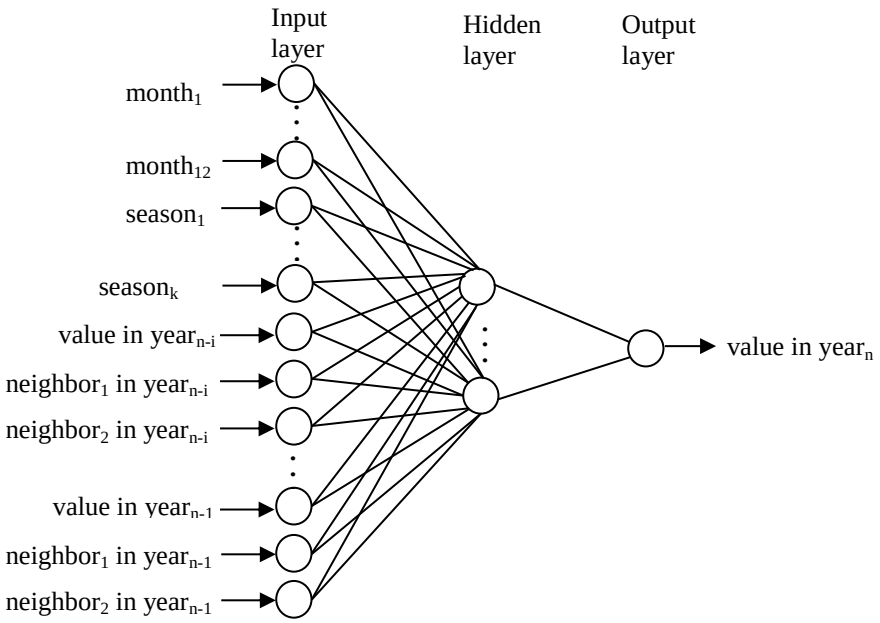


Fig. 1. Graphic representation of the proposed ANN structure.

3 Experimental results

To compare the models we forecast the tourism demand in two destinations with totally opposite characteristics, thus to know the differences that could mean the use of ANN models in forecasting demand. Mexico is a tourist destination with a very stable demand behavior while Santa Lucía de Cuba has a demand behavior complex and unstable, where the behavior of tourism demand can vary greatly from year to year. It is in destinations with these characteristics where the general accuracy of the forecasting models cannot be regarded as highly accurate.

The results obtained by the proposed model in forecasting each of the aforementioned destinations were compared with the ones found by an ANN model with two different forecasting horizons: ANN-month and ANN-year. The first used the 12 months immediately before to the month to be forecast, it means that for predict the month *t* is used: month *t-1*, month *t-2*, month *t-3*,…, month *t*-12. It is important to note that the prediction obtained with ANN-month has only a month in advance. The second variant used the 12 months that have a difference of at least one year with the month to be predicted, it means that for obtain the month *t* is used:

month *t-13*, month *t-14*, month *t-15*,…, month *t-24*. The ANN-year forecast with a year in advance. These results were also compared with those obtained by an ARIMA model with a forecasting horizon of one month in advance. This comparative analysis is carried out in all cases between the data of the original series and the corresponding estimated values for each model.

The criteria used in order to compare the accuracy of the different models are mean absolute percentage error (MAPE), due to its widespread use to assess the performance of the forecast [14], the Median and Huber's M-estimator, the latter two do not show the sensitivity of MAPE when faced with the existence of outlier values far from the centre of the distribution.

3.1 Definition of training, validation and test sets

The data corresponding to the tourist destinations of Santa Lucía de Cuba [11] and Mexico [2] were divided into three sets, 96 months which correspond to the training set, 12 months for the validation set and 12 months for the test set.

In the case of the tourist destination Santa Lucía de Cuba, for the proposed ANN model, ANN-month and ANN-year the observations from January 2001 to December 2008 were defined as the training set, the ones corresponding to January to December 2009 as the validation set, and from January to December 2010 as the test set. Meanwhile, for Mexico, the period ranging from January 2003 to December 2010 is considered as the training set, from January to December 2011 as the validation set, and from January to December 2012 as the test set.

Due to the fact that the proposed ANN model uses information referring to the three previous years, the training set will then have 60 patterns; on the other hand, the ANN-month model, as it only uses information from the 12 previous months, has 84 patterns in its training set, and the ANN-year, that uses information referring to the 24 previous months, has 72 patterns in its training set. Whereas, for all models both the validation set and the test set have 12 patterns.

3.2 Assessment of results

In order to obtain the best network architecture of the proposed ANN, ANN-month and ANN-year models for each destination, different training sessions were carried out, in each of which one or two hidden layers were taken, with the number of neurons varying in each of these layers between 1 and 30, avoiding a complex network structure which would impede their generalization. In the learning process the algorithms used were Back Propagation, which is widely used in different problems [7], [16] and the conjugated gradient. The former was applied independently or followed by the latter, which has the advantage of rapid convergence. In each case the number of iterations in the training varied and the network with the best performance in forecasting the validation set was chosen (see Table 1). The activation function used in the hidden neurons was the sigmoidal one and in the output neuron the linear one.

In order to choose the best ARIMA model in each destination, all the ARIMA models were constructed ARIMA(p,d,q)x(sp,sd,sq)$_{12}$ by varying the parameters *p*, *d*, *q*, *sp*, *sd* and *sq* between 0 and 2, values greater than 2 are generally not necessary [4]. The training set were used, without transforming and by applying the natural logarithm transformation; so as to calculate the coefficients of each model and the least error in the forecast of the validation set was chosen as the best model (see Table 1).

Table 1. ANN and ARIMA models obtained for forecasting each destination.

Model / Destination	proposed ANN	ANN-month	ANN-year	ARIMA
	input-hidden-output layers			
Santa Lucía	24-2-1	12-4-1	12-10-1	ARIMA $(2,0,0)x(2,1,2)_{12}$
Mexico	24-2-1	12-4-1	12-10-1	ARIMA $(2,0,1)x(1,0,2)_{12}$

In accordance with the classification given in [8] the absolute relative errors of the forecasts that are lower than 10% are considered 'highly accurate', between 10 and 20%, are considered 'good', between 20 and 50% are considered 'reasonable', and greater than 50%, 'imprecise/not very reliable'.

Below we analyse the forecasting capacity for each model, first of all in a non-consolidated destination (Santa Lucía de Cuba) and then in a consolidated one (Mexico), taking into account the aforementioned criterion in order to classify the quality of the forecast and the characteristics peculiar to each type of destination.

The neural network model applied was Multilayer Perceptron (MLP). In this study no change was made in the MLP model, but is propose a set of new inputs not previously considered and which are relevant to achieve an improvement in prediction. The input values considered in our work allow to describe more precisely the behavior of series with a marked instability. In order to test the relevance of the variables: month name and season, in the effectiveness of the forecast, we performed a sensitivity analysis to the proposed model. We constructed several subsets with the input variables and was selected the three subset that obtained the best accuracy in the forecast.

Table 2 shows the results of the sensitivity analysis. In the table, $N_{i,j}$ is the value of the *j*-th neighbor month in the year *n-i*, and V_i is the value of the month to predict but in the year *n-i*. The variable month name and season were selected in 3 networks and season was placed in the first rank. A ratio value equal to 1 or less indicates that the input variable is irrelevant and a higher value indicates more importance. The Rank indicates the order of importance of the input variables. Table 3 shows the results of the sensitivity analysis performed to all variables that are proposed, this results correspond to the tourism demand forecasting in Mexico. In the case of St. Lucia all input variables turn out to be relevant, see Table 4.

In Table 5, we can see the results obtained by the different models in the tourist destination Santa Lucía. Our propose approach achieve a prediction with a MAPE of

23.39%, based on the study of Lewis [8] this is a reasonable prediction. Since the instability of tourism demand in Santa Lucia and the results obtained by other models, we can consider that results achieved by our proposal are reasonable. Although the proposed ANN has a MAPE 5% higher than ANN-month, must be taken into account that has a forecast horizon 12 times higher than the model ANN-month, which is an important advantage for decision making.

Table 2. Results obtained for Mexico destination by creating subsets of independent variables.

	Month	Season	V_3	$N_{1,3}$	$N_{2,3}$	V_2	$N_{1,2}$	$N_{2,2}$	$N_{1,2}$
Ratio.1	1.34	1.52	1.44				1.36		
Rank.1	4	1	2				3		
Ratio.2	1.90	2.57	1.05	1.23				1.26	
Rank.2	2	1	5	4				3	
Ratio.3	1.32	2.42	1.53		1.01	1.10	1.36		0.99
Rank.3	4	1	2		6	5	3		7

Table 3. Results obtained for Mexico destination using all input variables of the model.

	Month	Season	V_3	$N_{1,3}$	$N_{2,3}$	V_2	$N_{1,2}$	$N_{2,2}$	V_1	$N_{1,1}$	$N_{1,2}$
Ratio	1.69	2.13	1.03	1.10	1.00	1.00	1.01	1.01	1.18	1.02	1.01
Rank	2	1	5	4	11	10	7	8	3	6	9

Table 4. Results obtained for Santa Lucía destination using all input variables of the model.

	Month	Season	V_3	$N_{1,3}$	$N_{2,3}$	V_2	$N_{1,2}$	$N_{2,2}$	V_1	$N_{1,1}$	$N_{1,2}$
Ratio	1.45	2.83	1.53	1.09	1.04	1.49	1.48	1.36	1.23	1.27	1.06
Rank	5	1	2	9	11	3	4	6	8	7	10

The results obtained with the ARIMA model for this destination are shown in Table 5. The accuracy achieved for this model in the validation set is lower than that obtained by the ANN models. This inadequate fit to the validation data is reflected to a greater extent in the real forecast which it is capable of achieving in the test set, as even ARIMA, which carries out the forecast a month in advance, achieves results inferior to the proposed ANN. Similar results to the ARIMA models in forecasting demand were obtained in [4].

One of the reasons for the difference between the results obtained in each of the sets is due to the fact that from 2007 to 2009 there was a continuous growth in tourism demand but in 2010 there was a sharp drop of 25% (see Fig. 2). As the ANN-month model uses real values for the months prior to the month to be forecast, it is capable of gradually identifying this decrease, however the proposed ANN model only has the information from previous years to model the behaviour of the demand. However, compared with results obtained by the ANN-year the proposed model has a better performance because forecast with an error of 4.5% lower, although both predict with the same period of time.

Table 6 shows the results obtained for all models in the validation and test sets corresponding to tourism demand in Mexico. In the validation set, the proposed ANN structure had a better fit than the others (see Fig. 3). Furthermore, in the test set it has a less error than the obtained by ANN-month, even though it forecast with a greater time in advance. Although the ANN-year structure predicts with the same period of anticipation, has a 1.1% greater error than the obtained by the proposed ANN structure. In general terms, the forecast for both models in each set can be classified as highly precise.

Table 5. Forecasts obtained with the ANN and ARIMA models in Santa Lucía de Cuba.

	Validation set			**Test set**		
	MAPE	Median	Huber's M-estimator	MAPE	Median	Huber's M-estimator
Proposed ANN	10.14	5.08	5.38	23.39	17.92	23.33
ANN-month	7.59	7.02	6.91	18.18	11.86	12.46
ANN-year	8.25	4.25	4.81	27.93	24.87	23.93
ARIMA	13.82	10.16	12.49	34.51	33.14	30.64

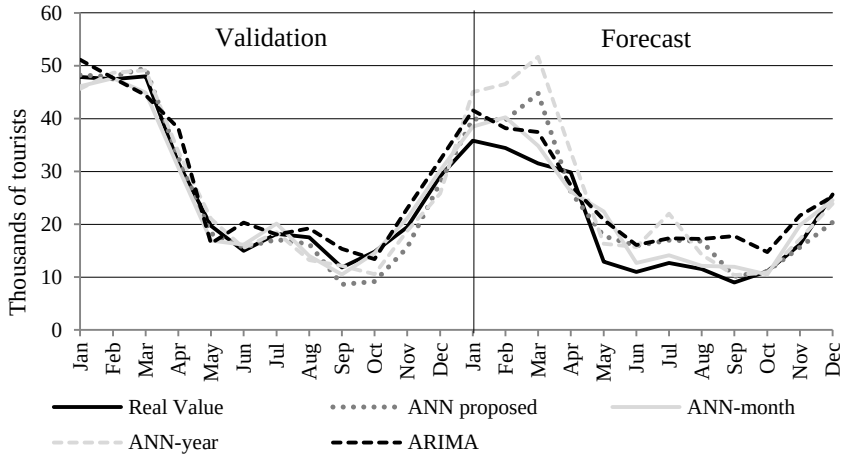


Fig. 2. Real values and forecasts obtained with the ANN and ARIMA models in Santa Lucía de Cuba.

Table 6. Forecasts obtained with the ANN and ARIMA models in Mexico.

	Validation set			**Test set**		
	MAPE	Median	Huber's M-estimator	MAPE	Median	Huber's M-estimator
Proposed ANN	1.34	0.69	0.87	3.51	3.19	2.98
ANN-month	6.98	6.6	6.85	4.07	1.77	2.35
ANN-year	6.83	4.99	5.26	4.59	3.77	3.96
ARIMA	6.52	5.56	5.67	4.22	3.04	3.27

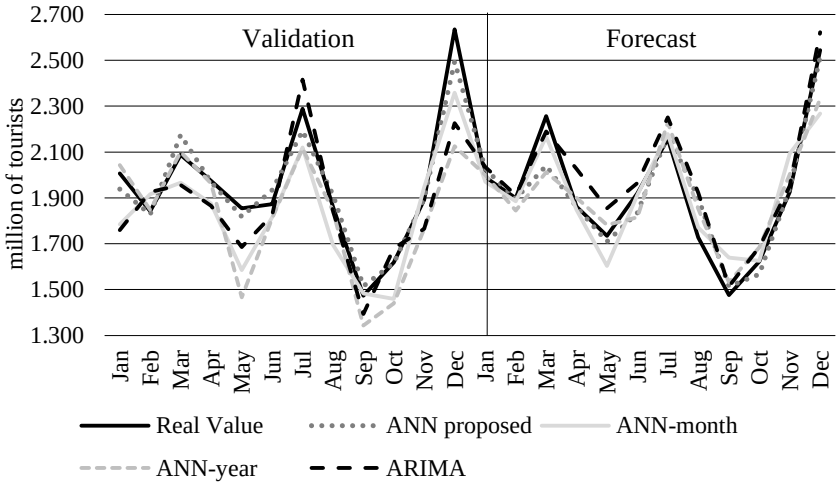


Fig. 3. Real values and forecasts obtained with the ANN and ARIMA models in Mexico.

4 Conclusions

In this study we proposed an ANN model which by using time series, different arrival seasons and values of months with similar behaviour, enabled us to obtain a demand forecast up to a year in advance. The idea of using the month and the season as input values has the aim to characterize the time series with other features that will provide more information to the neural network and improve its ability to detect patterns in time series behaviour. The input values incorporated to our network model enhance its performance mainly in unconsolidated tourist destinations, which have a high variability in its behaviour and hence, the time series have a high degree of non-stationarity. The proposed ANN was compared with an ARIMA model and two neural network models: ANN-month and ANN-year, they have a forecasting horizon of one month and one year, respectively. These last two ANN models only consider as inputs the values of the previous months. Our experimental results show that our proposal improves the performance of ANN-month and ANN-year. First, the forecasting horizon of the proposed ANN is longer than ANN-month and the forecasting accuracy is competitive. Second, the accuracy obtained by our proposal outperforms ANN-year.

The results obtained in forecasting demand in a consolidated and non-consolidated tourist destination indicate that the proposed model may be effective in forecasting demand and obtaining better results than those obtained by traditional ARIMA model and the ANN-year model. Moreover, the forecast carried out by the proposed model is similar to that obtained by the ANNs that use the 12 months prior to the one to be forecast, but with one year in advance.

5 References

1. Aladag, C. H., Yolcu, U., Egrioglu, E.: A new multiplicative seasonal neural network model based on particle swarm optimization. Neural Processing Letters, 37(3), 251-262 (2013)
2. BANXICO. Información económica. Banco de México, México (2013)
3. Bishop, C. M.: Neural networks for pattern recognition. Oxford University Press, Oxford (1995)
4. Cho, V.: A comparison of three different approaches to tourist arrival forecasting. Tourism Management, 24, 323-330 (2003)
5. Fernandes, P. O., Teixeira, J. P., Ferreira, J. M., Azevedo, S. G.: Modelling Tourism Demand: A Comparative Study between Artificial Neural Networks and the Box-Jenkins Methodology. Romanian Journal of Economic Forecasting 5, 30-50 (2008)
6. Kim, T. Y., Kyong, J. O., Kim, C., Do J. D.: Artificial neural networks for non-stationary time series, Neurocomputing 61, 439-447 (2004)
7. Law, R.: Back-propagation learning in improving the accuracy of neural network-based tourism demand forecasting. Tourism Management 21, 331-340 (2000)
8. Lewis, C. D.: Industrial and Business Forecasting Method. Butterworth Scientific, London (1982)
9. Lin, C.-J., Chen, H.-F., Lee, T.-S.: Forecasting Tourism Demand Using Time Series, Artificial Neural Networks and Multivariate Adaptive Regression Splines: Evidence from Taiwan. International Journal of Business Administration 2, 14-24 (2011)
10. Montaño, J. J., Palmer, A., Muñoz, P.: Artificial neural networks applied to forecasting time series. Psicothema 23, 322-329 (2011)
11. ONE. Anuario estadístico de Cuba. Oficina Nacional de Estadística, Cuba (2011)
12. Palmer, A., Montaño, J. J., Sesé, A.: Designing an artificial neural network for forecasting tourism time series. Tourism Management, 27, 781-790 (2006)
13. Teixeira, J. P., Fernandes, P. O.: Tourism Time Series Forecast -Different ANN Architectures with Time Index Input. Procedia Technology 5, 445–454 (2012)
14. Witt, S. F., Witt, C. A.: Modelling and Forecasting Demand in Tourism. Academic Press, London (1992)
15. Yolcu, U., Egrioglu, E., Aladag, C. H.: A New Linear & Nonlinear Artificial Neural Network Model for Time Series Forecasting. Decision Support Systems, 54(3), 1340-1347 (2013)
16. Zhang, G., Patuwo, B. E., Hu, M. Y.: Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting, 14, 35–62 (1998)

Comparative Analysis of Associative Memories on Agricultural Context

Mario Aldape-Pérez, José Antonio Estrada Pavía,
Oscar Camacho-Nieto, and Joel Omar Juarez Gambino

Center for Computing Innovation and Technology Development. CIDETEC
National Polytechnic Institute, IPN
Mexico City, Mexico
maldape@ieee.org; jaestradap@gmail.com; oscarc@cic.ipn.mx;
omarjg82@gmail.com
URL: http://www.aldape.mx

Abstract. Associative memories have a number of properties, including a rapid, compute efficient best-match and intrinsic noise tolerance that make them ideal for many applications. In this paper we will compare Alpha-Beta associative memory against CHAT associative memory in order to find out which one is better for a real agronomic application focused on herbicides.

Keywords: Associative Memories, Alpha-Beta, CHAT, Herbicides, Pattern Recognition

1 Introduction

This paper focuses on classification of pesticides. Synthetic organic pesticides are used to control weeds, insects, and other organisms in a wide variety of agricultural and non-agricultural settings [1]. Particularly this paper is focused on herbicides that are used to control weeds.

Nowadays the agronomists have a lot of herbicides to use, as well as a lot of information related to product application. Each herbicide has its own characteristics, and more important has a set of weed and crops in where it can be used, herein lies a classification problem, a lot of information and difficult access to it. Since 1997 this problem has been taken into account.

On 1997 Artificial Neural Networks (ANN) have proven to be an effective alternative to deal with complex agricultural systems [2]. After this first approach, complex intelligent agronomic systems have been developed. In 2003 a herbicide application using neural networks and fuzzy logic was developed; results of this project illustrated the potential benefits of using image processing methods, ANNs and fuzzy logic to develop herbicide application maps for precision farming [3].

Later on 2004 a work was developed in order to investigate the potential of using neural-network classifiers to discriminate weed species in field crops. It has been found that the Back Propagation Neural Network (BPNN) classifier

Advances in Computer Science
Research in computing Science 83, 2014 pp. 93-103

achieved classification accuracies of 96.7% which exceed discriminant classification procedures in relatively simple network topologies [4].

In 2005 an application of ANN was developed for prediction of pesticide occurrence in rural domestic wells with reduced information. With the available information, a four layer BPNN can be employed to detect pesticide occurrence in wells with 89% accuracy [1].

Some authors describe associative memories as weightless neural networks, however associative memories have intrinsic properties that make them suitable for a wide variety of problems [5],[6], [7], [8]. The fundamental purpose of an associative memory is to recall complete patterns from input patterns, which may be affeted with additive, subtractive or mixed noise [9]. Associative memories have two functional phases, the learning phase, and the recalling phase, in the learning phase the associative memory is built with a set of previously classified patterns one input pattern and its corresponding output pattern.

One of the most relevant associative memory work was developed by Hopfield in 1982 [10] his model demonstrates that the interaction of simple processing elements, similar to neurons, give place to the rise of collective computational properties, such as the stability of memories [9]. The associative memories models to be compared in this paper are named Alpha-Beta Associative Memory and CHAT Associative Memory.

2 Alpha-Beta Associative Memories

The purpose of any associative memory is to recall correct output patterns from input patterns, which may be altered with additive, subtractive or mixed noise. The concepts used in this section are presented in [9].

An associative memory M is a system that relates input patterns and output patterns as follows: $x \longrightarrow \boxed{\mathbf{M}} \longrightarrow y$ with x and y, respectively, the input and output pattern vectors. Each input vector forms an association with its corresponding output vector. For each k integer and positive, the corresponding association will be denoted as: (x^k, y^k). Associative memory M is represented by a matrix whose ij-th component is m_{ij} [11]. Memory M is generated from an *a priori* finite set of known associations, called the fundamental set of associations. If μ is an index, the fundamental set is represented as: $\{(x^\mu, y^\mu) \mid \mu = 1, 2, ..., p\}$ with p as the cardinality of the set. The patterns that form the fundamental set are called fundamental patterns. If it holds that $x^\mu = y^\mu \ \forall \mu \in \{1, 2, ..., p\}$ M is auto-associative, otherwise it is heteroassociative; in this case, it is possible to establish that $\exists \mu \in \{1, 2, ..., p\}$ for which $x^\mu \neq y^\mu$. If we consider the fundamental set of patterns $\{(x^\mu, y^\mu) \mid \mu = 1, 2, ..., p\}$ where n and m are the dimensions of the input patterns and output patterns, respectively, it is said that $x^\mu \in A^n$, $A = \{0, 1\}$ and $y^\mu \in A^m$. Then the j-th component of an input pattern is $x_j^\mu \in A$. Analogously, the j-th component of an output pattern is represented as $y_j^\mu \in A$. A distorted version of a pattern x^k to be recuperated will be denoted as $\widetilde{x}^k$. If when feeding an unknown input pattern x^ω with $\omega \in \{1, 2, ..., k, ..., p\}$

Table 1. Alfa and Beta Operators.

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

x	y	α(x,y)
0	0	1
0	1	0
1	0	2
1	1	1

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

x	y	β(x,y)
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

to an associative memory M, it happens that the output corresponds exactly to the associated pattern y^{ω}, it is said that recuperation is perfect.

Alpha-Beta Associative Memories mathematical foundations are based on two binary operators: α and β. Alpha operator is used during the learning phase while Beta operator is used during the recalling phase. The mathematical properties within these operators, allow the $\alpha\beta$ associative memories to exhibit similar characteristics to the binary version of the morphological associative memories, in the sense of: learning capacity, type and amount of noise against which the memory is robust, and the sufficient conditions for perfect recall [12]. First, we define set $A = \{0, 1\}$ and set $B = \{0, 1, 2\}$, so α and β operators can be defined as in Table 1.

These two binary operators along with maximum ($\vee$) and minimum ($\wedge$) operators establish the mathematical tools around the Alpha-Beta model. The definitions of α and β exposed in Table 1, imply that: α is increasing by the left and decreasing by the right, β is increasing by the left and right, β is the left inverse of α. According to the type of operator that is used during the learning phase, two kinds of Alpha-Beta Associative Memories are obtained. If maximum operator ($\vee$) is used, Alpha-Beta Associative Memory of type MAX will be obtained, denoted as M; analogously, if minimum operator ($\wedge$) is used, Alpha-Beta Associative Memory of type min will be obtained, denoted as W [9]. In any case, the fundamental input and output patterns are represented as follows:

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

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

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

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

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

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

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

$$y\nabla_\alpha x^t = y \oplus x^t = y\Delta_\alpha x^t.$$

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

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

2.1 Learning Phase

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

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

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

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

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

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

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

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

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

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

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

2.2 Recalling Phase

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

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

Alpha-Beta Associative Memory of type MAX:

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

Alpha-Beta Associative Memory of type min:

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

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

3 CHAT Associative Memory

Associative Memory CHAT is based on the combination of two associative memories, the Lernmatrix from Steinbuch[13] and Linear Associator from Kohonen[11]. By combining these two associative memories the Associative Memory CHAT can accept real values in the components of the input vectors and these vectors are not required to be orthonormal, to obtain those new properties the learning phase from Linear Associator and the recalling phase from Lernmatrix were combined [14].

The algorithm for memory CHAT is as follows:

1. Lets define a fundamental set of input patterns of dimension n with real values on their components (as in the Linear Associator), this pattern are organized in m different classes.
2. For each input pattern in the class k define a vector made of zeros except the $k - th$ coordinate, where the value is one (as in the Lernmatrix).
3. Calculate the medium vector from the set of sample patterns with the expression.

$$\bar{x} = \frac{1}{p} \sum_{\mu=1}^{p} x^\mu$$

With $x^1, x^2, x^3, ..., x^p$ as the set of input patterns
With $\bar{x}$ as the medium vector for all input patterns
With p as the total number of patterns to be used

4. Take the components from the medium vector as the center of a new set of coordinate axes.
5. Move all the patterns from the fundamental set.

$$x^{\mu'} = x^{\mu} - \bar{x}$$

With $x^{\mu'}$ as the displaced pattern
With x^{μ} as the original pattern
With $\bar{x}$ as the medium vector for all input patterns

6. Apply the learning phase as described on [14]. This phase is similar to the Linear Associator learning phase.

The learning phase involves two stages:

Stage 1 Calculate each one of the p associations (x^{μ}, y^{μ}) in order to find the matrix $y^{\mu} \cdot (x^{\mu})^t$ with dimentions $m \times n$.

$$y^{\mu} \cdot (x^{\mu})^t = \begin{pmatrix} y_1^{\mu} \\ y_2^{\mu} \\ \vdots \\ y_m^{\mu} \end{pmatrix} \cdot (x_1^{\mu}, x_2^{\mu}, ..., x_n^{\mu})$$

$$y^{\mu} \cdot (x^{\mu})^t = \begin{pmatrix} y_1^{\mu}x_1^{\mu} & \cdots & y_1^{\mu}x_j^{\mu} & \cdots & y_1^{\mu}x_n^{\mu} \\ y_2^{\mu}x_1^{\mu} & \cdots & y_2^{\mu}x_j^{\mu} & \cdots & y_2^{\mu}x_n^{\mu} \\ \vdots & & \vdots & & \vdots \\ y_i^{\mu}x_1^{\mu} & \cdots & y_i^{\mu}x_j^{\mu} & \cdots & y_i^{\mu}x_n^{\mu} \\ \vdots & & \vdots & & \vdots \\ y_m^{\mu}x_1^{\mu} & \cdots & y_m^{\mu}x_j^{\mu} & \cdots & y_m^{\mu}x_n^{\mu} \end{pmatrix}$$

Stage 2 Add the p matrices to obtain the memory M

$$M = \sum_{\mu=1}^{p} y^{\mu} \cdot (x^{\mu})^t = [m_{ij}]_{m \times n}$$

so that the ij-th component of memory M is expressed as follows:

$$m_{ij} = \sum_{\mu=1}^{p} y_i^{\mu} x_j^{\mu}$$

7. Move all the patterns to be classified to the new axis

$$x^{\mu'} = x^{\mu} - \bar{x}$$

With $x^{\mu'}$ as the displaced pattern
With x^{μ} as the original pattern
With $\bar{x}$ as the medium vector for all input patterns

8. Apply the recalling phase as in [14]. This phase is similarly to the Lernmatrix recalling phase.

The recalling phase is devoted to find the class that belongs to an input vector $x^\omega \in A^n$ given. Find the class means to obtain the components from the vector $y^\omega \in A^p$ that belongs to the pattern x^ω ; with the construction method of the vectors y^μ, class label should be obtained without ambiguity.

The i-th coordinate y_i^ω from the vector with the class $y^\omega \in A^p$ is obtained as shown in the next expression, where $\bigvee$ is the maximum operator

$$y_i^\omega = \begin{cases} 0 & if \quad \sum_{j=1}^{n} m_{ij} \cdot x_j^\omega = \bigvee_{h=1}^{p} \left[\sum_{j=1}^{n} m_{hj} \cdot x_j^\omega \right] \\ 1 & \text{other case} \end{cases}$$

This memory is being improved nowadays by combining CHAT memory with different coding techniques, like in [15]. This new form of CHAT memory can improve the classification accuracy.

4 Herbicide Dataset

The dataset used along experimental phase is an extraction from the Dictionary of Agrochemical Specialities (DEAQ). The DEAQ can be consulted online in [16], the main characteristics of 40 herbicides were distributed in 3473 instances. The features are: toxicity, action method, active ingredient, incompatibility, reentry period, counter indications and first aid.

In the dataset the patterns are structured as follows:

{"weed", "crop", "toxicity", “reentry period ", "herbicide"}

Where "weed", "crop", "toxicity" and "reentry period" are the principal attributes to be analyzed and "herbicide" is the most suitable option. The dataset contains 68 weeds, 50 crops, 2 kinds of toxicity, and 6 different reentry periods.

5 Experimental Phase

The fundamental set made from 40 herbicides distributed in 3473 instances was taken on all the experiments. Two experiments were performed with the associative models. The experimental procedure is as follows: first a k-fold cross validation classification was performed with CHAT Associative Memory in order to test the recovery rate over the fundamental set, then the learning model was changed to Alpha-Beta Associative Memory where recovery rate was evaluated.

In order to determine the classification performance of other existing models in the literature, a set of experiments were performed with "WEKA 3: Data Mining Software in Java" [17]. WEKA is an open source application under the GNU General Public License, free available on internet. On all the tests with WEKA the same conditions and validation schemes were aplied. The algorithms

were trained with all the fundamental set and then the fundamental set was recovered. In Table 2 all the resullts are compared. In the following paragraph the experiments with associative models are explained.

The first approach was to use CHAT Associative Memory. This associative model is known to be a good choice to classify patterns in bi-class problems, this has been tested by classifying each of the products into one of the two possible classes: "is the product" or "is not the product", this experimental approach was performed using a different memory for each product. CHAT Associative Memory was created only with all the training patterns from a specific product, for the "not product" examples, the pattern that doesn't have the product was taken, until it complete the same number of correct patterns. This experiment gave good results, with a k-fold cross validation with k=10, classification efficiency from all the products was 80.27%.

The next model was a cascade model made with a set of CHAT Associative Memories, each CHAT Associative Memory was trained with a set of products and "not products". In this approach the memories were organized from most efficient to the less efficient, this in order to avoid error propagation in the model, then CHAT Associative Memories were trained with the crop feature, so now there was a set of memories that only would be used with each different crop, using this model. This approach was performed in order to know how much the new model can recover from the learned phase, the result was a 21.01% . This implies that this model can recover from its fundamental set, however as the goal was to select a product, with this result this cannot be done in a competitive way even if the user gave a pattern contained in the fundamental set.

This forced us to seek for a more efficient solution, which was to change the classification mode to Alpha-Beta Associative Memories. First all the patterns were codified with The Johnson-Möbius code to improve the efficiency from Alpha-Beta Associative Memories[5], then the autoassociative Alpha-Beta Asociative Memories were selected, due to the fact that autoassociative Alpha-Beta Asociative Memories can recover entirely the fundamental set learned, classification efficiency reached 100% . This implies that the 3473 instances were correctly classified.

6 Experimental Results

Although WEKA 3: Data Mining Software in Java[17] has more than seventy well known algorithms implemented, only the two best-performing algorithms from each category were considered for comparison purposes. According to the type of learning scheme, each of these can be grouped in one of the following types of classifiers: Bayesian classifiers, Functions based classifiers, Meta classifiers, Lazy based classifiers and Decision Trees classifiers.

Decision trees algorithms can classify the fundamental set with an averange of 71.14%, the bayesian algorithms could classify the fundamental set with a BayesNet but not with a NaiveBayes, the best category for classify the fundamental set was lazy, with an average of 72.27%, the function based classifiers are

Table 2. Eficiency recovering fundamental set.

Algorithm	Instances correctly classified	Percentage
Trees		
RandomForest	2505	72.13%
NBTree	2437	70.16%
Bayesian		
BayesNet	2484	71.52%
NaiveBayes	1637	47.13%
Functions		
MultilayerPercerpron	2026	58.33%
Logistic	1897	54.62%
Lazy		
IB1	2510	72.27%
IBK	2510	72.27%
Meta		
Bagging	2444	70.37%
AttributeSelectedClassifier	2361	67.98%
Associative Memories		
Alpha-Beta	**3473**	**100%**
CHAT	730	21.01%

not good enough to classify the fundamental set with an averange of 56.47%. On the associative models CHAT Associative Memory was not the best solution even though it was mixed in different models trying to improve the classification rate, but best of all algorithms was the Alpha-Beta Associative Memories with a 100% of correct classification from the fundamental set. This imply that everything reminds learned on the Alpha-Beta Associative Memories.

7 Conclusions and Ongoing Research

Alpha-Beta Associative Memories can recall the entire fundamental set, this give us a big opportunity to develop a wide range of applications. In this paper the Alpha-Beta Associative Memory and CHAT Associative Memory were compared, in this particular case, an agronomic application was created to make recommendations about a herbicide. The best memory to use in this agronomic application was the Alpha-Beta memory, due its characteristics. On each intelligent application developed an important point to consider is the classification performance. Finally this paper shows that a problem in an agricultural context can be solved using a specific characteristic of a classification technique, in this case Alpha-Beta Associative memories.

Currently, we are investigating how to make better associative memories using feature selection in order to include them in more fields of science.

References

1. Sahoo, G., Ray, C., Wade, H.: Pesticide prediction in ground water in north carolina domestic wells using artificial neural networks. Ecological Modelling **183** (2005) 29 – 46
2. Hashimoto, Y.: Applications of artificial neural networks and genetic algorithms to agricultural systems. Computers and Electronics in Agriculture **18** (1997) 71 – 72 Applications of Artificial Neural Networks and Genetic Algorithms to Agricultural Systems.
3. Yang, C.C., Prasher, S.O., Landry, J.A., Ramaswamy, H.S.: Development of a herbicide application map using artificial neural networks and fuzzy logic. Agricultural Systems **76** (2003) 561 – 574
4. Burks, T., Shearer, S., Heath, J., Donohue, K.: Evaluation of neural-network classifiers for weed species discrimination. Biosystems Engineering **91** (2005) 293 – 304
5. Cornelio Yañez Edgardo, Felipe-Riveron, I.L.Y., Flores-Carapia, R.: A novel approach to automatic color matching. In Martinez-Trinidad, J., ed.: CIARP 2006. LNCS (2006) 529–538
6. Yáñez-Márquez, C., Cruz-Meza, M.E., Sánchez-Garfias, F.A., López-Yáñez, I.: Using alpha-beta associative memories to learn and recall RGB images. In: Advances in Neural Networks - ISNN 2007, 4th International Symposium on Neural Networks, ISNN 2007, Nanjing, China, June 3-7, 2007, Proceedings, Part III. (2007) 828–833
7. Román-Godínez, I., López-Yáñez, I., Yáñez-Márquez, C.: Classifying patterns in bioinformatics databases by using alpha-beta associative memories. In: Biomedical Data and Applications. Springer-Verlag Berlin Heidelberg (2009) 187–210
8. Aldape-Pérez, M., Román-Godínez, I., Camacho-Nieto, O.: Thresholded learning matrix for efficient pattern recalling. In: CIARP '08: Proceedings of the 13th Iberoamerican congress on Pattern Recognition, Berlin, Heidelberg, Springer-Verlag (2008) 445–452
9. Acevedo-Mosqueda, Yañez-Marquez, L.Y.: Alpha. beta bidirectional associative memories: theory and applications. Neural Processing Letters (2007)
10. JJ, H.: Neural networks and physical systems with emergent collective computational abilities. Proc Nat AcadSci (1982)
11. Kohonen, T.: Correlation matrix memories. IEEE Transactions on Computers (1972)
12. Acevedo-Mosqueda, Yañez-Marquez, L.Y.: A new model of bam: Alpha-beta bidirectional associative memories. Lecture Notes in Computer Science (LNCS) (2006)
13. Steinbuch, K., Frank, H.: Nichtdigitale lernmatrizen als perzeptoren. Biological Cybernetics **1** (1961) 117–124
14. Montero, R.S.: Clasificador hibrido de patrones basado en la lernmatrix de steinbuch y el linear associator de anderson-kohonen. Master's thesis, Instituto Politecnico Nacional Centro de Investigacion en Computacion (2003)
15. Uriarte-Arcia AV, López-Yáñez I, Y.M.C.: One-hot vector hybrid associative classifier for medical data classification. PLoS ONE **9** (2014) e95715
16. PLM: Diccionario de especialidades agroquimicas [online]. http://www.elcamporadio.com/source/ (2014)
17. 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** (2009) 10–18

Simulation & Modeling

Álgebra Geométrica Conforme para construir diagramas de Voronoi

Netz Romero, Ricardo Barrón Fernández

Centro de Investigación en Computación del Instituto Politécnico Nacional, México
jromero_a13@sagitario.cic.ipn.mx, rbarron@cic.ipn.mx

Resumen. Para dividir un espacio de acuerdo a las regiones de Voronoi, se han utilizado diversas técnicas, desde los métodos incrementales, divide y vencerás, de barrido entre los más importantes. En este trabajo se plantean los conceptos básicos del Álgebra Geométrica Conforme como herramienta matemática, para aplicarlos a una metodología propuesta donde definimos el concepto de "esfera hueca" y así construir los diagramas de Voronoi en forma incremental. La estructura de esferas huecas será la encargada de construir el diagrama y se relacionará con el empleo de los objetos geométricos que nos proporciona el Álgebra Geométrica Conforme. Dicha herramienta matemática nos muestra diversas ventajas frente al espacio euclidiano y dan una excelente solución al cómputo gráfico.

Palabras clave: Diagrama de Voronoi, Álgebra Geométrica Conforme, esfera hueca, objetos geométricos.

Abstract. A way to divide a space according to Voronoi regions, different techniques have been used since the incremental methods, divide and conquer, sweep line among the most important. In this paper we use the basic concepts of Conformal Geometric Algebra as mathematical tool, to apply a proposed methodology where we define a new concept called "hollow sphere" and build Voronoi diagrams incrementally. The structure of hollow spheres will be responsible for building the diagram and relate to the employment of the geometric objects that provides the Conformal Geometric Algebra. Such mathematical tool shows several advantages over the Euclidean space and provide an excellent solution to the graphic computer.

Keywords: Voronoi diagram, Conformal Geometric Algebra, hollow spheres, geometric objects.

1 Introducción

Los diagramas de Voronoi son las estructuras que nombra así el matemático Georgy Voronoi en 1908. Desde finales del siglo XIX los diagramas de Voronoi han realizado significativa influencia en aplicaciones como áreas de matemáticas, físicas, geología, meteorología, ecología, anatomía, astronomía por mencionar las principales áreas de interés. En las últimas décadas, gracias a los aportes de los avances en tecnología se

Advances in Computer Science
Research in computing Science 83, 2014 pp. 105-116

han desarrollado nuevas áreas de investigación en los que han realizado aplicaciones de los diagramas de Voronoi [1]. Antes de que se establecieran los conceptos matemáticos de los diagramas de Voronoi, en 1844 Hermann Grassmann publica su obra "Teorema de la Extensión" donde expresa su visión matemática para la geometría, pero su trabajo no fue reconocido como debería ser en su tiempo. William K. Clifford combinó las ideas del producto exterior de Grassmann y los cuaterniones de Hamilton. Aun así sus contemporáneos no supieron utilizar de manera adecuada las herramientas propuestas por Clifford y utilizaron el álgebra vectorial de Josiah W. Gibbs y Oliver Heaviside. Sin embargo no fue hasta los años de 1960 cuando David Hestenes [2] quien buscaba mejores herramientas matemáticas para las ciencias físicas (mecánica cuántica y relatividad) fue quien reconoció la importancia del álgebra geométrica. A partir de ahí se generaron trabajos de investigación en diversas áreas como las matemáticas, la física, aplicaciones para la ingeniería y las ciencias computacionales [3].

En este trabajo se plantearán los principios y conceptos para poder construir el diagrama de Voronoi implementando el uso del Álgebra Geométrica Conforme (AGC). El AGC se visualiza como una alternativa del espacio euclidiano para dar solución a problemas de computación gráfica, y por sus propiedades nos permite manejar intuitivamente objetos geométricos y operaciones entre ellos. Para aprovechar esta herramienta, presentaremos también en este trabajo una novedosa técnica para construir las regiones de Voronoi, donde se utiliza una esfera como estructura principal para su construcción. De esta manera se asocia el cuerpo geométrico de una esfera a la representación de una esfera en el AGC como una de sus primitivas figuras básicas.

2 Diagramas de Voronoi utilizando el concepto de esfera hueca

2.1 Diagramas de Voronoi

El diagrama de Voronoi consiste en dividir el espacio en regiones, donde cada región está relacionada a un sitio de un conjunto finito de ellos, de tal manera que cada región del espacio está constituida por los puntos más cercanos a él que a cualquier otro. El diagrama de Voronoi en un espacio de dos dimensiones está formados por aristas, vértices y regiones. Una arista está representada como un bisector perpendicular de dos sitios más cercanos, con la condición de que la arista pase por el centro de un círculo si éste contiene a dos sitios cercanos. Un vértice puede ser representado como el centro de un círculo definido por tres sitios. Y las regiones acotadas del diagrama tienen forma de polígonos convexos. Ver la figura 1 para comprender las propiedades del diagrama de Voronoi.

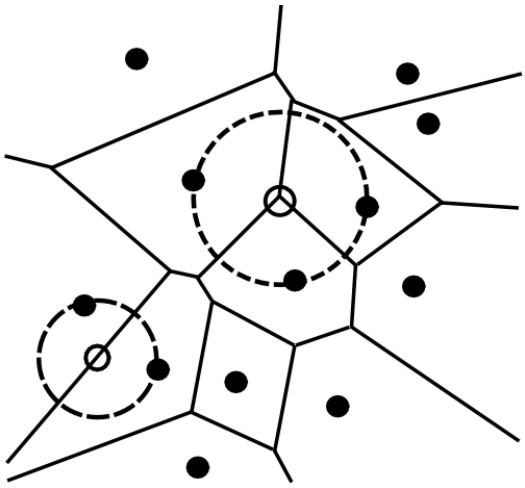


Fig. 1. Diagrama de Voronoi en dos dimensiones.

Tomaremos la siguiente notación para definir el diagrama de Voronoi de primer orden. La distancia euclidiana entre dos puntos p y q se denota como $dist(p,q)$. Sea $P = \{p_1, p_2, \dots, p_N\}$ un conjunto N de sitios distintos en un espacio, definiremos la región de Voronoi como:

$$Vor(p_i) = \{x \in \mathbb{R}^2 \mid dist(x,p_i) < dist(x,p_j), \forall\, i \neq j\} \quad (1)$$

2.2 Propiedades de las esferas huecas

El concepto de esfera hueca es una idea propuesta para resolver la construcción del diagrama de Voronoi utilizando principalmente la esfera como cuerpo geométrico, debemos de aclarar que las esferas denotan un cuerpo en tres dimensiones, pero las siguientes explicaciones e ilustraciones se tomarán en cuenta en un plano. Por lo que la esfera será acotada a un círculo. El término n-esfera (donde n representa el número de dimensiones) corresponde a una esfera en un espacio euclídeo de cualquier dimensión, donde su centro está ubicado en un punto c y posee un radio r, por lo que:

$$\mathbb{S}^n = \{x \in \mathbb{R}^{n+1} \mid dist(x,c) = r\} \quad (2)$$

Como el grado del diagrama de Voronoi es de tres y está en un espacio con $n = 2$, se establece la cantidad de tres sitios para construir una esfera S, donde $S \in \mathbb{S}^1$ (círculo), siempre que los tres sitios no sean colineales. Estableceremos las siguientes propiedades en que una esfera hueca es una esfera S, que:

- En su interior no contiene ningún sitio de P.
- En su superficie contiene n + 1 sitios de P.
- Su centro representa un vértice de Voronoi.
- Llamaremos esfera frontera a aquella esfera que comparte n sitios.

2.3 Algoritmo de construcción

Antes de explicar la metodología que se va a utilizar para construir la estructura del diagrama, vamos a identificar dos casos particulares. El diagrama se irá generando conforme se vayan agregando sitios al plano o espacio, y estos sitios se agregarán de manera secuencial uno por uno. Como primer paso se configuran tres sitios y a partir de ellos se genera una esfera inicial que cuenta con un polígono interno, entonces tenemos:

- Caso 1. a) Cuando un sitio cae dentro de la esfera hueca y dentro del polígono interno, ver figura 2a. b) Cuando un sitio cae dentro de la esfera hueca pero fuera del polígono interno, ver figura 2b.
- Caso 2. Cuando un sitio cae fuera de la esfera hueca, ver figura 2c.

Con la esfera inicial, estableceremos solamente el caso 2a, esto para evitar varios escenarios que dificulten la creación de esferas huecas. Además de que la esfera inicial, con este caso, se crean las aristas que están determinadas por un solo vértice.

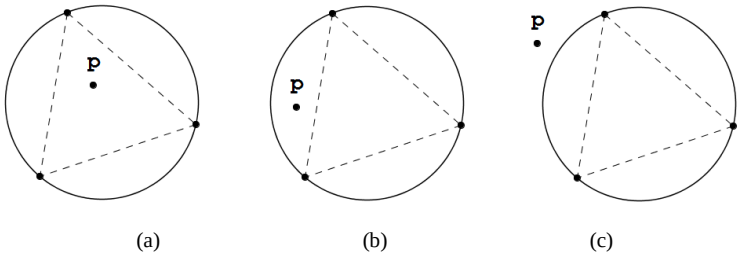


Fig. 2. En esta figura se muestra la clasificación de un nuevo sitio.

Empezamos con un diagrama de Voronoi que se construye con las propiedades de las esferas huecas (figura 3a), como se observa no es necesaria la generación de una triangulación Delaunay pero sí el principio planteado por Deloné [4]. A continuación se agrega un nuevo sitio (llamado Q) a la estructura inicial, lo importante es identificar las esferas en donde cae el nuevo sitio. En el ejemplo, se observa que son tres esferas huecas (ver figura 3b) y se reconocen los sitios que construyen las esferas afectadas (puntos A, C, F, E y D), ver figura 3c.

El procedimiento para detectar si un nuevo punto está dentro de las esferas huecas afectadas es similar al empleado por el algoritmo de Bowyer [5] y Watson [6], pero con la diferencia de que ellos trabajan con la malla triangular y detectan los círculos afectados para reconstruir la triangulación Delaunay. Las esferas seleccionadas se eliminan de la estructura del diagrama y los puntos afectados se ordenan alrededor del nuevo sitio introducido, en la figura 3c se muestran dichos puntos a través de un polígono que encierra el sitio Q, estos puntos concuerdan con los vecinos naturales explicados por Sibson [7]. Después se procede a generar las nuevas esferas huecas con estos puntos, pero de manera ordenada, por ejemplo se traza la primera esfera con los puntos frontera A y C junto con el nuevo punto agregado Q, ver figura 3d. De la misma manera se continua con los puntos C, F y Q después con F, E y Q y así sucesivamente. Con los

centros de las recién creadas esferas se calcula la nueva región, la cual se muestra sombreada en la figura 3e. Finalmente se reconstruye el diagrama con su nueva región generada por el nuevo sitio, ver figura 3f.

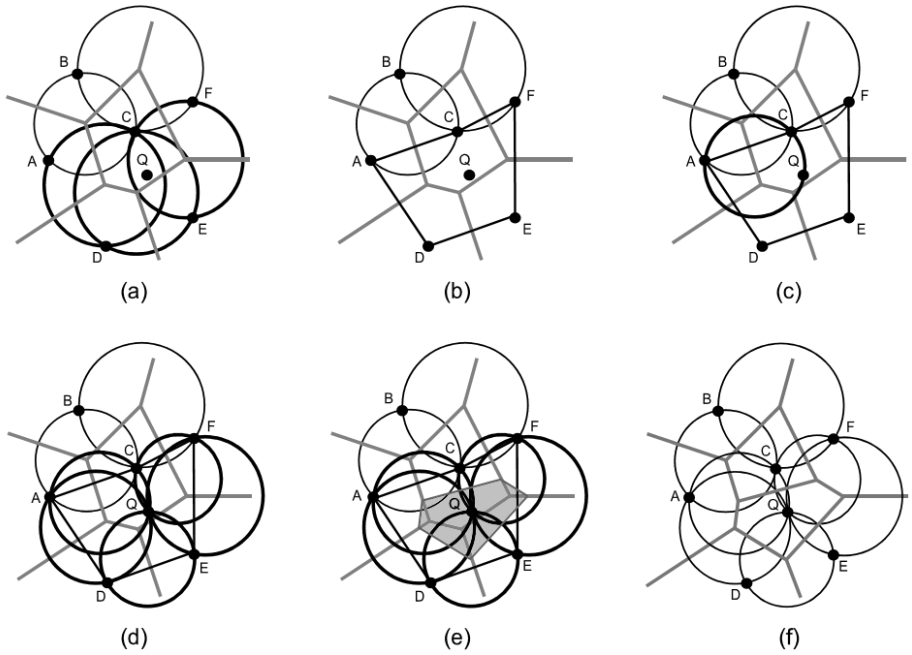


Fig. 3. Metodología para la construcción del diagrama de Voronoi empleando la esfera hueca.

3 Álgebra Geométrica Conforme

El modelo conforme se presenta como una alternativa al espacio euclidiano para resolver problemas geométricos, sin dejar todas las propiedades características del espacio euclidiano. Si deseamos trabajar con objetos geométricos en 3D, tendremos que utilizar un espacio euclidiano en $\mathbb{R}^3$, por lo que el espacio conforme requiere de cinco dimensiones. Una razón de este cambio es poder formular los problemas más intuitivamente en un espacio con dos dimensiones de más. El AGC puede utilizar tres vectores euclidianos básicos e_1, e_2 y e_3 y cuenta con dos adicionales e_0 y e_∞, donde el primero representa el origen euclidiano y el segundo representa al infinito. Las entidades básicas formadas por el álgebra geométrica se llaman blades (segmentos de subespacios vectoriales), y los 32 blades que se llegan a formar en las cinco dimensiones del AGC se muestra en la tabla 1.

Tabla 1. Lista de los 32 blades que se usan en el AGC.

Grado	Término	Blades	No.
0	scalar	1	1
1	vector	$e_1, e_2, e_3, e_0, e_\infty$	5
2	bivector	$e_1 \wedge e_2$, $e_1 \wedge e_3$, $e_2 \wedge e_3$, $e_1 \wedge e_\infty$, $e_2 \wedge e_\infty$, $e_3 \wedge e_\infty$, $e_1 \wedge e_0$, $e_2 \wedge e_0$, $e_3 \wedge e_0$, $e_0 \wedge e_\infty$	10
3	trivector	$e_1 \wedge e_2 \wedge e_3$, $e_1 \wedge e_2 \wedge e_0$, $e_1 \wedge e_2 \wedge e_\infty$, $e_1 \wedge e_3 \wedge e_0$, $e_1 \wedge e_3 \wedge e_\infty$, $e_1 \wedge e_0 \wedge e_\infty$, $e_2 \wedge e_3 \wedge e_0$, $e_2 \wedge e_3 \wedge e_\infty$, $e_2 \wedge e_0 \wedge e_\infty$, $e_3 \wedge e_0 \wedge e_\infty$	10
4	quadvector	$e_1 \wedge e_2 \wedge e_3 \wedge e_\infty$, $e_1 \wedge e_2 \wedge e_3 \wedge e_0$, $e_1 \wedge e_2 \wedge e_0 \wedge e_\infty$, $e_1 \wedge e_3 \wedge e_0 \wedge e_\infty$, $e_2 \wedge e_3 \wedge e_0 \wedge e_\infty$	5
5	pseudoscalar	$e_1 \wedge e_2 \wedge e_3 \wedge e_0 \wedge e_\infty$	1

Una vez que están definidos los elementos en este espacio, el AGC provee una gran variedad de objetos geométricos básicos que tienen dos representaciones la IPNS (Inner Product Null Space) y la OPNS (Outer Product Null Space) (ver referencia [8]), la lista se observa en la tabla 2 y éstas representaciones son duales una de la otra.

Tabla 2. Lista de primitivas geométricas del AGC.

Entidad	Representación IPNS	Representación OPNS
Punto	$P = \mathbf{x} + \frac{1}{2}\mathbf{x}^2 e_\infty + e_0$	
Esfera	$S = P - \frac{1}{2}r^2 e_\infty$	$S^* = P_1 \wedge P_2 \wedge P_3 \wedge P_4$
Plano	$\pi = \mathbf{n} + d e_\infty$	$\pi^* = P_1 \wedge P_2 \wedge P_3 \wedge e_\infty$
Círculo	$Z = S_1 \wedge S_2$	$Z^* = P_1 \wedge P_2 \wedge P_3$
Linea	$L = \pi_1 \wedge \pi_2$	$L^* = P_1 \wedge P_2 \wedge e_\infty$
Par de Puntos	$Pp = S_1 \wedge S_2 \wedge S_3$	$Pp^* = P_1 \wedge P_2$

Tenemos que $\boldsymbol{x}$ y $\boldsymbol{n}$ representan entidades en n dimensiones, por ejemplo $\boldsymbol{x}$ se obtiene por la siguiente combinación básica de vectores:

$$\boldsymbol{x} = x_1\, e_1 + \, x_2\, e_2 + \cdots + \, x_n\, e_n \tag{3}$$

4 AGC para construir el diagrama de Voronoi

4.1 Representación de objetos geométricos en AGC

Un **punto**, para representarlo en el espacio conforme, se toma el punto original $\boldsymbol{x} \in \mathbb{R}^n$ el cual tiene que ser extendido a un espacio de $n + 2$ dimensiones, y se denota como:

$$P = \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^2 e_\infty + e_0 \tag{4}$$

donde $\boldsymbol{x}^2$ es la norma cuadrada o el producto escalar de

$$\boldsymbol{x}^2 = x_1^2 + \, x_2^2 + \cdots + x_n^2 \tag{5}$$

Para una **esfera**, en la forma IPNS, se representa con un punto P en el centro de la esfera y el radio r, queda de la siguiente forma:

$$S = P - \frac{1}{2}r^2 e_\infty \tag{6}$$

Si el punto P lo empleamos como la ecuación 4, tenemos:

$$S = \boldsymbol{x} + \frac{1}{2}(\boldsymbol{x}^2 - r^2)e_\infty + e_0 \tag{7}$$

4.2 Operaciones necesarias del AGC para construir el diagrama de Voronoi

En la explicación de la metodología de los diagramas de Voronoi, las esferas huecas fueron presentadas como círculos, esto debido al ejemplo de trabajar en un espacio de dos dimensiones. Sin embargo el término esfera sigue quedando en pie para trabajar en dimensiones iguales o mayores a dos. A continuación mencionaremos las principales operaciones necesarias para construir los diagramas de Voronoi y los ejemplos serán para un espacio de 3D. Se explicarán brevemente algunos conceptos matemáticos, dejando solamente las operaciones que son necesarias, para una mayor referencia a esta herramienta matemática se pueden checar las referencias [9] [10].

Distancia entre dos puntos

Esta es una operación básica e indispensable para muchos de los cálculos que se realizan en la metodología propuesta, principalmente para encontrar distancias entre los puntos y determinar cuáles son los puntos que van a pertenecer a una esfera. Tomaremos dos puntos P y Q y efectuando el producto punto entre los vectores es:

$$P \cdot Q = \left(\boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^2 e_\infty + e_0\right) \cdot \left(\boldsymbol{q} + \frac{1}{2}\boldsymbol{q}^2 e_\infty + e_0\right) = \boldsymbol{p} \cdot \boldsymbol{q} - \frac{1}{2}\boldsymbol{p}^2 - \frac{1}{2}\boldsymbol{q}^2$$

Si $\boldsymbol{p}$ y $\boldsymbol{q}$ tienen la forma de la ecuación 3, y tomamos los tres vectores básicos, tenemos:

$$P \cdot Q = -\frac{1}{2}(\boldsymbol{p} - \boldsymbol{q})^2 \tag{8}$$

El resultado del producto punto de dos puntos (ecuación 8) multiplicado por -2, se identifica con el cuadrado de la distancia Euclidiana.

$$-2(P \cdot Q) = (\boldsymbol{p} - \boldsymbol{q})^2 \tag{9}$$

Ajuste de una esfera a través de puntos

En la metodología es importante identificar a las esferas huecas, conforme se inserten los puntos en el espacio, por lo que es necesario calcular el centro y radio como característica de cada esfera. Por ejemplo en la figura 4 se aprecia una esfera compuesta por cuatro puntos. Necesitamos de cuatro puntos para obtener las características de una esfera, sean esos cuatro puntos $\boldsymbol{p}_i \in \mathbb{R}^3$, $i \in \{1, 2, 3, 4\}$ y representaremos a un punto como la ecuación 4 y a una esfera como la ecuación 6. El producto interno nos provee una medida de distancia, y buscaremos un ajuste de mínimos cuadrados de:

$$\sum_{i=1}^{4}[\boldsymbol{P_i}\cdot\boldsymbol{S}]^2=\sum_{i=1}^{4}\left[\boldsymbol{p_i}\cdot\boldsymbol{s}-\frac{1}{2}\boldsymbol{p_i^2}-\frac{1}{2}(\boldsymbol{s^2}-r^2)\right]^2 \tag{10}$$

Para desear el mínimo de la ecuación 10, se deriva con respecto a los componentes s_1, s_2, s_3, y s_4 (donde $s_4=(\boldsymbol{s^2}-r^2)/2$) e igualando a cero, nos proporciona un sistema de cuatro ecuaciones y cuatro incógnitas, cuyo sistema es:

$$\begin{pmatrix}\sum_{i=1}^{4}p_{i,1}p_{i,1} & \sum_{i=1}^{4}p_{i,2}p_{i,1} & \sum_{i=1}^{4}p_{i,3}p_{i,1} & -\sum_{i=1}^{4}p_{i,1}\\ \sum_{i=1}^{4}p_{i,1}p_{i,2} & \sum_{i=1}^{4}p_{i,2}p_{i,2} & \sum_{i=1}^{4}p_{i,3}p_{i,2} & -\sum_{i=1}^{4}p_{i,2}\\ \sum_{i=1}^{4}p_{i,1}p_{i,3} & \sum_{i=1}^{4}p_{i,2}p_{i,3} & \sum_{i=1}^{4}p_{i,3}p_{i,3} & -\sum_{i=1}^{4}p_{i,3}\\ -\sum_{i=1}^{4}p_{i,1} & -\sum_{i=1}^{4}p_{i,2} & -\sum_{i=1}^{4}p_{i,3} & \sum_{i=1}^{4}1\end{pmatrix}\cdot s=\begin{pmatrix}\frac{1}{2}\sum_{i=1}^{4}\boldsymbol{p_i^2}p_{i,1}\\ \frac{1}{2}\sum_{i=1}^{4}\boldsymbol{p_i^2}p_{i,2}\\ \frac{1}{2}\sum_{i=1}^{4}\boldsymbol{p_i^2}p_{i,3}\\ \frac{1}{2}\sum_{i=1}^{4}\boldsymbol{p_i^2}\end{pmatrix} \tag{11}$$

El centro de la esfera corresponde a los valores de s_1, s_2 y s_3, mientras que el radio se determina de la siguiente forma:

$$r^2=\ s_1^2+\ s_2^2+s_3^2-2s_4 \tag{12}$$

Fig. 4. Con cuatro puntos podemos construir una esfera

En el caso de que sólo queramos construir círculos, como el ejemplo mostrado anteriormente en dos dimensiones, solamente quitamos un punto y con tres puntos se calcula el centro y radio de un círculo. Se modifica la ecuación 11 para trabajar con tres puntos. A continuación se muestra un segmento de código en lenguaje de programación java que resuelve el ajuste de un círculo con tres puntos.

```
// Matriz que contiene los tres puntos
double pts[][] = {{p1.getXplano(), p1.getYplano()},
 {p2.getXplano(), p2.getYplano()},{p3.getXplano(), p3.getYplano()}};
// Iteración que calcula las sumatorias
for(int i = 0; i < 3; i++) {
       a[0][0] += pts[i][0] * pts[i][0];
       a[0][1] += pts[i][0] * pts[i][1];
       a[0][2] += -pts[i][0];
       a[1][0] += pts[i][0] * pts[i][1];
       a[1][1] += pts[i][1] * pts[i][1];
       a[1][2] += -pts[i][1];
       a[2][0] += -pts[i][0];
       a[2][1] += -pts[i][1];
       a[2][2] += 1
```

```
        d = Math.pow(pts[i][0],2) + Math.pow(pts[i][1],2);
        b[0] += 0.5 * d * pts[i][0];
        b[0] += 0.5 * d * pts[i][1];
        b[0] += 0.5 * d;  }
// Método que resuelve un Sistema de Ecuaciones Lineales
resuelveSEL(a, b, x, 3);
// Se obtiene el centro del círculo
xcent = x[0];
ycent = x[1];
// Se obtiene el radio del círculo
radio = Math.sqrt(x[0] * x[0] + x[1] * x[1] - 2 * x[2]);
```

Determinar si un punto está dentro de una esfera

Un punto importante, es identificar en cuáles esferas huecas tiene contenido el nuevo punto que ha sido insertado para generar una nueva región de Voronoi. En el AGC usaremos el producto punto entre un punto y una esfera para determinar si está dentro, en o fuera de la esfera. Sea un punto P y sea S una esfera, el producto punto de estos dos cuerpos geométricos es:

$$P \cdot S = \left(\boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^2 e_\infty + e_0\right) \cdot \left(\boldsymbol{s} + \frac{1}{2}(\boldsymbol{s}^2 - r^2)e_\infty + e_0\right)$$

$$P \cdot S = -\frac{1}{2}(\boldsymbol{s} - \boldsymbol{p})^2 + \frac{1}{2}r^2 \tag{13}$$

El resultado del producto punto de un punto con una esfera (ecuación 13) multiplicado por -2, se identifica con la potencia de un punto.

$$-2(P \cdot Q) = (\boldsymbol{s} - \boldsymbol{p})^2 - r^2 \tag{14}$$

De la ecuación 13 identificamos mediante las siguientes relaciones, la ubicación del punto con respecto a la esfera y en la figura 5 se observan los tres casos.

- Si $P \cdot S > 0$, P está dentro de la esfera, ver figura 5a.
- Si $P \cdot S = 0$, P está sobre de la esfera, ver figura 5b.
- Si $P \cdot S < 0$, P está fuera de la esfera, ver figura 5c.

En seguida se muestra un segmento de código en java para determinar si un punto cae dentro de un círculo o no.

(a) (b) (c)

Fig. 5**.** La operación nos ayuda a determinar si el punto está dentro, sobre o fuera de la esfera.

```
// Se realiza el producto punto de P y S
double ps = Math.pow(s.getRadio(),2) –
  ( Math.pow(c.getXplano(),2) + Math.pow(c.getYplano(),2) -
 2 * (s.getXcent() * p.getXplano() + s.getYcent() * p.getYplano()) +
 Math.pow(s.getXplano(),2) + Math.pow(s.getYplano(),2) );
// Se identifica la condición del producto punto
if( ps > 0 )  return 1;     // P está dentro del círculo
if( ps == 0 ) return 0;     // P está sobre el círculo
if( ps < 0 )  return -1;    // P está fuera del círculo
```

4.3 Implementación realizada y ventajas del AGC

Se ha desarrollado un programa que resuelve la metodología de las esferas huecas empleando las características del espacio euclidiano. Se utilizó el lenguaje de programación Java con bibliotecas gráficas de JOGL. Se probaron módulos que utilizan los conceptos del AGC, como los segmentos de código mostrados anteriormente, y los resultados fueron los mismos para la generación de las regiones de Voronoi. En la figura 6 se muestra un resultado de la aplicación gráfica del programa.

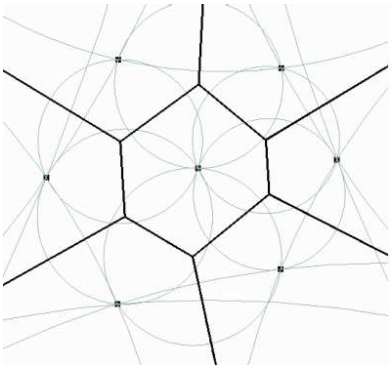


Fig. 6. Resultado de la interfaz gráfica del programa

Algunas de las características que benefician al AGC son:

- Unificación de diversos sistemas matemáticos, además de que se preservan ángulos y distancias.
- Manejo intuitivo y simple para manejar objetos geométricos, como puntos, líneas, planos, círculos y esferas.
- Manejo intuitivo de las operaciones entre los cuerpos geométricos.

5 Conclusiones y trabajo futuro

Este trabajo es una contribución fundamental donde se replantean las técnicas que existen para construir los diagramas de Voronoi, de tal manera que sea posible asociar cuerpos geométricos sencillos y poderosos. Dichos cuerpos geométricos formarán la estructura principal para crear las regiones de Voronoi en forma incremental. El AGC nos

proporciona un mundo en el cual los puntos y esferas se representan como vectores, además de operaciones entre ellos que facilitan sus aplicaciones y de esta forma se puede trabajar con una matemática uniforme y coherente que de salida a una representación gráfica.

Como trabajo a futuro, se necesita implementar totalmente las matemáticas del AGC para realizar todos los cálculos de las esferas huecas que nos permitan construir el diagrama de Voronoi. Como se mencionaba, se acoplaron módulos que resuelven aplicaciones específicas utilizando AGC, todo esto en una aplicación gráfica bajo el lenguaje de programación Java. También, se puede pensar en paralelizar los cálculos de las propiedades de las esferas huecas para obtener ventaja en los tiempos de procesamiento.

Agradecimientos

Para la elaboración de este trabajo se agradece el apoyo otorgando por Instituto Politécnico Nacional, a través del proyecto SIP20141047 y al programa BEIFI.

Referencias

1. Du, Q., Faber V. and Gunzburger, M..: Centroidal Voronoi Tessellations: Applications and Algorithms, SIAM Review, vol. 41, no. 4, 637-676 (1999)
2. Hestenes, D., and Garret E.: Clifford Algebra to Geometric Calculus. Kluwer: Springer Verlag (1984)
3. Bayro, E.: Handbook of Geometric Computing. Applications in Pattern Recognition, Computer Vision, Neurocomputing and Robotics, Springer (2005)
4. Deloné , B. N.: Sur la sphere vide, Bulletin of the Academy of Sciences of the U.S.S.R., no. 6, 793-800 (1934)
5. Bowyer, A.: Computing dirichlet tessalations, The Computer Journal, vol. 24, no. 2, 162-166 (1981)
6. Watson. D. Computing the n-dimensional delaunay tessalation with application to voronoi polytopes, The Computer Journal, vol. 24, no. 2, 167-172 (1981)
7. Green, P. J., Sibson, R. R.: Computing Dirichlet tessellations in the plane. Comput. J., 21:168–173 (1978)
8. Hitzer, E.: Euclidean Geometric Objects in the Clifford Geometric Algebra of Origin, 3-Space, Infinity Bulletin of the Belgian Mathematical Society. (2004)
9. Hilderbrand, D.: Foundations of Geometric Algebra Computing, Springer (2013)
10. Vince, J.: Geometric Algebra for Computing Graphics, Springer (2008)

AC 2-dimensional como Modelo de una Membrana Vibrante

I. Huerta-Trujillo[1], E. Castillo-Montiel[1], J.C. Chimal-Eguía[1], N. Sanchez-Salas[2] and J.A. Martinez-Nuño[3]

[1] Laboratorio de Simulación y Modelado
Centro de Investigación en Computación, Instituto Politécnico Nacional,
Av. Juan de Dios Bátiz, Esq. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo,Del. Gustavo A. Madero, D. F., C. P. 07738, México..
[2] Sección de Estudios de Posgrado
Escuela Superior de Física y Matemáticas, Instituto Politécnico Nacional,
Av. Instituto Politécnico Nacional Edificio 9, Unidad Profesional Adolfo López Mateos, Col. Zacatenco. Del. Gustavo A. Madero. D. F., C. P. 07738, México.
[3] Sección de Estudios de Posgrado
Escuela Superior de Cómputo, Instituto Politécnico Nacional,
Av. Juan de Dios Bátiz, Esq. Miguel Othón de Mendizábal, Col. Lindavista. Del. Gustavo A. Madero. D. F., C. P. 07738, México.

Abstract. En este trabajo se propone un modelo de Autómata Celular (AC) 2-dimensional y el proceso de obtención de la regla de evolución, el modelo obtenido se compara contra la solución analítica de una ecuación diferencial parcial hiperbólica de dos dimensiones, lineal y homogénea, la cual modela una membrana vibrante con condiciones iniciales y de frontera específicas, se analiza el espectro de frecuencia así como el error entre los datos arrojados por el modelo de AC contra los datos proporcionados por la evaluación de la solución a la ecuación diferencial.

Keywords: Autómatas celulares, ecuaciónes diferenciales parciales, cómputo paralelo, membrana vibrante.

1 Introducción

Un autómata celular (AC), es un sistema dinámico discreto, que consiste en un arreglo de células (nodos) en alguna dimensión d [3]. Wolfram [9] los define como idealizaciones matemáticas de sistemas físicos, cuyo espacio y tiempo son discretos, en donde las cantidades físicas se pueden circunscribir a un conjunto finito de valores.

La definición de los AC lleva implícitamente asociado otros conceptos, como espacio y localidad de influencia. Se asume que el sistema representado está distribuido en el espacio y que regiones cercanas tienen mayor influencia entre si, que otras que se encuentren apartadas dentro del sistema [1].

Advances in Computer Science
Research in computing Science 83, 2014 pp. 117-132

1.1 Definición Formal de Autómata Celular

Definición 1 *Una "lattice" o "retícula" es un arreglo uniforme, generalmente infinito [9], formado por objetos idénticos llamados "células" [7]. Este arreglo puede ser n-dimensional, pero para efectos de simulación de sistemas naturales se implementa de* 1, 2 *ó* 3 *dimensiones, de tamaño finito.*

Definición 2 *Un AC es una 4-tupla* $AC = (L, S, V, \Phi)$ *donde:*

L: *Es una retícula regular y* $L = \{c \in \mathbb{C}^d\}$ *para una lattice d-dimensional.*
S: *Es el conjunto finito de todos los posibles estados de las células,* $c \in L$.
V: *Es el conjunto finito de células que definen la vecindad para una célula.*
Φ: $S^d \to S$, *es una función de transición aplicada simultaneamente a las células que conforman el lattice.*

La actualización en el estado de las células requiere que se conozca el estado de las células vecinas [2].

Definición 3 *Una vecindad para una célula* $c \in L$ *es* $V(c) = \{k_1, k_2, \cdots, k_n \mid k_j \in L, j = 0, 1, \ldots, n\}$*; es decir, una vecindad es un conjunto de células para las cuales la célula c es el punto de referencia para el área de influencia.*

2 Modelo clásico de Membrana Vibrante

La obtención de la ecuación de movimiento para una membrana [6] se basa en las suposiciones de que es delgada y uniforme con rigidez despreciable, que es perfectamente elástica, sin amortiguamiento y que vibra con desplazamientos de amplitud pequeños. La ecuación que regula las vibraciones transversales de una membrana esta dada por [4]:

$$\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} \tag{1}$$

donde $u(x,y,t)$ es la deflexión de la membrana, y $c^2 = \frac{T}{\rho}$, siendo ρ la densidad de masa de la membrana [8] y T la tensión de le membrana por unidad de longitud. La ecuación (1) se denomina *ecuación de onda en dos dimensiones*.

Se considera una membrana cuadrada (Fig:1), definimos las condiciones de frontera como sigue:

$$u(x,y,t) = 0 \text{ para } x = 0,\ x = b,\ y = 0,\ y = b \quad \forall\ t \tag{2}$$

Las condiciones iniciales se definen como:

$$u(x,y,t) = f(x,y) \tag{3}$$

$$\frac{\partial u(x,y,t)}{\partial t}\Big|_{t=0} = g(x,y) \tag{4}$$

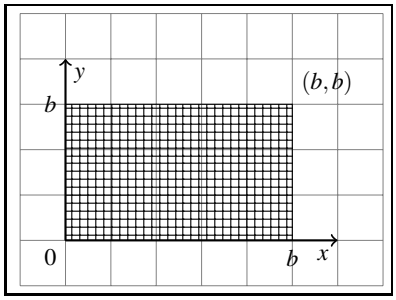


Fig. 1. Membrana rectangular de longitud $b \times b$.

donde $f(x,y)$ y $g(x,y)$ son el desplazamiento y la velocidad inicial de la membrana respectivamente.

Si definimos las condiciones iniciales como:

$$u(x,y,t) = xy(x-b)(y-b) \tag{5}$$

$$\frac{\partial u(x,y,t)}{\partial t}\Big|_{t=0} = 0 \tag{6}$$

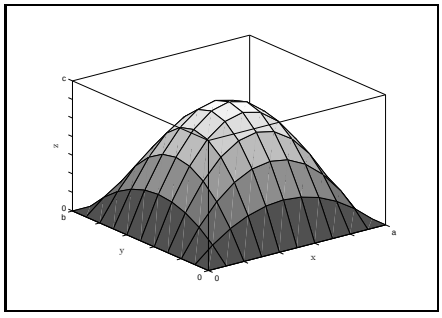


Fig. 2. Condiciones iniciales y de frontera del sistema de membrana.

Entonces la membrana inicia del reposo con una distribución espacial que puede verse en la gráfica de la figura 2.

La solución a la ecuación diferencial (1) con las condiciones iniciales y de frontera descritas anteriormente es:

$$u_{m,n}(x,y,t) = [G_{m,n}\cos(ck_{m,n}t) + H_{m,n}\operatorname{sen}(ck_{m,n}t)]\operatorname{sen}\left(\frac{m\pi x}{b}\right)\operatorname{sen}\left(\frac{n\pi y}{b}\right) \tag{7}$$

Donde $m = 1,2,3,\ldots$, $n = 1,2,3,\ldots$ y $k_{m,n} = \frac{\pi\sqrt{m^2+n^2}}{b}$.

La solución plasmada en la ec. (7) depende directamente de los valores que m y n tomen, luego entonces cada combinación de estas es una solución particular a (1). Por el principio de superposición la combinación lineal de las soluciones es también una solución a la ecuación diferencial luego entonces se tiene:

$$u(x,y,t) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} u_{m,n}(x,y,t) \tag{8}$$

Usando series de Fourier, la solución final es:

$$u(x,y,t) = \frac{64b^4}{\pi^6} \sum_{\substack{m=1 \\ m=\text{impar}}}^{\infty} \sum_{\substack{n=1 \\ n=\text{impar}}}^{\infty} \frac{1}{m^3 n^3} \cos\left(\frac{\pi\sqrt{m^2+n^2}}{b} ct\right) \text{sen}\left(\frac{m\pi x}{b}\right) \text{sen}\left(\frac{n\pi y}{b}\right) \tag{9}$$

3 Discretización del modelo analítico

3.1 Análisis previo

Supongamos que una membrana es una sucesión de puntos con masa específica unidos por resortes, cada punto de la membrana se encuentra unido a sus cuatro vecinos ortogonales, donde la masa de la membrana se encuentra distribuida en los puntos de unión y no en los resortes, y el borde de la membrana se encuentra sujeto a una superficie.

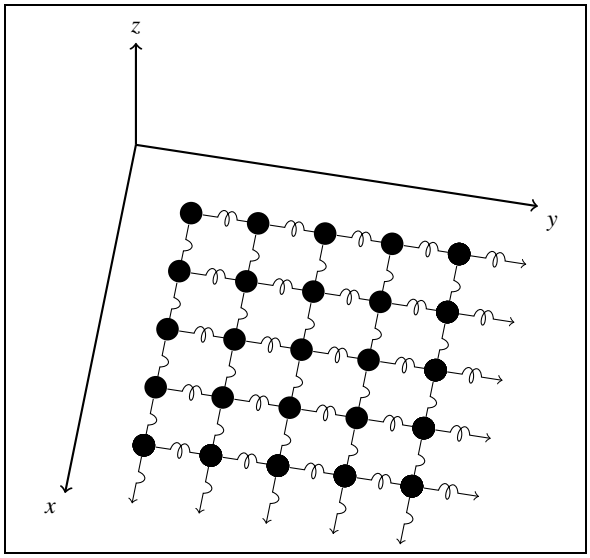


Fig. 3. Representación de una membrana como una sucesión de masas y resortes, cada masa esta unida a sus vecinos norte, sur, este y oeste.

Llamaremos d_e o distancia de equilibrio, a la longitud del resorte que une a dos masas y el cual se encuentra en estado de reposo, esa longitud será la que separe a las masas.

Para un sistema de membrana vibrante con condiciones iniciales de posición descrita en la ecuación (5) y partiendo del reposo tenemos que cada punto de unión de la membrana se encuentra sometido a cuatro fuerzas que actúan en dirección de cada uno de los vecinos ortogonales que llamaremos m_c, m_n, m_s, m_e, m_w para la partícula central, norte, sur, este y oeste (west) respectivamente. como se muestra en la figura

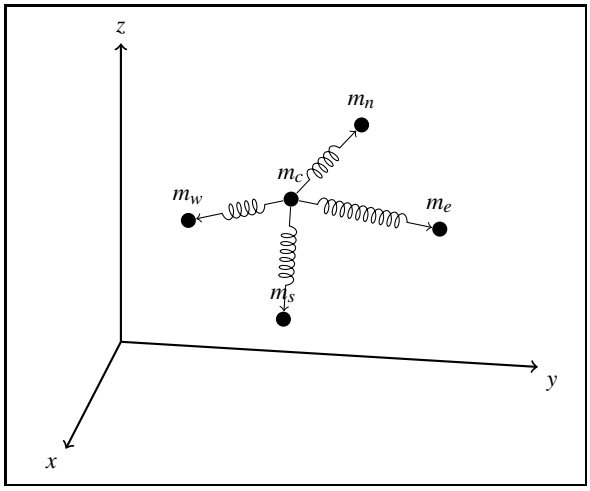


Fig. 4. Representación de la vecindad von Newmann del AC, la célula central y sus cuatro vecinos ortogonales, así como la dirección en la que ejercen la fuerza los vecinos.

Es necesario conocer la fuerza que cada vecino ejerce sobre m_c. Siguiendo el procedimiento definido por [5], procedemos a calcular para un vecino.

Tomando en cuenta la figura tenemos que:

$$\begin{array}{c} \overrightarrow{r_c} + \overrightarrow{\Delta r_w} = \overrightarrow{r_w} \\ \Rightarrow \overrightarrow{\Delta r_w} = \overrightarrow{r_w} - \overrightarrow{r_c} \end{array} \tag{10}$$

El vector $\overrightarrow{\Delta r_w}$ se puede expresar como el producto de un vector unitario por el módulo del mismo, entonces:

$$\overrightarrow{\Delta r_w} = |\overrightarrow{\Delta r_w}| \widehat{\Delta r_w}$$

Sabemos que $|\overrightarrow{\Delta r_w}|$, representa la distancia de separación de las masas, es posible escribir este escalar como la suma de la distancia de equilibrio o longitud del resorte incrementando la deformación sufrida por el mismo resorte debido al cambio de posición de la masa central, por lo que tenemos:

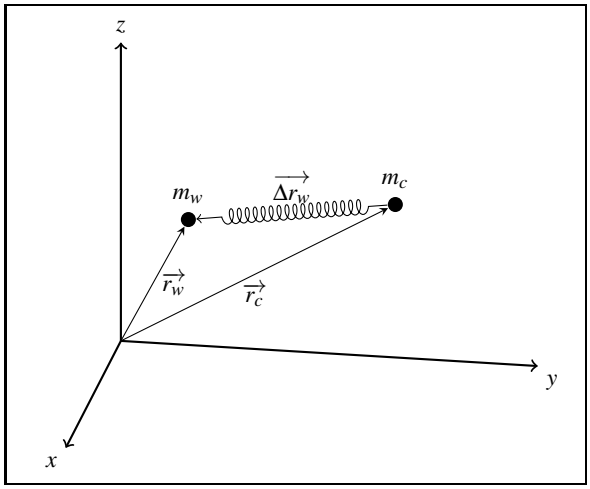


Fig. 5. Representación de los vectores asociados con la célula central, un vecino y la dirección de la fuerza ejercida por el resorte en dirección a la célula vecina.

$$|\overrightarrow{\Delta r_w}|\widehat{\Delta r_w} = (d_e + \Delta d_e)\widehat{\Delta r_w}$$

Donde Δd_e es el valor de la deformación del resorte, la cual es necesario calcular con el objeto de conocer el incremento de la fuerza desde el punto de equilibrio al punto de análisis, por lo que tenemos lo siguiente:

$$\begin{aligned} & |\overrightarrow{\Delta r_w}|\widehat{\Delta r_w} = (d_e + \Delta d_e)\widehat{\Delta r_w} \\ \Rightarrow\ & |\overrightarrow{\Delta r_w}|\widehat{\Delta r_w} = d_e\widehat{\Delta r_w} + \Delta d_e\widehat{\Delta r_w} \\ \Rightarrow\ & \Delta d_e\widehat{\Delta r_w} = |\overrightarrow{\Delta r_w}|\widehat{\Delta r_w} - d_e\widehat{\Delta r_w} \\ \Rightarrow\ & \Delta d_e\widehat{\Delta r_w} = \left(|\overrightarrow{\Delta r_w}| - d_e\right)\widehat{\Delta r_w} \end{aligned} \tag{11}$$

Dado que ambos elementos de la ecuación (11) son vectores, los representamos en sus componentes como sigue:

$$\Delta d_e\widehat{\Delta r_w} = (\Delta x_w, \Delta y_w, \Delta z_w) \tag{12}$$

$$\left(|\overrightarrow{\Delta r_w}| - d_e\right)\widehat{\Delta r_w} = \left(|\overrightarrow{\Delta r_w}| - d_e\right)\left(\frac{x_w - x_c}{|\overrightarrow{\Delta r_w}|}, \frac{y_w - y_c}{|\overrightarrow{\Delta r_w}|}, \frac{z_w - z_c}{|\overrightarrow{\Delta r_w}|}\right) \tag{13}$$

Igualando miembro a miembro de las ecuaciones vectoriales (12) y (13), tenemos:

$$\Delta x_w = \left(|\overrightarrow{\Delta r_w}| - d_e\right)\frac{x_w - x_c}{|\overrightarrow{\Delta r_w}|} \tag{14}$$

$$\Delta y_w = \left(|\overrightarrow{\Delta r_w}| - d_e\right) \frac{y_w - y_c}{|\overrightarrow{\Delta r_w}|} \tag{15}$$

$$\Delta z_w = \left(|\overrightarrow{\Delta r_w}| - d_e\right) \frac{z_w - z_c}{|\overrightarrow{\Delta r_w}|} \tag{16}$$

De esta forma obtenemos las componentes rectangulares Δx_w, Δy_w, y Δz_w que corresponden a los incrementos de desplazamiento en los ejes coordenados x, y y z para la masa m_c para $\overrightarrow{\Delta r_w}$, los valores de los componentes para los vectores $\overrightarrow{\Delta r_n}$, $\overrightarrow{\Delta r_s}$ y $\overrightarrow{\Delta r_e}$ se calculan de forma análoga.

Continuando con el análisis, por la *ley de Hooke* para un sistema de masa-resorte en una dimensión tenemos que:

para un sistema masa resorte en una dimensión, tenemos que:

$$F = -k\Delta x \tag{17}$$

En general, para la partícula m_c, existen tres fuerzas ejercidas por m_w en dirección de $\overrightarrow{\Delta r_w}$ debido a los componentes x, y y z del vector, y así para cada una de las partículas m_n, m_s y m_e en las direcciones $\overrightarrow{\Delta r_n}$, $\overrightarrow{\Delta r_s}$ y $\overrightarrow{\Delta r_e}$ respectivamente. Sustituyendo los valores encontrados en la ecuación (14) y los respectivos de los vecinos de m_c, en la ecuación (17) obtenemos:

$$F_x = -k_n\Delta x_n - k_s\Delta x_s - k_e\Delta x_e - k_w\Delta x_w$$

Suponiendo que los resortes que unen a las masas de la membrana son iguales, entonces $-k_n = -k_s = -k_e = -k_w == -k$ entonces:

$$F_x = -k(\Delta x_n + \Delta x_s + \Delta x_e + \Delta x_w) \tag{18}$$

Análogamente:

$$F_y = -k(\Delta y_n + \Delta y_s + \Delta y_e + \Delta y_w) \tag{19}$$

$$F_z = -k(\Delta z_n + \Delta z_s + \Delta z_e + \Delta z_w) \tag{20}$$

Siendo estas las fuerzas que actúan sobre m_c, que define su aceleración en el momento que m_c se encuentra oscilando. Dado que se conoce la fuerza, utilizando *la segunda ley de Newton* $\overrightarrow{F} = m\overrightarrow{a}$, y la ecuación de velocidad para un movimiento uniformemente acelerado, tenemos que la velocidad final para m_c está dada en función de su velocidad inicial y de la aceleración que siente la partícula en un instante de tiempo t, en consecuencia:

$$\begin{aligned} \overrightarrow{v_f} &= \overrightarrow{v_i} + \overrightarrow{a}t \\ &= \overrightarrow{v_i} + \frac{\overrightarrow{F}}{m}t \end{aligned}$$

Separando el vector de velocidad en sus componentes rectangulares y retomando los valores de fuerza de las ecuaciones (18), (19) y (20), se encuentran las componentes de velocidad para m_c siendo estos:

$$v_{fx} = v_{ix} + \frac{F_x}{m}t \tag{21}$$

$$v_{fy} = v_{iy} + \frac{F_y}{m}t \tag{22}$$

$$v_{fz} = v_{iz} + \frac{F_z}{m}t \tag{23}$$

Con esto obtenemos la velocidad de m_c transcurrido un tiempo t, esto permite realizar el cálculo de la nueva posición de la partícula para el mismo instante de tiempo, utilizando la ecuación de movimiento uniformemente acelerado, tenemos:

$$x_{fi} = x_i + v_{ix}t + \frac{1}{2}\frac{F_x}{m}t^2 \tag{24}$$

$$y_{fi} = y_i + v_{iy}t + \frac{1}{2}\frac{F_y}{m}t^2 \tag{25}$$

$$z_{fi} = z_i + v_{iz}t + \frac{1}{2}\frac{F_z}{m}t^2 \tag{26}$$

Las ecuaciones de velocidad y posición son las que se emplean en la defición de la función de evolución para el AC propuesto.

4 Modelo de membrana vibrante usando un AC 2-dimensional

Basado en el análisis desarrollado en la sección 3.1, se define el modelo de AC para un sistema de membrana vibrantes fijo en los extremos de una longitud $l \times l$ como: Una 4-tupla $AC = (L, S, V, \Phi)$ donde cada célula $m_c \in L$ está definida por su masa, posición inicial y su velocidad, cuando la membrana se encuentra en reposo, siendo:

L: Es una retícula regular 2-dimensional.

$$S: = \left\{ \begin{array}{l} \text{Célula fija} \\ \overrightarrow{P^t_{m_{i,j}}} : \text{vector de posición en tiempo } t \\ \overrightarrow{V^t_{m_{i,j}}} : \text{velocidad en el tiempo } t. \end{array} \right\} \forall m_{i,j} \in \mathbb{L}^2$$

V: $V = \{(m_n, m_s, m_c, m_e, m_w)\}$

Φ: $\mathbb{R}^3 \to \mathbb{R}^3$,

$$\Phi : \left\{ \begin{array}{l} \text{a)} \overrightarrow{P^{t+1}_{m_{cf}}} = \overrightarrow{P^t_{m_{ci}}} + \overrightarrow{Vi^t_{m_{ci}}} t + \dfrac{\sum_{v=1}^{4} \overrightarrow{F^t_{v_c}}}{2m} t^2 \\ \\ \text{b)} \overrightarrow{Vf^{t+1}_{m_c}} = \overrightarrow{Vi^t_{m_{cf}}} + \dfrac{\sum_{v=1}^{4} \overrightarrow{F^t_{v_c}}}{m} t \end{array} \right\}$$

donde : $\sum_{v=1}^{4} \overrightarrow{F^t_{v_c}}$ es la fuerza que los vecinos de m_c
ejercen sobre ella en el tiempo t.
$\overrightarrow{P^{t+1}_{m_{cf}}}$, es la posición final de la célula en el espacio, y
$\overrightarrow{Vf^{t+1}_{m_c}}$, es la velocidad final en el tiempo $t+1$.

La función de evolución Φ, está compuesta por dos reglas fundamentales, ambas reglas se aplican simultáneamente a todas las células que conforman el lattice.

La regla *a)* define la posición de la célula en el tiempo $t+1$, tomando la velocidad en el tiempo t, esta posición se actualiza, siendo la nueva posición inicial para $t+2$ y así sucesivamente. Similarmente, para *b)* la velocidad final para el tiempo $t+1$ se actualiza, siendo la velocidad inicial para el tiempo $t+2$.

Un punto importante en la definición del modelo es la constante de restitucón para los resortes empleada en la ecuación (17), a diferencia de otros modelos [5] en el cual la constante de restitución era relativamente simple de calcular, en este modelo nos enfrentamos a un problema, el que el arreglo de masas y resortes no presentan un patrón serial sino que presentan un arreglo de mallas, para obtener este valor procedemos de la siguiente manera:

Análogamente al método para reducción de mallas eléctricas, para obtener una constante de restitución general iniciamos definiendo como se pueden reducir arreglos de tipo "Delta" a "Estrella" y viceversa, tomando como base la figura 4.

Definición 4 (Transformación Delta a Estrella) *Definimos la tranfomación de un arreglo de resortes en patrón delta hacia un patrón estrella como:*

$$k_1 = \frac{k_a k_c}{k_a + k_b + k_c} \tag{27a}$$

$$k_2 = \frac{k_b k_c}{k_a + k_b + k_c} \tag{27b}$$

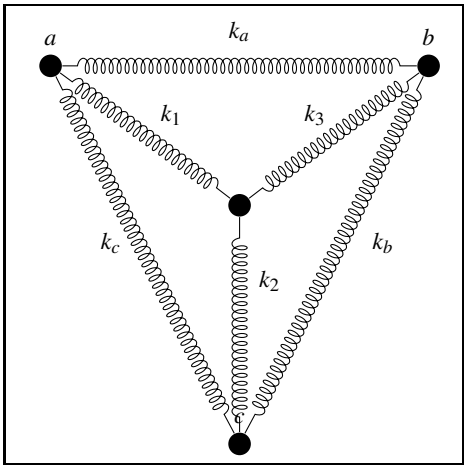


Fig. 6. Representación de un arreglo de resortes en forma de Delta-Estrella.

$$k_3 = \frac{k_a k_b}{k_a + k_b + k_c} \tag{27c}$$

Donde k_1, k_2, k_3 son las nuevas constantes de restitución que tendrán los resortes al realizar la transformación y k_a, k_b, k_c son las constantes de restitución que sirven como base para el cambio.

Definición 5 (Transformación Estrella a Delta) *Definimos la tranfomacón de un arreglo de resortes en patrón estrella hacia un patrón delta como:*

$$k_a = \frac{k_1 k_2 + k_1 k_3 + k_2 k_3}{k_2} \tag{28a}$$

$$k_b = \frac{k_1 k_2 + k_1 k_3 + k_2 k_3}{k_1} \tag{28b}$$

$$k_c = \frac{k_1 k_2 + k_1 k_3 + k_2 k_3}{k_3} \tag{28c}$$

Donde k_a, k_b, k_c son las nuevas constantes de restitución que tendrán los resortes al realizar la transformación y k_1, k_2, k_3 son las constantes de restitución que sirven como base para el cambio.

Es necesario saber como tratar los resortes que se encuentren conectados en serie o en paralelo, para esto definimos lo siguiente:

Definición 6 (Resortes conectados en serie) *Dado un grupo de resortes unidos en un patrón serie, definimos la constante de restitución total del arreglo como:*

$$\frac{1}{k_t} = \frac{1}{k_1} + \frac{1}{k_2} + \ldots + \frac{1}{k_n} \tag{29}$$

Donde k_t es la constante de restitución total del arreglo y $k_1, \ldots, k_n$ son las constantes de restitución de cada uno de los resortes que forman el arreglo.

Definición 7 (Resortes conectados en paralelo) *Dado un grupo de resortes unidos en un patrón paralelo, definimos la constante de restitución total del arreglo como la suma de las constantes de restitución de todos los resortes que formen el arreglo.*

4.1 Cálculo de la constante k

Se procede de la siguiente forma para calcular el valor de la constante de restitución k de los resortes que unirán a las masa correspondientes.

El proceso inicia suponiendo que la membrana tiene una constante de elasticidad con un valor igual a k_t. Si se inicia con una membrana representada por el AC propuesto conformada por 2×2 nodos, luego entonces siguiendo la reducción para resortes conectados en serie y en paralelo obtenemos que:

$$k = k_t$$

Siguiendo el proceso para un AC de 3×3 nodos, encontramos que:

$$k = \frac{3k_t}{2}$$

Si se continua podemos ver que la relación se presenta de la forma:

$$k = \frac{n}{2}k_t$$

donde k es la constante de los resortes, k_t la constante de elasticidad de la membrana y n el número de nodos para un AC de $n \times n$ nodos.

5 Simulación

Para realizar la simulación con el AC propuesto, tomamos como base una membrana con las siguientes características:

- Longitud: 10×10 cm. estirada 20% de su longitud.
- Densidad: 0.1
- Tensión: 20 N/m
- Nodos: 50×50 nodos para el AC.

Las condiciones iniciales para el AC, son las mismas que las descritas para la EDP de la que se derivó el modelo.

Se presenta el desplazamiento obtenido por la EDP de la membrana que osciló por $1s$ y de la cual se tomaron 1×10^4 muestras, también se presenta el desplazamiento que se obtuvo del AC propuesto para la misma célula durante el mismo tiempo y con el mismo número de muestras.

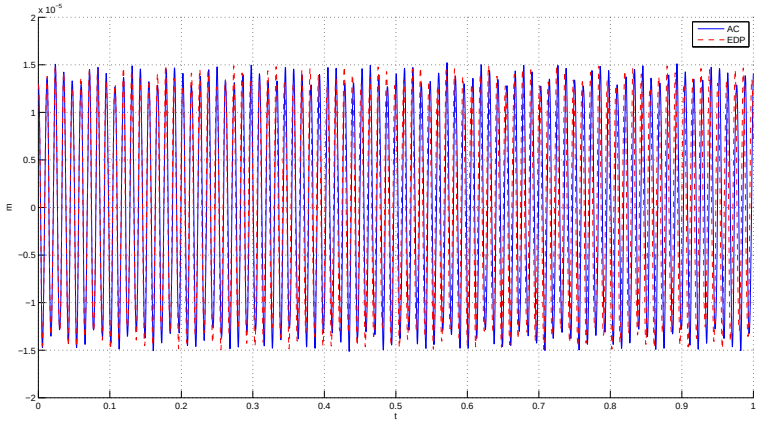


Fig. 7. Gráfica de superposición de la simulación realizada por el AC y la EDP, se presenta el desplazamiento de la célula $(\frac{n}{2}, \frac{n}{2})$ en z para ambos casos.

Las figuras 7 y 8, muestran la superposición de las gráficas obtenidas de las oscilaciones arrojadas por el AC (en línea continua) y por la EDP (en línea punteada) así como un acercamiento a la misma superposición.

De forma cualitativa, existe una correspondencia de fase entre los dos modelos para la misma célula.

Obteniendo los espectros de frecuencia para ambas señales y graficándolos, se observar que existe una congruencia entre los espectros de frecuencia arrojados por el AC y la EDP (ver figura 9). En la gráfica 10 puede verse que la frecuencia fundamental para ambos modelos se encuentra alrededor de los $85Hz$.

5.1 Error

Para calcular el error cuadrático medio entre las medidas generadas por el AC y los valores de referencia que nos proporciona la EDP, empleamos la formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Vac_i - Vr_i)^2$$

donde Vac_i es el i-ésimo valor estimado por el AC y Vr_i es el valor de referencia tomado de la EDP.

Realizando el cálculo obtenemos que $MSE = 4.4189 \times 10^{-11}$.

6 Conclusiones

Se puede concluir que escalar un modelo de AC de cuerda vibrantes a dos dimensiones, para modelar membranas y es factible extender el modelo a tres dimensiones. La di-

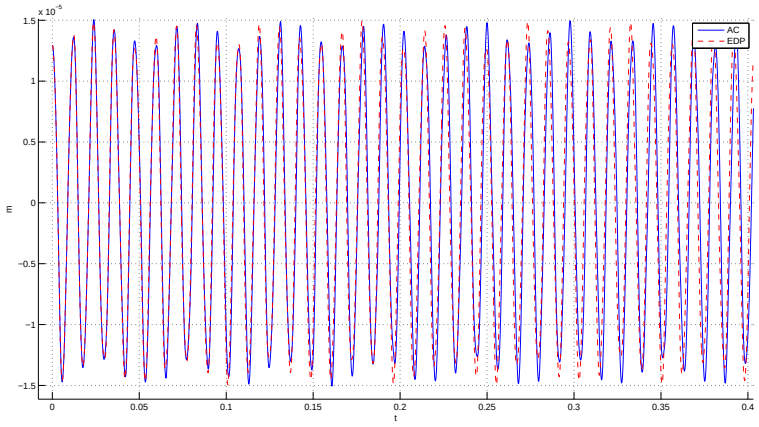


Fig. 8. Acercamiento de la gráfica de superposición de la simulación realizada por el AC y la EDP, se presenta el desplazamiento de la célula $(\frac{n}{2}, \frac{n}{2})$ en z para ambos casos.

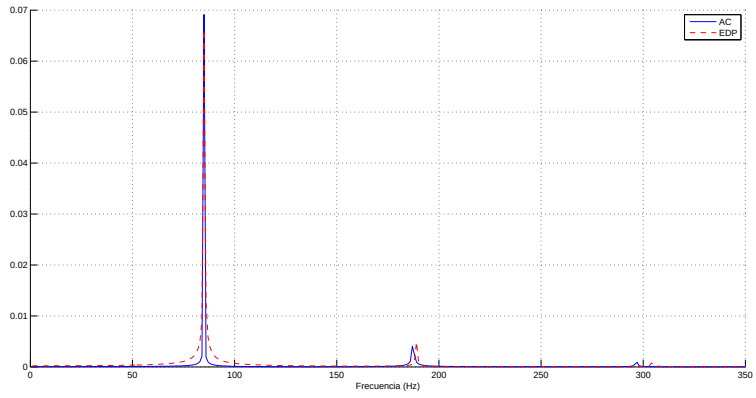


Fig. 9. Gráfica de superposición del espectro de frecuencias de la simulación realizada por el AC y la EDP.

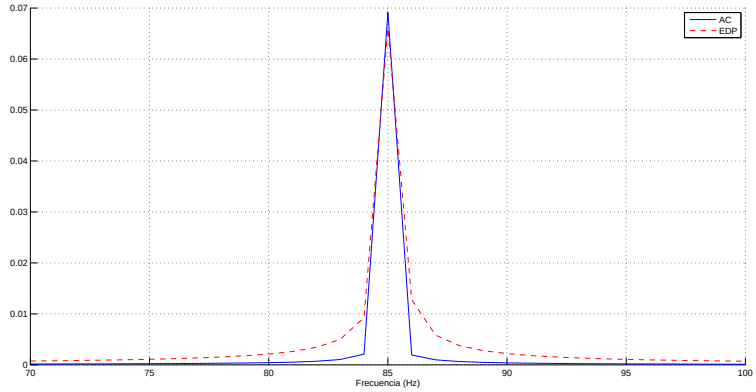


Fig. 10. Acercamiento de la gráfica de superposición del espectro de frecuencias de la simulación realizada por el AC y la EDP.

ficultad radica en obtener la relación existente entre la elasticidad del material que se modela y la constante de restitución de los resortes internos del AC.

El modelo AC para la membrana, queda liberado de las condiciones iniciales, pues no es necesario redefinirlo debido a algún cambio en estas por lo que es posible definir las condiciones iniciales y simular el comportamiento del sistema de forma inmediata, a diferancia de la EDP que es sensible a dichas condiciones para su solución.

La expansión del modelo de una a dos dimensiones, trae consigo el incremento de la complejidad computacional del modelo, de lineal en el caso de una dimensión, a una complejidad $O(n^2)$ debido a que el espacio del AC ahora es una matriz de $n \times n$. En este sentido se justificaría la paralelización del modelo, si definimos m como el número de hilos de ejecución que ayuden a la evolución del AC por unidad de tiempo, entonces la complejidad del AC quedaría como $O(\frac{n^2}{m})$ y si hacemos a m lo suficientemente grande tal que $m \to n$, entonces la complejidad se reduce a $O(n)$ lo que justifica la paralelización del modelo.

Agradecimientos

Los autores agradecen al CONACYT, a la COFAA-IPN (proyecto número 20143959) y EDI-IPN por el apoyo proporcionado a este trabajo.

References

[1] Bar-Yam, Y.: “Dynamics of Complex Systems”. Addison-Wesley, USA (1997)
[2] Chopard, B., et al.: “Cellular Automata Modeling of Physical Systems”. Cambridge University Press, USA (1998)
[3] Espericueta, R.: “Cellular Automata Dynamics Explorations in Parallel Processing”. Bakersfield College, USA (1997)

[4] Hsu, H.P.: “Análisis de Fourier”. Prentice Hall, México (1998)
[5] Huerta-Trujillo, I., Chimal-Eguía, J., Sánchez-Salas, N., Martínez-Nuño, J.: Modelo de autómata celular 1-dimensional para una edp hiperbólica. Research in Computing Science 58(ISSN: 1870-4069), 407 – 423 (November 2012)
[6] Kinsler, L.: Fundamentals of acoustics. Wiley (2000)
[7] Mitchell, M.: “Computation in cellular automata: A selected review”. Nonstandard Computation pp. 95–140 (1998), weinheim: VCH Verlagsgesellschaft
[8] Van Walstijn, M., Mullan, E.: Time-domain simulation of rectangular membrane vibrations with 1-d digital waveguides. pp. 449–454 (2011)
[9] Wolfram, S.: “statistical mechanics of cellular automata”. Reviews of Modern Physics (55), 601–644 (1983)

Modeling, Computer Aided Design, and Construction of a Furuta Pendulum Test-Bed

Mayra Antonio-Cruz[1], Ramón Silva-Ortigoza[1],
Carlos Alejandro Merlo-Zapata[1], Victor Manuel Hernández-Guzmán[2],
Celso Márquez-Sánchez[1], and Jacobo Sandoval-Gutiérrez[1]

[1] Instituto Politécnico Nacional, CIDETEC, Área de Mecatrónica.
Unidad Profesional Adolfo López Mateos, CP. 07700, México, DF, Mexico.

[2] Universidad Autónoma de Querétaro, Facultad de Ingeniería.
CP. 76150, Querétaro, Qro., Mexico.

Abstract. This paper presents a Furuta pendulum as a test-bed to experimentally validate automatic control strategies or theoretical concepts associated with nonlinear systems. Herein, the modeling, computer aided design, and construction of a Furuta pendulum test-bed is introduced step-by-step. The deduction of the Furuta pendulum mathematical model is achieved by using Lagrange equations of motion. In contrast with other works, this model deduction includes an analysis of the system kinematics. Furthermore, a computer aided design of the Furuta pendulum is carried out via the software SolidWorks, based on such a design a test-bed is built. Numerical simulations of the Furuta pendulum model are performed via Matlab-Simulink. Moreover, with the intention of verifying that the test-bed built behaves according to the model herein deduced, experimental tests with the test-bed in open-loop are carried out by using Matlab-Simulink, ControlDesk, and a DS1104 board from dSPACE.

Keywords: Underactuated system; Furuta Pendulum; Test-Bed; Modeling; Computer Aided Design; Construction.

1 Introduction

In last decades, control engineering researchers have had a strong interest on the underactuated mechanical systems. This interest is due to such systems exhibit various problems that can be observed in industrial applications, such as external disturbances and nonlinear behaviors under different operation conditions [1]. Particularly, the Furuta Pendulum –also known as rotary inverted pendulum– is a mechanism that has two degrees of freedom (DOF) and two rotational joints. It is essentially integrated of three elements: a motor and two bars called *arm* and *pendulum*. The motor's shaft is connected to one end of the *arm*, which causes the *arm* to be moved angularly in the horizontal plane, whereas the *pendulum* is joined to the free end of the *arm* through a link that can move freely and allows

Advances in Computer Science
Research in computing Science 83, 2014 pp. 133-144

the rotation of the *pendulum* in the vertical plane. This mechanism is a popular device that has been used both as benchmark for the analysis of nonlinear control and for educational purposes (see, for example [2–8]).

Literature associated with the modeling and construction of the Furuta pendulum is as follows. Regarding the modeling, Acosta [9] described a quasi-conservative dynamic model, derived from the classical mechanics, which allows designing all the controllers as if the system was conservative. Another work provided by Cazzolato and Prime [10], introduces a dynamics of the Furuta pendulum considering a full inertia tensor. That dynamics was derived by using two methods: a Lagrangian formulation and an iterative Newton-Euler formulation. Jadlovská and Sarnovský [1] presented an application of a general procedure to derive a mathematical model of the rotary inverted pendulum with an arbitrary number of pendulum links. To design such a model Lagrangian equations and a Rayleigh dissipation function were used. The validity of the mathematical model generated by the application were shown via numerical simulations. As regards the construction, Allotta *et al.* [11] constructed two different prototypes of the Furuta pendulum with the intention of providing test-beds for the laboratories of mechatronics and complex dynamics, and systems control of the University of Florence. In addition, the dynamic parameter identification of the real prototype was carried out. In the study of García-Alarcón *et al.* [12], a procedure to achieve the parameter identification of an experimental system associated with the Furuta pendulum, a computer aided design and the system built were shown. Also, in order to validate such a procedure, results from numerical simulations of the system dynamic model were compared with experimental results from the system built. On the other hand, a work that describes step-by-step both the modeling and construction of a Furuta pendulum prototype was introduced recently by Antonio-Cruz *et al.* [13]. In that work numerical simulations of the Furuta pendulum mathematical model were performed. Likewise, the prototype built was experimentally tested to show its real behavior.

Having undertaken the literature review, it was found that few works have been exclusively dedicated to the modeling, computer aided design, and construction of a Furuta pendulum. Also, in such works, the Furuta pendulum mathematical model is not described step-by-step and most of them do not include the kinematic analysis of the system. Furthermore, to the authors' knowledge, papers where the computer aided design along with the construction of a Furuta pendulum test-bed are described step-by-step have not been reported until now. The aforementioned could be of great help, since to implement or validate automatic control strategies in real time, the modeling, computer aided design, and construction of a Furuta pendulum test-bed are required. Thus, in order to contribute in this direction, this paper presents the modeling, computer aided design, and construction step-by-step of a Furuta pendulum test-bed, including the corresponding experimental verification.

The remaining of the paper is structured as follows. Section 2 deals with the deduction of the Furuta pendulum mathematical model; whereas, the design and construction of the system under study is treated in Section 3. The numerical

simulations of the Furuta pendulum model and the experimental tests of the test-bed built are shown in Section 4. Lastly, the conclusion is given in Section 5.

2 Modeling

A graphical representation of the Furuta pendulum is shown in Figure 1. There, θ_0 is the *arm* angular position measured with respect to an arbitrary position, θ_1 is the *pendulum* angular position measured with respect to the upright position, τ is the torque (applied to the *arm*) generated by the electric motor, I_0 is the *arm* inertia (when it turns around one of its ends) and the motor inertia, L_0 is the *arm* length, m_1, l_1, and J_1 are the mass, the center of mass location, and the *pendulum* inertia, respectively. Lastly, $g = 9.81$ m/s^2 represents the gravity acceleration.

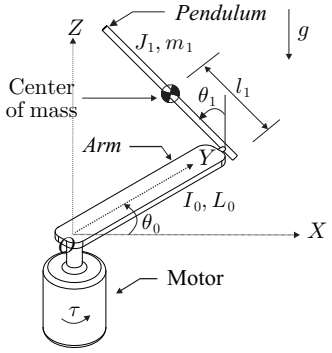


Fig. 1. Furuta pendulum.

As the Furuta pendulum is a two DOF system, its dynamic model is given by two Lagrange equations of motion, which are defined by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_0}\right) - \frac{\partial L}{\partial \theta_0} = \tau, \tag{1}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \frac{\partial L}{\partial \theta_1} = 0, \tag{2}$$

where $\dot{\theta}_0$ is the *arm* angular velocity, $\dot{\theta}_1$ the *pendulum* angular velocity, and L the system Lagrangian determined as

$$L = K - V, \tag{3}$$

being K and V the kinetic energy and potential energy, respectively, of the Furuta pendulum system.

On the one hand, K is the sum of the kinetic energy of the *arm* and the *pendulum*, which are, respectively, defined as follows:

$$K_0 = \frac{1}{2} I_0 \dot{\theta}_0^2, \tag{4}$$

$$K_1 = \frac{1}{2} J_1 \dot{\theta}_1^2 + \frac{1}{2} m_1 v_1^T v_1, \tag{5}$$

where v_1 is the linear velocity of the *pendulum* center of mass. Hence, an analysis of the Furuta pendulum kinematics is required. Then, from Figure 2, the location of the *pendulum* center of mass is determined by

$$x = [x_x,\ x_y,\ x_z]^T, \tag{6}$$

where x_x, x_y, and x_z are defined as follows:

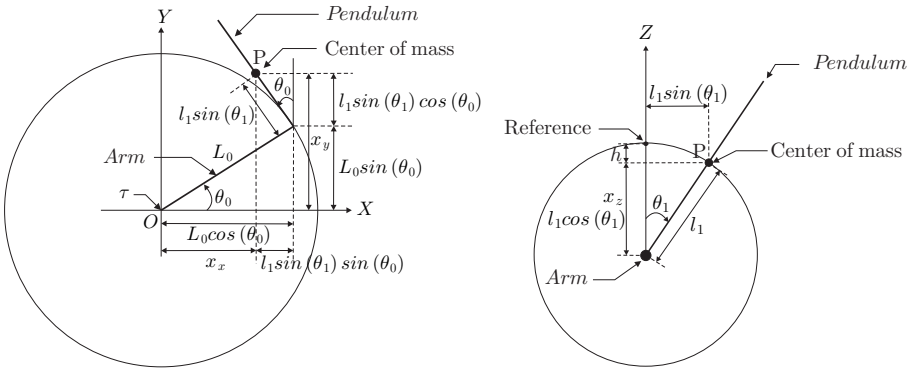
$$x_x = L_0 cos(\theta_0) - l_1 sin(\theta_1) sin(\theta_0), \qquad x_y = L_0 sin(\theta_0) + l_1 sin(\theta_1) cos(\theta_0),$$
$$x_z = l_1 cos(\theta_1).$$

Thus, v_1 is given by

$$v_1 = [\dot{x}_x,\ \dot{x}_y,\ \dot{x}_z]^T, \tag{7}$$

being

$$\dot{x}_x = -\dot{\theta}_0 L_0 sin(\theta_0) - l_1 \left(\dot{\theta}_0 sin(\theta_1) cos(\theta_0) + \dot{\theta}_1 sin(\theta_0) cos(\theta_1)\right),$$
$$\dot{x}_y = \dot{\theta}_0 L_0 cos(\theta_0) + l_1 \left(\dot{\theta}_1 cos(\theta_0) cos(\theta_1) - \dot{\theta}_0 sin(\theta_0) sin(\theta_1)\right),$$
$$\dot{x}_z = -\dot{\theta}_1 l_1 sin(\theta_1).$$



(a) Projection of the *arm* and *pendulum* in the horizontal plane.

(b) Projection of the *pendulum* in the vertical plane.

Fig. 2. Free body diagram of the system.

After replacing (7) in (5) and reducing the resulting expression, the following is found:

$$K_1 = \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}m_1\left[\left(\dot{\theta}_0 L_0\right)^2 + \left(l_1\dot{\theta}_0 sin\left(\theta_1\right)\right)^2 + \left(l_1\dot{\theta}_1\right)^2 + 2\dot{\theta}_0\dot{\theta}_1 L_0 l_1 cos\left(\theta_1\right)\right].$$

Therefore, the Furuta pendulum kinetic energy K is given by

$$\begin{aligned} K &= K_0 + K_1, \\ &= \frac{1}{2}I_0\dot{\theta}_0^2 + \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}m_1\left[\left(\dot{\theta}_0 L_0\right)^2 + \left(l_1\dot{\theta}_0 sin\left(\theta_1\right)\right)^2 + \left(l_1\dot{\theta}_1\right)^2 + \right. \\ &\quad \left. + 2\dot{\theta}_0\dot{\theta}_1 L_0 l_1 cos\left(\theta_1\right)\right]. \end{aligned} \tag{8}$$

On the other hand, V is the sum of the potential energy of the *arm* and *pendulum*. Since the *arm* is moved on the horizontal plane, its potential energy is constant and can be considered equal to zero. Hence, the Furuta pendulum potential energy V is reduced to the *pendulum* potential energy, that is:

$$V = -hm_1 g = m_1 g l_1\left(cos\left(\theta_1\right) - 1\right). \tag{9}$$

Then, from (3), which has associated to (8) and (9), and after carrying out the corresponding derivatives in the equations system (1), (2), the dynamics of the Furuta pendulum is found as follows:

$$\alpha\ddot{\theta}_0 + \beta\dot{\theta}_0\dot{\theta}_1 + \gamma\ddot{\theta}_1 - \sigma\dot{\theta}_1^2 = \tau, \tag{10}$$

$$\gamma\ddot{\theta}_0 + (m_1 l_1^2 + J_1)\ddot{\theta}_1 - \frac{1}{2}\beta\dot{\theta}_0^2 - m_1 g l_1 sin\left(\theta_1\right) = 0, \tag{11}$$

where $\ddot{\theta}_0$ is the *arm* angular acceleration, $\ddot{\theta}_1$ is the *pendulum* angular acceleration,

$$\begin{aligned} \alpha &= I_0 + m_1 L_0^2 + m_1 l_1^2 sin^2\left(\theta_1\right), & \gamma &= m_1 L_0 l_1 cos\left(\theta_1\right), \\ \beta &= m_1 l_1^2 sin(2\theta_1), & \sigma &= m_1 L_0 l_1 sin\left(\theta_1\right). \end{aligned}$$

3 Computer aided design and construction

The elements that integrate the Furuta pendulum test-bed and the procedure followed in the construction of such a test-bed are presented below.

As shown in Figure 3, the Furuta pendulum test-bed is composed –in general– of three blocks, namely: *subsystems*, *power stage*, and *data acquisition and processing*. The block *subsystems* refers to two subsystems, the first one corresponds to a DC permanent magnet motor and two encoders; whereas, the second corresponds to a mechanical structure, which includes the mechanical elements that are, directly and indirectly, moved by the DC motor and those that hold the system. The block *power stage* is integrated by two power electronic devices, which as a whole provide energy to the DC motor. The block *data acquisition and processing* is associated with Matlab-Simulink, ControlDesk, and a DS1104 board from dSPACE, which allow the acquisition and processing of the data provided by the encoders.

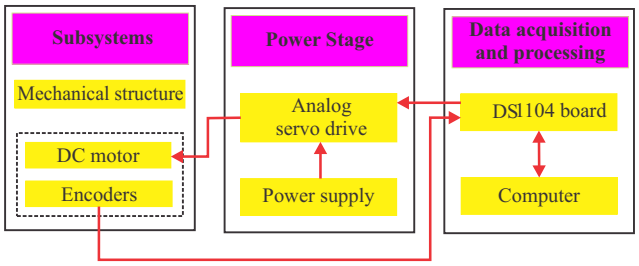


Fig. 3. Block diagram of the test-bed.

3.1 Subsystems

This Section describes the DC motor, encoders, and mechanical structure. The DC motor provides angular movement to the *arm*, the encoders are employed to sense the angular position of the *arm* and the *pendulum*, and the mechanical structure was designed via the software SolidWorks, which is a tool for mechanical design in 3D.

DC motor and encoders. Know the mechanical and electric characteristics of the DC motor is not an easy task; therefore, numerical simulations of the Furuta pendulum mathematical model were performed with the intention of determining the required torque to move the *pendulum* around its upright position (see Section 4). These simulations showed that the required torque is 0.17 Nm. Thus, a 14204 Brush DC motor from Pittman was used.

Regarding the encoders, the encoder that allows sensing the *arm* position is included in the DC motor chassis and it has a 500 CPR resolution; whereas, the encoder associated with the *pendulum* is an ITD 01 A 4 Y 1 optical mini encoder fabricated by Baumer with 1024 CPR as maximum resolution. Both encoders are of the incremental type.

Mechanical structure. The mechanical elements that compose the Furuta pendulum test-bed were drawn and assembled, virtually, by using SolidWorks, since this software includes advanced functions that facilitate the part modeling, create assemblies, and generate plans easily and quickly. Also, SolidWorks allows specifying the material properties for each part of the Furuta pendulum test-bed. Thus, a computer aided design of such a test-bed was generated as shown in Figure 4(a).

According to Figure 4(a), the description of each part that integrates the mechanical structure is as follows:

① The *Arm* was manufactured from a T–6061 T–6 aluminum round bar with a $5/8''$ diameter.

② The *Pendulum* was made from a T–6061 aluminum tube with a 3/8″ outer diameter and a T–6061 aluminum sheet. Furthermore, the *pendulum* includes the shaft that must be mounted into the *pendulum* encoder. The shaft was manufactured from a C–1018 round AISI with a 1/2″ diameter.

③ This part corresponds to the DC motor with encoder described in Section 3.1.

④ This part is associated with the *pendulum* encoder described in Section 3.1.

⑤ The encoder holder is used to hold the encoder of the *pendulum*. This holder was made from a T–6061 aluminum sheet. This part includes a 628/6–2z deep groove ball bearing from SKF.

⑥ The upper sheet is used to hold the DC motor with encoder and was manufactured from a stainless steel sheet.

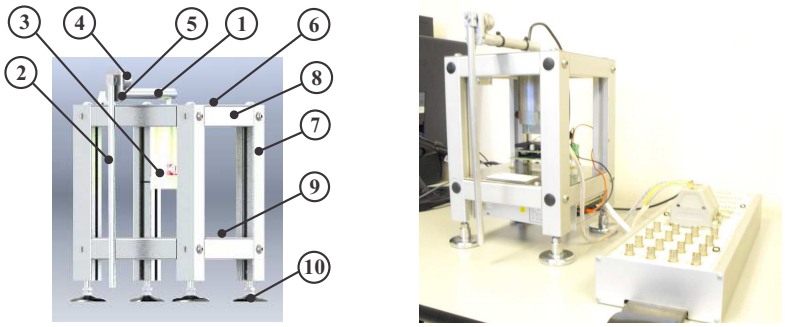
⑦ The vertical aluminum profiles were made from Bosch tubular profiles. The weight of the remaining parts that compose the mechanical structure rests on such vertical profiles.

⑧ The horizontal aluminum profiles keep apart the vertical profiles to a certain distance, providing structural stability to the test-bed.

⑨ The bottom sheet, manufactured from a stainless steel sheet, is used to hold the power supply that provides energy to the electrical and electronic devices.

⑩ The leveling legs are used to even the mechanical structure on a surface, which allows avoiding undesired movements of the test-bed.

The test-bed built can be seen in Figure 4(b).



(a) Computer aided design of the test-bed.

(b) Real test-bed connected to the DS1104 board.

Fig. 4. Furuta pendulum test-bed.

It is important to mention that the design of the Furuta pendulum test-bed was carried out in such a way that another configuration of pendulum can be set (see [14]).

3.2 Power Stage

As was mentioned previously, the block *Power Stage* consists of two power electronic devices. The first one refers to a switched power supply of the HF100W–SF–24 model. This power supply provides energy to the DC motor, by means of the second power electronic device, that is, an analog servo drive, which is used to isolate the block *data acquisition and processing* from the DC motor. Also, this servo drive amplifies the current of a signal provided by the *data acquisition and processing* block, being the amplified signal the DC motor input signal. The servo drive is fabricated by Advanced Motion Controls in the model AZ12A8DDC. An important characteristic of such a servo drive is that it includes an internal control-loop, which avoids losses when the amplified signal is delivered to the DC motor.

3.3 Data acquisition and processing

This Section describes the connection between the test-bed and the DS1104 board from dSPACE. This board was selected due to the integration software between Matlab-Simulink and ControlDesk, that is, the board firmware.

In order to connect the test-bed with the DS1104 board, a block diagram is programmed in the Matlab-Simulink environment. Such a program contains the blocks to generate the signal that has to be amplified by the analog servo drive. Also, the program includes the blocks related to the ports of the DS1104 board where the incremental encoders and the analog servo drive are connected. Latter, the program is executed by means of ControlDesk, which allows the acquisition and processing of the data provided by the encoders and by the DC motor input signal.

4 Simulations and experimental results

Numerical simulations associated with the model (10)-(11) and experiments obtained by using the test-bed built of the Furuta pendulum are presented below.

4.1 Results

In the implementation of the numerical simulations of the dynamics of the Furuta pendulum, that is, (10)-(11), the following values of the parameters were used:

$$I_0 = 0.4592 \text{ x } 10^{-3} \text{ Kgm}^2, \quad l_1 = 0.1475 \text{ m}, \quad L_0 = 0.1414 \text{ m},$$
$$J_1 = 0.2755 \text{ x } 10^{-3} \text{ Kgm}^2, \quad m_1 = 0.038 \text{ Kg}.$$

Such values were obtained directly from the test-bed built. The numerical simulations consisted in apply a constant torque to the *arm* during a period of time equal to 0.2 s, moving it from an arbitrary position, achieving that the *pendulum* reaches a position between ± 1 rad around the upright position from its natural

equilibrium point, that is, $\pm\pi$ rad. As regards to the experimental tests similar conditions to those used in the simulations were considered. Both simulations and experimental tests were carried out in open-loop. The numerical simulations were performed by using Matlab-Simulink and the experimental tests were carried out via Matlab-Simulink, ControlDesk, and a DS1104 board.

The simulation results are shown in Figure 5, also this figure includes the experimental results obtained from the test-bed built. With the purpose of differentiating the simulation results from the experimental ones, the following nomenclature was used. For the simulation results the variables θ_0 and θ_1 are denoted as θ_{0s} and θ_{1s}, respectively. Likewise, θ_{0e} and θ_{1e} denote the experimental results of the variables aforementioned. Whereas, the input of the system, τ, is denoted as τ_s for the case of the simulations and for the experimental results is defined as τ_e.

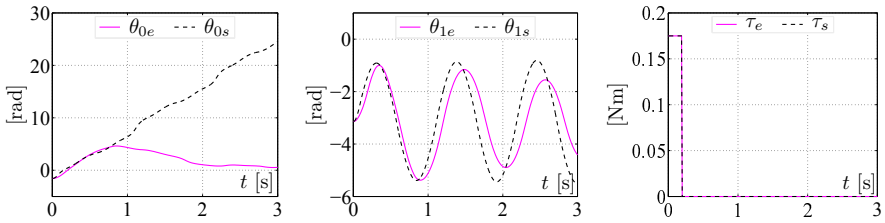


Fig. 5. Simulations and experimental results.

4.2 Discussion

The differences found when comparing the simulation and experimental results are mainly due to the friction forces, which are not considered in the deduction of the mathematical model (10)-(11). Also, another factor that limits the angular movement of the *arm* in real time experiments, that is θ_{0e}, is the cable of the encoder that senses the variable θ_1, temporarily placed in such a way that offers a minimum resistance to the *arm* movement.

5 Conclusion

This paper has introduced step-by-step the modeling, computer aided design, and construction of a Furuta pendulum test-bed. Also, experimental tests on the test-bed built were carried out with success, since the obtained results are in accordance with the ones obtained from numerical simulations of the Furuta pendulum mathematical model. The aforementioned has the propose of facilitating the modeling and construction of a Furuta pendulum test-bed, which can

be used to experimentally validate automatic control strategies and study some features of nonlinear systems.

Regarding the future work, in the deduction of the Furuta pendulum mathematical model, consider the dynamics of the actuator and driver would be interesting. Some works associated with the dynamics of actuators and drivers are [15–18].

Acknowledgments

The work of Mayra Antonio-Cruz, Carlos Alejandro Merlo-Zapata, and Celso Márquez-Sánchez was supported by CONACYT-México and BEIFI scholarships. R. Silva-Ortigoza acknowledges financial support from Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional (SIP-IPN), SNI-México, and the IPN programs EDI and COFAA. Likewise, Victor Manuel Hernández-Guzmán thanks financial support from SNI-México. Finally, Jacobo Sandoval-Gutiérrez acknowledges the financial support from Cátedras CONACYT-México and CIDETEC-IPN.

References

1. S. Jadlovská and J. Sarnovský, "Modelling of classical and rotary inverted pendulum systems–a generalized approach," *Journal of Electrical Engineering*, vol. 64, no. 1, pp. 12–19, 2013.
2. F. J. Muñoz-Almaraz, E. Freire, and J. Galán-Vioque, "Bifurcation behavior of the Furuta pendulum," *International Journal of Bifurcation and Chaos*, vol. 17, no. 8, pp. 2571–2578, 2007.
3. P. X. La Hera, L. B. Freidovich, A. S. Shiriaev, and U. Mettin, "New approach for swinging up the Furuta pendulum: Theory and experiments," *Mechatronics*, vol. 19, no. 8, pp. 1240–1250, 2009.
4. J. Sandoval, R. Kelly, and V. Santibáñez, "Interconnection and damping assignment passivity-based control of a class of underactuated mechanical systems with dynamic friction," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 7, pp. 738–751, 2011.
5. P. Seman, B. Rohal'-Ilkiv, M. Juhás, and M. Salaj, "Swinging up the Furuta pendulum and its stabilization via model predictive control," *Journal of Electrical Engineering*, vol. 64, no. 3, pp. 152–158, 2013.
6. J. Aracil, J. Á. Acosta, and F. Gordillo, "A nonlinear hybrid controller for swinging-up and stabilizing the Furuta pendulum," *Control Engineering Practice*, vol. 21, no. 8, pp. 989–993, 2013.
7. M. Demirtas, Y. Altun, and A. Istanbullu, "Virtual laboratory for sliding mode and PID control of rotary inverted pendulum," *Computer Applications in Engineering Education*, vol. 21, no. 3, pp. 400–409, 2013.
8. V. M. Hernández-Guzmán, R. Silva-Ortigoza y R. V. Carrillo-Serrano, *Control Automático: Teoría de Diseño, Construcción de Prototipos, Modelado, Identificación y Pruebas Experimentales*, Colección CIDETEC–IPN, México, DF, México, 2013. [Online]. Available: http://www.controlautomatico.com.mx

9. J. Á. Acosta, "Furuta's pendulum: A conservative nonlinear model for theory validation and practise," *Mathematical Problems in Engineering*, vol. 2010, Article ID 742894, 29 pages, 2010.
10. B. S. Cazzolato and Z. Prime, "On the dynamics of the Furuta pendulum," *Journal of Control Science and Engineering*, vol. 2011, Article ID 528341, 8 pages.
11. B. Allota, L. Pugi, and F. Bartolini, "Reinforcement neural network for the stabilization of a Furuta pendulum," in *Proc. 2nd European Conference on Mechanism Science* (*EUCOMES*), Cassino, Italy, Sep. 17–20, 2008, pp. 287–294.
12. O. García-Alarcón, S, Puga-Guzmán, and J. Moreno-Valenzuela, "On parameter identification of the Furuta pendulum," in *Proc. International Meeting of Electrical Engineering Research* (*ENIINVIE*), Ensenada, B. C., Mexico, Mar. 28–30, 2012, pp. 77–84.
13. M. Antonio-Cruz, R. Silva-Ortigoza, C. A. Merlo-Zapata, M. G. Villarreal-Cervantes, D. Muñoz-Carrillo, and V. M. Hernández-Guzmán, "Modeling and construction of a Furuta pendulum prototype," *IEEE 2014 International Conference on Mechatronics, Electronics and Automotive Engineering* (*ICMEAE*), Morelos, Mexico, Nov. 2014. Article in Press.
14. C. A. Merlo-Zapata, M. Antonio-Cruz, R. Silva-Ortigoza, H. Taud, I. Rivera-Zárate, D. Muñoz-Carrillo, and V. M. Hernández-Guzmán, "Modeling and construction of an inertia wheel pendulum test-bed," *IEEE 2014 International Conference on Mechatronics, Electronics and Automotive Engineering* (*ICMEAE*), Morelos, Mexico, Nov. 2014. Article in Press.
15. R. Silva-Ortigoza, J. R. García-Sánchez, J. M. Alba-Martínez, V. M. Hernández-Guzmán, M. Marcelino-Aranda, H. Taud, and R. Bautista-Quintero, "Two-stage control design of a Buck converter/DC motor system without velocity measurements via a $\Sigma - \Delta$-modulator," *Mathematical Problems in Engineering*, vol. 2013, Article ID 929316, 11 pages, 2013. [Online]. Available at `http://dx.doi.org/10.1155/2013/929316`
16. R. Silva-Ortigoza, C. Márquez-Sánchez, F. Carrizosa-Corral, M. Antonio-Cruz, J. M. Alba-Martínez, and G. Saldaña-González "Hierarchical velocity control based on differential flatness for a DC/DC Buck converter–DC motor system," *Mathematical Problems in Engineering*, vol. 2014, Article ID 912815, 12 pages, 2014. [Online]. Available at `http://dx.doi.org/10.1155/2014/912815`
17. R. Silva-Ortigoza, V. M. Hernández-Guzmán, M. Antonio-Cruz, and D. Muñoz-Carrillo, "DC/DC Buck power converter as a smooth starter for a DC motor based on a hierarchical control," *IEEE Transactions on Power Electronics*, vol. 30, no. 2, pp. 1076–1084, 2015. [Online]. Available with DOI: `10.1109/TPEL.2014.2311821`
18. V. M. Hernández-Guzmán, R. Silva-Ortigoza, and D. Muñoz-Carrillo, "Velocity control of a brushed dc-motor driven by a DC to DC Buck power converter," *International Journal of Innovative Computing, Information and Control*, vol. 11, no. 2, 13 pages, 2015.

Reviewing Committee

Alejo Macías Miguel Ángel
Altamirano Álvaro
Argüelles Cruz Amadeo José
Barrón Fernández Ricardo
Bonilla Licea Daniel
Botello Castillo Alejandro
Calvo Castro Francisco Hiram
Carreto Arellano Chadwick
Castillo Montiel Erandi
Chimal Eguía Juan Carlos
Cleofas Sánchez Laura
Cruz Cortés Nareli
Delgado Hernández Julio Carlos
Felipe Riverón Edgardo Manuel
Figueroa Nazuno Jesús Guillermo
Godoy Calderón Salvador
Guevara Martínez Elizabeth
Gutierrez Aldana Alfonso
Gutierrez García Juan Jesús
Guzmán Lugo José Giovanni
Horacio Dominguez Christian
Huerta Trujillo Iliac
Juárez Gambino Joel Omar
Landassuri Moreno Victor Manuel
Lazcano Salas Saul
López Ramírez Blanca Cecilia
Martínez Ibarra Mario Iván
Martínez Luna Gilberto Lorenzo
Márquez de Silva Sergio Antonio
Márquez Molina Miguel
Menchaca Méndez Rolando
Mendoza Mendoza Julio Alberto
Miranda Jiménez Sabino
Morales Escobar Saturnino Job
Moreno Armendáriz Marco Antonio
Navarrete Manzanilla Niels Henrik
Orantes Jiménez Sandra Dinora
Orozco Aguirre Héctor Rafael
Pineda Briseño Anabel
Quintana López Maricela
Rubio Espino Elsa
Ruiz Ibañez Victor Antonio
Salinas Rosales Moisés
Sidorov Grigori Olegovich
Sossa Azuela Juan Humberto
Suárez Guerra Sergio
Tamariz Flores Edna Iliana
Téllez Castillo Germán
Torrealba Meléndez Richard
Valle Chavez Abel
Yáñez Márquez Cornelio

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
noviembre de 2014
Printing 500 / Edición 500 ejemplares

ISSN: 1870-4069
http://rcs.cic.ipn.mx

RCS
Research in Computing Science