# Advances in Computing Science

# Research in Computing Science





***Research in Computing Science*** es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 105,** noviembre 2015. 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 105**, November 2015. Printing 500.



Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 105

# Advances in Computing Science

**Juan Carlos Chimal Eguía,**
**Erandi Castillo Montiel,**
**Iliac Huerta Trujillo,**
**Julio Cesar Rangel Reyes (eds.)**

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2015

**ISSN: 1870-4069**



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

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



Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico

# Editorial

The purpose of this volume is to present the most recent advances in selected areas of Computer Science. The works included in this volume were carefully selected by the editors on the basis of the blind reviewing process being the main criteria for selection the originality and technical quality. This issue of the journal Research in Computing Science will be useful for researches and students working in various areas of Computer Science, as well as for all readers interested in enrichment of their knowledge in this field.

All received papers that were submitted for evaluation for the special issue were reviewed by 2 independent members of the editorial board of the volume or additional reviewers. In general, the acceptance rate was 48%. This volume contains revised and corrected versions of 15 accepted papers.

We would like express our gratitude to all people who help to elaborate this volume. First, the authors of the papers for the technical excellence of their works that guarantees the quality of this publication. We also want to thank the members of the editorial board for their hard work in evaluating and selecting the best 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 preparation of this publication. Also we want to give special recognition to the Centro de Investigación en Computacion of the Instituto Politécnico Nacional (CIC-IPN) for their support in publication of this volume. The submission, reviewing, and selection process was supported for free by the EasyChair system, www.EasyChair.org.

Juan Carlos Chimal Eguía
Erandi Castillo Montiel
Iliac Huerta Trujillo
Julio Cesar Rangel Reyes

November 2015

# Table of Contents

Page

# A Software Architecture for Defining a Methodological Approach to Develop Collaborative Applications

Mario Anzures-García[1], Luz A. Sánchez-Gálvez[1], Miguel J. Hornos[2], Patricia Paderewski-Rodríguez[2]

[1] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computacion, Puebla, Mexico

[2] Universidad de Granada, Departamento de Lenguajes y Sistemas Informáticos, E.T.S.I. Informática y de Telecomunicación, Granada, Spain

{mario.anzures, sanchez.galvez}@correo.buap.mx, {mhornos,patricia}@ugr.es

**Abstract.** This paper presents a software architecture-based methodological approach to develop collaborative applications. Today, the use of collaborative applications has spread to various domains, as they facilitate communication, collaboration, and coordination between several users. These applications require mechanisms to support and model communication activities and processing of information, vital in the dynamic nature to the group. In this paper, the use of a software architecture is recommended to develop collaborative applications. This architecture for specifying the structure and behavior through the application, providing a shared meeting space to simplify and agile the group work. Thus, it is possible to support dynamic group structure. In addition, specification tables are proposed to simplify the development of this kind of applications; since the developers to complete the table are analyzing the necessary elements required to build an application, so performing requirements analysis, design, and displayed as would the final application. A case study to validate the software architecture is proposed.

**Keywords:** Methodological Approach, Software Architecture, Collaborative Applications, Specification Tables, Group Work.

## 1 Introduction

The use of a software architecture allows us to have a global perspective of the software applications, since it knows its components what do them and how are related, as well as, the environment in which interacts these. This knowledge leads us to identify and analyze the necessary components of the application to develop. In this way, it is possible to get any application requirements. Therefore, it can be handled to the development of collaborative applications, which provides a shared interface that allows a group of people to achieve a common goal. Consequently, in this paper, a collaborative application for managing the departmental tests is developed as a case study to implement a software architecture.

The necessary elements to create a collaborative application are specified by this model in four layers; which provide four essentials aspects: the group, the cornerstone of the group work; the interaction to control and manage the shared objects to the application and between different users of the group: the application presents several views to visualize the interaction carried out by the group in the stages that application contains; and the adaptation to adjust the application with respect to the produced changes through group interaction.

In order to facilitate the development of the collaborative applications, this model supplies specification tables, so it is possible to define which elements will have the application of an intuitive manner, even this can be made by any inexperienced person in this domain. Thus, this model can be used to specify requirements, to outline the design and implementation.

These requirements identified in the table inform how the application elements will be distributed and executed in each involved stage in this. In this paper, the table elements are the base of the requirements analysis, since each they are part of the application for managing the departmental tests, and therefore, these determine the design and implementation of the same. Thus, the software architecture can be used how a methodological approach to develop this type of applications. This approach is made up by four parts: requirements specification, sketch creation, code production, and application test.

The rest of the paper is organized as follows: Section 2 describes briefly the collaborative applications; Section 3 explains the used software architecture, and the derived specification table of the same; Section 4 presents the case study, in which software architecture is implemented using a methodological approach for building a collaborative application for managing the departmental tests. Section 5 outlines the conclusions and future work.

## 2 Collaborative Applications

A collaborative application is a computer-based application that supports a group of people to achieve a common goal and provides services to support the work of users through a shared environment interface [1]. Collaborative applications provide the shared workspace, where they will perform group work; therefore, it must provide the communication, collaboration, and coordination of the users. Different terms to denote the shared workspace have been used, such as conversations [2], local [3], places [4], spaces [5], conferences [6, 7], and meetings [8, 9]. In general, all these terms denote a group of individuals, geographically distributed, which share a common interest to perform common tasks. In this paper the term session to denote the shared workspace is used.

Collaborative applications provide a mechanism to control and manage sessions, called session management, which allows you to define sessions via a user interface, through which users establish a connection; that is, users to join, leave, invite someone to, and exclude someone from a session. Generally, these mechanisms only specify how the group work will be organized. However, it is important to support and define different styles for group work. Thus, if the style imposed by the system is accepted or unsuitable for group work, you should be changed to one that meets your needs. For this reason, this model uses an ontology to model the session management policies [10] that allows to support different styles of group work.

A variety of tools (such as Groupkit [11], ANTS [9], and SAGA [12]), architectures (e.g., Clock [13], and Clover [14]), and methodologies (AMENITIES [15], ClAM [16], and TOUCHE [17]), which allows to develop collaborative applications. However, these do not specify the steps to develop this kind of applications, and they are not flexible enough to adjust to the group changing needs.

## 3 Software Architecture

Software architecture is 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 [18]. A variety of architectural styles, can be identified in a software architecture. A style is each recognized generic pattern in relation to systems group; of another manner, a style describes and provides the basic property of an architecture, as well as; it establishes the limits for its evolution. One example of architectural style is a layered style, which is organized hierarchically, and it is characterized by a sense of development "bottom-up", so that lower layers provide resources that are used by upper layers, according to their particular needs. A layer is a software technique for structuring the software architecture that can be used to reflect different abstraction levels in the architecture.

A layered style is ideal for supporting the development of collaborative applications, since it leads to break down a complex problem into a set of smaller problems and simpler to solve. Therefore, in this paper a layered software architecture will be used to develop the distributed components of a management system for departmental tests.

The layered software architecture (see Figure 1) has been derived from performed analysis about: Task Analysis [19], Activity Theory [20], Coordination Theory [21], Conceptual Model [22]; and Distributed Cognition [23]. These related works supply a set of ideas and concepts to manage the group interaction of the collaborative applications. Fundamentally, these studies consider four principal aspects in a collaborative application: group, their interaction, the application itself, and its adaptation. Therefore, the software architecture contains four layers: Group Layer, Interaction Layer, Application Layer, and Adaptation Layer.

The first aspect is a key to the performance of the work carried out to achieve the common goal. The group must present an organizational structure to support the *division of labour*, which indicate the actions that the group members (users) should make in relation to the established roles for each of them. This organization must be governed by a police, which defines the roles (Role) that users can play. These roles establish the set of rights/obligations (R/O) and status (St) of the user; whom can execute tasks (T), which are comprised of Activities (A) that use the prevailing shared resources (R).

The second aspect is elemental to provide the communication, coordination, and collaboration between the users. Accordingly, it must establish the session (Ss), which is the shared workspace where the interaction is carried out. Furthermore, it must make available for the awareness group and group memory through a notification (Nt) mechanism, which informs users of and registers every change in the shared resources used in each activity. Finally, it must ensure the consistency of the resources being shared, facilitating the manipulation of the users' permissions, which are granted, in accordance with established organizational structure, and a concurrency (Cc) mechanism.

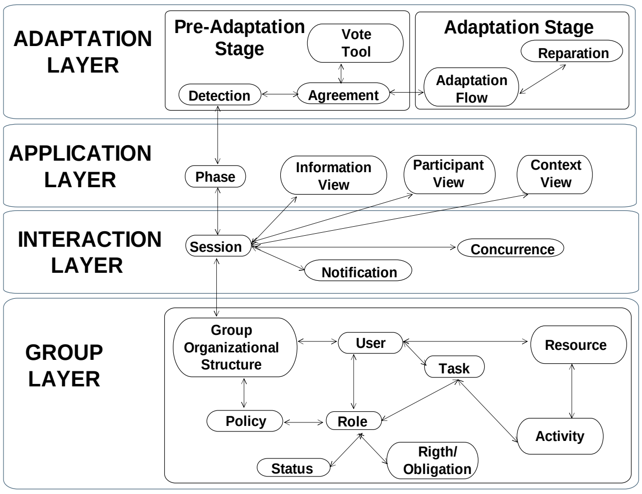



**Fig. 1.** Layered software architecture for building collaborative applications.

The third aspect allows us to show the generated information and interaction in the collaborative application. This is presented in stages (they are defined as each of the collaboration moments [22]) on views (which are user interfaces). Three views are considered in this model: Information View (IV) that displays the user information, Participant View (PV), exhibiting the changes in shared resources and, therefore, provide group awareness, and Context View (CV) shows the group memory, i.e., the change history of shared resources.

In the fourth aspect, the views are adjusted to produced changes by interaction between users and of these with the own application. For doing this adaptation, a detection process monitors the session, determining whether an activity requires to carry out the adaptation. Only if it is an adaptable process, in a non-hierarchical organizational style, an agreement process is executed, and a vote tool is used for reaching a consensus on whether an adaptation process should be performed. When an adaptation or adaptive process should be executed, an adaptation flow process and one reparation —which returns each component to their previous state and notifies users that adaptation cannot take place— will be executed.

The software architecture is mainly focused on the design and implementation of software structures, abstractly defining components that perform a task, their interfaces, and communication between them, in order to meet adequately functional and non-functional requirements of an application. For this reason, this software architecture facilitates the requirements' specification, which will do by a table; which is based principally on the architectural model proposed here, MetaOntology [27], and agile methodologies [28].

Specification Table allows us to: collect all the requirements and agile the design of collaborative applications; reduce the learning curve in the process of creating of this kind of applications, since it is only necessary to complete the table with elements that are intuitive even for any inexperienced person in the domain CSCW; establish how will be access control to application collaborative since these tables are classified by stages, delineating the roles that can participate in each of visualize the collaborative application, since it is possible to define which elements will have the application user interfaces.

The table (see figure 2) contains the elements' specification of the Group Layer (except the organizational structure of the group, policy and user); Interaction Layer, Application Layer. With respect to Adaptation Layer, two columns only are set, one to indicate whether "there or not adaptation" (TA), and another to describe "What is this?" (W?).

## 4 Case Study

In the Autonomous University of Puebla (BUAP) have sought different ways to improve or increase the quality of student learning, one of these mechanisms is the realization of departmental tests. Which aim to homogenize the teaching of a subject, i.e., that all teachers will cover the same percentage of the academic program. The Faculty of Computer Science carries out departmental tests in different areas of knowledge; however, a departmental test requires a shared workspace to that involved teachers perform group work. For this reason, this paper proposes the development of a collaborative application for managing departmental tests using a software architecture. This application is intended to minimize the time and effort that engaged teachers in the enforcement of departmental tests. Several actors involved in this type of tests are considered: Manager (Mg —he/she is responsible to configure the application, establishing who plays the other four roles, existing areas and what subjects are part of these—); Area Coordinator (AC —he/she registers to EC, and schedules the professors' meetings related with the same subject—); Test Coordinator (TC —he/she organizes the completion of each test, requesting and agreeing the tests number to make, dates and questions of these; then he/she posting the test and the classroom where each Professor will apply it—); Professor (P —he/she proposes and vote date in that the test will be performed, as well as the exercises that it will contain—), and Student (Su — he/she consults date and classroom where the test will make, as well as its scores of each subject—). In general, the five roles must register to join at the session, which is provided by application user interfaces. The collaborative application for managing the departmental tests presents four stages: Application Configuration, Test Preparation, Test Elaborating, and Test Results.

Once it has been explained the case study, then it will prove a software architecture-based methodological approach to develop collaborative applications.

### 4.1. A Software Architecture-based Methodological Approach

A methodological approach is proposed to simplify and agile the development of a collaborative application. This approach derives of the software architecture mentioned above, and consists in the following steps:

- To elaborate of the requirements specification.
  - Specifying the elements of the Group Layer, for this, the ontological model of the session management policies can be applied. However, for some developers complete an ontology is difficult. For this reason, it is convenient to use a specification table, in which these elements can be laid.
  - Identifying the elements of the Interaction Layer, which must be listed in the specification table.
  - Recognizing the elements of the Application Layer, which must be registered in the specification table.
  - Determining the elements of the Adaptation Layer, which must be enumerated in the specification table.
  - Generate a unique specification table containing the elements corresponding to the four layers of software architecture proposed here.
- To create a sketch of how the application would display.
  - Organizing of the specification table by stages.
  - Determining the user's access control according to the roles that can participate in each stage.
  - Establishing the elements of each user interface, considering the resources and users interact in it.
  - Defining what and how data must be stored.
  - Carrying out a schema of the user interfaces and stored data.
- To implement the collaborative application.
  - Making the schema where will be data stored.
  - Developing the necessary user interfaces.
  - Building each of the web services required to implement the collaborative application.
  - Making the composition of these web services.
- To test of application.
  - Performing the necessary proofs to deliver the required application.

**Table 1.** Requirements specification of the test preparation stage.

| GROUP LAYER | | | | | | INTERACTION LAYER | | | APPLICATION LAYER | | | | ADAPTATION LAYER | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Role | St | R/O | TASK | ACTIVITY | RESOURCE | Ss | Nt | Cc | IV | PV | CV | STAGE | T | W? |
| | | Authentication AC | Authentication | geting into data | Text box | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | sending data | Acept Button | | χ | √ | χ | χ | χ | | χ | adding AC |
| | | Registering TC | Registering TC | filling record | Form | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | sending data | Acept Button | | χ | √ | χ | χ | χ | | χ | adding CE |
| AC | 3 | Eliminating TC | Removing TC | choosing data | Coordinater UI | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | deleting data | Acept Button | | χ | √ | χ | χ | χ | | χ | removing CE |
| | | Consulting TC | Consulting TC | choosing data | Coordinater UI | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | showing data | Acept Button | | χ | √ | χ | χ | χ | Test Prepara _tion | χ | showing data |
| | | Modifying TC | Modifying TC | choosing data | Form | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | modifying data | Acept Button | | χ | √ | χ | χ | χ | | χ | updating CE |
| TC | 2 | Authentication TC | Authentication | geting into data | Text box | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | sending data | Acept Button | | χ | √ | χ | χ | χ | | χ | adding AC |
| AC | 3 | Scheduling Meeting | Proposing Meeting Date | writing date | | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | posting date | Scheduling UI | | χ | √ | χ | χ | χ | | χ | adding date |
| | | | Seting Meeting Date Highest Voted | choosing date | | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | loading date | Scheduling | | χ | √ | χ | χ | χ | | χ | adding date |

### 4.1.1. Requirements Specification

This is the first step of the methodological approach, for which will used the specification table. In this paper, the tables of each stage (see table 1 to 4) are shown directly by space issues.

### 4.1.2. Creation of an Application Sketch

This is the second step of the methodological approach. Only, the tables of each stage (see table 1 to 4) are displayed, it is not possible to exhibit the other elements referent to application sketch by space issues.

**Table 2.** Requirements specification of the application configuration stage.

| GROUP LAYER | | | | | | INTERACTION LAYER | | | APPLICATION LAYER | | | | ADAPTATION LAYER | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Role | St | R/O | TASK | ACTIVITY | RESOURCE | Ss | Nt | Cc | IV | PV | CV | STAGE | T | W? |
| | | Authentication Mg | Authentication | geting into data | Text box | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | sending data | Acept Button | | χ | √ | χ | χ | χ | | χ | geting into Ad |
| | | Registering AC | Registering AC | filling record | Form | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | sending data | Acept Button | | χ | √ | χ | χ | χ | | χ | adding AC |
| | | Eliminating AC | Removing AC | choosing data | Coordinater UI | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | deleting data | Acept Button | | χ | √ | χ | χ | χ | | χ | removing AC |
| | | Consulting AC | Consulting AC | choosing data | Coordinater UI | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | showing data | Acept Button | | χ | √ | χ | √ | √ | | √ | nothing |
| | | Modifying AC | Modifying AC | choosing data | Formulario | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | modifying data | Acept Button | | χ | √ | χ | χ | χ | | χ | updating AC |
| | | Registering Area | Registering Area | filling record | Form | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| Mg | 1 | | | sending data | Acept Button | | χ | √ | χ | χ | χ | | χ | adding area |
| | | Eliminating Area | Removing Area | choosing data | Coordinater UI | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | deleting data | Acept Button | | χ | √ | χ | χ | χ | Application Configura_ tion | χ | removing area |
| | | Consulting Area | Consulting Area | choosing data | Form | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | modifying data | Acept Button | | χ | √ | χ | χ | χ | | χ | updating area |
| | | Registering Subject | Registering Subject | filling record | Form | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | sending data | Acept Button | | χ | √ | χ | χ | χ | | χ | adding subject |
| | | Eliminating Subject | Removing Subject | choosing data | Coordinater UI | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | deleting data | Acept Button | | χ | √ | χ | χ | χ | | χ | removing subject |
| | | Consulting Subject | Consulting Subject | choosing data | Coordinater UI | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | showing data | Acept Button | | χ | √ | χ | χ | χ | | χ | modifying subject |
| | | Modifying Subject | Modifying Subject | choosing data | Form | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | modifying data | Acept Button | | χ | √ | χ | χ | χ | | χ | updating subject |
| | | Registering P | Registering P | filling record | Form | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | sending data | Acept Button | | χ | √ | χ | χ | χ | | χ | adding P |
| MG, TC | 1, 2 | Eliminating P | Removing P | choosing data | Coordinater UI | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | deleting data | Acept Button | | χ | √ | χ | χ | χ | | χ | removing P |
| | | Modifying P | Modifying P | choosing data | Form | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | modifying data | Acept Button | | √ | √ | χ | χ | χ | | χ | updating P |

**Table 3.** Requirements specification of the elaborating test stage.

| GROUP LAYER | | | | | | INTERACTION LAYER | | | APPLICATION LAYER | | | | ADAPTATION LAYER | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Role | St | R/O | TASK | ACTIVITY | RESOURCE | Ss | Nt | Cc | IV | PV | CV | STAGE | T | W? |
| TC, P | 3,4 | Departmental tests Dates | Proposing tests Date | geting into date | test UI | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | posting date | | | χ | √ | χ | χ | χ | | χ | adding date |
| | | | Seting test Date Highest Voted | choosing date | | Not shared | χ | √ | χ | χ | χ | | χ | showing data |
| | | | | loading date | | | χ | √ | χ | χ | χ | | χ | adding date |
| | | Authentication P | Authentication | geting into data | Text box | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | sending data | Acept Button | | χ | √ | χ | χ | χ | | χ | adding AC |
| | | Consulting Proposals | Consulting Proposals | choosing data | Coordinater UI | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | showing data | Acept Button | | χ | √ | χ | χ | χ | | χ | showing data |
| P | 4 | Vote by test Date | Choosing test Date | choosing date | Scheduling UI | shared | χ | χ | χ | χ | χ | | χ | showing data |
| | | | | vote date | Text box | shared | χ | χ | χ | χ | χ | | χ | showing data |
| | | | | sending vote | Acept Button | | χ | χ | χ | χ | χ | | χ | adding vote |
| | | Elaborating Departamental test | Proposing Number of Questions | geting into date | | | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | posting data | | | χ | √ | χ | χ | χ | | χ | adding data |
| | | | Setting number, the most voted | choosing number | | | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | loading number | test UI | shared | χ | √ | χ | χ | χ | | χ | adding data |
| TC | 3 | | Consulting proposals | visualizing question | | | χ | χ | χ | χ | χ | Elaborating test | χ | showing data |
| | | | Posting test questions the most voted | choosing question | | | χ | √ | χ | χ | χ | | χ | showing data |
| | | | | loading question | | | χ | √ | χ | χ | χ | | χ | adding test |
| P | 4 | Proposing Questions test | Loading proposal of test Question | choosing file | test UI | shared | χ | χ | χ | χ | χ | | χ | showing data |
| | | | | Subir file | | | χ | χ | χ | χ | χ | | χ | adding question |
| TC, P | 3,4 | Consulting Proposals | Consulting Proposals | choosing data | Coordinater UI | Not shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | showing data | Acept Button | | χ | √ | χ | χ | χ | | χ | showing data |
| | | Downloading Proposals of test | Downloading test Exercises | choosing file | test UI | shared | χ | χ | χ | χ | χ | | χ | showing data |
| | | | | downloading file | | | χ | χ | χ | χ | χ | | χ | downloading file |
| P | 4 | Vote by Questions that test will contain | Choosing test Question | choosing question | | shared | χ | χ | χ | χ | χ | | χ | showing data |
| | | | | vote by question | Text box | | χ | χ | χ | χ | χ | | χ | showing data |
| | | | | sending vote | Acept Button | | χ | χ | χ | χ | χ | | χ | adding vote |
| | | Posting Notice | Posting notice to an Individual or Group | writing notice | Text box | shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | posting notice | Acept Button | | χ | √ | χ | χ | χ | | χ | adding notice |
| TC, P | 3,4 | Chat | Posting message | writing messageage | Text box | Not shared | χ | √ | χ | χ | χ | | χ | showing data |
| | | | | posting messageage | Acept Button | | χ | √ | χ | χ | χ | | χ | adding message |

**Table 4.** Requirements specification of the test results stage.

| GROUP LAYER | | | | | | INTERACTION | | | APPLICATION LAYER | | | | ADAPTATION LAYER | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Role | St | R/O | TASK | ACTIVITY | RESOURCE | Ss | Nt | Cc | IV | PV | CV | STAGE | T | W? |
| | | Posting test | Loading test | loading file | test UI | shared | χ | χ | χ | χ | χ | | χ | loading file |
| TC | 3 | | | posting file | | | χ | χ | χ | χ | χ | | χ | adding score |
| | | Posting classroom | Posting classroom where the tests will be done | writing classroom | Text box | Not shared | χ | √ | χ | χ | χ | | χ | showing data |
| | | | | posting classroom | Acept Button | | χ | χ | χ | χ | χ | | χ | adding classroom |
| P | 4 | Posting Scores | Loading Scores | loading file | test UI | shared | χ | χ | χ | χ | χ | | χ | loading file |
| | | | | posting file | | | χ | χ | χ | χ | χ | | χ | adding score |
| Su, P, TC, AC | 5, 4, 3, 2 | Downloading Scores | Downloading Scores | choosing file | test UI | shared | χ | χ | χ | χ | χ | | χ | showing data |
| | | | | downloading file | | | χ | χ | χ | χ | χ | | χ | showing data |
| | | Statistics | Genereting Reports | downloading statistics | | Not shared | χ | √ | χ | χ | χ | | χ | showing data |
| | | | | creating report | File | | χ | √ | χ | χ | χ | | χ | showing data |
| P, TC, AC | 4, 3, 2 | | | loading report | | | χ | √ | χ | χ | χ | test Results | χ | adding report |
| | | Posting Statistics | Loading Statistics | loading file | test UI | shared | χ | χ | χ | χ | χ | | χ | loading file |
| | | | | posting file | | | χ | χ | χ | χ | χ | | χ | adding score |
| Su, P, TC, AC | 5, 4, 3, 2 | Chatear | Posting messageage to to an Individual or Group | writing messageage | Text box | shared | χ | √ | χ | χ | χ | | χ | showing data |
| | | | | posting messageage | Acept Button | | χ | √ | χ | χ | χ | | χ | adding message |
| P, TC, AC | 4, 3, 2 | Posting Notice | Posting notice to an Individual or Group | writing notice | Text box | shared | √ | √ | χ | √ | √ | | √ | nothing |
| | | | | posting notice | Acept Button | | χ | √ | χ | χ | χ | | χ | adding notice |
| | | | | | Scheduling UI | | χ | √ | χ | χ | χ | | χ | showing data |
| Su, P, TC, AC | 5, 4, 3, 2 | Visualizing or downloading dates, questions number and questions proposed, meeting dates, test, chat, messageages, classroom, ratings, and statistics | | consulting date, test, chat, score, messageage and classroom | Scheduling | shared | χ | √ | χ | χ | χ | | χ | showing data |
| | | | | | test UI | | χ | √ | χ | χ | χ | | χ | showing data |
| | | | | | Coordinater UI | | χ | √ | χ | χ | χ | | χ | showing data |

### 4.1.3. Implementation of the Collaborative Application

This is the third step of the methodological approach. Only, some user interfaces see Figure 2 to 4) of the application are presented by space issues. As seen in Figures 2 to

4, user interfaces are the result of sketch derivative of the elements placed on the specification table. Although, the application for managing the departmental tests is developing, the proofs already have been performed.

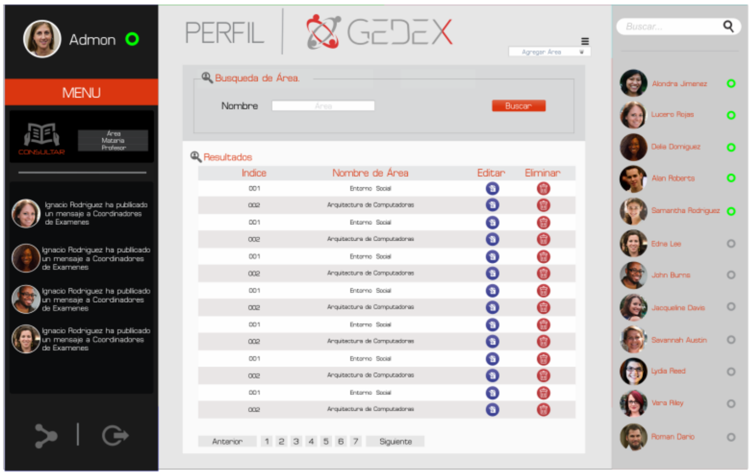



**Fig. 2.** User Interface of the subjects by area.

## 5 Conclusions and Future Work

This paper has presented a methodological approach based on layered software architecture for developing Collaborative Applications. The approach is enriched with layered software architecture, which offers the sufficient guidelines to build this kind of applications by four layers. These separate this construction on four concerns: group, interaction, application, and adaptation.

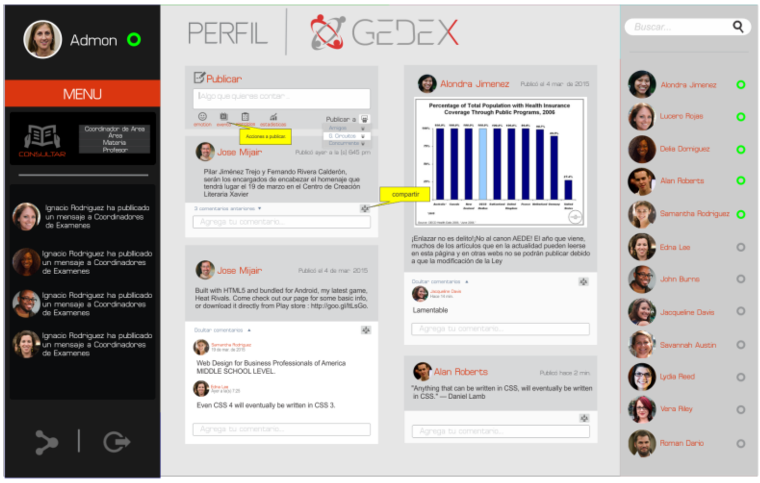



**Fig. 3.** User Interface of the professor profile and chat.

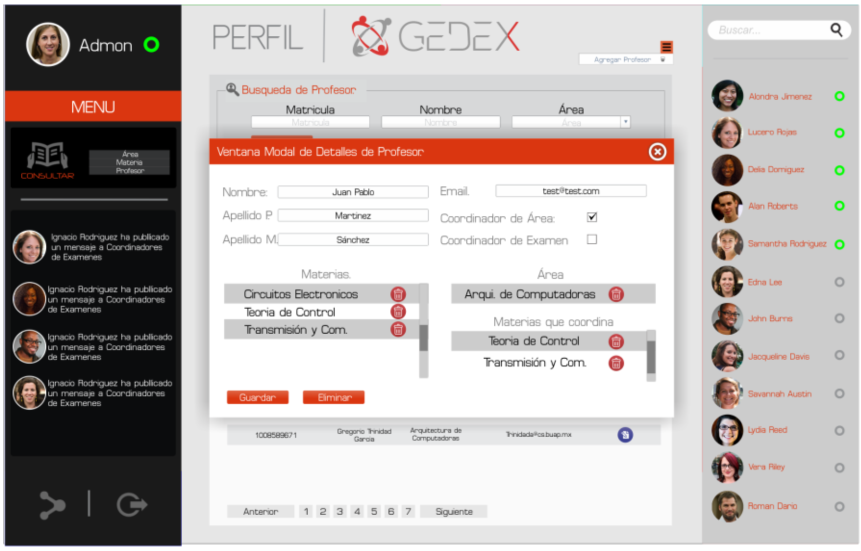



**Fig. 4.** User Interface of the searching of professor.

The methodological approach proposes four phases: requirements specification, sketch creation, implementation, and proof. Which is founded on specification tables that define the elements that will have the application of an intuitive manner, even this can be made by any inexperienced person in this domain, but with application knowledge to carry out. By applying this methodological approach has been implemented the collaborative application of management of departmental tests.

The future work is orientated to establish in a manner detailed this methodological approach.

## References


1. Ellis, C.A., Gibbs, S.J., Rein, G.L.: Groupware: some issues and experiences. Communications of the ACM, Vol. 34-1, pp. 39–58 (1991)
2. Kaplan, S.M., Carroll, A.M.: Supporting collaborative processes with conversation builder. Computer Communications, 15(8), pp. 489–501 (1992)
3. Fitzpatrick, G., Kaplan, S.M., Tolone, J.: Work, locales and distributed social worlds. In: Proceedings ECSCW, pp. 1–16 (1995)
4. Fitzpatrick, G., Kaplan, S.M., Mansfield, T.: Physical spaces, virtual places and social worlds: A study of work in the virtual. In: Proceedings CSCW, pp. 334–343 (1996)
5. Beaudouin-Lafon, M.: Beyond the workstation: Mediaspaces and augmented reality. In: Proceedings of the Conference on People and computers IX, 9, pp. 9–18 (1994)
6. Rajan, S., Venkat, R.P., Vin, H.M.: A formal basis for structured multimedia collaborations. In: Proceedings of the 2nd IEEE International Conference on Multimedia Computing and Systems, pp. 194–201 (1995)

7. Venkat R.P., Vin, H.M.: Multimedia conferencing as a universal paradigm for collaboration. In: L. Kjelldahl (ed.), Multimedia: Systems, Interaction and Application, 1st Eurographics Workshop, Springer-Verlag, pp. 173–185 (1991)
8. Edwards, W.K.: Session management for collaborative applications. In: Proceedings CSCW, pp. 323–330 (1994)
9. García, P., Gómez, A.: ANTS framework for cooperative work environments. IEEE Computer Society Press, 36(3), 56–62 (2003)
10. Anzures-García, M., Sánchez-Gálvez, L.A., Hornos, M., Paderewski-Rodríguez, P.: Ontology-Based Modelling of Session Management Policies for Groupware Applications. Lecture Notes on Computer Science, Vol. 4739, pp. 57–64, Springer-Verlag, (2007)
11. Roseman, M., Greenberg, S.: Building Real-time Groupware with GroupKit, a Groupware ToolKit. ACM Trans. Computer-Human Interaction, Vol. 3, 66–106 (1996)
12. Fonseca, B., Carrapatoso, E.: SAGA: A Web Services Architecture for Groupware Applications. In: Proc. of the CRIWG, LNCS 4154, Springer-Verlag, pp. 246–261, (2006)
13. Graham, T.C.N., Urnes, T.: Integrating Support for Temporal Media in to an Architecture for Graphical User Interfaces. In: Proc. of the International Conference on Software Engineering (ICSE'97), ACM Press, Boston, USA, pp. 172–182 (1997)
14. Laurillau, Y., Nigay, L.: Clover Architecture for Groupware. In: Proc. of the ACM Conference on CSCW, New Orleans, Louisiana, USA, pp. 236–245 (2002)
15. Gea, M., Gutierrez, F.L., Garrido, J.L., Canas, J.J.: AMENITIES: Metodología de Modelado de Sistemas Cooperativos. In: COLINE02, Workshop de Investigaci6n sobre nuevos paradigmas de interacción en entornos colaborativos aplicados a la gestión y difusión del Patrimonio cultural, Granada, Spain (2002)
16. Molina, A.I., Redondo, M.A., Ortega, M., Hope, U.: ClAM: A methodology for the development of groupware user interfaces. Journal of Universal Computer Science (2007)
17. Penichet, V.M.R., Lozano, M.D., Gallud. J.A.: An Ontology to Model Collaborative Organizational Structures in CSCW Systems. In: Engineering the User Interface, Springer, pp. 127–139 (2008)
18. Garlan, D., Shaw, M.: An introduction to software architecture. Advances in Software Engineering and Knowledge Engineering, 1, pp. 1–39 (1994)
19. Van Welie, M., van der Veer, G.C., Eliëns, A.: An Ontology for Task World Models, Design, Specification and Verification of Interactive System. Springer Computer Science, 57–70 (1998)
20. Kuutti K.: The concept of activity as a basic unit of analysis for CSCW research. In: Proceedings of the Second European Conference on CSCW (1991)
21. Ellis, C., Wainer, J.A.: Conceptual model of groupware. In: Proceedings of the 1994 ACM Conference on CSCW, pp. 79–88 (1994)
22. Hollan, J., Hutchins, E., Kirsh, D.: Distributed cognition: toward a new foundation for human-computer interaction research. ACM Transactions on Computer-Human Interaction (TOCHI) Special issue on HCI in the new millennium, Vol. 7-2 (2000)

23. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: From Ontological Art Towards Ontological Engineering. In: Spring Symposium on Ontological Engineering of AAAI, Stanford University, California, pp. 33–40 (1997)
24. Abrahamsson, P., Salo, O., Ronkainen, J.: Agile software development methods: Review and analysis. VTT Electronics (2002)

# Clasificador no supervisado para series de tiempo

E.A. Santos-Camacho, J.G. Figueroa-Nazuno, J.C. Chimal Eguía

Instituto Politécnico Nacional, Centro de Investigación en Computación (CIC), México

araceli.libelula@gmail.com, {jfn,chimal}@cic.ipn.mx

**Resumen.** Una serie de tiempo es una secuencia de datos numéricos que describen fenómenos naturales o artificiales, por lo que su análisis es de gran interés en diversas áreas del conocimiento, siendo de gran utilidad en el desarrollo industrial, social y científico. Una alternativa para su análisis se presenta en la clasificación, que permite tener agrupaciones con características similares, sin embargo, existe una gran cantidad de fenómenos en la vida real que no cuentan con una clasificación previa o establecida, por lo que los clasificadores supervisados no pueden ser aplicados en este tipo de problemas, y es por ello que surge la necesidad de utilizar un enfoque de clasificación no supervisado. En este trabajo se propone una metodología, que permita obtener la clasificación no supervisada de un conjunto de series de tiempo utilizando un enfoque no supervisado, la cual consta de dos etapas: en la primera, se diseñaron doce técnicas diferentes que contemplan la búsqueda de series representativas y en la segunda, se propone un algoritmo de ensamble para obtener la agrupación final, mezclando las agrupaciones obtenidas en la etapa anterior.

**Palabras clave:** clasificación no supervisada, serie de tiempo, ensamble de algoritmos.

## Unsupervised Classifier for Time Series

**Abstract.** A time series is a sequence of numerical data describing natural and/or artificial phenomena, therefore its analysis is of great interest in various areas of knowledge, being very useful in industrial, social and scientific development. An alternative for the analysis of time series is presented in classification techniques that allow the creation of groups of similar characteristics, however, there is a lot of phenomena in real life that do not have a prior or established classification, therefore supervised classifiers cannot be applied in such problems. This is why unsupervised classification approach arises, which allow us creating groups from a given set of data without any prior knowledge. In this work a methodology is proposed that allows the unsupervised classification of a set of time series using an unsupervised approach. This method consists of two stages: first, it was designed twelve different search techniques to find representative time series and second, an assembly algorithm is proposed for the final grouping, mixing clusters obtained in the previous stage.

**Keywords:** unsupervised classification, time series, assembly algorithms.

## 1. Introducción

En la actualidad son innumerables las aplicaciones que se pueden citar en diferentes áreas de investigación y la industria, donde los datos están representados en forma de series de tiempo. En los últimos años se ha visto una dramática explosión en la cantidad de series de tiempo, por lo que su análisis desempeña un papel muy importante, ya que con él es posible entender los fenómenos que describen.

Se denomina "serie de tiempo" a un conjunto de datos de un cierto fenómeno o ecuación, registrados secuencialmente.

Una alternativa que permite conocer el comportamiento y dinámica de un conjunto de series de tiempo, se ha presentado en el problema de clasificación, sin embargo, es preciso mencionar que la mayoría de los fenómenos encontrados en la vida real, no tienen una clasificación como tal y es por eso que la clasificación no supervisada, ha traído gran interés.

La clasificación es organizar y categorizar los objetos en clases o grupos diferentes no etiquetados, los cuales deben ser coherentes u homogéneos [1, 2].

Es importante comentar que uno de los retos en la clasificación de series de tiempo, está dada por su estructura, generalmente al clasificar fenómenos descritos por Atributos, el orden de ellos no afecta el resultado, sin embargo, las series de tiempo conservan un orden o temporalidad, lo que no permite cambiar la posición de los datos, por lo que algoritmos que trabajan con atributos no pueden ser aplicados en este tipo de problemas.

Por lo anterior, en este trabajo se presenta una metodología de clasificación no supervisada y de forma libre para series de tiempo, basada en el desarrollo de distintos algoritmos que son ensamblados para obtener la agrupación final, con el propósito de que este proceso pueda revelar objetos/categorías desconocidas que ayuden a un mejor entendimiento de los datos, resaltando la estructura inherente al agrupar un conjunto de serie de tiempo.

## 2. Serie de tiempo

Una serie de tiempo, es un conjunto de datos numéricos, obtenidos a partir de una observación experimental o mediante el cálculo numérico de ecuaciones, es decir, una serie de tiempo es un conjunto de la forma $ST = \{x_1, x_2, \ldots\ldots, x_t, \ldots\ldots x_N\}$.

De lo anterior algunas características sobresalientes deben ser consideradas en el análisis de las series de tiempo, las cuales se describen a continuación:

– Dimensionalidad: son los grados de libertad en las series de tiempo, es decir, si se trabaja en el espacio binario, en los reales u otro.
– Tamaño: es la cantidad de datos que conforman una serie de tiempo.
– Representación: una serie de tiempo puede ser representada en un plano cartesiano, donde el eje 'y' representa el valor (o magnitud) y el eje 'x' es un índice consecutivo que corresponde a cada valor (el cual puede ser tiempo u otra variable) en la serie de tiempo, por lo tanto, las series de tiempo se encuentran en 1½ dimensión.
– Estructura: las series de tiempo contienen picos, los cuales no son derivables ni integrables.

## 3. Comparación entre series de tiempo

La comparación de series de tiempo se centra en la búsqueda de similitud de patrones semejantes, por lo que para realizar el análisis de semejanza entre dos series de tiempo es necesario que éstas sean cuantificables, es decir, asignarles un valor numérico. De acuerdo a Krantz [3, 4, 5] "medir es el acto o proceso de asignar un número a un fenómeno, con base en alguna regla".

En [6] se presentan las siguientes definiciones:

- "La similitud es una cantidad que refleja la fuerza o intensidad de relación, entre dos objetos".
- "La distancia es la medida de disimilitud entre dos objetos y se refiere a la discrepancia entre dos objetos, basada en varias características analizadas. También puede interpretarse como una medida de desorden, entre dos objetos".

En este trabajo se emplean técnicas de distancia para dar un valor de semejanza entre un par de series de tiempo, por lo que la relación con la similitud queda dada por: "Poca distancia equivale a Poca diferencia, lo que equivale a Gran similitud".

## 4. Ensamble de algoritmos de agrupación

Por el teorema de *No Free Lunch* [7], que nos dice que si un algoritmo funciona bien para un problema dado, este no va a tener los mismos resultados para otro problema, en este trabajo se presentan diferentes técnicas de agrupación de series de tiempo y para obtener la agrupación final, se aplica un algoritmo de ensamble.

El ensamble de algoritmos de agrupación [8], es generado por un conjunto de algoritmos de agrupación denominado "agrupación base", el cual combina las salidas de los algoritmos de la "agrupación base" tal que la información útil codificada en cada algoritmo de agrupación, es aprovechada al máximo para obtener el agrupamiento final.

Comúnmente los métodos de ensamble son aplicados principalmente porque son capaces de impulsar algoritmos débiles y mejorar la estimación aleatoria.

En términos generales obtener una agrupación es relativamente fácil, ya que cualquier algoritmo de partición genera una agrupación, mientras que la mayor dificultad radica en la combinación de los algoritmos, por lo que para obtener éxito en el ensamble de algoritmos, la clave está en cómo se expresa la información dada por los “agrupación base” y como está es ensamblada.

## 5. Metodología

La metodología propuesta para la agrupación de series de tiempo considerando un enfoque no supervisado y de forma libre, se describe de la siguiente manera:

1. Se selecciona un conjunto de series de tiempo.
2. Se selecciona una medida de distancia.
3. Se aplican las técnicas de agrupación propuestas (sección 6.3).
4. Se aplica el algoritmo de ensamble modificado (sección 6.4), a las agrupaciones obtenidas en el paso 3.
5. Se evalúa la agrupación final.

### 5.1. Conjunto de datos

En este trabajo se tomaron 3 conjuntos de datos en los que se contemplan datos Sintéticos, Aleatorios y un conjunto de datos con que contempla la transformación de objetos 3D a 1D, donde 1D corresponde a una serie de tiempo.

#### 5.1.1. Conjunto de datos sintético

Para ejemplificar los problemas más comunes de las series de tiempo se utiliza un conjunto de datos sintético con 21 series de tiempo, el cual contempla 3 estructuras diferentes (tres grupos), que son: Cuadros, Senos y Cuadros; a los cuales se les hicieron diferentes modificaciones para representar los problemas de escala, desfase, ruido y combinación de ellas.

#### 5.1.2. Conjunto de datos aleatorios

La distribución normal o distribución de Gauss es sin duda la más importante y la demás aplicación en todas las distribuciones continuas, ya que es bastante adecuada para describir la distribución de muchos conjuntos de datos que ocurren en la naturaleza, la industria y la navegación, entre otros. Por esta razón se generó un conjunto de datos conformado por 180 series de tiempo, con distribución normal de forma aleatoria; con el objetivo es contar con un conjunto de datos controlado, se generaron 9 grupos.

### 5.2. Conjunto de datos objetos 3D

En Computación la representación de datos espaciales de una figura 3D está dada por la definición de malla de polígonos, la cual es muy popular para modelos tridimensionales debido a su simplicidad. Se tomó un conjunto de datos correspondiente al trabajo realizado en [9], donde realizan la transformación de objetos 3D a 1D bajo la siguiente idea "Se coloca el objeto 3D en un cubo y dado un orden predeterminado, se registra la distancia de forma consecutiva".

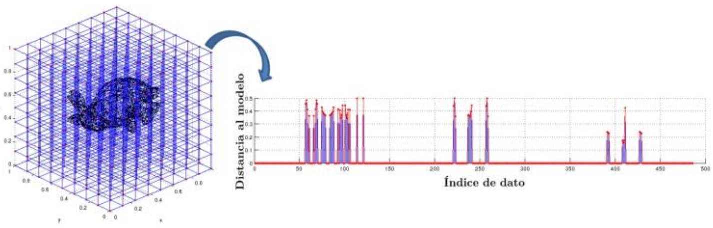



**Fig. 2.** Conjunto de datos de objetos 3D, extraída de [9].

El conjunto de datos está conformado por 5 clases que incluye a: delfines, perro, rostros, copas y pistolas, con un total de 40 series de tiempo, donde cada serie de tiempo tiene 1014 datos.

### 5.3. Medidas de distancia

Las medidas de distancia utilizadas e implementadas en este trabajo, contemplan la comparación directa utilizando la distancia de *Minkowski* (ecuación 1) e indirecta *Fast Dynamic Time Warping* o Fréchet evolucionado [10].

$$d = \sqrt[\lambda]{\sum_{k=1}^{n} |x_k - y_k|^{\lambda}}, \tag{1}$$

donde $x, y$ corresponden a la serie 1 y 2 respectivamente, $k$ es el valor contenido en esa posición en la serie de tiempo, $n$ es la cardinalidad de la serie de tiempo y $\lambda$ es el orden de distancia a calcular, si es 1 corresponde a distancia *City Block*, 2 es *Ecuclidiana* y >= 3 *Minkowski*.

### 5.4. Técnicas de agrupación

Los algoritmos de agrupación propuestos, están basados en la búsqueda de series de tiempo representativas dado un conjunto de datos, a través de la relación que existe entre la serie de tiempo representativa y una medida de distancia hacia todas las demás. Para formar cada una de los grupos encontradas en el conjunto de datos, se emplea un criterio de corte estadístico [11].

### 5.5. Agrupación mediante serie representativa

Para la agrupación de series de tiempo, en este trabajo nos centramos en la búsqueda de series representativas, para lo cual se consideran los siguientes cuatro criterios:

– Aleatorio: se escoge de forma aleatoria una serie de tiempo.
– Mínima distancia: se obtiene la distancia de cada serie de tiempo contra todas las demás y aquella que tenga la mínima distancia con alguna otra.
– Máxima distancia: se obtiene la distancia de cada serie de tiempo contra todas las demás y aquella que tenga la máxima distancia con alguna otra.
– Centroide: por cada serie de tiempo se obtiene la distancia a todas las demás y se promedian, se toma como serie representativa a la serie con el menor promedio.

Por cada criterio propuesto se presentan a continuación tres métodos para la agrupación de series de tiempo, con un enfoque no supervisado y de forma libre:

- **Método de agrupación 1**

Este método considera la agrupación de un conjunto de datos dado, empleando solo una serie de tiempo denominada serie representativa.

1. Obtención de serie representativa: se toma la serie representativa dependiendo del criterio deseado (aleatorio, mínima distancia, máxima distancia o centroide), del conjunto de datos y su distancia con respecto a todas las demás.
2. Criterio de corte: para generar las agrupaciones se ordena de menor a mayor las distancias de la serie representativa contra las demás y se aplica el criterio de corte empleando el umbral establecido en 2.

- **Método de agrupación 2**

Este método modifica el criterio de corte, ya que se podrían tener agrupaciones en los extremos (un grupo o tantos grupos como series de tiempo existentes en el conjunto de datos), dependiendo de la distribución de los datos. Dado el método 1 se modifica el paso 2 y se agrega el paso 3.

2. Criterio de corte: para generar las agrupaciones se ordena de menor a mayor las distancias de la serie representativa contra las demás y se aplica el criterio de corte empleando el umbral 0.5 (en cada iteración se aumentara el umbral en 0.5, hasta llegar a 3.5).
3. Evaluación: dada la agrupación con un umbral de corte, se evalúa mediante el índice I (la agrupación final será la que tenga el Índice I mayor).

- **Método de agrupación 3**

Este método examina la posibilidad de que exista, más de una serie representativa. Dado el método 2, se modifica el paso 2 y 3, y se agrega el paso 4.

2. Criterio de corte: para generar las agrupaciones se ordena de menor a mayor las distancias de la serie representativa contra las demás y se aplica el criterio de corte empleando el umbral en 2.
3. Agrupación final: del paso 2 se obtiene el primer grupo formado y se agrega a la agrupación final, después estas series se eliminan del conjunto de datos.
4. Posteriormente se repite el paso 1, 2 y 3, hasta que no existen series de tiempo en el conjunto de datos.

### 5.6. Método de ensamble

Dado que en este trabajo se proponen varios algoritmos para la agrupación de series de tiempo, se presenta el inconveniente de tener diversas agrupaciones, por lo que se hace necesario implementar un método que permita obtener una sola agrupación. En este trabajo se propone la modificación del método de ensamble por re-etiquetado [12], las modificaciones realizadas son:

– No se re-etiquetan los grupos.
– No se fija la cardinalidad de grupos finales como lo hace el método, basado en votación.
– Se examinan grupos con cardinalidad >3.
– Se considera un grupo en la agrupación, siempre y cuando este aparezca en la solución en más de 6 de las técnicas propuestas.

## 6. Análisis experimental

Se aplicaron las 12 técnicas de agrupación propuestas y posteriormente el algoritmo de ensamble en los tres conjuntos de datos, los resultados obtenidos se presentan en la tabla 1. Así mismo se examinan los conjuntos de datos utilizando el algoritmo k-medoids, con el objetivo de tener un referente de comparación de resultados.
En los resultados obtenidos se puede observar que el método de agrupación propuesto obtuvo mejor precisión que el algoritmo más *k-medoids*, independientemente de la técnica de distancia empleada.

Para ejemplificar una de las agrupaciones obtenidas en la figura 3, se presenta la agrupación del conjunto de datos de objetos 3D, en donde se presenta al objeto 3D que representa cada serie de tiempo, donde se puede apreciar en los resultados, que los objetos se agrupan correctamente, teniendo solo seis objetos mal agrupados; si bien no se logran obtener los 5 grupos existentes en el conjunto de datos como lo haría el algoritmo *k-medoids*, esto no afecta a la propuesta, debido a que los grupos formados son consistentes, es decir, los grupos contienen elementos que pertenecen a la misma clase. Cabe mencionar que la agrupación empleada fue utilizando la transformación de estos objetos 3D a una dimensión.

**Tabla 1.** Precisión obtenida utilizando el ensamble de algoritmos.

| Técnicas de distancia | Sintético | Aleatoria | Imágenes 3D |
|---|---|---|---|
| *City Block* | 80.95 | 100 | 82.5 |
| *Euclidiana* | 76.19 | 100 | 85 |
| *Minkowski* | 80.95 | 100 | 57.7 |
| FDTW | 95.23 | 100 | 77.5 |
| *K-medoids* | 47.61 | 80 | 57.5 |

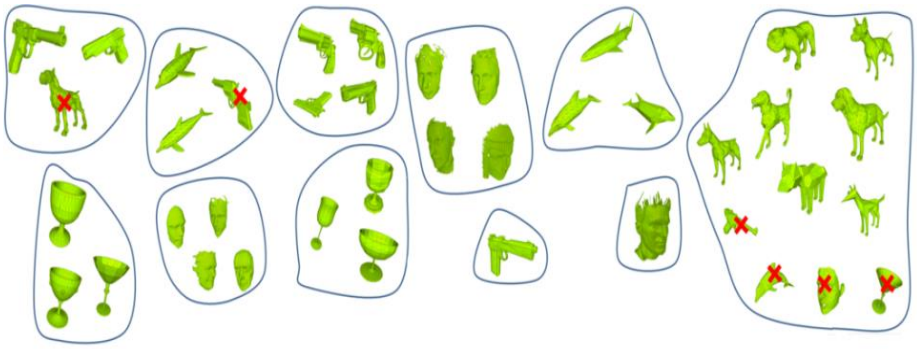


**Fig. 1.** Agrupación del conjunto de datos 3D empleando el método de ensamble y técnicas de agrupación propuestas y la medida de distancia *City Block*.

## 7. Conclusiones

El problema de clasificación no supervisada de series de tiempo, consiste en organizar las series de tiempo que son similares y distinguir entre aquellas que no lo son. Considerando el teorema de *No Free Lunch*, en este trabajo se presenta la agrupación de series de tiempo bajo el enfoque no supervisado y de forma libre, usando doce técnicas que involucran 4 criterios diferentes y finalmente para obtener una agrupación final, se combinan las salidas de las técnicas propuestas mediante la modificación del método de ensamble por re-etiquetado. Los resultados obtenidos por el ensamble de algoritmos supera en todos los casos probados al algoritmo *k-medoids*, lo que indica el potencial del método propuesto; además una de las aplicaciones de este método sobresaliente es la agrupación de objetos 3D con transformación a 1D, en donde además de tener semejanza en 1D en los grupos formados, al identificar el objeto 3D correspondiente, estos pertenecen a la misma categoría.

## Referencias


1. Yang, Y., Chen, K.: Time series clustering via RPCL network ensemble with different representations. IEEE Transactions on Systems, Man, and Cyberneticspart C. Applications and Reviews (2010)
2. Vilar, J.A., Alonso, A.M., Vilar, J.M.: Non-linear time series clustering based on non-parametric forecast densities. Computational Statistics & Data Analysis (2010)
3. Krantz, D.H., Luce, R.D., Suppes, P., Tversky, A.: Foundations of measurement. Vol. 1. Additive and polynomial representations. New York: Academic (1971)
4. Suppes, P., Krantz, D.H., Luce, R.D., Tversky, A.: Foundations of measurement. Vol. 2. Geometrical, threshold, and probabilistic representations. San Diego, CA: Academic, (1989)
5. Suppes, P., Krantz, D.H., Luce, R.D., Tversky, A.: Foundations of measurement. Vol. 3. Representation, axiomatization, and invariance. San Diego, CA: Academic (1990)
6. Teknomo, K.: Similarity measurement. [http://people.revoledu.com/kardi/tutorial/Similarity/index.html] (Agosto, 2015)
7. Wolpert, D.H., Macready, W.G.: No Free Lunch Theorems for Optimization. IEEE Trans. Evolutionary Computation, Vol. 1, No. 1, pp. 67–82 (April 1997)
8. Zhou, H.: Ensemble Methods Foundations and Algorithms. Machine Learning & Pattern Recognition Series, Chapman & Hall/CRC, Microsoft Research Ltd. Cambridge, UK, pp. 135–156 (2012)
9. Valle Chávez, Figueroa Nazuno, J.: Recuperación y Comparación de Figuras en 3D. Tesis de la Maestría en Ciencias de la Computación del Centro de Investigación en Computación del Instituto Politécnico Nacional (2012)
10. Santos Camacho, E.A.: Clasificación de series de tiempo mediante una comparación elástica. Tesis de licenciatura, Centro Universitario UAEM Valle de México (2012)
11. Renteria Agualimpia, W., Figueroa Nazuno, J.: Análisis e Identificación de Cambios Abruptos en Arreglos Unidimensionales. IV Congreso Internacional en Tecnologías Inteligentes y de la Información CITII, pp. 1–11 (2008)

# Design of an Artificial Neural Network to Detect Obstacles on Highways through the Flight of an UAV

Dariel A. Islas Guzmán[1,2], J. Rodrigo Córdova Alarcón[1],
Adrián Alcántar Torres[2], Mario A. Mendoza Bárcenas[1]

[1] Instituto Politécnico Nacional, Centro de Desarrollo Aeroespacial, Mexico

[2] Instituto Politécnico Nacional, Escuela Superior de Física y Matemáticas, Mexico

dariel.islas@hotmail.com, jcordovaa@ipn.mx,
adrian.alcantar.torres@gmail.com, mmendozab@ipn.mx

**Abstract.** Due to several risks that involve travelling in highways, such as vehicle collision, natural disasters and other obstacles, the need of implementing an early warning system which detects obstacles to provide security has been growing in the last years. Artificial neural networks based systems have been successfully applied in obstacle detection through image processing and recognition. To address this issue, a Multilayer Feed-forward Network (MFN) was designed to detect obstacles in highways from a zenith perspective. *Backpropagation* algorithm is used to supervise the training of the proposed neural network by minimizing the square error function via descending gradient criteria. Performance analysis was performed by using binarized, grayscale and RGB images with constant size and considering several contexts. The proposed neural network was trained and tested offline. In a next stage of the project, this neural network will be implemented into the on-board computer of unmanned aerial vehicles (UAV's) as an early warning platform.

**Keywords:** Artificial neural network, image recognition, backpropagation, obstacle detection.

## 1 Introduction

The improvement of highway driving safety by reducing the number and severity of incidents has become a great challenge in recent years [9]. Obstacles on highways such as overturned cars, mudslides, landslides, lack of road sections and fallen trees, represent a danger for drivers.

Nowadays vehicles have been equipped with technological tools, which are able to determine the presence of an obstacle on the road and thus transmit a warning to the road users, with a certain degree of reliability. For example in [15,11,3] the obstacle detection is performed through stereoscopic vision. Other obstacle detection systems use radar-based methods such as ClearWay$^{TM}$, developed by Navtech Radar. Also, autonomous and semi-autonomous land vehicle systems have been developed to accomplish this objective [20,5,13].

Recently the UAV's are affordable platforms for the implementation of several applications and opens up opportunities to develop an efficient and effective obstacle detection system for highways. Obstacle detection task performed during the flight of an UAV can be raised as a problem of pattern recognition, which is one of the main topics in computational vision and artificial intelligence. In order to solve the pattern recognition problem, the development of an algorithm which assigns any image in a representative class is performed [4,16].

There are a large number of papers which present several types of obstacle detection methods. Stereo vision, laser radar and computer vision are the most common methods of obstacle detection [23]. In [17], a Multi-Layer Perceptron (MLP) to detect traffic incidents from changes of quantitative data (traffic quantity per minute, velocity, occupancy rate) was proposed. They confirmed its effectiveness by adding enough combinations of similar average change patterns in order to increase the recognition rate and reduce the number of false detections. In [23] a Convolutional Neural Network it was proposed to detect contours of obstacles in streets. In [13] an unmanned ground vehicle to detect obstacles during off-road autonomous navigation by stereo vision was presented. Referring to [14], an obstacle detection system based on vision by the flight of a UAV has been implemented by segmenting sky and non-sky images and finding obstacles in segmenting sky afterwards.

The images for the detection of obstacles in previous work, are from an aerial perspective, while the problem from an overhead perspective has not been addressed. Therefore, we refer to an obstacle as any visible object in a highway from zenith perspective in this work in the next paragraphs.

In this work, the development of a neural network based system for detecting obstacles on highways is presented. We designed and analyzed the performance of the neural network by modifying its topology and using three different types of images: binarized, gray scaled and RGB images. With the obtained results, an obstacle detection system can be implemented into the on-board computer of an UAV by pattern recognition, which is being developed in our institute, seeking high degree of reliability, low operating cost and flexibility for use in any type of computational infrastructure for the prevention of highways accidents.

## 2 Artificial Neural Networks and Pattern Recognition

Pattern recognition is the mechanism of relating things and classifying them into different categories by taking and explaining decisions [21]. To solve pattern recognition problem, several approaches has been developed and implemented into a great variety of applications [2]. In this work, neural networks are used as our approach.

The neural network model is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use [8,7].

The information-processing unit of a neural network is called neuron, each neuron is formed by a vector of signals inputs $x = [x_1, x_2, \ldots, x_n]^T$, a vector of weights $w = [w_1, w_2, \ldots, w_n]^T$ and a nonlinear mapping $f : \mathbb{R} \mapsto [0, 1]$ called transfer function, that is non-decreasing, such that $\lim_{x\to\infty} f(x) = 1$ and $\lim_{x\to-\infty} f(x) = 0$ (e.g. Sigmoid Function). A neuron $k$ is described mathematically by the following equations.

$$y_k = f(u_k + b_k), \qquad u_k = \sum_{j=1}^{m} w_{kj} x_j, \tag{1}$$

where $y_k$ is the response of the $k$-th neuron due an input signal $x$, $f$ is the transfer function and define the output of the neuron, $u_k$ is a linear combination between input signal and synaptic weights taking $m$ units having connections to $j$-th unit, $b_k$ is the bias, whose value allows to control the behavior of the layer, and contribute for a successful learning.

Learning is a process by which the free parameters of a neural network are adapted though a process of simulation by the environment in which the network is embedded.

Considering $j = 1, \ldots, k$ neurons outputs associated with each input signal, the error signal at the output of $j$-th neuron at $n$ iteration is defined by $e_j(n) = d_j(n) - y_j(n)$, where $d_j(n)$ is the desired output for $j$-th neuron and $y_j(n)$ is the output of $j$-th neuron at $n$ iteration given by equation 1.

The total error energy obtained by the sum of all neurons in the output layer is defined by

$$E(n) = \frac{1}{2} \sum_j (e_j(n))^2. \tag{2}$$

By the training process, the network parameters are adjusted under the combined influence of the training vector and the error signal. The adjustment of the parameters is carried out iteratively until the neural network emulate the training vector. The *backpropagation* algorithm is efficient for training MLP with training vector, the weights are updated according of the minimization of the total error energy (Eq.2). Then, each weight value is softly adjusted in the direction of the negative of the gradient according with the following equation

$$\Delta W_{ki} = -\eta \frac{\partial E}{\partial W_{ki}} \quad \Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}}, \tag{3}$$

where $\eta$ is a proportionality constant called the learning-rate, the smaller the learning-rate, the smaller the changes to the weight from one iteration to the next. With the aim to accelerate the learning process, a momentum term $\alpha$ is incorporated in updating weight, the momentum term keep the moving in a certain direction, adding a fraction of the previous change to the new one as follows $\alpha\Delta w_{ki}(n) = \alpha(w_{ki}(n) - w_{ki}(n-1))$.

In the case of single hidden layer, $W_{ki}$ denotes the connection of the $i$-th neuron in input layer to $k$-th neuron of the hidden layer, and similarly $w_{jk}$ denotes the connection of the $k$-th neuron to the $j$-th neuron which belongs to the output layer.

For measuring network efficiency, we consider the Mean Square Error (MSE) criteria, which represents the cost function as a measure of learning performance for a given training set, MSE is updated at each epoch and is defined by the equation (4), where is considered N training patterns:

$$MSE = \frac{1}{N} \sum_{p=1}^{N} E(n). \qquad (4)$$

After the training of the network, a validation is performed with a set of patterns that are not used during the learning process, with the goal to appreciate the quality of the net for pattern recognition [8,22,10].

## 3 Experiments and Results

In this section, we present our experiments and results of the proposed MFN for obstacle detection by images recognition, built by a single hidden layer and log-sigmoid like transfer function. The *backpropagation* algorithm is used for the training of the network. A single hidden layer network with enough neurons is an universal approximator [10]. If the machine for pattern recognition is designed as a MFN using a supervised learning algorithm, the task of feature extraction is performed by the computational units in the hidden layers of the network [8].

### 3.1 Data Set

For the training phase and validation of the network, a database consisting of images of highways with obstacles and without obstacles was built. Such images were acquired via the Internet. We consider images with natural sunlight, an average size of $1000 \times 500$ pixels and taken from a zenith perspective. We also consider images with obstacles such as cars, mudslides, landslides and lack of road sections, as well as images without the above items. Examples of these images are shown in elements of the Figure 1.

The images of the database are samples of constant dimensions of $71 \times 71$ pixels, some with context, i.e. the images have characteristics of the environment in which the highway is located. Contextual influences on object recognition are evident when local features are insufficient because the object is small, occluded or camouflaged [18].

For the database, two subsets of images are created: training and validation. These subsets are formed from a percentage of the total image to consider, for each image 14 samples from seven shifts and seven rotations are obtained. Recognition systems improve efficiency if distortions of the images are included in the training patterns: displacements and rotations [12,1]. This article proposed

(a) (b)

**Fig. 1.** Images database, (a) obstacle image, (b) unobstructed image.

that the rotations are performed with an integer random angle of between 10 and 350 degrees, considering that the route of the UAV will be defined and images taken may have some rotation of the initial reference frame, displacements are considered so that samples are not single context. In Figure 2 can see a set of samples with context, rotation and displacement.

**Fig. 2.** Sample images.

### 3.2 Training Patterns

Three types of image pre-processing have been considered to form the training patterns.

1. The images were converted to grayscale and then binarized by the Otsu method [1].
2. The images were converted to grayscale and normalized according to the mean and standard deviation for each sample.
3. The images are considered in the RGB color space and normalized according to the mean and standard deviation for each sample.

Each sample is a subset of the scanned image, so we consider the samples as a matrix of $71 \times 71$ elements, where each element is the pixel value at that point, each matrix corresponding to each sample, is written as a column vector, transposing each row of the matrix in a column vector of $5041 \times 1$ for the first two pre-processing and $15,123 \times 1$ for the latter. These vectors contain the information that the net use to classify each pattern.

[1] A description of the algorithm to find global binarization levels can be found in [6]

### 3.3 ANN Topology

The structure of the proposed ANN is MFN. In Figure 3, a scheme of information processing of an image in the RGB color space is shown.

In this application, the number of entries is associated with each digitized sample. In order to propose the best treatment for the classification of the images, experiments with each of the pre-processed images are performed, considering a training set of 2385 patterns for training and 477 patterns for validation of the recognition obtained from the net. There is only one output, whose value corresponds to the absence (0) or presence (1) of an obstacle.

For the determination of the number of neurons in the hidden layers, there are different heuristic criteria. As the number of inputs increases, the ratio of hidden-layer neurons to inputs decreases [19]. Successive iterations of testing and error were performed to find a efficient and effective topology. The proportion of the training and validation sets corresponds to 80% and 20% respectively of the images in database. The parameters of the ANN are set small for each experiment, the learning-rate $\eta = 0.01$ and momentum factor $\alpha = 0.001$. The hidden and output layer have a log-sigmoid like transfer function.

**Table 1.** Simulation with different preprocessed images.

| Binarized images | | | | | |
|---|---|---|---|---|---|
| Topology | MSE | Epoch | Time | Training | Recognition |
| 5041 : 1000 : 1 | 0.0091 | 41 | 6h 3m | 64.31% | 56.21% |
| 5041 : 2000 : 1 | 0.0059 | 57 | 7h 55m | 57.62% | 51.94% |
| 5041 : 2000 : 500 : 1 | 0.013 | 28 | 7h 45m | 54.38% | 51.94% |
| Grayscale images | | | | | |
| 5041 : 2000 : 1 | 0.0051 | 74 | 16h 25m | 55.30% | 50.27% |
| 5041 : 1000 : 500 : 1 | 0.014 | 75 | 5h 57m | 47.33% | 48.05% |
| RGB images | | | | | |
| 15123 : 1 | 0.0239 | 124 | 81.70s | 95.22% | 73.80% |
| 15123 : 500 : 1 | 0.0001 | 124 | 6h 42m | 100% | 82.73% |
| 15123 : 300 : 1 | 0.0005 | 103 | 5h | 99.91% | 81.81% |
| 15123 : 4000 : 1 | 0.0005 | 200 | 60h 48m | 99.91% | 76.50% |
| 15123 : 500 : 100 : 1 | 0.0041 | 75 | 14h 30m | 97.31% | 78.33% |

As shown in Table 1, by using binarized and grayscale images, the training process last more than 6 hours and an incomplete training on the total pattern is achieved. Considering the RGB images, the ANN achieved more than 95% of training at different time given the size of the database. We simulated nets with images of small size of $36 \times 36$ pixels but there are slight improvements in the training and validation process, which we associate that with this type of

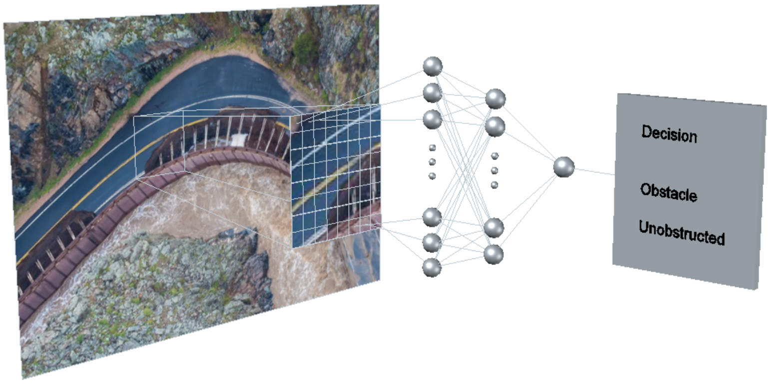



**Fig. 3.** Information processing. The image is taking from databse, then is considered in the RGB color space and normalized, the values resulting from this operation are the entries of the net, then the classifier that is the MFN proposed take a decision, about whether or not there is an obstacle in the image.

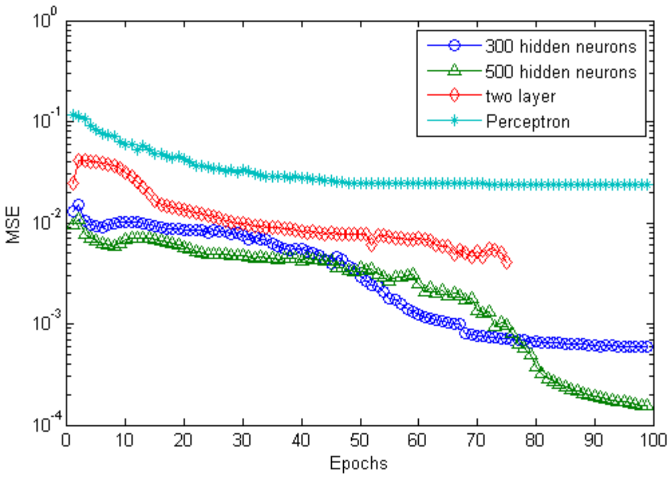



**Fig. 4.** Comparison of the learning curves associated with different topologies.

network structure is inadequate for treatment with highways images in grayscale and binarized.

In Figure 6, a set of images that the network could not classified correctly is shown. In the first three items, we can see that the obstacle is very small, so the lack of similar patterns in the training set result in misclassification. In the fourth image we can see that much of the sample is context, then the response of ANN result in misclassification.

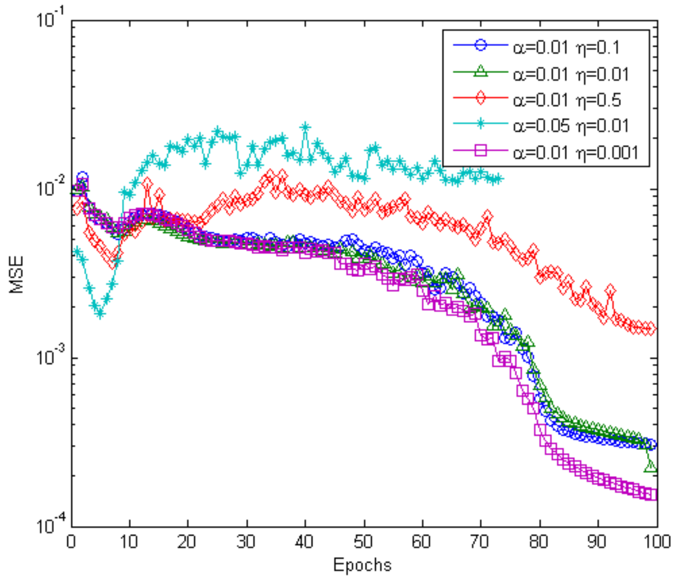



**Fig. 5.** Comparison of the different parameters.

**Fig. 6.** Misclassified images.

### 3.4 ANN Parameters

To find the learning rate and momentum factor of the parameters of the RNA, various tests were performed considering the same topology, weights are generated by the implementation in Matlab of the generator of observations of a uniform variable MT19937 with seed 2085268696. Via experimentation, we obtain the following results:

For values of the learning rate $\alpha \in [0.05, 1]$, the network has been unstable manner.
For values of the momentum $\eta \in [0.6, 1]$, the network has been unstable manner, due the factor is too large.
The best values for parameters are $\eta = 0.01$ and $\alpha = 0.001$

In Figure 5 a comparison of different parameters is shown, we plots the best learning curves selected for our experiments, which they consisted of simulating nets with different parameters.

## 4 Conclusions

In this work, a design of a MFN with a single hidden layer has been proposed. The MFN was trained by using normalized RGB images, as we have seen for treatment of grayscale and binarized images, a different structure of ANN is required, so we conclude that when objects in the images are indistinguishable with the background, the color of the objects is a powerful feature for pattern recognition.

For the optimal values of the learning-rate and momentum , we use the definition: The $\eta$ and $\alpha$ that, for either the worst-case or on average, yield converge to the global minimum in the error surface with the least number of epochs. With the results obtained, we propose an MFN with a single hidden layer with topology 15123:500:1, learning-rate $\eta = 0.01$ and momentum term $\alpha = 0.001$. The transfer function of the net is log-sigmoid for the hidden and output layer. With this topology and parameters we get the best result of recognition of 82.73% with images not used in training and of several highways.

As future work the MFN will be tested in real world controlled scenarios into the on-board computer of an UAV. The features of the considered obstacles are limited to static objects and cars. Dynamic obstacles are not considered in this work, so it is necessary to design an algorithm to make the recognition of cars in proper circulation and those that are an obstacle as detained cars and overturned cars. Regarding the training process, database should be expanded, because as we have seen the net misclassified small object associated with small obstacles on highways as persons or stones. In terms of image acquisition and processing, filtering processes are necessary for image reduction noise and thus obtain images suitable for classification.

## References


1. Baidyk, T., Kussul, E.: Redes neuronales, visión computacional y micromecánica. Editorial Itaca (2009)
2. Basu, J.K., Bhattacharyya, D., hoon Kim, T.: Use of artificial neural network in pattern recognition. International Journal of Software Engineering and Its Applications (2010)
3. Bertozzi, M., Broggi, A.: A parallel real-time stereo vision system for generic obstacle and lane detection. IEEE Transactions on Image Processing 7(1), 62–81 (1998)
4. Bishop, C.M.: Pattern Recognition. Clarendon Press, Birmingham, UK (1995)
5. Chen, J., Zhao, P., Liang, H., Mei, T.: Motion planning for autonomous vehicle based on radial basis function neural network in unstructured environment. Journal Sensors (2014)
6. Chen, Z., Tu, Y.: Improved image segmentation algorithm based on OTSU algorithm. International Journal of Advancements in Computing Technology (IJACT) 4(15) (2012)
7. Cruz, P.P.: Inteligencia artificial con aplicaciones a la ingeniería. Alfaomega, México (2010)

8. Haykin, S.: Neural Networks:a comprehensive foundation. Prentice-Hall Inc., Ontario, Canada (1998)
9. Hinojosa, C.: Utilizando sistemas de gestión inteligente para mejorar la seguridad vial. Versión electrónica presentada en Actualidad vial, Zaragoza, España (2013)
10. Hornik, K., Stinchcombe, M.: Multilayer feedforward networks are universal approximators. Neural Networks 2(5), 359–366 (1989)
11. Huha, K., Park, J., Hwang, J., Hong, D.: A stereo vision-based obstacle detection system in vehicles. Optics and Lasers in Engineering 46(2), 168–178 (2008)
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceeding of the IEEE 86(11), 2278–2344 (1998)
13. Liu, F., Cao, B.: A neural network enhanced stereo vision obstacle detection and avoidance system for unmanned ground vehicle. 2nd International Conference on Advances in Computer Science and Engineering (2013)
14. McGee, T., Sengupta, R., Hedrick, K.: Obstacle detection for small autonomous aircraft using sky segmentation. Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA) pp. 4679–4684 (2005)
15. Nedevschi, S., Danescu, R., Frentiu, D., Marita, T., Oniga, F., Pocol, C., Schmidt, R., Graf, T.: High accuracy stereo vision system for far distance obstacle detection. In: IEEE Intelligent Vehicles Symposium. Parma, Italy (2004)
16. Ochoa, J.A.C., Trinidad, J.F.M.: Reconocimiento de patrones. Komputer Sapiens III(II) (2011)
17. Ohe, I., Kawashima, H., Kojima, M., Kaneko, Y.: A method for automatic detection of traffic incidents using neural networks. Bulletin of Mathematical Biophysics (1995)
18. Oliva, A., Torralba, A.: The role of context in object recognition. TRENDS in Cognitive Sciences (2007)
19. Priddy, K.L., Keller, P.E.: Artificial neural networks: an introduction. The International Society for Optical Engineering, Bellingham, USA (2005)
20. Raju, S., Sanjay, K., Kumar, T.S., Madhini, B.: Semi autonomous vehicle to prevent accident. International Journal of Technology Enhancements and Emerging Engineering Research 2(5), 42–45 (2014)
21. Ramirez Q., J.A., Chacón M., M.I.: Redes neuronales artificiales para el procesamiento de imágenes, una revisión de la última década. Revista de ingeniería eléctrica, electrónica y computación 9(1) (2011)
22. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review 65(6) (1958)
23. Yu, H., Hong, R., Huang, X., Wang, Z.: Obstacle detection with deep convolutional neural network. Sixth International Symposium on Computational Intelligence and Design (2013)

# Estabilización orbital de un robot móvil con ruedas tipo uniciclo: síntesis y validación experimental

Maugro Hernández-Ruiz[1], Iliana Marlen Meza-Sánchez[2]

[1] Tecnológico Nacional de México, Instituto Tecnológico de Ensenada, Ensenada BC, México

[2] Cátedra CONACYT, Instituto Tecnológico de Ensenada, Ensenada BC, México,

mhernandez@ite.edu.mx, immezasa@conacyt.mx

**Resumen.** En este trabajo se propone el desarrollo de un controlador por rediseño de Lyapunov para generar movimiento periódico en un robot móvil con ruedas (rmr) tipo uniciclo. Se utiliza la solución de un sistema dinámico de segundo orden que exhibe un ciclo límite para generar la órbita deseada; este sistema dinámico permite especificar el radio de la órbita y el sentido de giro mediante la configuración de dos parámetros. A partir de este análisis, se diseña un controlador basado en el modelo cinemático del robot móvil para asegurar su convergencia hacia el movimiento periódico deseado y se realiza un análisis de estabilidad para la dinámica en lazo cerrado. Se presentan resultados experimentales para demostrar la efectividad y robustez del esquema propuesto utilizando una plataforma de visión en tiempo real.

**Palabras clave:** Estabilización orbital, robot móvil tipo uniciclo, control por rediseño de Lyapunov.

## Orbital Stabilization for a Unicycle-Type Wheeled Mobile Robot: Synthesis and Experimental Validation

**Abstract.** In this work, a Lyapunov-based synthesis and control design to induce periodic motion in a unicycle-type wheeled mobile robot (wmr) is under study. The solution of a second-order dynamical system is used as desired trajectory to achieve orbital stabilization. The selected dynamical reference system exhibits a limit cycle behavior which allows to move from one orbit to another by simply changing the parameters regarding radius of the orbit and direction of rotation. From this analysis, a kinematic tracking controller is designed for which closed-loop dynamics and stability analysis are provided to demonstrate control problem resolution based on error dynamics. Performance and robustness issues for the proposed framework are illustrated by experimental results using a real-time vision control system.

**Keywords:** Orbital estabilization, unicycle-type mobile robot, Lyapunov redesign control.

## 1. Introducción

La estabilización orbital en la Robótica es un problema de control que ha sido abordada en su mayoría para sistemas mecánicos y bípedos con la finalidad de generar trayectorias cíclicas parametrizadas entre eslabones así, como también en la generación de patrones de caminata [1,2,3]. Recientemente, sobre robots móviles con ruedas, se han desarrollado múltiples aplicaciones basados en este enfoque. Por ejemplo, en [4] se presenta el desarrollo de un controlador para navegación autónoma y evasión de obstáculos utilizando un método heurístico en línea combinado con un filtro extendido de Kalman (EKF) para generar elipses alrededor de los óbstaculos; un algoritmo de control óptimo para seguimiento y regulación para robots móviles diferenciales no holonómicos utilizando balizas de RF es descrito en [5]; el problema de estabilización por control adaptivo utilizando un sistema de visión es desarrollado en [6]. Problemas más complejos son abordados por ejemplo por [7], en donde se presenta una propuesta de solución para coordinación de múltiples robots en formación hacia un objetivo donde las trayectorias son generadas por un oscilador neuronal; la implementación de comportamientos cooperativos utilizando control descentralizado para seguimiento de trayectorias periódicas es propuesto en [8].

Como puede observarse, la estabilización orbital es un problema de control que permite el desarrollo de nuevas propuestas de solución para problemas cada vez más complejos. La teoría de control aborda este problema bajo dos esquemas principales; ya sea buscando soluciones periódicas dentro de la dinámica del sistema generadas por variación paramétrica como por ejemplo propone [9], o utilizando sistemas dinámicos de referencia que de manera intrínseca exhiban este comportamiento como se presenta en [10]. El trabajo que se presenta pertenece a esta última forma.

En esta propuesta, se desarrolla un controlador para resolver el problema de regulación con una trayectoria deseada que presenta velocidades de desplazamiento y movimiento angular constantes en estado estable. La generación de la trayectoria priódica está definida por un sistema dinámico de segundo orden que presenta una órbita estable de manera natural.

Tradicionalmente, la generación de movimiento periódico se define como la resolución de un problema de seguimiento para una trayectoria descrita por velocidades lineal y angular constantes (Véase por ejemplo [11]); sin embargo, en esta propuesta se toma ventaja de la dinámica de un sistema dinámico de referencia que converje a un órbita periódica, con sentido de giro y dirección configurables, para desarrollar un controlador que resuelve un problema de regulación para imponer estabilización orbital en un robot móvil con ruedas tipo uniciclo. En este caso, no es necesario garantizar la convergencia de los estados del robot ya que el objetivo de control está especificado con respecto al radio y dirección del movimiento deseado. La ventaja de establecer este planteamiento consiste en que el controlador no necesita compensar la velocidad de la trayectoria deseada y dado que el sistema dinámico de referencia converje a un ciclo límite, es suficiente con garantizar estabilidad asintótica para un problema de regulación.

El presente trabajo se encuentra organizado de la siguiente manera. En la Sección 2 se presenta la descripción del problema de control que se pretende resolver, se describe el modelo cinemático del robot móvil y el sistema dinámico de referencia. La síntesis de control basada en rediseño de Lyapunov y el análisis de estabilidad en lazo cerrado se presenta en la Sección 3. Los resultados experimentales utilizando una plataforma de visión con procesamiento en tiempo real para el sistema bajo estudio se muestra en Sección 3. Finalmente, resumimos las conclusiones y comentarios finales en la Sección 5.

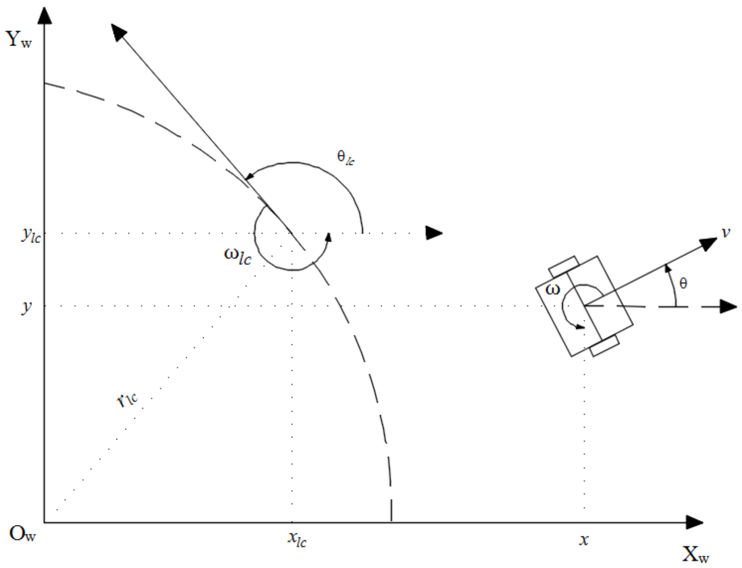



Fig. 1: Postura del rmr definida por $q = [x\ y\ \theta]^T$ y movimiento deseado descrito por $q_{lc} = [x_{lc}\ y_{lc}\ \theta_{lc}]^T$ con respecto al marco fijo $XOY_\omega$.

## 2. Descripción del problema de control

El objetivo de control es la generación de movimiento periódico en un robot móvil con ruedas tipo uniciclo por lo que se propone el uso de un sistema dinámico de segundo orden como modelo de referencia cuya dinámica converje a una trayectoria periódica alrededor del punto de equilibrio definido originalmente por el origen. A continuación, se presenta el modelo cinemático del rmr tipo uniciclo así como el modelo dinámico de referencia que ha sido seleccionado. En la Figura 1, se muestra el problema de control donde la trayectoria deseada y la postura del rmr están definidas por los vectores de estados $q_{lc}$ y $q$, respectivamente; ambas se encuentran descritas con respecto al marco fijo $\mathrm{XOY}_w$.

### 2.1. Modelo cinemático

Una de las principales características de los robots móviles con ruedas tipo uniciclo consiste en la restricción de movimiento en la cual sólo pueden desplazarse en dirección normal al eje de las ruedas. Esta característica propia de esta arquitectura puede expresarse matemáticamente como

$$\dot{y}\cos\theta - \dot{x}\,\mathrm{sen}\,\theta = 0, \tag{1}$$

donde $\dot{x}, \dot{y}, \theta \in \mathbf{R}$ corresponden a las velocidades lineales en el eje $x$, en el eje $y$, y al ángulo de orientación, respectivamente. Considerando la restricción de movimiento descrita por (1), el modelo cinemático del robot móvil con respecto al marco fijo $\mathrm{XOY}_w$, está definido como

$$\dot{q} = \begin{bmatrix} \cos\theta & 0 \\ \mathrm{sen}\,\theta & 0 \\ 0 & 1 \end{bmatrix} u, \tag{2}$$

donde las coordenadas generalizadas están descritas por el vector de estados $q = [x\ y\ \theta]^T$, donde $x, y \in \mathbf{R}$ corresponden a las posiciones con respecto al plano $\mathrm{XY}_w$, y la orientación con respecto al eje $\mathrm{X}_w$ es descrito por el ángulo $\theta$; $u = [v\ \omega]^T$ define al vector de entradas de control denotado por las magnitudes de las velocidades lineal y angular, respectivamente.

### 2.2. Modelo dinámico de referencia: ciclo límite

El modelo seleccionado por exhibir un movimiento periódico configurable a través de dos parámetros para definir el radio de la órbita y el sentido de giro es el introducido en [10]; sin embargo, es importante señalar que existe una gran variedad de sistemas dinámicos clásicos y algunos propuestos en literatura que presentan este comportamiento y pueden ser utilizado bajo el esquema de esta propuesta.

El sistema de referencia seleccionado exhibe un ciclo límite y su dinámica está definida por

$$\dot{x}_{lc} = x_{lc} - \delta y_{lc} - x_{lc}(x_{lc}^2 + y_{lc}^2)\frac{1}{r_{lc}^2} \tag{3}$$

$$\dot{y}_{lc} = \delta x_{lc} + y_{lc} - y_{lc}(x_{lc}^2 + y_{lc}^2)\frac{1}{r_{lc}^2}, \tag{4}$$

donde $r_{lc} \in \mathbf{R}$ es una constante positiva que define el radio de la órbita y $\delta = \{+1, -1\}$ al sentido de rotación del movimiento. Redefínase este sistema de referencia (3),(4), a un estructura equivalente dado por

$$\dot{q}_{lc} = \begin{bmatrix} \cos\theta_{lc} & 0 \\ \operatorname{sen}\theta_{lc} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{lc} \\ \omega_{lc} \end{bmatrix}, \tag{5}$$

donde $\dot{q}_{lc} = [\dot{x}_{lc} \ \ \dot{y}_{lc} \ \ \dot{\theta}_{lc}]^T \in \mathbf{R}^3$, corresponde al vector de velocidades del movimiento en los ejes $x$, $y$, $X_\omega$, respectivamente; $v_{lc}, \omega_{lc} \in \mathbf{R}$ se redefinen como las velocidades lineal y angular de (3), (4).

La descripción de la reformulación definida en (5) puede ser obtenida mediante [9]

$$v_{lc} = \sqrt{x_{lc}^2 + y_{lc}^2}, \ \ \omega_{lc} = \frac{\dot{x}\ddot{y}_{lc} - \dot{y}_{lc}\ddot{x}_{lc}}{\dot{x}_{lc}^2 + \dot{y}_{lc}^2}, \ \ \theta_{lc} = \operatorname{atan2}(\dot{y}_{lc}, \dot{x}_{lc}), \tag{6}$$

donde $\theta_{lc}$ corresponde al ángulo de orientación del sistema dinámico considerandolo básicamente como si se tratara de un robot móvil de referencia; $\ddot{x}_{lc}$, $\ddot{y}_{lc}$ se refiere a las derivadas temporales de (3), (4), respectivamente.

## 3. Síntesis de control y análisis de estabilidad

Como se muestra en la Figura 1, el ángulo de orientación es periódico con un periodo definido por $T = 2\pi$. Además, considerando las propiedades del sistema de referencia definido por (3) y (4), el problema puede ser formulado como un control de posición a lo largo de la trayectoria deseada. Obsérvese que dado que el sistema de referencia posee un ciclo límite estable, esto implica que cualquier trayectoria, sin importar donde inicie, convergerá hacia una órbita de radio $r_{lc}$ en un sentido de rotación definido por $\delta = \{1, -1\}$.

En consecuencia, el objetivo de control está descrito como el diseño de una ley de control $(v, \omega)$, tal que el robot móvil converja a la órbita generada por (3), (4); esto es

$$\lim_{t\to\infty} ||r_{lc} - r|| = 0 \tag{7}$$

donde $k \in \mathbb{Z}$ y el error de la distancia en $x$, $y$ del robot móvil con respecto a la órbita están definidas por $r_{lc} = \sqrt{x_{lc}^2 + y_{lc}^2}$ y $r = \sqrt{x^2 + y^2}$.

Primero, defínase el vector de errores de estado $q_e = T(\theta)(q_c - q) \in \mathbf{R}^3$ donde $q = [x, y, \theta]^T$, $q_c = [x_c, y_c, \theta_c]^T$ corresponden a los vectores de postura del robot móvil y al de la trayectoria deseada, respectivamente. Por lo tanto, el error de posición está dado por

$$q_e = T(\theta)(q_c - q) = \begin{bmatrix} \cos\theta & \operatorname{sen}\theta & 0 \\ -\operatorname{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} \tag{8}$$

donde $x_e = (x_c - x), y_e = (y_c - y)$, $\theta_e = (\theta_c - \theta)$. La dinámica del error puede ser obtenida a través de la derivada temporal de (8) como

$$\dot{q}_e = \begin{bmatrix} \cos\theta_e & 0 \\ -\operatorname{sen}\theta_e & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} + \begin{bmatrix} -1 & y_e \\ 0 & -x_e \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix},$$

o reformulada de manera equivalente,

$$\dot{x}_e = v_{lc}\cos\theta_e + y_e\omega - v, \quad (9)$$

$$\dot{y}_e = v_{lc}\operatorname{sen}\theta_e - \omega x_e, \quad (10)$$

$$\dot{\theta}_e = \omega_e, \quad (11)$$

donde $\omega_e = \omega_{lc} - \omega$.

Para garantizar el cumplimiento del objetivo de control imponiendo la convergencia de la dinámica del error (8) a la variedad (7), se propone una ley de control dada por

$$v = v_{lc}\cos\theta_e + K_x x_e \quad (12)$$

$$\omega = \omega_{lc} + K_\omega v_{lc} y_e + K_\omega \operatorname{sen}\theta_e (1 + K_y y_e^2) \quad (13)$$

donde $K_x, K_y, K_\omega > 0$.

Es importante señalar que la propuesta de control resuelve el problema de regulación de postura del robot móvil con respecto al sistema de referencia que exhibe un ciclo límite estable; i.e. el objetivo es garantizar estabilidad con respecto a (7).

**Teorema 1.** *Sea el sistema libre de perturbaciones (2) restringido por (1) y con dinámica de error definida por (8), donde se aplica la ley de control descrita por (12), (13) y está sujeto a una trayectoria definida como una sucesión de posiciones con respecto a la solución del sistema dinámico de segundo orden (5). Entonces, el sistema en lazo cerrado dado por (2), (12), (13) es globalmente asintóticamente estable con* $K_x, K_y, K_\omega > 0$.

*Demostración.* Propóngase la siguiente función candidata

$$V = \frac{1}{2}(x_e^2 + y_e^2) + K_\omega^{-1}(1 - \cos\theta_e) \quad (14)$$

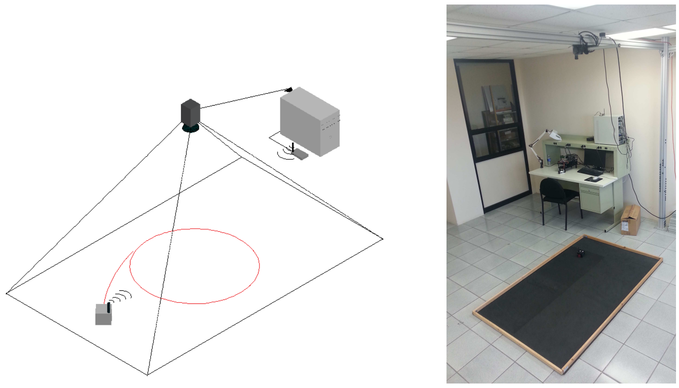
donde $K_\omega > 0$ es una constante positiva. Entonces, la derivada temporal de (14) está dada por

$$\dot{V} = x_e\dot{x}_e + y_e\dot{y}_e + K_\omega^{-1}\operatorname{sen}\theta_e\dot{\theta}_e. \quad (15)$$

Tomando en cuenta los errores en la dinámica (9), (10), (11), y sustituyendo (12), (13) en (15) tenemos

$$\dot{V} = -K_x x_e^2 - K_\omega^{-1}\operatorname{sen}^2\theta_e(1 + K_y y_e^2) \leq 0 \quad (16)$$

con $K_x, K_y, K_\omega > 0 \in \mathbf{R}$.



(a) (b)

Fig. 2: Configuración experimental utilizando una plataforma de visión con procesamiento en tiempo real: (a) esquema, b) sistema físico de laboratorio.

Inspirado en la línea de razonamiento de [9], podemos establecer que dada (16), la función candidata de Lyapunov propuesta $V(t)$ definida en (14) es decreciente y por lo tanto, posee un límite superior conforme $t \to \infty$; i.e. (14) es una función $\mathcal{L}_\infty$. Esta propiedad permite concluir entonces que el vector de error de estados $[x_e \quad y_e \quad \theta_e]^T$, sus derivadas temporales $[\dot{x}_e \quad \dot{y}_e \quad \dot{\theta}_e]^T$, así como tambiíen la ley de control (12),(13) son funciones $\mathcal{L}_\infty$ dado que $K_x$, $K_y$, $K_\omega$ son constantes y $v_{lc}$, $\omega_{lc}$ del sistema de referencia también son funciones acotadas. La estabilidad global asintótica para $x_e$ y $\theta_e$ puede ser rápidamente concluida mediante el cálculo de la integral de (16) para establecer que $x_e$ y $\theta_e$ son funciones $\mathcal{L}_2$, y aplicar la versión extendida del Lema de Barbalat (Véase Apéndice A).

Con el resultado anterior, dado que $\lim_{t\to\infty} \theta_e = 0$ y la igualdad (11) se cumple, la dinámica del error para la velocidad angular $\omega_e$ tambien converge a cero; por lo tanto, sustituyendo (13) en (11) se obtiene que $\lim_{t\to\infty} K_\omega v_{lc} y_e \to 0$. En consecuencia, se concluye también que $\lim_{t\to\infty} y_e = 0$ debido a que $v_{lc}$ es uniformemente continua y diferente de cero, y $K_\omega$ es una constante positiva. Resumiendo ambos resultados, el sistema en lazo cerrado (2), (12), (13) es globalmente asintóticamente estable y la validez del Teorema 1 queda demostrada. □

*Remark 1.* Considerando que el origen del sistema dinámico de segundo orden utilizado como referencia es un punto de equilibrio inestable, el robot móvil no puede ser inicializado en este punto. En la práctica, pequeñas perturbaciones y errores de precisión numérica en las mediciones permiten evitar esta situación; además es posible agregar de manera artificial una pequeña desviación a la medición de postura.

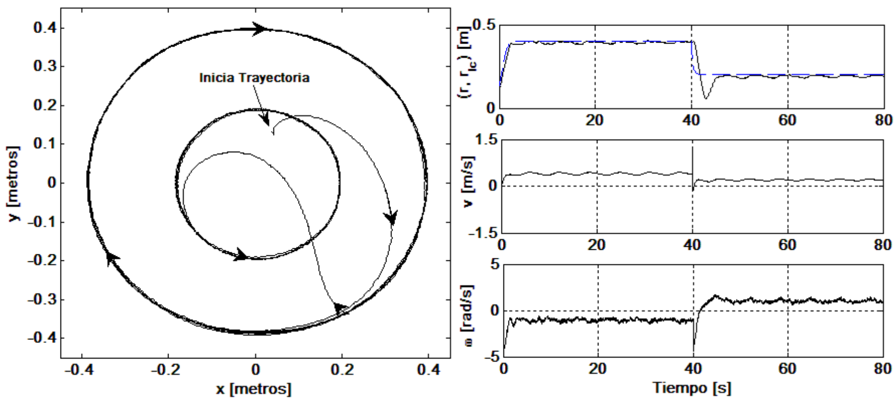



Fig. 3: Resultado experimental para el caso sin perturbaciones con condiciones iniciales $q(0) = [0{,}0394 \quad 0{,}1316 \quad 2{,}338]^T$. En $(r, r_{lc})$, la línea sólida corresponde al radio $r$ [m] del movimiento de rmr y la punteada al radio $r_{lc}$ ={0.25,0.4} [m] del sistema de referencia (3),(4) con $\delta$ ={-1,1}.

## 4. Resultados experimentales

Se realizaron experimentos utilizando una plataforma de visión en tiempo real para validar la efectividad del diseño de control propuesto. Esta plataforma se encuentra en el Laboratorio de Robótica Avanzada del Instituto Tecnológico de Ensenada basada en el sistema propuesto en [12] y se muestra en la Figura 2. Este sistema utiliza comunicación inalámbrica de RF para cerrar el lazo de control mediante un transmisor conectado al puerto paralelo de la PC y el tiempo de muestreo está definido como $t_m$ =0.015[s].

Los experimentos realizados consideran dos casos: a) el caso no perturbado en donde el robot móvil se mueve libremente de una órbita hacia otra, y b) el caso perturbado en donde manualmente se altera la trayectoria del robot móvil sacandolo de la órbita deseada. La trayectoria de referencia es generada mediante una integración numérica con un tiempo de muestreo $t_m$ =0.015 [s] en la plataforma de visión en tiempo real para obtener la solución del sistema dinámico de referencia.

Las condiciones iniciales para los experimentos sin perturbaciones y el perturbado son $q_1(0) = [0{,}0394 \quad 0{,}1316 \quad 2{,}338]^T$, $q_2(0) = [-0{,}6101 \quad 0{,}1910 \quad -0{,}3591]^T$, respectivamente. Las mediciones para $x$ y $y$ son en metros, y en radianes para el ángulo de orientación $\theta$. Las velocidades iniciales fueron definidas como $q_i = [0 \quad 0 \quad 0]^T$, $i = 1, 2$. El subíndice 1,2 se utiliza para el caso sin perturbaciones y el caso perturbado, respectivamente. Los valores de los parámetros del sistema dinámico de referencia son $r_{lc}$ =0.4 [m] y $\delta = -1$ para los primeros 40 [s], para posteriormente cambiar la órbita a un radio de $r_{lc}$ =0.25 [m] e invertir el sentido de giro con $\delta = 1$. Las ganancias seleccionadas para el

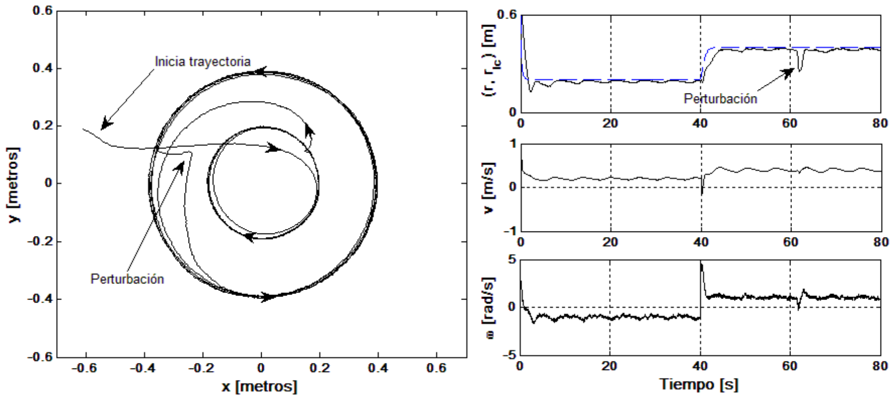



Fig. 4: Resultado experimental para el caso perturbado en $t$ ≈63[s] con condiciones iniciales $q(0) = [-0{,}6101 \quad 0{,}1910 \quad -0{,}3591]^T$. En $(r, r_{lc})$, la línea sólida corresponde al radio $r$ [m] del movimiento de rmr y la punteada al radio deseado $r_{lc}$ ={0.25,0.4} [m] del sistema de referencia (3),(4) con $\delta$ ={-1,1}.

controlador (12), (13) fueron $K_x = 0{,}4$, $K_y = 1$, y $K_\omega = 3$.

El resultado experimental para el caso no perturbado se muestra en la Figura 3 donde el controlador propuesto cumple con el objetivo de control estabilizando asintóticamente el movimiento periódico sobre el robot móvil. Además, en el inicio de su trayectoria presenta un desplazamiento en reversa para incorporarse a la órbita deseada. La gráfica $(r, r_{lc})$ muestra la comparación entre el radio deseado y el radio generado por el movimiento del rmr; $v$ y $\omega$ muestran las velocidades lineal y angular del robot. El resultado del experimento para el caso perturbado se muestra en la Figura 4; para este experimento, se desplaza manualmente al robot fuera de la órbita aproximadamente en $t$ = 63 [s] y el controlador logra que este recupere el movimiento periódico deseado. De ambas Figuras, podemos observar también que se logra la estabilización orbital sin importar si la postura inicial de robot móvil se encuentra fuera o dentro de la órbita deseada.

## 5. Conclusión y comentarios finales

Se ha presentado el diseño y la validación experimental de un controlador para regulación basado en rediseño de Lyapunov para inducir estabilización orbital en un robot móvil con ruedas tipo uniciclo. Además, se ha demostrado estabilidad asintótica del sistema en lazo cerrado. Los resultados experimentales utilizando una plataforma de visión con procesamiento en tiempo real ha demostrado su eficiencia al lograr el objetivo de control para un movimiento deseado generado por un sistema dinámico de referencia de segundo orden a pesar de que la solución de este último ha sido integrada numéricamente. El trabajo futuro pretende extender la estrategia de control propuesta para resolver problemas más complejos como control de formación y navegación autónoma mediante la generación de campos potenciales definidos por ciclos límites alrededor de los obstáculos.

**Agradecimientos.** Los autores desean agradecer al CONACYT Proyecto Cátedras 2459 denominado "Redes de Robots Móviles Colaborativos", al Tecnológico Nacional de México, al Instituto Tecnológico de Ensenada y a PRODEP.

## A. Lema de Barbalat y versión extendida

El lema de Barbalat es un resultado teórico bien conocido utilizado para analizar estabilidad en sistemas no lineales. Su forma original y su versión extendida se definen a continuación [9,13].

**Lemma 1.** *Sea* $f : \mathbf{R} \to \mathbf{R}$ *una función uniformemente continua en* $[0, \infty)$*. Supóngase que* $\lim_{t\to\infty} \int_0^t f(\tau)d\tau$ *existe y es finito. Entonces,* $f(t) \to 0$ *conforme* $t \to \infty$*.*

**Lemma 2.** *Sea* $f, \dot{f} \in \mathcal{L}_\infty$ *y* $f \in \mathcal{L}_p$ *para alguna* $p = [1, \infty)$ *entonces* $f(t) \to 0$ *conforme* $t \to \infty$*.*

## Referencias

1. Meza-Sánchez I.M., Aguilar L.T., Shiriaev A., Freidovich L., Orlov Y.: Periodic motion planning and nonlinear H-infinity tracking control of a 3-DOF underactuated helicopter. Int Journal of Syst Science 42(5), 829–838 (2011)
2. Garofalo G., Ott C., Albu-Schaffer A.: Orbital stabilization of mechanical systems through semidefinite Lyapunov functions. In: American Control Conference (ACC), pp. 5715–5721, Washington, DC, USA (2013)
3. Gamus B., Or Y.: Dynamic Bipedal Walking under Stick-Slip Transitions. SIAM (Society for Industrial and Applied Mathematics) J. Applied Dynamical Systems, 14(2), pp. 609-–642 (2015)
4. Vilca J., Adouane L., Mezouar Y.: Reactive Navigation of a Mobile Robot Using Elliptic Trajectories and Effective Online Obstacle Detection. Gyroscopy and Navigation, 4(1), pp. 14–25 (2013)
5. Miah M.S., Gueaieb W.: RFID-Based Mobile Robot Trajectory Tracking and Point Stabilization Through On-line Neighboring Optimal Control. J Intell Robot Syst, 78, pp. 377-–399 (2015)
6. Yang F., Wang C.: Adaptive stabilization for nonholonomic mobile robots with uncertain dynamics and unknown visual parameters. Transactions of the Institute of Measurement and Control, 37(2), pp. 282–288 (2015)
7. Sajjad M., Youngjin C: Neural oscillator-based multi-robot coordination algorithm to catch-Observe-Protect a target. In: IEEE Int Conf on Mechatronics and Automation (ICMA), pp. 1418–1423, Beijing, China (2015)

8. Sabattini L., Secchi C., Cocetti M., Levratti A., Fantuzzi C.: Implementation of Coordinated Complex Dynamic Behaviors in Multirobot Systems. IEEE Trans on Robotics, 31(4), 1018–1032 (2015)
9. Blazic, S.: On Periodic Control Laws for Mobile Robots. IEEE Trans on Industrial Electronics, 61(7), 3660–3670 (2014)
10. Hara N., Kokame H., Konishi K.: Circular Periodic Motion Generation for Mobile Robots Using Limit Cycle Systems. In: American Control Conference (ACC), pp. 4271–4276, Baltimore, USA (2010)
11. Wang Y., Miao Z., Zhong H., Pan Q.: Simultaneous Stabilization and Tracking of Nonholonomic Mobile Robots: A Lyapunov-Based Approach. IEEE Trans on Control Systems Technology, 23(4), pp. 1440–1450 (2015)
12. Bugarin E., Kelly R.: RTSVC: Real-time system for visual control of robots. Int. J. Imaging Syst. Technol., 18, 251–256 (2008)
13. Khalil H. K.: Nonlinear Systems. Second edition, Prentice-Hall (1996)

# Experimental-based Analysis of the Effect of Channel Errors in the Cluster Formation Phase in Wireless sensor Networks

Edgar Romo Montiel, Mario E. Rivero-Ángeles, Herón Molina Lozano, Rolando Menchaca Méndez

Instituto Politécnico Nacional, Centro de Investigacion en Computacion Mexico

eromom0900@alumno.ipn.mx, {erivero, hmolina, rmen}@cic.ipn.mx

**Abstract.** In this paper we present an experimental analysis of the performance of the cluster formation phase of a wireless sensor network. Unlike previous works, this analysis considers the effect of channel errors overran adaptive transmission probability scheme that is used in the cluster formation phase. Particularly, we study the impact of channel errors over the energy consumption and the average cluster formation time. The cluster scheme under study is based on the well-known LEACH protocol, which organizes the nodes of a wireless sensor network into groups in order to distribute the energy consumption of each node. However, in order to reduce processing time and energy consumption, we propose a variant of LEACH protocol where nodes are chosen to be cluster heads or cluster members based only on the order of arrival of nodes (called LEACH MOD), instead of whether or not a node has been a cluster head in the past.

**Keywords:** Wireless sensor network, adaptive transmission probability, LEACH protocol, channel error, energy consumption, cluster formation.

## 1 Introduction

Wireless Sensor Networks (WSNs) are technologies that provide a tool to have control, knowledge and monitoring of different sorts of physical variables of interest. WSNs allow collecting information of the environment depending on the application. Furthermore, WSNs use small nodes that can be installed inside a diversity of spaces, like a wall, a roof, under the pavement, among others. Unlike other wireless networks, WSNs require energy efficient communication among nodes in order to extend the system performance and lifetime. Indeed, in these networks, nodes can be located in places where they cannot be easily reached. As such, once the energy of a node is depleted, its battery cannot be replaced. When a certain number of nodes no longer function, the network loses its functionality. Also, nodes are provided only with a battery of small size and capacity [1]. Specifically, the main problems associated to energy depletion of nodes are: Information losses decrease and holes in the coverage area and broken links among nodes or base station (sink). Building on this, the energy consumption is even more critical in such scenarios where the coverage area is particularly wide, in that case, this increases due to the high-energy transmissions of nodes in order to reach the sink node.

To reduce the energy consumption in the system and distribute the energy load among nodes, a clustered-based protocol can be used. In this work, a clustering protocol is proposed, based on the well-known LEACH[2] protocol, called LEACH-MOD, which operates as follows: a) Nodes are chosen to be cluster heads or cluster members only based on the order of arrival of the nodes to reduce processing time and energy consumption. b) Nodes transmit in the cluster formation phase with a transmission probability $\tau$, which is adapted in each time slot and varies according to the number of nodes remaining to transmit their control packet. As such, the transmission probability increases in each successful transmission. In each time slot the sink calculates the adequate transmission probability based on the estimation of nodes attempting to transmit, even more, it is considered that a channel is not error-free and has a probability to obtain false negative or false positive errors described in following sections.

The rest of the paper is organized as follows: Section 2 describes the generalities of LEACH protocol, continued by the Proposed clustering scheme in section 3 where are described each consideration for this proposal. The parameters considered in the system model are given in section 4 and section 5 shows the results obtained by such system model. Finally, a brief conclusion is given in section 6.

## 2 LEACH Protocol

LEACH is a protocol that organizes nodes inside a wireless sensor network in clusters in order to distribute the energy load of each node. Under this scheme, all nodes organize themselves to take part in a unique cluster as cluster head (CH) or as cluster member (CM) [3]. The cluster head of each cluster (group of nodes) has the function of gathering the sensed data of each node that belongs to the cluster. When the cluster head has gathered all the cluster information, it sends the whole data to the sink.

We can easily see that CHs consume more energy since they have to be constantly receiving data from the CMs, and perform high-energy transmissions to the sink node once that information of the cluster is gathered. This means that CHs drain their energy faster. For that reason, LEACH gives and rotates the function of cluster heads among all nodes in the network in order to balance the energy consumption equitably. This rotation occurs periodically in time spaces called rounds and considers nodes that have not been CHs in previous round to be CHs. The LEACH protocol considers that each round is composed by two phases: the cluster formation phase and steady state phase [4]; after each round, clusters are broken and the cluster formation starts again selecting new CHs. In the cluster formation phase nodes are selected either as CHs or CMs and the steady state phase is where clusters are already established and data transmission is achieved from CMs to CH inside each cluster. In the former phase, a random access protocol is used. Specifically, the slotted NP-CSMA (Non-Persistent Carrier-Sense Multiple Access) scheme is applied, where nodes transmit in each time slot with probability $\tau$. In the steady state phase, a TDMA based scheme is preferred since the CH already knows the number of nodes that will report data in the cluster. Hence, transmissions can be done in an orderly fashion. Nodes will choose to become cluster heads according to the following equation [2]:

$$T(n) = \begin{cases} \dfrac{P}{1 - Px\left(r mod \frac{1}{P}\right)} & if\ n \in G \\ 0 & otherwise \end{cases} \qquad (1)$$

Where *P* is the percentage of cluster heads into the network, *r* is the number of current cluster heads and *G* is the group of nodes that have not been cluster head in a previous time. In this way the protocol allows the adequate distribution of energy consumption in each node. Fig. 1 shows how the network is organized in clusters, where nodes with a blue circle are the cluster heads.

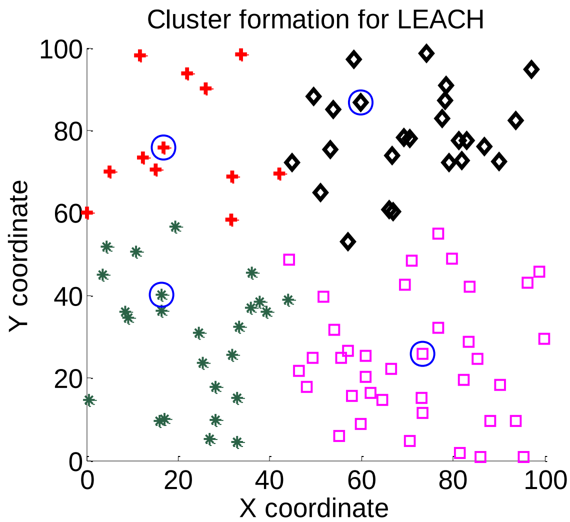



**Fig. 1.** Network organized in cluster for LEACH [5].

Note that there exist a complete set of protocols for wireless network [6] and especially for WSN [7], [8], [9] and each one of them has certain advantage over the others. Therefore, WSN have flexibility to choice any of those protocols according to the application in order to obtain its best performance [10]. This constantly generates a wide spectrum of research topics.

## 3 Proposed Clustering Scheme

In this work, a clustering protocol based on the well-known LEACH protocol is proposed and studied. This clustering protocol named LEACH-MOD, works as follows. Nodes are chosen to be cluster heads or cluster members only based on the order of arrival of nodes. Specifically, in the cluster formation phase, nodes transmit in each time slot according to a geometric process with probability τ, which is broadcasted by the sink according to its estimation of the outcome of the previous. In any given time slot, if there are two or more transmissions, a collision occurs and none of the transmitted packets are received successfully. Consequently the packets involved in the collision have to be retransmitted. On the other hand, if only one packet is transmitted in a time slot, it is considered to be successful and all the nodes in the network recognize such transmission. Building on this, the LEACH-MOD scheme considers that the first $N_{CH}$ nodes that successfully transmit their packet become cluster heads while the rest become cluster members. Fig. 2 shows how the network is organized in clusters based on the LEACH-MOD scheme, where nodes with a blue circle are cluster head.

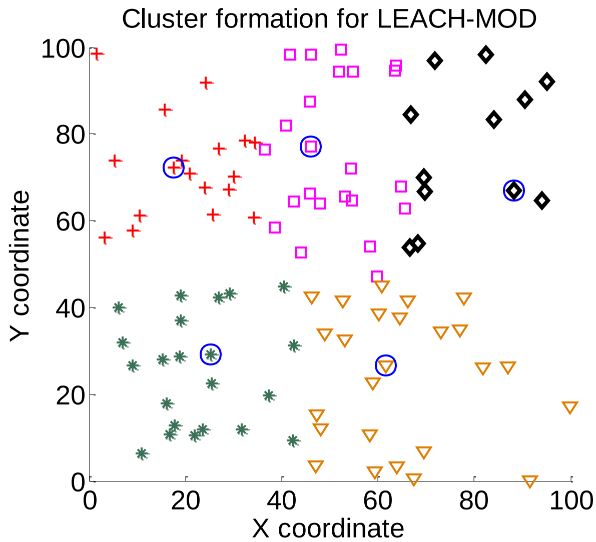



**Fig. 2.** Network organized in cluster for LEACH-MOD **[5]**.

The main reason to consider this modification to the LEACH protocols is that, in this protocol, each node calculates its transmission probability according to the nodes that have been CHs before. As such, nodes need to calculate this transmission probability throughout the operation of the network, which entails extra energy consumption and increases complexity. Furthermore, it is possible that each node calculates a different transmission probability. On the other hand, the LEACH-MOD approach does not require this transmission probability computation since the sink computes and broadcast it. It is important to mention that the LEACH protocol ensures that no node can become CH in successive rounds. This cannot be guaranteed by the proposed protocol. However, since nodes transmit according to a geometric process with parameter $\tau$, the probability that a node is elected as a CH in consecutive rounds is very low. Also, it can notice that neither LEACH nor LEACH-MOD protocols ensure a uniform distribution of clusters in the system area.

On a more detailed study on the effect of the transmission probability in LEACH-MOD, it is possible to see that when the number of nodes with a pending transmission is relatively high, the use of a high value of $\tau$ causes a high amount of collisions inside the cluster formation process, entailing a quick energy drain for all nodes in the system due to each node has to retransmit again. On the other hand, the use of a low transmission probability in the same scenario entails a lower number of collisions. However, as the nodes successfully transmit their control packet, the number of pending transmissions decreases accordingly. Hence, the low value of $\tau$ is no longer adequate when only a few nodes remain to transmit their packet since a high idle transmission period occurs. Such that, it is clear that the value of the transmission probability has to vary according to the number of remaining nodes in the cluster formation procedure. As such, we propose an adaptive transmission strategy.

### 3.1 Adaptive Transmission

As mentioned before, for the adaptive transmission probability strategy it was proposed that the value of $\tau$ varies according to the number of nodes with a pending transmission. Hence, the transmission probability of each node depends on the inverse of the number of neighbor nodes for each time slot. In this way, with a high number of contending nodes, the transmission probability is low, and vice versa. From this, each node computes the value of $\tau$ as $\tau_A = \frac{1}{i}$. Where $i$ is the number of remaining nodes attempting to transmit in the cluster formation phase. This adaptive transmission probability scheme was studied considering an error-free channel (see [5]), where it is assumed that all nodes know the exact number of nodes attempting a transmission. As it can be seen, in order to calculate the value of $\tau$ the sink estimates the number of nodes with a pending transmission in each time slot of the cluster formation phase for each round, as sink does not have energy restriction issues can perform this estimation and broadcast it to the nodes in the system. This estimation can be done by detecting each successful transmission and assuming that the number of nodes in the system has not changed, i.e., that all initial nodes are still correctly functioning. However the successful transmission detection is not error-free in a wireless environment due mainly to fading, interference and noise.

### 3.2 Channel Error

In order to see how the system is affected by channel errors, we study the effect of error estimation on the performance of the adaptive probability scheme. Specifically, it is considered that two estimation errors can occur in such environment: false positive and false negative estimation. The latter corresponds to the case when the sink node does not detect a transmission while the former corresponds to the case when the sink node detects a transmission but in reality no such transmission occurred. As such, the estimation of $\tau$ can have a higher (in case of false negative estimation) or lower value (in case of false positive estimation) than the actual value $i$. In this work, we assume that either false positive and false negative estimation errors have a probability to occur called PeP and PeN respectively. Different values of these probabilities are considered with the objective of analyzing the effect on the network of a noisy channel. Since the sink calculates the number of remaining nodes in the cluster formation phase according to the outcome of the previous time slot, we differentiate the actual number of nodes, *i*, from the estimation, *i*'. Specifically, the channel errors can impact the cluster formation phase as follows:

1. When there are no transmissions
   a. A false positive occurs, then the sink estimates ***i'*-1** nodes attempting to transmit, i.e. the sink assumes that a successful transition occurred.
   b. A false negative occurs, then the estimation of the sink, ***i'***, is the same.
   c. It is not possible that both the false positive and false negative errors occur at the same time.
   d. Neither false positive nor false negative occur, then there is not a change and the sink estimation, ***i'***, remains unchanged.

2. When there is one transmission
   a. A false positive occurs, then the sink detects a collision and its estimation ***i'*** is the same. Note that in this case, the node that transmitted assumes that its transmission suffered a collision (when in fact it did not) and retransmits its packet in a future time slot.
   b. A false negative occurs, then the sink does not detect the transmission and its estimation, ***i'***, is unchanged.
   c. It is not possible that both the false positive and false negative errors occur at the same time.
   d. Neither false positive nor false negative errors occur, then there is a successful transmission and the sink estimates ***i'*-1**.In this case, the actual remaining nodes with pending transmission, ***i***, decreases in one.

3. When there are two or more transmissions
   a. A false positive occurs, then the sink detects a collision and its estimation ***i'*** remains unchanged.
   b. We assume that a false negative cannot occur since the energy level in two or more transmission is sufficiently high that it is always detected as a collision. Then, the estimation ***i'*** remains unchanged.
   c. It is not possible that both the false positive and false negative errors occur at the same time.
   d. Neither false positive nor false negative occur, then there is a collision and the sink estimation ***i'*** remains unchanged.

From the previous discussion, we consider that the transmission probability in the cluster formation phase is now calculated as $\tau_A = \frac{1}{i'}$. Note that the transmission probability varies in each time slot according to the estimated outcome of the previous time slot. It is important to note that *i* always is bigger or equal to *i'*. The case where *i* is equal to *i'*, implies that there have not been errors. As more errors occur, *i'* will be smaller than *i*. The critical case is when *i'* is by far smaller than *i*. In this case, there will be high transmission probability for a high number of nodes that attempt a transmission, causing a high collision rate. The lowest value of *i'* is 0. In this case, $\tau_A$ takes the value of 0.95 in order to prevent a catastrophic scenario where nodes transmit in each time slot.

## 4 Model of the System

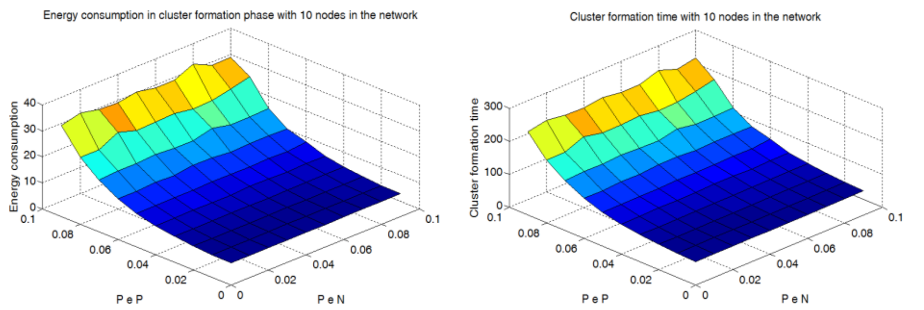
In this section, the main system parameters and assumptions are presented. It is considered a square coverage area of 100 meters per side with a variable number of nodes in order to analyze the performance of the cluster formation process under different scenarios. The energy consumption model considered is described in detail. We considered normalized energy units as values of energy required for the packet transmission and reception. This is basically to have a general energy consumption model that can be easily scaled to any commercial equipment. Building on this, we assume the following:

- The energy required to transmit a packet in the cluster formation phase (CF) is $E_{tx}^{CF}$= 0.02 units.
- The energy required to receive a packet in the cluster formation phase (CF) is $E_{tx}^{CF}$= 0.01 units.
- The energy required to transmit a packet from a cluster member to a cluster head in the steady state is $E_{tx}^{Steady}$ = 0.01 units.
- The energy consumed by the cluster head in order to receive a packet from a cluster member in the steady state is $E_{tx}^{Steady}$ = 0.01 units.
- The energy consumed by the cluster head to transmit a packet to the sink node (CH→Sink), is $E_{tx}^{CH\rightarrow Sink}$= 0.2 units.

Note that the energy required to transmit a packet is always higher than the energy required to receive such packet. Also the packet transmission from the cluster head to the sink node is the most energy consuming. Besides, we consider that the sink has an infinite energy supply. Furthermore, as a novel feature of this work, a non-ideal wireless channel is considered where channel errors can occur. Specifically, all nodes transmit directly to the sink node, and it determines whether a successful transmission, a collision or an empty slot occurs. As such, channel errors can alter the perception of sink of the outcome of the slot (due to interference and noise in the channel) in two major manners: a) The probability to hear a transmission when there is not one (false positive), is PeP; b) The probability to not hear an actual transmission (false negative) when a node sends a packet, is PeN.

## 5 Simulation Results

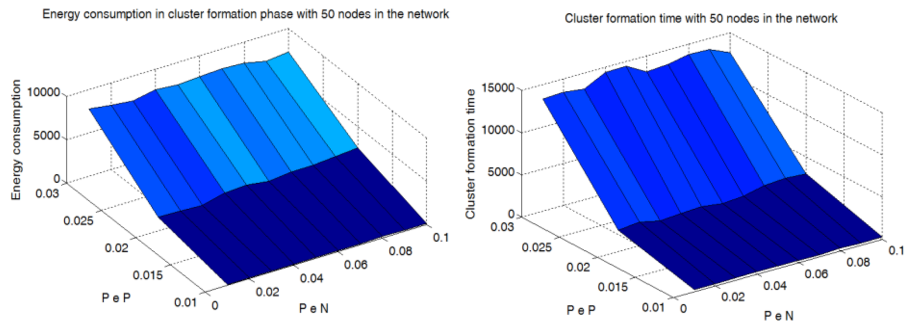
In this section some relevant numerical results are shown. Such results were obtained through a simulation based on discreet events according to each stage of the network life developed in C++ language with the parameters described in section 4. The main performance parameters considered in this section are the average cluster formation time and average energy consumption. Fig. 3a shows the average energy consumption in the system for 10 nodes inside the network and different probabilities of false positives and false negatives, both probabilities ranging from 0.01 to 0.1.It can be seen that the false positive probability has a major impact on the system performance compared to the effect of the false negative probability. While variations of the false negative probability barely affects the energy consumption. A similar behavior is shown in Fig. 3b which depicts the cluster formation time for the same 10 nodes network. This effect is not equal for both probabilities. Since false positive means a transmission, when none nodes transmit, the estimation of nodes attempting to transmit decreases causing high transmission probabilities for high number of nodes which entails more collisions on the system; causing a higher impact the false positive probability than the false negative probability. On the other hand, with the false negative probability, which means that a transmission is not detected, nodes retransmit the packets that were not correctly received by the sink with the optimal transmission probability because the estimation does not change.




| a) | b) |
|---|---|

**Fig. 3. a)** Average cluster energy consumption under an adaptive probability of transmission with different channel errors, **b)** Average cluster formation time under an adaptive probability to transmission with different channel errors.

We also consider the case with 50 and 100 nodes in the WSN. When 50 nodes integrate the network, the affectation of both probabilities increases. Fig. 4a shows the average energy consumption for 50 nodes network with probabilities of false positive from 0.01 to 0.03 and probability of false negative from 0.01 to 0.1. Also, Fig. 4b shows the average cluster formation time for the same network.

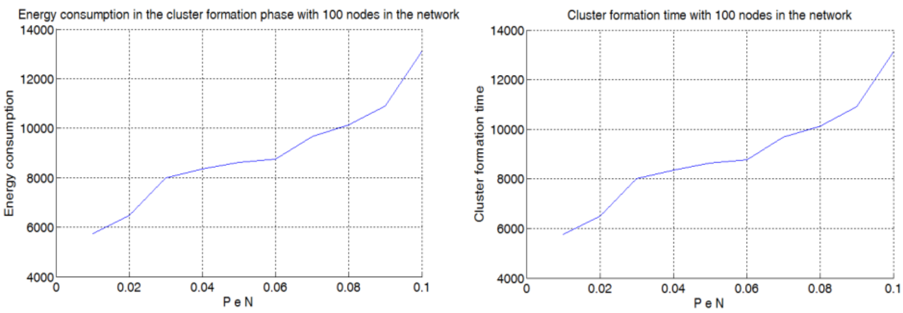



a) b)

**Fig. 4. a)** Average cluster energy consumption under an adaptive probability of transmission, **b)** Average cluster formation time under an adaptive probability of transmission.

Note that for the scenario with 50 nodes in the network, the energy consumption is in the order of 10000 units when the false positive probability is 0.03, while in the case where there are 10 nodes, the energy consumption is in the order of 10 units for the same value of false positive probability. The rationale behind these results is due to the density of the system. For instance, an error probability of 0.03 implies that for 100 transmissions, 3 of them will have error. Specifically, for the case of 10 nodes and assuming that all of them try to transmit, this probability of 0.03 represent that less than one transmission (0.3), will have error, but for the case of 50 nodes network, it implies that one or two transmissions (1.5) will have error. In such case, the rate of collisions increase more in the 50 nodes network since the sink detects more transmissions at the same time, even if only one node is transmitting. As the network density increases, the affectation of false positive and false negative probabilities too increases, as shown in Fig. 5a and in Fig. 5b. The following figures were obtained from a 100 nodes network with static probability of false positive of 0.01 and only changing the values of false negative probability, which ranges from 0.01 to 0.1.Unlike previous results, in the100 nodes network, the affectation of false negative probability entails high rates of energy consumption and cluster formation time which are in the order of 10000 units.

Furthermore, the most error-free scenario where both probabilities are 0.01 has an energy consumption and cluster formation time of 5800 units and this increases to 13000 units. From these results it can be seen that the effect of the error estimation is important, especially for highly dense networks.




a) b)

**Fig. 5. a)** Average energy consumption under an adaptive probability of transmission, **b)** Average cluster formation time under an adaptive probability of transmission.

## 6 Conclusions

In this work, the LEACH-MOD protocol under an adaptive transmission probability with a noisy channel is studied. From the results obtained and presented, it is clear that a simple CH selection, like the LEACH-MOD protocol, can achieve a good system performance with energy consumption and complexity reductions even in the presence of errors in the wireless channel. Also, the use of the adaptive transmission strategy is of high importance to reduce energy consumption at the cluster formation phase. However, the presence of errors causes that energy consumption and cluster formation time increase, something that, inherently, impact the performance of our system.

Building on the obtained results, it is possible to see that errors in the channel affect more when the network increases in number of nodes. Furthermore, the effect of the false positive probability causes a higher impact than the false negative probability due to the fact that the false positives change the estimation of nodes attempting to transmit resulting in a high transmission probability when there is a high number of nodes remaining to transmit contrary to the false negative probability that does not change the estimation and nodes continue to transmit with the previously estimated probability.

## References

1. Baronti, P., Pillai, P., Chook, V.W.C., Chessa, S., Gotta, A., Fun Hu, Y.: Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards. Computer Communications, 30, 1655–1695 (2007)
2. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: System Sciences, Proceedings of the 33rd Annual Hawaii International Conference, pp. 10 (2000)
3. Elhabyan, R.S., Yagoub, M.C.E.: Weighted tree based routing and clustering protocol for WSN. In: Electrical and Computer Engineering (CCECE), 26th Annual IEEE Canadian Conference, pp.1–6 (2013)
4. Nguyen, T.T., Shieh, C.S., Dao, T.K., Wu, J.S., Hu, W.C.: Prolonging of the Network Lifetime of WSN Using Fuzzy Clustering Topology. In: Robot, Vision and Signal Processing (RVSP), Second International Conference, pp. 13–16 ( 2013)
5. Romo-Montiel, E., Rivero-Ángeles, M.E., Villordo-Jiménez, I.: Effect of the error estimation of nodes in the cluster formation phase in wireless sensor networks with adaptive transmission probability. Research in Computing Science, 75, 9–18 (2014)
6. Handy, M.J., Haase, M., Timmermann, D.: Low energy adaptive clustering hierarchy with deterministic cluster-head selection. In: Mobile and Wireless Communications Network, 4th International Workshop, pp. 368–372 (2002)
7. Kredo, K., Mohapatra, P.: Medium access control in wireless sensor networks. Computer Networks, 50, 961–994 (2007)
8. Johnson, D.B., Maltz, D.A.: Truly seamless wireless and mobile host networking. Protocols for adaptive wireless and mobile networking. Personal Communications, IEEE, 3, 34–42 (1996)
9. Lin, R., Wang, Z., Sun, Y.: Energy efficient medium access control protocols for wireless sensor networks and its state-of-art. In: Industrial Electronics, IEEE International Symposium, pp. 669–674 (2004)
10. Lee H.C.: Towards a general wireless sensor network platform for outdoor environment monitoring. Sensors, IEEE, pp. 1–5 (2012)

# Flexible Rule-Based Programming for Autonomic Computing

José Oscar Olmedo-Aguirre[1], Marisol Vázquez-Tzompantzi[2]

[1] Cinvestav-IPN, Department of Electrical Engineering, Mexico City, Mexico

[2] Cinvestav-IPN, DCTS, Mexico City, Mexico

oolmedo@cinvestav.mx, mvazquezt@cinvestav.mx

**Abstract.** The ECAP rule programming language DLRL is currently being developed for architecting autonomic systems by coupling deduction and interaction. Three of the fundamental properties of autonomic systems, namely self-configuration, self-optimization and self-healing, are provided by DLRL: high-level program specification that can be user-defined for self-configuration; program introspection that allows to reactively adapt on-line program behavior for self-optimization, and program interaction that provides communication and coordination with the surrounding environment in order to detect deviations from their expected behavior for self-healing. The DLRL, programming model extends pure Prolog by including the modal actions of dynamic logic in the consequent predicates of conditional forward rules. It combines some the well-known refinements of resolution along with syntactically guided control strategies to represent and enact problem specifications dealing with stateless and state-based descriptions. The main contribution of this work consists in showing the benefits for architecting autonomic systems in a single, uniform and expressive mulit-paradigm programming language for rapidly changing demands of complex problems in distributed settings.

**Keywords:** Logic programming, interaction, autonomic computing, self-management, dynamic logic.

## 1 Introduction

As the complexity of developing and maintaining distributed applications has steadily increased, new approaches for software design are needed to provide a simple yet resourceful conceptual framework. The traditional off-line feedback for reconfiguring and reprogramming software needs to be replaced by an on-line self-management software. The software needs to adapt to the rapidly changing demands of modern applications and still keeping the desired quality of service within reasonable costs during operation. IBM introduced the main properties of self-management, known as the self-* properties of an autonomic system: self-configuration, self-optimization, self-healing and self-protection [1, 2]:

- Self-configuration. An autonomic system has the ability to configure itself according to high-level goals stated in a declarative manner, by specifying what is desired, not necessarily how to accomplish it.
- Self-optimization. An autonomic system pursues the best use of resources. After some automated reasoning, the autonomic system may conclude to initiate a change to the system behavior, either reactively by pursuing the user demands or pro-actively by systematically satisfying its own goals, in an attempt to improve performance or quality of service.
- Self-healing. An autonomic system detects, analyzes and diagnoses (potential) problems. The kind of problems detected can be low-level like hardware malfunctioning or high-level like a non-responsive software module. Being distributed, the autonomic system may take advantage of the available redundancy in both hardware and software to fix the problem, by switching to a trusted redundant component or by downloading and installing a software update. Because this replacement process is a potential risk for various kinds of malware intrusion, self-healing must perform exhaustive checking that ensures the system exhibits the expected trusted behavior before becoming an operational component in its distributed setting.
- Self-protection. An autonomic system protects itself not only from intrusion attacks but also from users who inadvertently make changes that opposes to the overall system purposes and behavior. The system opportunely verifies and adjusts its settings on security, privacy and data protection based on the rise of unexpected patterns of external activity.

`DLRL` is a symbolic rule-based programming language where self-configuration, self-optimization and self-healing, are respectively provided by:

- high-level program specification that can be user-defined
- program introspection that allows to reactively adapt on-line program behavior
- program interaction that provides communication and coordination with the surrounding environment in order to detect deviations from their expected behavior

The property of self-protection is believed by the authors that can be derived from the others three in the setting of an appropriate architecture. `DLRL` is an ECAP programming language with deductive, introspective and interactive capabilities that provide a run-time system that is able to reconfigure and recompile program code on demand whenever the changes of the surrounding environment compromises the goals of the program purpose. ECAP stands for Event-Condition-Action-Postcondition an extension of usual ECA rules commonly found in database management systems where the Postcondition imposes additional constraints to the result of the Action. `DLRL` borrows from `Indeed` [5] its state-based forward conditional rules and from `DLProlog` [6, 7] the dynamic logic modalities [4] for efficient imperative program execution. Thus, it is able to combine efficiently stateless and state-based reasoning along with coordinated interaction. The computational model comprises various forms of resolution-based inference procedures, like SLD-resolution, UR-resolution and positive hyper-resolution [10, 11], to describe respectively stateless deduction and state-based transitions. The coordination model consists of a transactional global memory of ground predicates along with a strategy for the theorem prover to control program execution by syntactically guided rule selection. In addition, the set of support restriction strategy [11] coordinates the input and output of facts with the shared memory, maintaining the coherence of the current state of the program.

The paper is organized as follows. In section 2, a brief revision of the related work is presented, In section 3, we illustrate the forward and backward reasoning schemes that arise from the computational model with a programming example. Next, in section 4, the syntax and the declarative semantics of the the `DLRL` programming language is presented. Finally, in section 5 some remarks are given to conclude.

## 2 Related Work

Let us briefly explore other approaches that can be compared with ours: resolution theorem provers, constraint logic programming and coordination logic programming. The resolution-based theorem prover *Prover9* [10, 11] comprises a number of refinements of resolution along with a set of control strategies to prune the explosive generation of intermediate clauses. However, Prover9 does not account for interaction. The set of all instantaneous descriptions essentially corresponds to the set of support strategy. In Prover9, a clause is selected and removed from the set of support to produce a new set of clauses deduced from the axioms of the theory. Then, after simplifying a new clause by demodulation and possibly discarding it by either weighting, backward or forward subsumption, the new clause is placed back to the set of support. *Concurrent Constraint Programming* (CCP) [8] proposes a programming model centered on the notion of constraint store that is accessed through the basic operations 'blocking $ask$' and 'atomic $tell$'. Blocking $ask(c)$ corresponds to the logical entailment of constraint $c$ from the contents of the constraint store: the operation blocks if there is not an enough strong valuation to decide on $c$. In this respect, the blocking mechanism is similar to the one used in `DLRL` to obtain the set of ground facts that match with the left-hand side of some rule. Besides, the constraint store shares some similarities with the global memory of ground facts. However, operation $tell(c)$ is more restrictive than placing ground atoms in the global memory because constraint $c$ must be logically consistent with the constraint store. *Extended Shared Prolog* (ESP) [3] is a language for modeling rule-based software processes for distributed environments. ESP is based in the PoliS coordination model that extends Linda with multiple tuple spaces. The language design seeks for combining the PoliS mechanisms for coordinating distribution with the logic programming Prolog. Coordination takes place in ESP through a named multiset of passive and active tuples. They correspond to the global memory of facts in DLRL although no further distinction between passive and active ground facts is made. ESP also extends Linda by using unification-based communication and backtracking to control program execution.

## 3 A Programming Example

As autonomic systems perceive the surrounding environment through *sensors* and act upon it through *effectors*, their interaction can effectively be decoupled by a global shared transactional memory consisting of a multiset of basic facts (ground predicates) that can be implemented following the interaction approach of Wegner [10]. By abstracting away interaction from deduction, the inherently complex operational details of sensors and effectors become irrelevant. The behavior of each user-defined system is described by a set of backward and forward rules that describe the exchange of information through the global memory. As an example, consider the problem of converting a sequence of digits to obtain its numeric value where the sequence is fragmented in subsequences that are dispersed across a large partitioned region. This problem is representative of a class of problems that deal with summarizing information coming from disperse geographical regions, like those that deal with calculating the average temperature, humidity or atmospheric pressure of a large region, for example. In order to perform the conversion, the subsequences need to be retrieved and incorporated into the calculations in a coordinated manner. Because of the underlying indeterminism, sometimes one method may lead to shorter processing time for some regions than for others. The problem addressed here can be stated as finding and applying the faster method suitable for some region and to incorporate this partial result into the overall calculation.

For the conversion, a recursive definition of the basic arithmetic operators is given next. Figure 1 shows theory *Natural* for the natural numbers written in DLRL, closely similar to those written in pure Prolog. Similar rules, not presented here, can be given for arithmetic multiplication. This theory uses *backward rules* that have the general form $P \Leftarrow P_1, \ldots, P_n$, where $P, P_1, \ldots, P_n$ are atomic predicates, $P$ is the consequent, or simply the head of the rule and $P_1, \ldots, P_n$ is the antecedent, or simply the body of the rule, with $n \geq 0$. The ellipsis stands for the conjunction of predicates $P_1, \ldots, P_n$ if $n > 0$ and for `true` if $n = 0$. The logical propositions of the theory are built upon infix predicates $=$, $<$, and $\leq$, whose recursive definitions are given by backward rules $N_1$ to $N_5$. *Natural* represents the deductive component of the interactive parser. Figure 2 shows a theory written in DLRL for a parallel *Parser* that extends *Natural*. This theory uses *forward rules* that have the general form $E_1, \ldots, E_n \,|\, C \Rightarrow [A]\, P$ with $n \geq 0$. The declarative reading of the forward rule is that, if appropriate predicates $P_1, \ldots, P_n$ have been placed in the shared memory such that each $E_i$ $(i = 1, \ldots, n)$ matches a distinct $P_j$ and their contents satisfy the condition

$C$, then the action (i.e. imperative program) $A$ is executed to obtain the values bound to the variables occurring in the postcondition $P$.

The conversion methods use forward rules to define a simple bottom-up parser whose syntactic entities are represented by ground atomic predicates. $T(n, t)$ asserts that symbol $t$ occurs at position $n$, while $E(n_1, n_2, x)$, with $n_1 \leq n_2$, asserts that the sequence of symbols from $n_1$ to $n_2$ forms a well-formed arithmetic expression whose evaluation is the integer value $x$.

The three conversion methods considered here are presented next. Figure 2 shows the rules of a method based on a parallel composition from smaller fragments that are adjacent to compose a larger one. Figure 3 shows the rules of a method based on a sequential conversion of digits, proceeding from left to right, starting with the delimiter '[' and ending with the delimiter ']'. Finally Figure 4 shows the single rule of a method that uses a sequential implementation of the method shown in Figure 3. The rules have been named consecutively as $R_1, R_2, \ldots, R_6$. In that follows, the user-defined rule $R_i : E_i | C_i \Rightarrow [A_i] P_i$ will be simply designated by its index $i$ instead of $R_i$ for brevity.

$$
\begin{array}{ll}
N_1 : & 0 + y = y \Leftarrow \\
N_2 : & (x+1) + y = (x+y) + 1 \Leftarrow \\
N_3 : & 0 \leq y \Leftarrow \\
N_4 : & (x+1) \leq (y+1) \Leftarrow x \leq y
\end{array}
$$

**Fig. 1.** Natural numbers using backward rules.

$$
\begin{array}{l}
R_1 : T(n, x) \mid digit(x) \Rightarrow [z := toInt(x)]\, N(n, n, z). \\
R_2 : N(n_1, n_2, x), N(n_3, n_4, y) \\
\qquad \mid n_1 \leq n_2, n_2 + 1 = n_3, n_3 \leq n_4 \\
\qquad \Rightarrow [z := x \times 10^{n_4 - n_3 + 1} + y] N(n_1, n_4, z). \\
R_3 : T(n_1, '[') , N(n_2, n_3, x), T(n_4, ']') \\
\qquad \mid n_1 + 1 = n_2, n_2 \leq n_3, n_3 + 1 = n_4 \\
\qquad \Rightarrow E(n_1, n_4, x).
\end{array}
$$

**Fig. 2.** Parallel parser of numbers.

The scheduler selects and applies rules according to the rules $S_1$, $EC_i$ , $AP_i$ and $M_1$ shown in Figure 5. Rule $S_1$ applies to a set of user-defined rules that are in conflict like the set $\{R_3, R_4, R_6\}$. The action of rule $S_1$ performs a linear search for those userd-defined rules that can be selected. From them, the one selected has the minimum average time of execution. This criterion helps not only to select the historically faster rule but also to avoid oscillations caused by

$$
\begin{array}{l}
R_4 : T(n, '[') \Rightarrow M(n, n, 0). \\
R_5 : N(n_1, n_2, x), T(n_3, t) \\
\qquad\qquad | \; n_1 \leq n_2, n_2 + 1 = n_3, digit(t) \\
\qquad\qquad \Rightarrow [z := x \times 10 + toInt(t)] \, N(n_1, n_3, z). \\
R_5 : N(n_1, n_2, x), T(n_3, ']') \\
\qquad\qquad | \; n_1 \leq n_2, n_2 + 1 = n_3 \\
\qquad\qquad \Rightarrow E(n_1, n_3, x).
\end{array}
$$

**Fig. 3.** Sequential parser of numbers with three rules.

$$
\begin{array}{l}
R_6 : T(n_1, '[') \\
\qquad \Rightarrow \left[ \begin{array}{l}
\texttt{int}\, x, t, n, v : \\
\left( \begin{array}{l}
x, n := 0, n_1 + 1; \\
\left( \begin{array}{l}
T(n,t)?; digit(t)?; \\
\quad v := toInt(t); \\
\quad x, n := 10 \times x + v, n + 1; \\
\quad \texttt{retract}(T(n,t))
\end{array} \right) *; \\
t = ']'?; \\
z, n_2 := x, n
\end{array} \right)
\end{array} \right] E(n_1, n_2, z).
\end{array}
$$

**Fig. 4.** Sequential parser of numbers with a single rule.

sporadic fluctuations. Note that rule evaluates asserted predicates $ready(i)$ and $average(i, t)$ to determine if rule $i$ can be selected for execution and to obtain the average execution time $t$, respectively. Once obtained the statistically faster rule $m$, the predicate $ready(m)$ is discarded to avoid applying the rule once more. A request for applying the faster rule is also established by asserting the predicate $do(m)$. shown in Figure 5.

The rule conflict solver and the rule executor are given by rules $EC_i$ derived from the event-condition parts and rules $AP_i$ derived from the action-postcondition parts of each user-defined rule $i$. The purpose of rules $EC_i$ and $AP_i$ is to produce the same effect caused by rule $E_i \,|\, C_i \Rightarrow [A_i] \, P_i$ though in this case by mediation of the rule scheduler $S$. In order to produce this effect, the user-defined rule $i$ is split into two parts at compilation-time, the event-condition and the action-postcondition that are later rejoined at run-time by the scheduler. The event-condition parts $E_i \,|\, C_i$ are embedded in rule $EC_i$, while the action-postcondition parts $[A_i] \, P_i$ are embedded in rule $AC_i$, producing in fact two set of rules. The purpose of the rules $EC_i$ is to determine for which of them there are available asserted predicates that unifies with the event part of rule and if such unifier satisfies the condition part of the user-defined rule $E_i \,|\, C_i \Rightarrow [A_i] \, P_i$. If such a unifier exists, the values bound to the variables are orderly passed to the corresponding rule executor in order to instantiate the variables used in the action-postcondition parts of the corresponding rule, producing in this way the same effect.

$$
S_1 \;:\; S \Rightarrow \left[\begin{array}{l}
\texttt{int}\ m, i, t\texttt{;} \\
(m, i) := (\infty, 0)\texttt{;} \\
\left(\begin{array}{l}
i < N\texttt{?;} \\
\mathit{ready}(i)\texttt{?;} \\
\mathit{average}(i, t)\texttt{?;} \\
\quad (t < m\texttt{?;} m := i \cup \texttt{skip})\texttt{;} \\
\quad i := i + 1
\end{array}\right)\texttt{*;} \\
\texttt{retract}(\mathit{ready}(m))\texttt{;} \\
\texttt{assert}(\mathit{do}(m))
\end{array}\right] S
$$

$$
\begin{array}{l}
EC_i \;:\; EC,\ E_i \mid C_i \\
\qquad \Rightarrow \left[\begin{array}{l}
\texttt{assert}(\mathit{ready}(i))\texttt{;} \\
\texttt{assert}(\mathit{input}(i, \mathit{fvlist}(E_i \cup C_i)))
\end{array}\right] EC
\end{array}
$$

$$
\begin{array}{l}
AC_i \;:\; AP, \mathit{do}(i), \mathit{input}(i, \mathit{fvlist}(EC_i)) \\
\qquad \Rightarrow \left[\begin{array}{l}
\texttt{assert}(\mathit{startedAt}(i, \mathit{now}()))\texttt{;} \\
\texttt{retract}(\mathit{input}(i, \_))\texttt{;} \\
A_i\texttt{;} \\
\texttt{assert}(P_i)\texttt{;} \\
\texttt{assert}(\mathit{endedAt}(i, \mathit{now}()))
\end{array}\right] AP
\end{array}
$$

$$
\begin{array}{l}
M \;:\; M, \mathit{average}(i, e), \mathit{startingAt}(i, s), \mathit{endingAt}(i, f) \\
\qquad \Rightarrow \left[\begin{array}{l}
\texttt{retract}(\mathit{average}(i, e))\texttt{;} \\
\texttt{retract}(\mathit{startingAt}(i, s))\texttt{;} \\
\texttt{retract}(\mathit{endingAt}(i, f))\texttt{;} \\
\texttt{assert}(\mathit{average}(i, \mathit{newaverage}(e, s, f)))
\end{array}\right] M
\end{array}
$$

**Fig. 5.** Scheduler rules.

In rule $EC_i$, the expression $fvlist(\{T(n,x), digit(x)\}) = [n,x]$, determined at compilation-time, produces the list of variable names in the textual order in which they appear in both the event and the condition parts of rule $i$. Thus for example, rule $EC_1$ shown in Figure 6 is obtained from rule $R_1$ in this manner. where for rule $P_1$, $fvlist(\{T(n,x), digit(x)\}) = [n,x]$ . Note that neither this rule nor any other rule of this set modifies the knowledge base of the asserted predicates as they only obtains the values bound to the variables and evaluates the condition with such values. The asserted predicate $ready(i)$ tells the scheduler that the user-defined rule $i$ is selectable, while the asserted predicate $input(i, vs)$ tells the executor $AP_i$ to bind its orderly list of variables with the list $vs$ of values by means of unification. The purpose of the rule executor $AP_i$ is to perform the action part of the user-defined rule $i$, given the values passed through the asserted predicate $input(1, [x,n])$ and then binding the variables to their respective values in the so-called input substitution. Then, after the action terminates with a binding of values to variables, the so-called output substitution, rule $AP_i$ asserts the instance of the postcondition $N(n,n,z)$ under the composition of both the input and the output substitutions. For example,

$$EC_1 : EC, T(n,x) \mid digit(x) \Rightarrow \begin{bmatrix} \texttt{assert}(ready(1))\texttt{;} \\ \texttt{assert}(input(1,[n,x])) \end{bmatrix} EC$$

$$AP_1 : AP, do(1), input(1,[n,x]) \Rightarrow \begin{bmatrix} \texttt{assert}(startedAt(1, now()))\texttt{;} \\ \texttt{retract}(input(1, \_))\texttt{;} \\ z = toInt(x)\texttt{;} \\ \texttt{assert}(N(n,n,z))\texttt{;} \\ \texttt{assert}(endedAt(1, now())) \end{bmatrix} AP$$

**Fig. 6.** From rule $P_1$, rules $EC_1$ and $AP_1$ are extracted and handled by the scheduler.

for the user-defined rule $R_1$, the rule $AP_1$, obtained as indicated before, is shown in Figure 6, where the asserted predicate $input(1,[x,n])$ helps to instantiate the variables of the input substitution rule $AP_1$.

Finally, rule $M_1$ describes the simple behavior of the monitor. The monitor simply determines the average execution time for the actions of each of the methods described in Figure 2 to 4. The execution time is calculated as the difference between the final time $f$ and the initial time $s$ of execution. The new average is recalculated from the previous one $e$ along with $s$ and $f$ using function $newaverage(e, s, f)$.

## 4 DLRL Formal Description

An experimental system for DLRL has been built to evidence the viability of the approach. The system consists of a parser with integrated type inference to decide whether the program constructs are well-formed. The computational model is described as a structured-operational semantics interpreter that calculates the next state of the shared memory.

Let $\Sigma = \bigcup_\alpha \Sigma_\alpha$ be a set of *constructor* (constant) names and let $\Xi = \bigcup_\beta \Xi_\beta$ be a set of *variable* names, each partitioned by the basic types `bool`, `int`, and `act`, among others. The following syntactic categories are built upon the signature $(\Sigma, \Xi)$:

| | |
|---|---|
| **Terms** $T(\Sigma,\Xi)$ | $T ::= x \mid c \mid c(T_1,\ldots,T_n)$ |
| **Predicates** $P(\Sigma,\Xi)$ | $P ::= \texttt{false} \mid \texttt{true} \mid T_1 = T_2 \mid p(T_1,\ldots,T_n)$ |
| **Goals** $G(\Sigma,\Xi)$ | $G ::= P \mid G_1 \wedge G_2$ |
| **Horn clauses** $B(\Sigma,\Xi)$ | $B ::= P \mid P \Leftarrow G \mid \forall x.B$ |
| **Events** $E(\Sigma,\Xi)$ | $E ::= P \mid E_1 \ , \ E_2$ |
| **Actions** $A(\Sigma,\Xi)$ | $A ::= \texttt{skip} \mid \texttt{fail} \mid G\texttt{?} \mid (A) \mid A_1\texttt{;}A_2 \mid A_1 \cup A_2 \mid A\texttt{*} \mid \texttt{int}\ x_1,\ldots,x_n :\ A \mid x_1,\ldots,x_n := T_1,\ldots,T_n \mid \texttt{assert}(p(T_1,\ldots,T_n)) \mid \texttt{retract}(p(T_1,\ldots,T_n))$ |
| **Modal actions** $A(\Sigma,\Xi)$ | $M ::= P \mid [A]\ M \mid \langle A \rangle\ M$ |
| **Forward rules** $F(\Sigma,\Xi)$ | $F ::= M \mid E \,\vert\, G \Rightarrow M \mid \forall x.F$ |

Variables occurring in an action $A$ are either *logical variables* or *local imperative variables*. Logical variables occurring in a clause are universally quantified, whereas local variables are introduced by declaration within an action. A declaration of local variables `int` $x_1, \ldots, x_n : A$ creates new local imperative variables whose scope and duration are restricted to the block $A$. A simple assignment $x := T$ evaluates the term $T$ in the current state and the resulting constant value is assigned to $x$. Logical and imperative variables are compatible in assignments of the same type, so they can appear in both sides of the assignment. Note however that logical variables can be defined at most once, whereas imperative variables can be redefined. A multiple assignment $x_1, \ldots, x_n := T_1, \ldots, T_n$ evaluates all the terms at the right-hand side in the current state and the resulting values are assigned to the corresponding variables at the left-hand side of the assignment. The action $\texttt{assert}(p(T_1, \ldots, T_n))$ introduces the predicate $p(T_1, \ldots, T_n)$ into the knowledge base making it valid, whereas the action $\texttt{retract}(p(T_1, \ldots, T_n))$ removes the predicate $p(T_1, \ldots, T_n)$ from the knowledge base making it invalid. Being these actions borrowed from standard Prolog, they are not intended to be blocking actions, like in other Linda-like coordination models, and therefore they can only succeed or fail. Modal necessity composition $[A]\ P$ means that after executing action $A$, postcondition $P$ is necessarily true.

In a signature $(\Sigma, \Xi)$ with variables, a *substitution* is a partial function $\sigma : \Xi \to T(\Sigma, \Xi)$, where $\sigma(x) \neq x$ for any variable $x \in \Xi$. $\{\}$ denotes the empty substitution. A *ground substitution* is a substitution $\sigma : \Xi \to T(\Sigma)$ valued on ground terms. For any variable $x \in \Xi$ and any substitution $\sigma$, let $x\sigma = \sigma(x)$ if $x \in \mathrm{dom}(\sigma)$ and $x\sigma = x$ otherwise. For any term $t \in T(\Sigma, \Xi)$, let $t\sigma$ be the term obtained by substituting any variable $x$ appearing in $T$ by $x\sigma$:

$$\begin{aligned}
x\{\} &= x \\
x\,\sigma &= \begin{cases} x & \text{if } x \notin \mathrm{dom}(\sigma) \\ \sigma(x) & \text{if } x \in \mathrm{dom}(\sigma) \end{cases} \\
c\,\sigma &= c \\
c(T_1, \ldots, T_n)\,\sigma &= c(T_1\,\sigma, \ldots, T_n\,\sigma) \\
[A]\,p\,\{\} &= [A]\,p \\
[A]\,p\,\sigma &= [\sigma_{:=}\,\texttt{;}\,A]\,p
\end{aligned}$$

where notation $[\sigma_{:=}]$ stands for the multiple assignment $x_1, \ldots, x_n := T_1, \ldots, T_n$ obtained from the substitution $\sigma = \{x_1 \mapsto T_1, \ldots, x_n \mapsto T_n\}$, for $1 \leq n$. Thus the substitution for a modal action $A$ is defined as the initial value that the variables take before the action starts its execution. The *composition* of two substitutions $\sigma_0, \sigma_1 \in \Xi \to T(\Sigma, \Xi)$, written $\sigma_0 \cdot \sigma_1$, is defined as

$$\sigma_0 \cdot \sigma_1 : x \mapsto \begin{cases} (x\sigma_0)\sigma_1 & \text{if } x\sigma_1 \notin \mathrm{dom}(\sigma_1) \\ x\sigma_1 & \text{if } x \in \mathrm{dom}(\sigma_1) - \mathrm{dom}(\sigma_0) \\ failure & \text{otherwise} \end{cases}$$

Besides the natural extension to terms $T(\Sigma, \Xi) \to T(\Sigma, \Xi)$, substitutions are also extended to predicates, goals, and both backward and forward rules.

The *backward computation* relation $\lhd \subset G(\Sigma, \Xi) \times (\Xi \to T(\Sigma, \Xi))$ consists of pairs relating goals and substitutions, where the substitutions are defined upon the variables occurring in a renamed variant of the rule. An *instantaneous description* $I \subset P(\Sigma) \times (\Xi \to T(\Sigma))$ relates ground predicates and ground substitutions, describing a portion of the current state of the shared memory. The substitutions keep a track of the bindings for all the variables that occurred in the renamed variant of each forward rule applied. The *forward computation* relation $\rhd \subset \mathcal{P}(I) \times \mathcal{P}(I)$ relates pairs of instantaneous descriptions. The transition relations are defined in Figure 7.

**Backward computation**

$$\frac{\begin{array}{c} P' \Leftarrow G' \in B(\Sigma, \Xi) \\ P\sigma' = P'\sigma' \end{array}}{(\{P\} \cup G, \sigma) \lhd (G'\sigma' \cup G\sigma', \sigma\sigma')}$$

**Forward computation**

$$\frac{\begin{array}{c} E_1, \ldots, E_n,\ P \mid G \Rightarrow [A]\, P' \in F(\Sigma, \Xi) \\ P_i\sigma_i = E_i \ \ (i \in 1, ..., n) \\ (G, \sigma_1 \ldots \sigma_n\sigma) \lhd^* (\{\}, \iota) \end{array}}{\begin{array}{c} \{(P_1, \sigma_1), \ldots, (P_n, \sigma_n), (P, \sigma)\} \cup I \\ \rhd \\ \{([\iota_{:=}; A; o_{:=}]\, P', \iota o)\} \cup I \end{array}}$$

**Fig. 7.** Operational semantics of backward and forward rules.

The backward computation rule describes a *refutation step* from $(\{P\} \cup G, \sigma)$ to $(G'\sigma' \cup G\sigma', \sigma\sigma')$ by replacing the head $P\sigma'$ with the body $G'\sigma'$ of the instance of the backward rule $P \Leftarrow G'$ under a suitable substitution $\sigma'$ such that $P\sigma' = P'\sigma'$. The new goal is an instance under $\sigma'$ of the body $G'$ and the remaining goal $G$, along with the new answer substitution obtained from the composition of $\sigma'$ with the previous one $\sigma$. In case that the application of the rule leads to a failure, another backward rule if any is selected and applied after backtracking to the previous goal and the previous substitution; otherwise, if no more rules can be selected, the backward computation terminates in failure.

The forward computation rule $E_1, \ldots, E_n, P \mid G \Rightarrow [A]\, P'$, with $n > 0$, can be selected for deducing the ground predicate $P'\iota o$ from $P\sigma$ only if the following three conditions hold: (i) there are $n$ ground predicates $P_1, \ldots, P_n$ already asserted in the shared memory, (ii) there are $n$ ground substitutions $\sigma_1, \ldots, \sigma_n$ that makes syntactically identical the corresponding instances of each event $E_i$ with an appropriate predicate $P_i$, i.e. equation $E_i\sigma_i = P_i\sigma_i$ holds for $1 \leq i \leq n$, and (iii) the composition $\sigma_1 \cdots \sigma_n\sigma$ of the $n$ substitutions along with $\sigma$ satisfies the goal $G$. Whenever these conditions are met, the forward rule can be applied. With the ground substitution $\sigma_1 \cdots \sigma_n\sigma$ starts the backward computation rule that leads to the input substitution $\iota$ with bindings for the new variables that $G$ may introduce. The instance under $\iota$ of the modal action $[A]\,P$ is then executed following the standard interpretation of the action connectives [4]. Assuming that $A$ terminates starting with the initial values given by $\iota$, the postcondition $P'$ becomes satisfied by the substitution $\iota o$, where $o$ is the output substitution produced by $A$ on the output variables. However, if the guard $G\sigma$ fails, another set of predicates asserted in the shared memory must be considered. If no more possible selections of predicates were possible for the forward rule, another rule is selected if any. If no more forward rules were applicable, the system would appear non-responsive until another predicate assertion were eventually produced in the shared memory.

## 5 Conclusions

The problem of coupling interaction in a resolution theorem prover with syntactically guided selection of the control strategy to be used has been presented in this paper. The experimental programming language `DLRL` has been designed to deal with state-based descriptions using forward rules and stateless deduction using backward rules. The programming model allows to combine backward and forward rule chaining in a simple and more efficient manner.

## References


1. P. Horn: Autonomous Computing: IBMs perspective on the state of Information Technology. IBM Research (2001)
2. IBM: An architectural blueprint for autonomic computing. Tech. Rep., IBM (2003)
3. P. Ciancarini: Coordinating Rule-Based Software Processes with ESP. ACM Trans. on Software Engineering and Methodology, 2(3), 203–227 (1993)
4. D. Harel, J. Tiuryn, D. Kozen: Dynamic Logic. Cambridge, MA, USA, MIT Press (2000)
5. O. Olmedo-Aguirre, G. Morales-Luna: Indeed: Interactive Deduction on Horn Clause Theories. In: Proceedings IBERAMIA 2002, LNAI, vol. 2527, pp. 151–160, Springer-Verlag (2002)
6. J. O. Olmedo-Aguirre, G. Morales-Luna: A Dynamic Logic-based Modal Prolog. In: Proceedings MICAI 2012, CPS IEEE Computer Society, pp. 3–9 (2012)
7. J. O. Olmedo-Aguirre: DL Prolog: Another Unifying Programming Language. Research in Computing Science, vol. 60, pp. 23–33 (2012)
8. V.A. Saraswat: Concurrent Constraint Programming. In: Records of 17th ACM Symposium on Principles of Programming Languages, San Franciso, CA, pp. 232–245 (1990)
9. P. Wegner: Interactive Software Technology. In: CRC Handbook of Computer Science and Engineering (1996)
10. L. Wos, R. Overbeek, E. Lusk, J. Boyle: Automated Reasoning. Introduction and Applications. McGraw-Hill (1992)
11. L. Wos, G. Pieper: A Fascinating Country in the World of Computing: Your Guide to Automated Reasoning. World Scientific Publishing Co. (1999)

# Gripper robótico antropomórfico a los dedos primero y segundo, sensible a la presión

Francisco O. Gonzalez-Espinosa[1], Erick D. de la Rosa-Montero[1], Carlos Rios-Ramirez[1], Yesenia E. Gonzalez-Navarro[2]

[1] Instituto Politécnico Nacional, UPIITA, Academia de Biónica, México

[2] Instituto Politécnico Nacional, UPIITA, Academia de Sistemas, México

ygonzalezn@ipn.mx

**Resumen.** El presente trabajo aborda el diseño y construcción de un gripper robótico con cinco grados de libertad, capaz de realizar la prensión cilíndrica palmar y por oposición terminal de la mano humana. El gripper cuenta con la implementación de un sistema sensorial de presión donde cada eslabón posee un sensor con los cuales establece al contacto si hay o no un objeto en el espacio de trabajo del prototipo. La información de la posición final de los actuadores es utilizada en un algoritmo de control basado en reconocimiento de patrones con el cual el robot determina el tipo de agarre que realiza y la presión que debe de ejercer para sostener el objeto. Finalmente se activa el mecanismo de desplazamiento lineal que demuestra la capacidad del prototipo de sostener y asir el objeto.

**Palabras clave:** Gripper, sistema sensorial, presión, reconocimiento de patrones.

# Robotic Anthropomorphic Gripper of the First and Second Fingers Sensible to Pressure

**Abstract.** This paper presented the designing and building of a robotic gripper with five degrees of freedom capable to perform cylindrical grasp and grasp by terminal opposition of the human hand. The gripper has a pressure sensorial system, where each link has a sensor to determine if exist contact with the object in the work space of the prototype. The final position information of the motors is using in the control algorithm of pattern recognition and this provides the information the grasp type that the robot is doing and the pressure that the robot must do to grasp and hold the objects. Finally the linear movement mechanism is activated to up.

**Keywords:** Gripper, sensorial system, pressure, patterns recognition.

# 1. Introducción

Un gripper es un dispositivo electro-mecánico, cuya función es agarrar y sostener objetos específicos. Son comúnmente usados como actuadores finales de brazos robóticos o de prótesis de miembro superior. Con el fin de optimizar las tareas de agarre y sujeción de los robots, los elementos del gripper deben ser adaptables a la pieza de trabajo, esto es, sostener objetos con diferentes características, lo cual se logra maximizando el área de contacto entre el gripper y el elemento a sujetar, implementando distintos grados de libertad en el gripper [1], utilizando elementos flexibles pero firmes en los eslabones como el efecto derivado del movimiento de la cola de un pez [2] o con el uso de membranas flexibles que se pueden endurecer o ablandar mediante la modulación de aire dentro de ellas [3]. No obstante un robot capaz de aprender de su percepción y experiencia del entorno puede ajustarse a la características del objeto [4], lo cual se logra con la implementación de un sistema sensorial que comunique al actuador con su entorno y pueda distinguir entre las diferentes superficies de contacto, la presión ejercida y las fuerzas que intervienen [5].

Con las características mencionadas, a continuación este trabajo presenta el diseño y construcción del gripper; la sección 2 consiste en modelar la pinza del robot como los dedos primero y segundo de la mano humana, posteriormente las secciones 3 y 4 abordan el diseño y construcción mecánica y electrónica, finalmente en las secciones 5 y 6 se muestran los resultados de implementar el algoritmo de reconocimiento de patrones en el sistema de control del prototipo.

# 2. Modelo matemático

## 2.1. Análisis robótico de la mano humana

La mano humana es un gripper natural universal, la cual se analiza como un robot considerando cada una de las falanges de los dedos como los eslabones del mismo, unidas por juntas rotacionales que producen el movimiento de flexión y extensión o abducción y aducción.

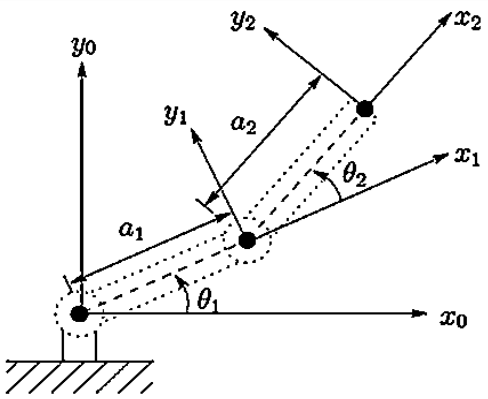



**Fig. 1**. Robot de dos grados de libertad.

En este trabajo, el análisis se restringe al primer y segundo dedo de la mano humana, considerando solamente el movimiento de flexión y extensión de cada dedo y que cada uno posee dos falanges.

Se modela cada dedo como un robot de dos grados de libertad, donde el primer eslabón, el que se encuentra anclado al origen de un sistema cartesiano de coordenadas corresponde a la falange proximal para ambos dedos y el segundo eslabón corresponde a la falange distal del primer dedo y a la falange medial del segundo dedo, por lo tanto las variables incluidas en este modelo son: el ángulo entre sistemas coordenados $x_0$ y $x_1$ ($\theta_1$); el ángulo entre sistemas coordenados $x_1$ y $x_2$ ($\theta_2$); la longitud del primer y segundo eslabón $a_1$ y $a_2$, ver Figura 1.

## 2.2. Análisis cinemático

El problema de la cinemática directa se refiere a la relación entre las articulaciones individuales del robot, la posición y la orientación del efector final [6]. Las variables de este análisis, mostradas en la Figura 1, son el conjunto de ángulos entre los eslabones en el caso de las juntas de rotación y la distancia entre eslabones en el caso de las articulaciones prismáticas. En la convención de Denavit Hartenberg cada eslabón del robot es representado por una matriz de transformación homogénea A, Ecuación (1), la cual está descrita por el producto de cuatro matrices básicas de transformación, rotación y traslación utilizando los ángulos de Euler, mostrada de forma explícita en la Ecuación (2). Finalmente el resultado del producto de todas las matrices A en su debido orden es la matriz de posición final T, Ecuación (3).

$$A = Rot_{z,\theta} \cdot Tras_{z,d} \cdot Tras_{x,a} \cdot Rot_{x,\alpha} \quad (1)$$

$$A = \begin{pmatrix} cos\theta & -sen\theta & 0 & 0 \\ sen\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & d_1 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & a_2 \\ 0 & 0 & 1 & a_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\alpha & -sen\alpha & 0 \\ 0 & sen\alpha & cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

$$T = A_1 \cdot A_2 \cdot \ldots \cdot A_{n-1} \cdot A_n \quad (3)$$

Siguiendo la convención de Denavit Hartenberg, cada dedo posee dos juntas. Por cada junta se tiene una matriz de transformación $A_1$ y $A_2$, Ecuación (4). Finalmente se tiene la expresión de posición del robot multiplicando la matriz $A_i$ correspondiente a cada junta del robot, donde el primer renglón corresponde a la falange proximal y el segundo la falange medial para cada dedo, Ecuación (5).

$$T = A_1 \cdot A_2 = \begin{bmatrix} \cos(\theta_1) & -sen(\theta_1) & 0 & a_1\cos(\theta_1) \\ sen(\theta_1) & \cos(\theta_1) & 0 & a_1\cos(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta_2) & -sen(\theta_2) & 0 & a_2\cos(\theta_2) \\ sen(\theta_2) & \cos(\theta_2) & 0 & a_2\cos(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$T = A_1 \cdot A_2 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & a_1\cos(\theta_1) + a_2\cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & a_1\sin(\theta_1) + a_2\sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

### 2.3. Análisis dinámico

El análisis dinámico del robot es de gran utilidad para calcular el torque necesario para cada actuador. Se realiza a partir de las ecuaciones de posición del mismo resultantes de la matriz de transformación $T$ y con las fuerzas que intervienen en él. La Ecuación (6) describe la ecuación general de torque.

$$F = \sum_{i=1}^{n} F_{rep_i}\Omega + \sum_{i=1}^{n} F_{att_i}\Omega = \sum_{i=1}^{n} J_{ai}^{T}\Omega, \quad (6)$$

donde Ω es la carga, $F_{rep}$ son las fuerzas repulsivas y $F_{att}$ son las fuerzas atractivas que intervienen y $J^T$ es el Jacobiano transpuesto de la ecuación de posición final de cada articulación del robot. Se calcula el Jacobiano de las funciones que se encuentran en la última columna de las matrices $A_1$ y $T$ respectivamente, Ecuaciones (7) y (8):

$$J_{A1} = \begin{bmatrix} -a_1 sen(\theta_1) & 0 \\ a_2\cos(\theta_2) & 0 \end{bmatrix} \quad (7)$$

$$J_T = \begin{bmatrix} -a_1 sen(\theta_1) - a_2 sen(\theta_1+\theta_2) & -a_1 sen(\theta_1+\theta_2) \\ a_1\cos(\theta_1) + a_2\cos(\theta_1+\theta_2) & a_2\cos(\theta_1+\theta_2) \end{bmatrix} \quad (8)$$

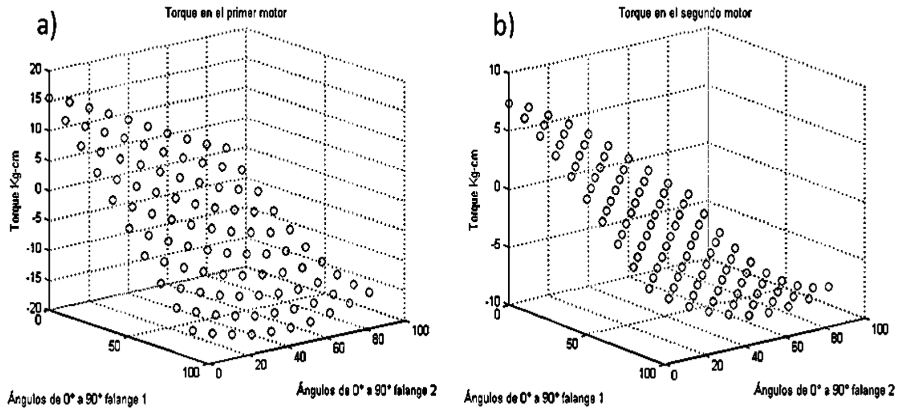



**Fig. 2.** Torque en los motores, a) primera falange, b) segunda falange.

Finalmente se realiza la suma de los productos de los Jacobianos transpuestos por el vector Ω que son las fuerzas que actúan en unidades Newton metro:

$$F = \begin{bmatrix} -2l_1\,sin(\theta_1) + 2l_1\,cos(\theta_1)\Omega_2 - l_2\,sin(\theta_1+\theta_2) + l_2\,cos(\theta_1+\theta_2)\Omega_2 \\ -l_1\,sin(\theta_1+\theta_2)\Omega_1 + l_2\,cos(\theta_1+\theta_2)\Omega_2 \end{bmatrix} \quad (9)$$

La Ecuación (9) se evalúa con un barrido de ángulos de 0° a 90° para cada articulación del gripper y en la Figura 2 se muestra el torque para cada posición que puede tomar cada motor correspondiente a las falanges, expresado en kilogramos por centímetro, con lo cual se seleccionaron motores con torque de 15 kg-cm. Para este cálculo se consideró la aplicación de una fuerza producida por una masa de dos kilogramos, también se incluye un coeficiente de fricción de 0.48 correspondiente al plástico ABS (Acrilonitrilo Butadieno Estireno) [7].

## 3. Diseño y construcción mecánica

### 3.1. Análisis de esfuerzos y criterio de falla

El diseño del gripper está íntimamente relacionado con las dimensiones de los motores y sensores que se implementaron. Este diseño se realizó en el software SolidWorks, Figura 3, con el cual se realizó un estudio de análisis de esfuerzos para cada una de las piezas que componen el gripper, con el fin de conocer la intensidad de las fuerza internas que actúan dentro del cuerpo, calcular las deformaciones del cuerpo y proveer un estudio de la estabilidad del mismo cuando está sometido a fuerzas externas, lo cual depende del tamaño de los miembros, sus deflexiones y su estabilidad dependen de las cargas internas y el material del que están hechas y utilizando la teoría del esfuerzo cortante máximo o criterio de Tresca, se predice el esfuerzo de falla de un material dúctil sometido a cualquier carga. En consecuencia para evitar la falla, el esfuerzo constante máximo debe ser menor o igual a la mitad de la resistencia a la tracción del material ($\sigma_y/2$) [8]. En la Tabla 1 se muestran los resultados de someter a las piezas del prototipo a la simulación de análisis de esfuerzos, considerando una fuerza de 20 Newtons, equivalente a más del doble de la fuerza a la cual será sometido el prototipo.

**Tabla 1.** Tensión máxima de Von Mises y factor de seguridad de los elementos del prototipo.

| **Pieza** | **Tensión de Von Mises Máxima $(N/m^2)$** | **Factor de Seguridad Mínimo (Tresca).** | **Nodo** |
|---|---|---|---|
| Base inferior | 3.89559e+006 | 7.09025 | 15022 |
| Falange proximal 1 parte inferior | 1.97519e+007 | 1.31537 | 15630 |
| Falange proximal 2 parte inferior | 1.87217e+007 | 1.38977 | 106 |
| Falange distal 1 parte inferior | 1.82166e+007 | 1.43566 | 70 |
| Falange distal 2 parte inferior | 1.79774e+007 | 1.45655 | 70 |
| Eslabón corto de 5.5 cm | 705789 | 31.8458 | 125 |
| Eslabón largo de 8.5 cm | 313511 | 77.6142 | 11246 |

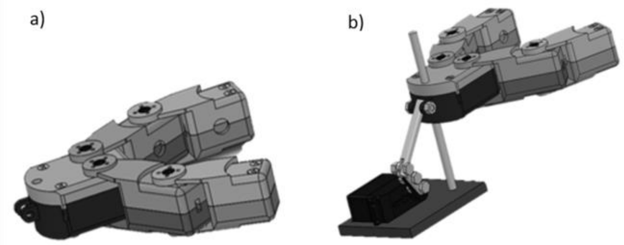



**Fig. 3.** Prototipo ensamblado, a) gripper completo, b) prototipo completo.

### 3.2. Maquinado e impresión 3D de los elementos constituyentes del prototipo

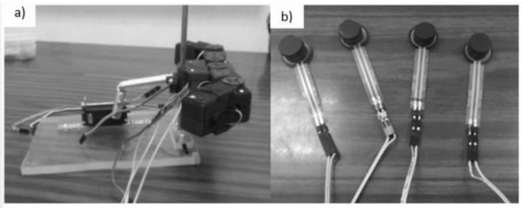



**Fig. 4.** Implementación, a) prototipo b) superficie de contacto en los sensores de presión.

La construcción de los elementos constituyentes del gripper se realizó mediante la impresión 3D. Por otro lado, los eslabones del mecanismo de desplazamiento lineal se maquinaron de forma convencional con ayuda del torno y la fresadora. En la Figura 4 se muestran los elementos maquinados, impresos y ensamblados. Además de la implementación del sistema sensorial, en el cual se coloca espuma de caucho en la superficie de contacto de cada sensor, brindando deformación al contacto.

## 4. Diseño e instrumentación electrónica

El prototipo es alimentado con tres baterías de 12 Volts a 2 amperios, las cuales son reguladas por tres fuentes conmutadas de bajada al voltaje de alimentación de los servomotores (6 volts a máximo 2 amperios). Están diseñadas con el circuito integrado LM2596 en la configuración recomendada por el fabricante. Con la finalidad de evitar algún corto circuito al conectar las baterías, se implementó un circuito sencillo con diodos e interruptores de encendido y apagado del prototipo.

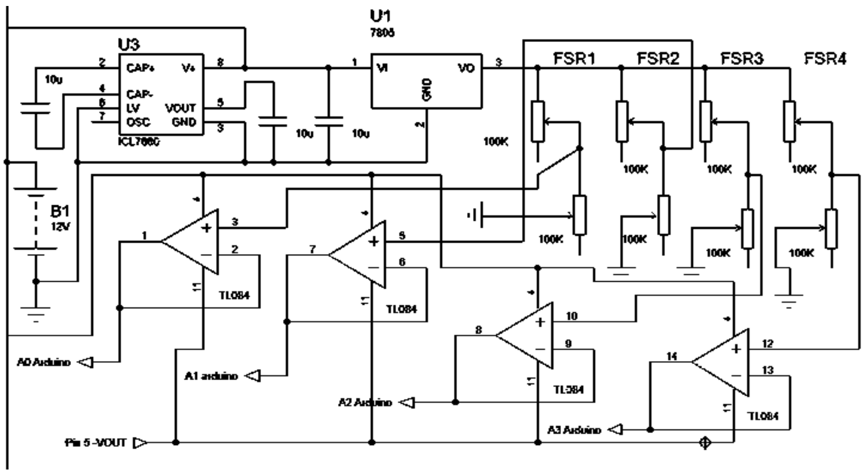



**Fig. 5.** Circuito completo de instrumentación de los sensores de presión.

El sensor de presión FSR tiene un comportamiento logarítmico que al incrementar la fuerza de aplicación en él, la resistencia disminuye [9]. Se usó la configuración recomendada por el fabricante en la que puede trabajar el sensor, en la cual el voltaje de salida es calculado por la Ecuación (10), que es un divisor de voltaje conectado a un seguidor de tensión construido con amplificadores operacionales, este último con el objetivo de acoplar impedancias y evitar la caída de tensión al conectarlo con el microcontrolador, el uso de amplificadores operacionales hace necesaria la inclusión de un circuito inversor de voltaje, matricula 7660, para evitar el uso de fuentes de tensión negativas de voltaje, el circuito electrónico completo de instrumentación de los sensores de presión se muestra en la Figura 5. Finalmente se utiliza un micro controlador ATMEGA328P en la etapa digital, debido a su implementación en la plataforma Arduino, el cual es el encargado de ser la unidad de lectura e interpretación de los sensores, realizar el reconocimiento de patrones y controlar los motores mediante modulación por ancho de pulso.

$$V_{salida} = \frac{V_{+}R}{(R + R_{FSR})} \tag{10}$$

## 5. Reconocimiento de patrones

### 5.1. Adquisición de patrones

Para entrenar el algoritmo de reconocimiento de patrones previamente se deben extraer las características más sobresalientes de los objetos a clasificar, en este caso, estas características son la forma de los objetos que el gripper puede sostener, la diferencia entre la posición inicial y final de la falange proximal del primer dedo AFP1, la diferencia entre la posición inicial y final de la falange distal del primer dedo AFD1, la diferencia entre la posición inicial y final de la falange proximal del segundo dedo AFP2, la diferencia entre la posición inicial y final de la falange distal del segundo dedo AFD2, y la presión registrada por el sensor, ejercida entre el objeto y el gripper, Tabla 2. Esto se logra con el prototipo ensamblado con las etapas mecánica y electrónica en conjunto con la inclusión de un programa en el micro controlador que permite realizar la lectura de los sensores de presión y controlar la posición de los motores. Apoyado con un circuito que consta de cinco botones, los cuales envían señales de control al micro controlador, estas señales son: reinicio, activar ascenso del mecanismo, activar movimiento de los eslabones del gripper, activar interrupción 1, activar interrupción 2. Una vez que se enciende el prototipo, el gripper toma la posición inicial, se presiona el botón que activa el movimiento de las falanges proximales de forma simultanea hasta hacer contacto con el objeto y se presiona el botón de la interrupción 1 para detener el movimiento de las falanges proximales y comenzar con el movimiento de las falanges distales, posteriormente al hacer éstas contacto con el objeto se presiona el botón 2 para detener finalmente el movimiento de las falanges, se toma lectura de los valores arrojados por los sensores y la información es visualizada en la pantalla de una computadora al igual que la posición final de cada motor, se presiona el botón que acciona el mecanismo ascendente y se demuestra la sujeción del objeto. Se presiona el botón de reinicio, el gripper toma la posición inicial y repite el proceso. Los datos provenientes del sensor con los que se trabaja han sido discretizados en el intervalo de cero a 1024.

**Tabla 2.** Diferencia en grados, posición de los motores.

| **Diferencias entre valores iniciales y finales de las características de los objetos** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Objeto | AFP2 | AFD2 | AFP1 | AFD1 | SFP2 | SFD2 | SFP1 | SFD1 |
| Cartón | -89 | -66 | 89 | 66 | -4 | 282 | -4 | 96 |
| Tarjeta | -89 | -64 | 89 | 64 | 2 | 144 | 2 | 88 |
| Caja | -89 | -58 | 89 | 58 | -5 | 105 | -5 | 58 |
| Sintra 3 mm | -89 | -64 | 89 | 64 | -6 | 191 | -6 | 107 |
| Acrílico 1 cm | -89 | -76 | 89 | 76 | -5 | 344 | -5 | 308 |
| Botella 200 gr | -58 | -115 | 58 | 115 | 62 | -5 | 62 | 210 |
| Botella 1 kg | -70 | -125 | 70 | 125 | 11 | 130 | 111 | -21 |
| Tostada | -89 | -65 | 89 | 65 | -1 | -2 | -1 | 137 |
| Huevo | -45 | -75 | 45 | 75 | 2 | -2 | 2 | 82 |
| Cubo Rubik | -77 | -71 | 77 | 71 | -6 | -2 | -6 | 394 |

### 5.2. Diseño del algoritmo de reconocimiento de patrones

Es necesario dividir el espacio de trabajo en cuatro regiones a clasificar que corresponden a la prensión por oposición terminal, prensión palmar cilíndrica, agarre rectangular y agarre irregular, Figura 6. Se agregan las dos últimas regiones debido a que el gripper por sus características mecánicas puede realizar estas prensiones, pero la mano humana no, por lo cual no pueden clasificarse en las regiones de prensión por oposición terminal y palmar cilíndrica, de forma que hacerlo podría generar un error en el sistema de control por solo tener dos regiones a clasificar en un amplio espacio de trabajo.

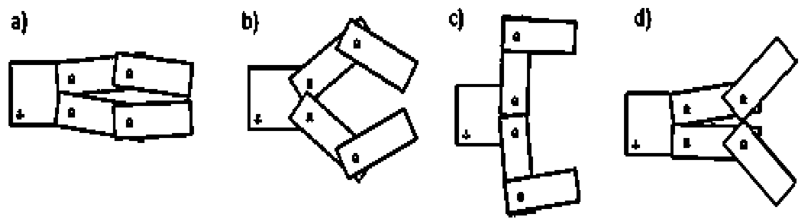



**Fig. 6.** Posiciones del gripper, a) prensión por oposición terminal, b) prensión palmar cilíndrica, c) agarre rectangular, d) agarre irregular.

### 5.3. Clasificador euclidiano modificado

Este clasificador resulta de aplicar el cuadrado a la distancia euclidiana, distancia mínima entre un punto y otro, Ecuación (11), y despreciar los términos no discriminantes de la ecuación desarrollada, Ecuación (12) [10].

$$D = \sqrt{|x_1 - z_1|^2 + |x_2 - z_2|^2} \tag{11}$$

$$f(x) = X^T \cdot Z - \frac{1}{2} Z^T \cdot Z \quad (12)$$

Cuando la distancia euclidiana calculada es mínima el valor de la Ecuación (12) es máximo. La función lineal que regionaliza el espacio de los patrones en dos clases separadas, Ecuación (13), es la diferencia entre dos patrones evaluados en la Ecuación (12). Siguiendo la convención de las Ecuaciones (14) y (15), el lugar geométrico de la función resultante es igual con cero, es la mediatriz del segmento que une los centroides de las dos clases.

$$f_d(x) = f_1(x) - f_2(x) \quad (13)$$

$$f_d(x) > 0 \rightarrow f_1(x) > f_2(x) \quad (14)$$

$$f_d(x) < 0 \rightarrow f_1(x) < f_2(x) \quad (15)$$

Se clasifica en una clase a los resultados con signo positivo y en otra a los resultados con signo negativo. De igual manera para más de dos regiones, la clasificación se realiza con una combinación de resultados con signo positivo y negativo. A partir de los datos AFP2, AFD2, AFP1 y AFD1 de la Tabla 2, se calculó el valor absoluto y posteriormente el promedio de estos valores y después se incrementó o decremento este valor, evaluándolos en las Ecuaciones (12) y (13) hasta llegar a la separación de las regiones, prensión por oposición terminal, prensión palmar cilíndrica, agarre rectangular y agarre irregular, obteniendo los patrones de entrenamiento: [60, 10], [30, 30], [5, 0] y [60, -30], respectivamente, donde las ecuaciones finales son las siguientes.

$$f_1(x) = 2[X_1\ X_2] \cdot \begin{bmatrix} 60 \\ 10 \end{bmatrix} - [60\ 10] \cdot \begin{bmatrix} 60 \\ 10 \end{bmatrix} = 120X_1 + 20X_2 - 3700 \quad (16)$$

$$f_2(x) = 2[X_1\ X_2] \cdot \begin{bmatrix} 30 \\ 30 \end{bmatrix} - [30\ 30] \cdot \begin{bmatrix} 30 \\ 30 \end{bmatrix} = 60X_1 + 60X_2 - 1800 \quad (17)$$

$$f_3(x) = 2[X_1\ X_2] \cdot \begin{bmatrix} 5 \\ 0 \end{bmatrix} - [5\ 0] \cdot \begin{bmatrix} 5 \\ 0 \end{bmatrix} = 10X_1 - 25 \quad (18)$$

$$f_4(x) = 2[X_1\ X_2] \cdot \begin{bmatrix} 60 \\ -30 \end{bmatrix} - [60, -30] \cdot \begin{bmatrix} 60 \\ -30 \end{bmatrix} = 120X_1 - 60X_2 - 4500 \quad (19)$$

Se calculan las dos funciones discriminantes utilizando la Ecuación (13) entre $f_1(x)$ y $f_2(x)$ y entre $f_2(x)$ y $f_3(x)$.

$$f_{12}(x) = f_1(x) - f_2(x) = 60X_1 - 40X_2 - 1900 \quad (20)$$

$$f_{23}(x) = f_2(x) - f_3(x) = 50X_1 + 60X_2 - 1775 \quad (21)$$

## 6. Implementación del sistema de control

Utilizando MATLAB se graficaron las regiones del clasificador divididas por dos funciones discriminantes, Figura 7. La pendiente negativa resultado de entrenar el algoritmo con los patrones [5, 0] y [30, 30]. La pendiente positiva es resultado de entrenar el algoritmo con los patrones [60, 10] y [30, 30].

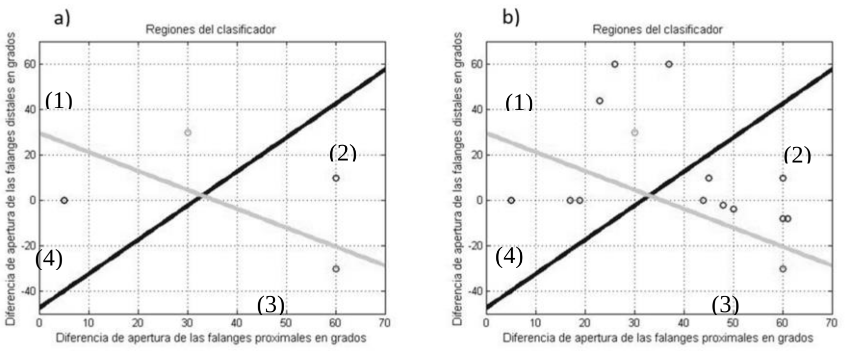



**Fig. 7.** Clasificador, a) Regiones, 1: cilíndrica, 2: plana, 3: irregular, 4: rectangular, b) Elementos clasificados.

**Tabla 3.** Resultados de la clasificación de distintos objetos.

| **Objeto** | **Resultado Ecuación 1** | **Resultado Ecuación 2** | **Ángulo 1 (°)** | **Ángulo 2 (°)** | **Tipo de prensión** |
|---|---|---|---|---|---|
| Tarjeta | 2080 | 795 | 60 | -8 | O. Terminal |
| Botella de aceite | 1260 | 485 | 50 | -4 | O. Terminal |
| Cubo rubik | 740 | 425 | 44 | 0 | O. Terminal |
| Tostada | 2080 | 795 | 61 | -8 | O. Terminal |
| Huevo | 1060 | 505 | 48 | -2 | O. Terminal |
| Pinzas | 380 | 125 | 38 | 0 | O. Terminal |
| Botella de 200 ml | -2740 | 3125 | 26 | 60 | P. Cilíndrica |
| Botella de 1 Lt | -2280 | 2015 | 23 | 44 | P. Cilíndrica |
| Botella de 600 ml | -2080 | 3675 | 37 | 60 | P. Cilíndrica |
| Eje de robot serpiente | -76 | -825 | 19 | 0 | Rectangular |
| Multímetro | -880 | -925 | 17 | 0 | Rectangular |

El algoritmo de control desarrollado sobre la plataforma Arduino, coloca al gripper en una posición inicial y se mantiene hasta recibir la señal de activación, después los motores que corresponden a las falanges proximales comienzan a moverse simultáneamente de modo que la apertura entre una y otra disminuye y realiza la lectura de todos los sensores de presión. Posteriormente existen dos escenarios. El primero si alguno de los sensores de presión que se encuentran en las falanges proximales detecta variación, se detendrán los motores activando los motores correspondientes a las falanges distales, los cuales se mueven de igual forma que los anteriores, disminuyendo la apertura entre ellos hasta que uno de los sensores colocados en estas falanges detecte algo o bien se toquen entre ellas.

El segundo escenario sucede cuando los sensores de presión colocados en las falanges proximales no detectan nada y se llega a una distancia mínima entre ellos provocando que los motores en las falanges distales se muevan como el caso anterior. Una vez que los motores del gripper se han detenido, la posición actual en grados de cada uno es evaluada en el algoritmo de reconocimiento de patrones y éste determina el tipo de prensión que realiza el gripper. Dependiendo del resultado, el gripper disminuye la distancia entre las falanges proximales y entre las falanges distales de modo que la presión entre el objeto y el gripper aumenta hasta llegar a un umbral, el cual es distinto para cada región. Finalmente se activa el mecanismo de desplazamiento lineal deslizando el gripper hacia arriba y lo mantiene hasta que llega una nueva señal de activación que deshace el agarre. Se probó el prototipo ensamblado y con el algoritmo de reconocimiento de patrones implementado en el sistema de control, obteniendo los resultados mostrados en la Figura 7 y la Tabla 3.

## 7. Conclusiones

Se desarrolló el modelo matemático de un gripper robótico antropomórfico a los dedos primero y segundo, analizando a cada dedo como un robot de dos grados de libertad, lo cual permite ubicar al robot en cualquier punto en su espacio de trabajo, además el modelo sirvió de base para el análisis dinámico del robot, lo que permitió conocer el torque necesario para cada motor. La respuesta logarítmica del sensor de presión contribuye al correcto funcionamiento del prototipo ya que cuando el sensor es sometido a una presión inicial se genera un cambio de valor con respecto al inicial, lo cual sirve para determinar de forma sencilla que el sensor entró en contacto con un objeto y que este cambio no es generado por ruido o interferencia. Se llevaron a cabo distintas pruebas de agarre y clasificación del sistema con objetos de distintas formas y pesos, presentando resultados satisfactorios. Una mejora al prototipo es que esté apoyado de un sistema de visión artificial el cual le indique dónde se encuentra el objeto y busque la mejor posición en que pueda sostenerlo.

## Referencias

1. Cheraghpour, F., Feizollahi, A.: Design, Fabrication and Control of a Three Finger Robotic Gripper. In: First International Conference on Robot, Vision and Signal Processing, pp. 280–283 (2011)
2. FESTO, Adaptative gripper DHDG. pp. 1–4 (2011)
3. Jiang, Y., Amed, J.R., Lipson, H., Saxena, A.: Learning hardware agnostic grasps for a universal jamming gripper. In: IEEE International Conference on Robotics an automation, pp. 2385–2391 (2012)
4. Jagannathan, S., Galna, G.: Adaptative Critic Neural Network Based Object Grasping Control Using a Three Finger Gripper. IEEE Transactions on neural networks, Vol 15, No. 2, pp. 395–407 (2004)

5. Sun, L., Shan, J.H., Meng, M., Zhang, D., Mei, T.: Application of Intelligent Flexible Skin Sensors for Interfacing with Robotic Pets. In: First IEEE International Conference on Nano/Micro Engineered and Molecular Systems, pp. 1527–1531 (2006)
6. Spong, M.W., Hutchinson, S., Vidyasagar, M.: Forward and inverse kinematics: Robot Modeling and Control. John Wiley and Sons, pp. 75–186 (2005)
7. bove-ag Plásticos y elastómeros. http://www.plasticosmecanizables.com/plasticos_coeficiente_friccion.html
8. Hibeler, R.C.: Transformación de deformación unitaria: Mecánica de Materiales. Pearson, pp. 542–545 (2006)
9. Interlink Electronics. http://www.interlinkelectronics.com/datasheets/ Datasheet_FSR.pdf
10. Theodoridis, S., Koutroumbas, K.: Classifiers Based on Bayes Decision Theory. Pattern Recognition, Elsevier, pp. 30–34 (2009)

# Identifying Topics about Leadership and Entrepreneurship using Topic Modelling

Silvia Beatriz González-Brambila, Josué Figueroa-González

Universidad Autónoma Metropolitana, Distrito Federal, Mexico

{sgb,jfgo}@correo.azc.uam.mx

**Abstract.** Terms like leadership and entrepreneurship have reached great importance and popularity. A lot of information has appeared related with these topics and searching for it has become a bit complicated for people interested in leadership and entrepreneurship. With so much information, it is difficult to identify a topic among the huge quantity of concepts involved in these themes. This work presents a process based on text processing, text mining and topic modelling for finding topics and patterns in interviews about these topics in order to identify the main themes that leaders and entrepreneurs talk about.

**Keywords:** Topic model, text mining, knowledge discover, leadership, entrepreneurship.

## 1 Introduction

Concepts like leadership and entrepreneurship have grown in the last years, nowadays there are research branches, schools of thoughts and a lot of information like books, speeches, conferences, workshops, etc. has appeared. Even, researches about characteristics that a person must have in order to be considered a leader or an entrepreneur have been done [9]. Many people are interested in these topics and look up for any information that allows them to become a leader or an entrepreneur.

There are a lot of topics related with these terms, for example: concepts that a leader or an entrepreneur must know, suggested books and authors for reading, conferences, seminars, and leaders or entrepreneurs experiences and talks. Internet has become one of the main sources of information for almost any topic, in any format. However, the amount of information makes difficult its search and its easy to be overwhelmed with a lot of concepts and ideas. Talks, conferences, speeches and interviews appear every day in different formats, as documents, web pages, video and audio. People like listening experts and well know leaders and entrepreneurs, but even this subset of information contains a lot of terms, concepts and ideas, so it becomes difficult to identify the main topics, especially when a person its not related with these two popular themes.

For this reason, a research over leadership and entrepreneurship is proposed using concepts related with text mining and topic models. The information considered for the work is contained in audio files, common known as podcast, a media that has grown considerably during the last years thanks to the Internet. The main contribution of this works is the analysis of concepts like leadership and entrepreneurship obtained from interviews with leaders and entrepreneurs using the Topic Model framework in order to identify what do they talk about.

This paper is composed by the following sections: section 2 presents the basic ideas of the elements and frameworks involved in the process of text analysis. Section 3 presents the description of the steps performed during the analysis considering as base Text Mining and Topic Modelling processes. Section 4 contains the results obtained in each step and their interpretation. Finally, section 5 contains the conclusions and future research.

## 2 Knowledge Discovery

Several techniques have appeared in order to manage, process, discover, understand and exploit knowledge from any kind of information and format. Many of these concepts have similar processes and performance, so it could be a little complicated distinguish one from another, for example: Knowledge Discover on Databases, Topic Modelling, and Data and Text Mining.

All of them can be used for data analysis and information retrieval which give a certain value to the users. However, even if their general process can look similar, there are specific differences that distinguish each other.

### 2.1 Knowledge Discover on Databases

Process for extracting useful information has received many names, for example: Data Mining, Text Mining, Knowledge Extraction or Information Discover, but nowadays these concepts are being considered as part of Knowledge Discover on Databases (KDD).

There are several definitions for KDD [7], but in general it can be resumed as the process for obtaining or discovering knowledge from data using different processes, for example: Data Mining which is considered as a step in the whole KDD process.

### 2.2 Data Mining

Data Mining is the process of applying specific algorithms for obtaining patterns in data [8]. Its main objectives can be classified in: verification and discovering.

In verification, the system only verifies a hypothesis, meanwhile in discovering, the goal is to automatically find new patterns in data. Applied, these two goals are transformed in prediction and description.

In order to reach these goals, Data Mining uses a set of techniques like: classification, regression, summarizing, dependencies modelling and other that can be grouped in: statistical for hypothesis validation or learning methods for recognizing patterns, however some of the techniques can be shared between both groups [5].

### 2.3 Text Mining

Text Mining consists in obtaining patterns of interest from documents or data, especially when these patterns are not so obvious even for experts [13]. The main difference with Data Mining, is that Text Mining works with non-structured data, for example, words in a document, and Data Mining works with structured data, like the stored in Data Bases [14].

Text Mining can be seen as an option for Data Mining for a special kind of data, it also can be used like a process similar to KDD, also, Text Mining can be used as a tool for structuring data contained in a group of documents so them can be analyzed with Data Mining techniques.

### 2.4 Topic Modelling

Topic Model [13] is a mathematical framework which main purpose is to discover the hidden patterns in a document in order to identify the topics that are in a group of documents for exploiting their knowledge. It assumes that in any document, some words will appear more frequently than other, and using some statistical analysis of these words, identifies the different topics that can be found in a group of documents, identifying the relationship between them.

This technique is used specially over non structured data and documents collections, for example: texts, web sites content, comments over social networks [11, 10] and e-mails. Topic Models algorithms can be used also for analyze genetic information and images.

Topic model algorithms can be classified into two main groups: sampling-based and variational, the most common algorithms are: Gibbs [4] for sampling and Variational Estimation Maximization (VEM) [15] for variational ones.

### 2.5 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [3] is a hierarchical Bayesian Model used for obtaining statistical relationships between groups of documents. LDA is based in two concepts:

- Any document is composed by a distribution of topics,
- Each topic is a distribution of words.

LDA assumes that a topic is a distribution over a fixed set of words, and those topics have been specified before data were generated. Then, the generation of a document is performed in the following way:

- A topic distribution is chosen,

– For each word in a document:
  - A topic from the topic distribution is chosen randomly,
  - A word is randomly chosen from the vocabulary distribution of the previous selected topic.

For LDA, a group of documents share the same topics, but each document presents them in different proportions. Documents are observable data; meanwhile the structure of topics, their vocabulary and distribution per document are the hidden data. LDA looks up for discovering the hidden data or structures using an inverse process to the one that generates the document.

## 3 Discovering Topics

Although the work of this paper was focused in analyze the content of audio files, podcast, their information was extracted to text files, so, the process for analyzing the information and finding the most common terms about leadership and entrepreneurship is related to Text Mining and Topic Model.

For the information analysis, it was used the software R, especially the libraries related with text processing and topic modelling: *tm*, *slam*, *topicmodels*, *snowballC*, and *clue*.

### 3.1 Obtaining and Integrating Information

A group of podcast related with leadership and entrepreneurship were obtained from different sources [1], also some transcription from interviews were considered [12, 6]. From the amount of available interviews, the selected ones talked about different topics, for example, products or business, and concepts about becoming and being a leader or an entrepreneur.

In the Integration step, the content of the audio was expressed as a text file using the application Adobe Premier Pro CC. Also, other text files were generated from the content of the transcription from the interviews. The analysis was performed with different kind of information: interviews with leaderships, with entrepreneurs and combining some interviews of both types.

The fact of choosing interviews which combine general concepts of leadership and entrepreneurship with concepts about different products and business, offered an interesting analysis of the information.

### 3.2 Selection, Cleaning and Transformation

This process, also known as data cooking, processes the data in order to obtain a more appropriate representation for their analysis. For the content obtained from audio files, punctuation marks didn't exist, but other elements which could be easy removed with some routines in R, like numbers, here appeared as words, so a special process had to be applied to eliminate them.

The *Stop Words*, which are words considered as the most common in a language, had to be filtered before the topic classification process. The same occurs with blank spaces and other group of words (pronouns, connectors and articles). A word can be reduced to its original root, this is called stream process, some tests were performed with steamed documents and other without applying this step.

The format of the interviews obtained from web content contains the names of the interviewer and the interviewed, after some tests, these names appeared as frequent words in many topics, so them had to be cleared from the text files. The amount of topics was tested from 2 to 20, obtaining similar results; finally, the decided quantity of topics chosen for the analysis was 8.

### 3.3 Topic Modelling

After the data cooking process, Topic Modelling framework was applied. With a fixed number of topics selected, the topic identification was realized with three models: VEM, VEM with an $\alpha$ fixed and Gibbs. Once the topics were obtained, a review of how close were the results for each model was performed using the concepts of Hellinger distance and the Linear Sum Assignment Problem (LSAP), this analysis was performed only for VEM with VEM using an $\alpha$ fixed and VEM with Gibbs.

### 3.4 Topic Interpretation

Once the topics and the vocabulary for each one were obtained, the next step was interpreting them. To do this, were considered the most 5 frequent words in each topic, and according to them, the main topics were identified. The results are shown in Section 4.

## 4 Results

In this section, are presented the results obtained in each step of processing the information.

### 4.1 Selection, Cleaning and Transformation

After processing the podcast and the interviews, a group of files was created, one composed only for data related with leadership, another about entrepreneurship and the last one with mixed interviews.

Each group was composed by ten interviews, and for the combined group, were randomly chosen five for each one of the other groups.

After the cleaning process performed with R routines, were obtained the following quantity of terms and documents: Leadership, 1960 terms and 176 documents. Entrepreneurship, 1368 terms and 146 documents. Mixed, 1987 terms and 183 documents.

### 4.2 Topic Modelling

For the Topic model process, were proposed 8 topics per group of interview, thought some tests were performed with different amounts of topics obtaining similar results. The Hellinger distance for entrepreneurship documents between terms identified using VEM and VEM with $\alpha$ Fixed was 0.27678, and with VEM and Gibbs was 0.74399. For leadership using VEM and VEM with $\alpha$ Fixed was 0.24620, and with VEM and Gibbs was 0.75104. Finally, the distance for mixed documents using VEM and VEM with $\alpha$ Fixed was 0.26176, and with VEM and Gibbs was 0.75303.

### 4.3 Topic Interpretation

In order to identify the topics obtained, were considered the 5 most used terms in each topic. Only the topics identified with VEM and Gibbs are presented because VEM and VEM with $\alpha$ Fixed models produced almost the same terms.

Some topics in the entrepreneurship interviews were related with a product or business, for this reason, these terms aren't shown in the results tables, which only present the terms and topics that talk about general concepts of entrepreneurship. These results are shown in Table 1 and Table 2.

**Table 1.** Relevant topics and terms identified with VEM for entrepreneurship interviews.

| Topic 1 | Topic 2 | Topic 4 | Topic 6 | Topic 7 | Topic 8 |
|---|---|---|---|---|---|
| business | time | companies | services | service | business |
| legal | portfolio | business | costs | companies | costs |
| money | career | consulting | cost | costs | think |
| disposable | work | source | reduction | customers | smart |
| working | make | product | consulting | time | idea |

Topics not considered talk more about a certain product or business containing terms like: restaurant, online, product and career, this is because some of the businesses owned by an entrepreneur were from food and online services.

Terms and topics identified with Gibbs model that are not included in the results contains terms like: online, site, software and transportation.

Topics and terms identified for entrepreneurship are very clear. Talking about this theme, some main aspects which are very important for an entrepreneur can be identified. For example:

- For starting a business, many entrepreneurs talk about studying the market and having an idea, Topics 8 in VEM and 7 in Gibbs models are related to this aspect.
- Financial aspects are very important too, Topic 1 in VEM model is related with them. Management its very important, Topic 6 in Gibbs model is related with this concept.

**Table 2.** Relevant topics and terms identified with Gibbs for entrepreneurship interviews.

| Topic 2 | Topic 3 | Topic 4 | Topic 5 | Topic 6 | Topic 7 |
|---|---|---|---|---|---|
| market | time | need | company | services | business |
| value | able | think | opportunity | costs | people |
| first | help | selling | products | source | think |
| new | sales | product | service | consulting | need |
| venture | better | customer | make | reduction | idea |

As the results show, some topics of every model are related with more than one concept because talking about entrepreneurship involves a lot of concepts that can appear in different ideas.

Topics and terms related with leadership are more general because interviews talked about general concepts of this theme, not about a certain product like in the entrepreneurship ones, for this reason, more topics are presented in the results. Table 3 and Table 4 present the topics for leadership interviews.

**Table 3.** Relevant topics and terms identified with VEM for leadership interviews.

| Topic 1 | Topic 2 | Topic 3 | Topic 5 | Topic 6 | Topic 7 |
|---|---|---|---|---|---|
| people | cash | years | design | cares | time |
| time | business | business | vision | leader | market |
| company | leadership | plan | process | patient | energy |
| process | people | data | organization | organization | important |
| organization | make | experience | people | will | world |

**Table 4.** Relevant topics and terms identified with Gibbs for leadership interviews.

| Topic 1 | Topic 2 | Topic 5 | Topic 6 | Topic 7 | Topic 8 |
|---|---|---|---|---|---|
| leaders | world | time | things | market | people |
| organization | data | work | need | change | business |
| will | across | years | want | global | environment |
| culture | well | understand | job | strategic | important |
| plan | much | think | patient | different | meetings |

Topics identified with leadership and their respective terms, also can be grouped according main themes of this concept.

- Topic 6 of VEM and Topic 1 in Gibbs can be related with the characteristics of a good leader must have.

- Topic 2 of VEM and 7 of Gibbs are related with financial aspects, this is because most of the interviews were about leaders in an business.
- Topics 3 from VEM and 5 from Gibbs can be identified as planning and time, concepts really important for a good leader.
- Topics 1 and 5 from VEM and Topic 8 from Gibbs models are related with organization of an enterprise or business.
- Topic 7 from VEM and Topics 2 and 6 from Gibbs can be related with concepts that leaders must manage.

Terms obtained from mixed interviews show a vocabulary more related with leadership with some words referring to entrepreneurship, like customers or business. With these interviews, VEM method produced better results than Gibbs. Notice than the terms about a specific product or business disappear. Table 5 and Table 6 present the topics for mixed interviews.

**Table 5.** Relevant topics and terms identified with VEM for mixed interviews.

| Topic 1 | Topic 4 | Topic 5 | Topic 8 |
|---|---|---|---|
| time | people | care | people |
| good | business | example | plan |
| need | process | patient | data |
| thing | things | leaders | talk |
| things | important | change | meetings |

**Table 6.** Relevant topics and terms identified with Gibbs for mixed interviews.

| Topic 1 | Topic 2 | Topic 4 |
|---|---|---|
| leaders | customers | business |
| organization | important | world |
| culture | certain | plan |
| leadership | different | care |
| might | know | working |

Using the Hellinger distance and LSAP framework, the topics from VEM and Gibbs with best relationship were obtained. The relationship between topics from every kind of interview and every model is presented in Table 7

## 5 Conclusions and Future Work

It's interesting to observe the results, where there are many terms which apparently don't have a lot in common between them; however, this is mainly because

**Table 7.** Main relationships between VEM and Gibbs models for leadership and entrepreneurship topics.

| Interviews | VEM | Gibbs |
|---|---|---|
| Leadership | Topic 6 | Topic 6 |
| | Topic 6 | Topic 6 |
| Entrepreneurship | Topic 2 | Topic 8 |

of the different products and business that entrepreneurs talked about in their interviews, something that didn't occur with the leadership interviews, where there are more topics related to the main theme.

There are some topics that clearly identify aspects related with these two themes. The fact of combining both types of interviews produced a result a little different than just merging the main terms obtained separately. It's important to notice that the main topic in the mixed interviews is related to leadership, with just some words about entrepreneurship, specially talking about products, money or business.

Talking about the steam process, applying this step reduces the amount of terms per topic, however sometimes it's difficult to understand the results viewing only a part of the word. It's more important to eliminate the most frequent words; besides the routines available in R, it's recommended to apply another cleaning process for this words.

In order to improve the analysis, more interviews could be considered. Also these interviews mustn't be related to a product or business, but concepts about becoming, being and thinking as a leader or entrepreneur. Using other kind of sources, like speeches or talks could also contribute for obtaining better results.

About the models used for the analysis, VEM and VEM Fixed, produced better results than Gibbs. VEM Fixed results aren't shown, but they are almost the same as VEM.

The most difficult part of the whole process is to identify the topics according the vocabulary contained in each one. It's necessary to rely on an expert in the topic, so the results can be interpreted in a better way.

As future works, once the main topics and terms about leadership and entrepreneurship are known, it's planned to structure them in order to perform a Data Mining analysis for knowing more specific information about a particular theme, for example:

- Evolution of topics over time,
- Books, conferences, talks or workshops that someone interested in the topics should read or attend,
- Concepts about leadership and entrepreneurship in a specific area, for example: sports, education, business, politics, etc.

Hundreds of interviews and other kind of information in many formats about almost any theme are appearing every minute over Internet. Extracting the content of audio files makes easier the process of obtaining information, because many of the data is being generated in audio and video files.

Another topic that can be very interesting for studying is the one related with saving money, analyzing interviews, speeches, conferences from experts in the called financial education.

Processing this content with the Topic Model framework, offers a great possibility for studying and discovering knowledge from this huge amount of information that otherwise could be difficult to explore.

## References

1. Audio Books, Podcasts, Videos, and Free Downloads to Learn From, http://www.bbc.co.uk/podcasts
2. BBC Podcast and Downloads, http://www.bbc.co.uk/podcasts
3. Blei, D.: Probabilistic Topic Models. Communications of the ACM. 4, 77–84 (2012)
4. Casella, G., George, E: Explaining the Gibbs Sampler. The American Statistician, 3, 167–174 (1992)
5. Chen, L., Sakaguchi, T., Frolick M.: Data Mining Methods, Applications, and Tools. Information Systems Management, 17, 65–70 (2006)
6. Entrepreneur interviews, http://www.entrepreneurship-interviews.com
7. Fayyad, U., Piatesky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery in Databases. AI Magazine, 3, 37–53 (1996)
8. Fayyad, U., Piatesky-Shapiro, G., Smyth, P.: The KDD Process for Extracting Useful Knowledge from Volumes of Data. Communications of the ACM, 11, 27–34 (1996)
9. Fernald, W. Jr., Solomon, G., Tarabishy.: A New Paradigm: Entrepreneurial Leadership. Southern Business Review, 2, 1–11 (2005)
10. Ghosh, D., Guha, R.: What are we tweeting about obesity? Mapping tweets with topic modelling and Geographic Information System. Cartography and geographic information science, 40, 90–102 (2013)
11. Hong, L., Davison, B. D.: Empirical study of topic modelling in twitter. In: Proceedings of the First Workshop on Social Media Analytics (2010)
12. Interviews with leaders articles and insights, http://www.mckinsey.com/insights/leading_in_the_21st_century/interviews_with_leaders
13. Mooney, R., Nahm,Y.: Text Mining with Information Extraction. Multilingualism and Electronic Language Management. In: Proceedings of the 4th International MIDP Colloquium (2005)
14. Ronen, F., et al.: Text mining at the term level. In: Principles of Data Mining and Knowledge Discovery, Springer Berlin Heidelberg, 65–73 (1998)
15. Tzikas, D. G., Likas, C. L., Galatsanos, N. P: The variational approximation for Bayesian inference. Signal Processing Magazine, 6, 131–146 (2008)

# Implementación sobre FPGA de la estrategia evolutiva CMA-ES para optimización numérica

Leopoldo Urbina, Carlos A. Duchanoy, Marco A. Moreno-Armendáriz,
Derlis Lara, Hiram Calvo

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

**Resumen.** En este trabajo se presenta la implementación sobre hardware del algoritmo *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES). Este algoritmo se basa en la adaptación de la matriz de covarianza que se focaliza inicialmente en una región del espacio de búsqueda particular y posteriormente se mueve o crece a lo largo del espacio de búsqueda, según sea conveniente para encontrar el valor óptimo. Los resultados experimentales muestran que dicha implementación será de gran utilidad para resolver problemas de optimización numérica en un sistema embebido.

**Palabras clave:** Estrategia evolutiva CMA-ES, implementación sobre FPGA, optimización numérica.

## FPGA Implementation of the Evolutive Strategy CMA-ES for Numerical Optimization

**Abstract.** In this paper, a hardware implementation of the *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) algorithm is presented. This algorithm is based on the adaptation of the covariance matrix, initially it is focused on a region of the particular search space and subsequently, it moves or grows along the search space, as appropriate to find the optimum value. The experimental results reveal that this implementation will be very useful to resolve numeric optimization issues in an embedded system.

**Keywords:** Evolutive strategy CMA-ES, numerical optimization, FPGA implementation.

## 1. Introducción

Uno de los objetivos más comunes de implementar algoritmos sobre alguna arquitectura o hardware especifico es el de obtener mayor velocidad de procesamiento de datos y menor utilización de recursos en un sistema, por lo general los algoritmos que se implementan de esta manera son los bio-inspirados que se caracterizan por consumir muchos recursos y tiempo de ejecución.

El propósito de este trabajo no es reducir el tiempo de procesamiento, el objetivo es implementar el algoritmo evolutivo CMA-ES sobre *Field-Programmable Gate Array* (FPGA), el cual pueda servir posteriormente para implementar todo un sistema de optimización numérica sobre un hardware específico, sin necesidad de una computadora de escritorio.

Se llama hardware evolutivo [7] a la integración del cómputo evolutivo y un dispositivo de hardware programable, cuyo objetivo es la reconfiguración "autónoma" de la estructura del hardware para mejorar el desempeño. En [16] se propuso la aceleración de un algoritmo genético (AG) implementándolo sobre un FPGA. Las características del AG implementado son: selección aleatoria de los padres, la cual mantiene el sistema de circuitos de selección; un modelo de memoria de estado estacionario, el cual ocupado un espacio constante al ser embebido en el chip; supervivencia de los cromosomas hijos más aptos sobre los cromosomas padres menos aptos, lo que promueve la evolución. El AG implementado sobre el FPGA está organizado en un proceso segmentado (pipeline) de seis estados, donde a cada estado se le asigna el mismo tiempo de procesamiento igual a un ciclo de reloj. Para sus experimentos, consideraron dos problemas, el problema del cubrimiento de conjuntos y el problema de plegamiento de proteínas.

En [15], se presenta el diseño de un algoritmo genético en *Very High Speed Integrated Circuit Hardware Description Language* (VHDL) llamado *Hardware-based Genetic Algorithm* (HGA) con el objetivo de implementarlo en hardware. Debido a los procesos de segmentación, paralelización y al no sobrecargo de funciones, un AG en hardware alcanza una velocidad significativa sobre un AG en software, lo que es especialmente útil cuando el AG es usado para aplicaciones de tiempo real, por ejemplo, planificación del disco y registro de imágenes.

En [10], se realiza el diseño y la implementación de un Algoritmo Genético Compacto sobre FPGA, los resultados logrados demuestran que tener un algoritmo de búsqueda como un Algoritmo Genético Compacto en hardware, es muy útil, ya que permite realizar optimización numérica global en tiempo real, y además permite incorporarlo en un chip que forme parte de una aplicación de mayor escala. Un ejemplo de aplicación se desarrolla en [8], donde un control automático inteligente para el aterrizaje de una aeronave es diseñado, basado en un algoritmo hibrido entre redes neuronales recurrentes y algoritmos genéticos. Otros trabajos relacionados se presentan en [1,9,4,11,3].

En este trabajo se logró la implementación del algoritmo CMA-ES en el Nios II. Este trabajo se desarrollará en las siguientes secciones: en la sección 2, se describirá a detalle la estructura del algoritmo CMA-ES; en la sección 3 se explicará el desafío que fue lograr la implementación del algoritmo; los resultados logrados por el trabajo se reportarán en la sección 4; la interpretación de resultados y las concluciones obtenidas, se justificarán en las secciones 5 y 6 respectivamente; por último, las ideas propuestas para continuar con el desarrollo de este trabajo, están redactadas en la sección 7.

## 2. Algoritmo CMA-ES

Las estrategias evolutivas fueron inventadas por Ingo Rechenberg [12] y Hans-Paul Schwefel [14] en los años 60's. El operador principal para mejorar el desempeño del individuo es la mutación. Este operador se basa en la distribución normal con media $\mathbf{m}$ y desviación estándar $\boldsymbol{\sigma}$. $\mathbf{m}$ y $\boldsymbol{\sigma}$ son parámetros de estrategia del algoritmo y se incluyen en el genoma del individuo. Por esta razón, la característica principal de las estrategias evolutivas es la auto-adaptación. El CMA-ES [5] es usado para problemas de minimización y se describe a continuación:

**Paso 1. Ajuste de parámetros de la estrategia.** Establecer el número de puntos de búsqueda $n$, el número de hijos $\lambda$, el número de padres $\mu$, los pesos $w_i, i = 1, \ldots, \mu$ y los parámetros $c_\sigma, d_\sigma, c_c, c_1$, y $c_\mu$ que es usada para cálculos complementarios.

**Paso 2. Inicialización.** Inicializar el número de generación g = 0, la matriz de covariancia $\mathbf{C}^{(0)} = \mathbf{I}$, el tamaño del paso $\sigma^{(0)} = 0.5$, el camino de evolución $\mathbf{p}_\sigma = 0, \mathbf{p}_c = 0$, y el valor inicial de la media $\mathbf{m}^{(0)} \in \mathbb{R}^n$.

**Paso 3. Inicio.** Generar una población de puntos de búsqueda usando una distribución normal, de la siguiente manera:

$$\mathbf{x^{(g+1)}}_k \sim \mathcal{N}(\mathbf{m^{(g)}}, (\sigma^{\mathbf{(g)}})^2 \mathbf{C^{(g)}}) \text{ for } k{=}1,\ldots,\lambda. \tag{1}$$

**Paso 4. Selección y recombinación.** Obtener el valor de aptitud $f(\mathbf{x_k^{(g+1)}})$ de los puntos de búsqueda y seleccionar los $\mu$ mejores puntos $\mathbf{x}_{i:\lambda}^{(g+1)}$ para $i = 1, \ldots, \mu$. Actualizar el valor de la media de la distribución usada para la búsqueda como sigue:

$$m^{(g+1)} = \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}^{(g+1)}. \tag{2}$$

**Paso 5. Adaptación de la matriz de covariancia.** La matriz de covarianza se actualiza como sigue.

$$\begin{aligned}\mathbf{C}^{(g+1)} &= (1 - c_1 - c_\mu)\mathbf{C}^{(g)} + c_1 \left(\mathbf{p}_c \mathbf{p}_c^T + \delta(h_\sigma)\mathbf{C}^{(g)}\right) \\ &+ c_\mu \sum_{i=1}^{\mu} w_i \left(\frac{\mathbf{x}_{i:\lambda} - \mathbf{m}}{\sigma}\right)\left(\frac{\mathbf{x}_{i:\lambda} - \mathbf{m}}{\sigma}\right)^T\end{aligned} \tag{3}$$

Donde,

$$
\begin{aligned}
\mathbf{p}_c^{(g+1)} &= (1-c_c)\mathbf{p}_c^{(g)} + h_\sigma \sqrt{c_\sigma(2-c_\sigma)\mu_{eff}} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma}, && (4) \\
c_c &= \frac{4+\mu_{eff}/n}{n+4+2\mu_{eff}/n}, && (5) \\
c_1 &= \frac{2}{(n+1.3)^2+\mu_{eff}}, && (6) \\
c_\mu &= {}_{min}\left({}_{1-c1,\alpha_\mu} \frac{\mu_{eff}-2+1/\mu_{eff}}{(n+2)^2+\alpha_\mu \mu_{eff}/2}\right), && (7) \\
\alpha_\mu &= 2, && (8) \\
\delta(h_\sigma) &= (1-h_\sigma)c_c(2-c_c), && (9) \\
h_\sigma &= \begin{cases} 1 & \text{if } \frac{\|\mathbf{p}_\sigma\|}{\sqrt{1-(1-c_\sigma)^{2(g+1)}}} \\ & < (1.4+\frac{2}{n+1})E\|\mathcal{N}(\mathbf{0},\mathbf{I})\|, \\ 0 & \text{otherwise.} \end{cases} , && (10) \\
E\|\mathcal{N}(\mathbf{0},\mathbf{I})\| &= \sqrt{n}\left(1-\frac{1}{4n}+\frac{1}{21n^2}\right). && (11)
\end{aligned}
$$

**Paso 6. Control del tamaño de paso.** El tamaño del paso se actualiza de la siguiente manera:

$$
\sigma^{(g+1)} = \sigma^{(g)} exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma\|}{E\|\mathcal{N}(\mathbf{0},\mathbf{I})\|}-1\right)\right), \tag{12}
$$

Donde,

$$
\begin{aligned}
\mathbf{p}_\sigma^{(g+1)} =& (1-c_\sigma)\mathbf{p}_\sigma^{(g)} + \sqrt{c_\sigma(2-c_\sigma)\mu_{eff}} \\
& \cdot \left(\mathbf{C}^{(g)}\right)^{-\frac{1}{2}} \frac{\mathbf{m}^{(g+1)}-\mathbf{m}^{(g)}}{\sigma^{(g)}}.
\end{aligned} \tag{13}
$$

**Paso 7. Criterio de finalización.** Si se cumple con alguna de las siguientes condiciones como lo es el máximo número de generaciones, el máximo número de evaluaciones de aptitud o se llega al valor de umbral de la aptitud; entonces se termina el proceso de búsqueda. Si no se cumple ninguna de las condiciones anteriores, entonces g = g +1 y se regresa al **Paso 3**.

En cada iteración del algoritmo, se actualizará la matriz de covarianza **C** y por lo tanto, también se actualizarán sus vectores y valores propios, esto significa que la magnitud del espacio de búsqueda de la solución deseada ira incrementando y adaptando en cada iteración, siendo así el espacio de búsqueda una elipse cada vez más enfocada (adaptada) a encontrar lo que se está buscando, como se puede apreciar en la Figura 1.

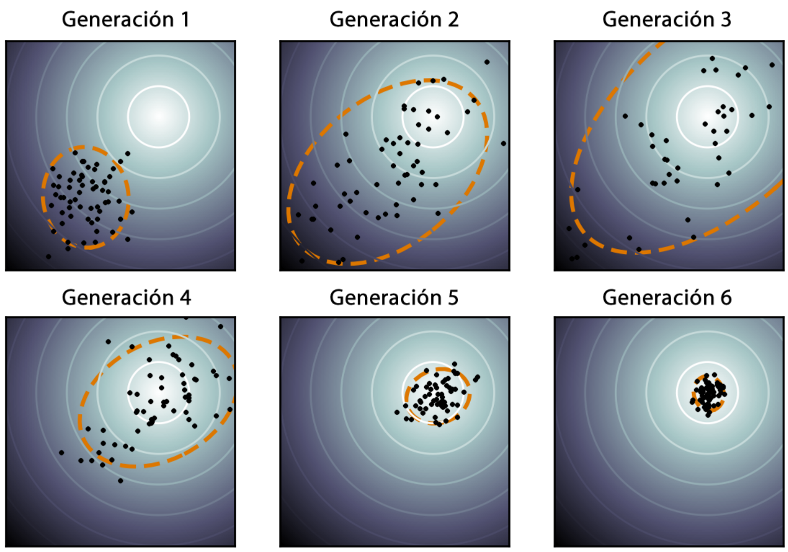



Fig. 1: Ejemplo de búsqueda adaptativa mediante la actualización de la matriz de covarianza C

## 3. Implementación sobre FPGA

Para llevar a cabo esta implementación fue necesario usar un procesador NIOS II [2]. El primer paso es realizar su inicialización con las siguientes características: Bloques de memoria de 4kbit de 32 bits, Dos puertos (uno de salida y otro de entrada) de 8 bits y el JTAG UART para tener comunicación con una computadora de escritorio. El segundo paso fue codificar el algoritmo de la sección 2, un problema que se encontró fue como realizar el cálculo de los eigenvectores y eigenvalores de matriz de covarianza **C**, después de revisar la literatura, se decidió usar el algoritmo QR, el cual se basa en la descomposición QR, desarrollada en la década de 1950 por John G.F. Francis (Reino Unido) y Vera N. Kublanovskaya (URSS).

Una vez implementado el Nios II y cargado el algoritmo CMA-ES sobre este, se verificó su correcto funcionamiento, un detalle a resolver fue la generación de la población inicial, esta se debe generar aleatoriamente, para hacerlo en lenguaje C se usa la función *rand*, esta función está programada con base a un algoritmo que toma como referencia un valor semilla y a partir de este se generan números que en realidad son pseudoaleatorios, por *default* la semilla es estática, es decir aunque se ejecute de nuevo el programa se obtendrán los mismo valores, para corregir esta situación se utiliza el reloj del sistema con la función *srand (time(0))*, con lo cual a cada ejecución la semilla tendrá un valor diferente y así los números pseudoaleatorios generados serán diferentes. Para más detalles se puede consultar [6].

## 4. Experimentos y resultados

Para realizar la evaluación de la implementación realizada, se usaron 7 funciones de aptitud, estas funciones representan problemas de optimización mono objetivo [13]. Además, se programaron versiones del CMA-ES en Matlab® y lenguaje C, con el fin de validar el funcionamiento del algortimo.

Tabla 1: Valores de inicialización del CMA-ES.

| **Valor** | **Uso** |
|---|---|
| $n = 4$ | Dimensión del problema y tamaño de la población |
| stopfitness $= 1 * E^{-11}$ | Valor mínimo a alcanzar en el valor de aptitud |
| stopeval $= N^2 * 10^3 = 16000$ | Máximo número de evaluaciones |
| $\lambda = 4 + floor(3 * log(n)) = 8$ | Número de hijos |
| $\sigma = 0.5$ | Tamaño del paso |
| $\mu = \frac{\lambda}{2}$ | Número de puntos para la recombinación |
| $w_i = log(\mu + 0.5) - log(1 : \mu)'$ | Pesos para la recombinación |

Nikolaus Hansen, el creador del algoritmo CMA-ES, explica detalladamente en [5] el funcionamiento del algoritmo, asi como consejos para su correcto uso e implementación. En la Tabla 1 se muestran los valores iniciales requeridos para la ejecucción del algoritmo CMA-ES.

Tabla 2: Valores finales de las funciones de aptitud.

| **Función** | **FPGA** | **Matlab®** | **Lenguaje C** |
|---|---|---|---|
| *Esfera* | 2.557E-07 | 7.221E-07 | 2.523E-09 |
| *Rosenbrock* | 9.998E-01 | 1.000E+00 | 1.000E+00 |
| *Elli* | -5.166E-08 | -3.456E-08 | -2.324E-09 |
| *Diffpow* | 6.739E-03 | 2.155E-03 | -5.409E-03 |
| *Cigar* | 1.183E-07 | -1.377E-07 | -8.210E-09 |
| *Tablet* | 6.536E-08 | 3.634E-07 | 1.954E-09 |
| *Cigtab* | -8.436E-09 | 8.827E-08 | 1.457E-03 |

Las pruebas del algoritmo CMA-ES sobre el Nios II se lograron pese a algunos inconvenientes, en el caso particular de las funciones *Cigtab*, *Diffpow* y *Tablet*, estas no convergían en la mayor parte de las ejecuciones, esto fue debido a que la dispersión de los valores aleatorios generados como solución influyen en la rapidez de convergencia del algorítmo, es decir, si los valores iniciales generados son muy dispersos entre si, es muy probable que al ser procesados por el CMA-ES no se llegue a una solución aceptada, ya que cabe la posibilidad de que los valores en la matriz de covarianza **C** sean ínfimos o excesivos, esto repercutiría en el aumento de recursos computacionales para obtener los eigenvalores y eigenvectores, a tal grado que se puede estar iterando por muchos ciclos sin conseguir un resultado aceptable. Para solucionar este problema, se puede ejecutar el algoritmo nuevamente esperando que la propuesta inicial de solución no sea tan dispersa y se pueda procesar por el CMA-ES de manera adecuada, o en su defecto de ser posible asignar una mayor cantidad de recursos en el diseño del Nios II. En nuestro caso se optó por omitir las ejecuciones que no tuvieron un resultado concluyente, es decir, solo se tomaron en cuenta las ejecuciones que hayan concluido de manera satisfactoria (llegar al valor de umbral de la aptitud).

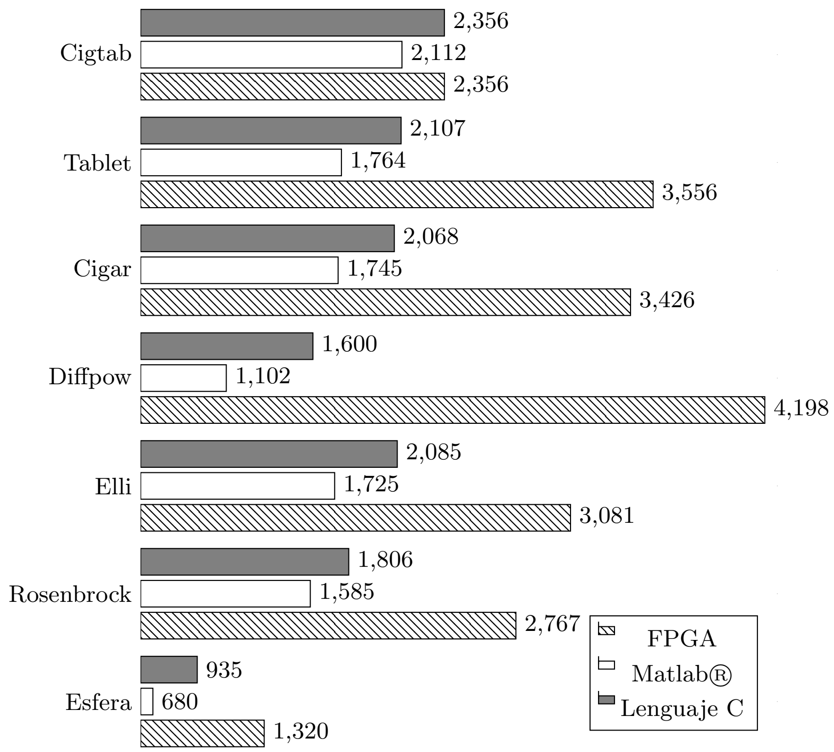



Fig. 2: Número de evaluaciones realizadas por la función de aptitud.

Se reportan tres resultados importantes del algoritmo CMA-ES, como son: número de iteraciones realizadas (Ver Figura 2), último valor obtenido para la función de aptitud (Ver Tabla 2), valor óptimo obtenído por el algoritmo (Ver Tabla 3). Por cada función de aptitud, el algoritmo se ejecutó 45 veces, solamente se reportó el promedio de cada función.

La Tabla 2 reporta el valor numérico de la función de aptitud evaluado por el valor óptimo, el valor de la función de aptitud debe ser cero o cercano a este, con excepción de la función *Rosenbrock* cuyo valor de aptitud debe tender a uno. La Tabla 3 contiene información referente al valor óptimo para cada función de aptitud, los valores alcanzados deben de ser lo más próximo a cero.

Tabla 3: Valor óptimo obtenído por el algoritmo.

| **Función** | **FPGA** | **Matlab®** | **Lenguaje C** |
|---|---|---|---|
| *Esfera* | 5.231E-12 | 5.669E-12 | 5.287E-15 |
| *Rosenbrock* | 6.191E-12 | 6.806E-12 | 1.821E-14 |
| *Elli* | 5.508E-12 | 4.611E-12 | 1.065E-14 |
| *Diffpow* | 6.286E-12 | 6.777E-12 | 6.072E-13 |
| *Cigar* | 5.270E-12 | 7.157E-12 | 1.377E-14 |
| *Tablet* | 6.810E-12 | 7.546E-12 | 1.057E-14 |
| *Cigtab* | 6.031E-12 | 5.549E-12 | 1.068E-13 |

### 4.1. Discusión

Los resultados obtenidos en los experimentos presentados demuestran que la estrategia evolutiva CMA-ES se implementó de manera exitosa sobre el FPGA, sin embargo, presenta ciertas desventajas respecto a su ejecución sobre una computadora personal, como ejemplo, la velocidad y capacidad de procesamiento en las computadoras personales es mayor en comparación al Nios II del FPGA.

Como ya se mencionó la velocidad del procesamiento del CMA-ES depende del hardware en el cual se está ejecutando, lo cual indica que aún y cuando se ejecute en una computadora, la rapidez con que llegue a una solución dependerá de las características de dicha computadora por ejemplo, el tipo de procesador y memoria RAM con los que cuente, otro aspecto importante de mencionar que también afecta la rapidez de convergencia, es la forma en que se haya programado el CMA-ES, por ejemplo para los experimentos realizados convergió más rápido la versión del CMA-ES programada en Matlab® que la versión en lenguaje C ejecutándose sobre una computadora, esto debido a que Matlab® cuenta con un lenguaje de programación propio que permite operaciones de vectores y matrices, mientras que en lenguaje C, se programaron versiones de dichas funciones que tardan un poco más en dar resultados, ya sea por el algoritmo programado o por la forma de crear arreglos o matrices en lenguaje C, que es mediante ciclos "*for*", esto depende de la experiencia y habilidad que se tenga en lenguaje C.

Analizando el gráfico presentado en la Figura 2, podemos concluir que el lenguaje de Matlab® presentó el mejor desempeño, a diferencia de la implementación en FPGA, esto se debe al modo en que se genera la población inicial en el FPGA. Los resultados reflejados en la Tabla 2, demuestran que el Lenguaje C presenta una mayor precisión, seguido por el FPGA, dejando al lenguaje de Matlab® en tercera posición, esto demuestra que la implementación realizada en el FPGA compite con los lenguajes de alto desempeño. En la Tabla 3 se reporta el valor óptimo obtenido para cada función, este valor está totalmente ligado a los resultados de la Tabla 2, por lo tanto se obtuvieron los mismos resultados el Lenguaje C obtuvo la mejor precisión, caso contrario del lenguaje en Matlab®.

El resultado de embeber el algoritmo CMA-ES en un FPGA es exitoso y comparado con los lenguajes C y Matlab®, es competitivo, ya que como en un principio se aclaró, el objetivo de este trabajo no es el de competir en tiempo de procesamiento, si no, embeber de manera funcional el algoritmo genético.

## 5. Conclusiones

Se ha logrado implementar un algoritmo evolutivo como el CMA-ES sobre FPGA, capaz de obtener los mejores individuos de una generación, los cuales servirán para diversas aplicaciones, basta modificar la función objetivo a evaluar por el CMA-ES, por la propia del sistema sobre el cual se quiera aplicar el modelo realizado.

La rapidez de convergencia de este tipo de heurísticas depende de los siguientes factores: el hardware en el cual son procesadas; la herramienta con la cual son programadas; la forma en la que son programadas y los valores aleatorios que adquiere la población inicial. Se concluyó que Matlab® es una herramienta eficiente que facilita la programación de este tipo de heurísticas, sin embargo para implementar algoritmos sobre el Nios II es necesario programarlos en C/C++ o ensamblador, la eficiencia de la implementación dependerá de la experiencia, habilidad y destreza que se tenga en estos lenguajes de programación, para programar funciones de manera eficiente. La ventaja de realizar implementaciones de este tipo es la de tomar señales directas del sistema real, lo que da un mayor de certidumbre. Al contrario de sólo realizarlo mediante un modelo matemático. Utilizar heurísticas para la solución a diversos problemas de optimización resulta una alternativa muy competitiva, puesto que no existe un método riguroso para realizar esta actividad, si además es apoyada con un hardware tan versátil como un FPGA, los resultados son positivos para este tipo de sistemas computacionales aplicados a diversa tareas de ingeniería.

## 6. Trabajo a futuro

Con el deseo de continuar con el desarrollo alcanzado por este trabajo, se proponen las siguientes ideas: Proponer otro algoritmo para la obtención de los valores y vectores propios de una matriz; modificar el método de generación aleatoria de la población inicial; por último, aplicar el sistema desarrollado en diversas tareas específicas, por ejemplo, la sintonización de controladores de diversos tipos.

**Agradecimientos.** Los autores agradecen el apoyo del Instituto Politécnico Nacional (SIP-IPN, COFAA-IPN, BEIFI-IPN) y del gobierno mexicano (SNI y CONACYT).

## Referencias

1. Affi, Z., Badreddine, E., Romdhane, L.: Advanced mechatronic design using a multi-objective genetic algorithm optimization of a motor-driven four-bar system. Mechatronics 17(9), 489–500 (2007)
2. Altera: Tutoriales sobre cómo usar el procesador Nios II (2015), `https://www.altera.com/`
3. Calle, A., Pazmiño, P., Ponguillo, R.: Algoritmo de detección de bordes en imágenes con Nios II (2014)
4. Cupertino, F., Mininno, E., Lino, E., Naso, D.: Optimization of position control of induction motors using compact genetic algorithms. In: IECON 2006, 32nd Annual Conference on IEEE Industrial Electronics. pp. 55–60. IEEE (2006)
5. Hansen, N.: The CMA evolution strategy: A tutorial. Vu le 29 (2005)
6. Hernández Lara, D.: Implementación sobre FPGA de la estrategia evolutiva CMA-ES pp. 1–139 (2014)
7. Higuchi, T., Yao, X.: Evolvable hardware, vol. 11. Springer Science and Business Media (2006)
8. Juang, J.G., Chiou, H.K., Chien, L.H.: Analysis and comparison of aircraft landing control using recurrent neural networks and genetic algorithms approaches. Neurocomputing 71(16), 3224–3238 (2008)
9. Kim, D.H., Abraham, A., Cho, J.H.: A hybrid genetic algorithm and bacterial foraging approach for global optimization. Information Sciences 177(18), 3918–3937 (2007)
10. Moreno-Armendáriz, M.A., Cruz-Cortés, N., Duchanoy, C.A., León-Javier, A., Quintero, R.: Hardware implementation of the elitist compact genetic algorithm using cellular automata pseudo-random number generator. Computers & Electrical Engineering 39(4), 1367–1379 (2013)
11. Raygoza, J.J., Ortega, S., Chirino, C.A., Rivera, J.: Implementación en hardware de un SVPWM en un soft-core Nios II, Parte I. e-Gnosis 7 (2009)
12. Rechenberg, I.: Evolution strategy: Optimization of technical systems by means of biological evolution. Fromman-Holzboog (1973)
13. Ross, O.H.M., Sepulveda, R.: High Performance Programming for Soft Computing. CRC Press (2014)
14. Schwefel, H.P.P.: Evolution and Optimum Seeking: The Sixth Generation. John Wiley & Sons, Inc. (1993)
15. Scott, S.D., Seth, S., Samal, A.: A synthesizable VHDL coding of a genetic algorithm. Tech. rep., Technical Report UNL-CSE-97-009, University of Nebraska-Lincoln (1997)
16. Shackleford, B., Snider, G., Carter, R.J., Okushi, E., Yasuda, M., Seo, K., Yasuura, H.: A high-performance, pipelined, FPGA-based genetic algorithm machine. Genetic Programming and Evolvable Machines 2(1), 33–60 (2001)

# Influence of the Luminance L* during Color Segmentation in the L*a*b* Color Space

Rodolfo Alvarado-Cervantes[1], Edgardo M. Felipe-Riveron [2], Vladislav Khartchenko[2], Oleksiy Pogrebnyak[2]

[1] Universidad Nacional Autónoma de Mexico, Centro de Investigaciones Teóricas, Facultad de Estudios Superiores Cuautitlan, Mexico

[2] Instituto Politécnico Nacional, Centro de Investigacion en Computacion, Mexico

rodolfo.alvarado.cervantes@gmail.com; edgardo@cic.ipn.mx; vlad@unam.mx; olek@pollux.cic.ipn.mx

**Abstract.** In this paper a study of the influence of luminance L* at the L* a* b* color space during color segmentation is presented. The behavior of different color image segmentation algorithms is studied, in particular, the techniques based on only the Euclidean metric of a* and b* and an adaptive color similarity function defined as a product of Gaussian functions in a modified HSI color space. For the evaluation purposes, synthetic images that accurately assess the performance of the color segmentation were particularly designed. The testing system can be used either to explore the behavior of a similarity function (or metric) in different color spaces or to explore different metrics (or similarity functions) in the same color space. From the obtained results it follows that the color parameters a* and b* are not independent on the luminance parameter L* as one may initially assume.

**Keywords**: Color image segmentation; L*a*b* color space; color metrics; color segmentation evaluation; synthetic color image generation.

## 1 Introduction

Image segmentation consists of entire image partitioning into different regions, which are similar in some predefined manner. It is an important and difficult task in image analysis and processing. All subsequent steps such as object recognition depend on the quality of segmentation [1].

For some time, the development of segmentation algorithms attracted remarkable consideration but the relatively fewer efforts on their evaluation and characterization [2, 3, 4, 5]. None of the published automatic segmentation algorithms is generally applicable to all types of images and different algorithms are not equally suitable for particular applications. For these reasons, the evaluation of different segmentation techniques and their characterization are very important subjects in the study of segmentation [3, 5].

Perceptually uniform color spaces such as L*a*b*, with the Euclidean metric to quantify color distances are commonly used in color image segmentation of natural scenes using histogram based or clustering algorithms among others [1].

To evaluate the segmentation performance of the Euclidian metric in the L*a*b color space, we designed a system that generates synthetic color images and the associated ground truth (GT) and evaluates the results with Receiver operating characteristics (ROC) curves [7]. The synthetic images where designed to evaluate the efficiency of achieved color information from given segmentation algorithms and are explained in detail in Section 3. A comparative study with an adaptive color similarity function defined as a product of Gaussian functions in a modified HSI color space [6] is presented in Section 4. Finally, conclusions are given in Section 5.

## 2 Previous Works

The first comprehensive survey on image segmentation evaluation methods is presented in [2]. It brings a coherent classification of existing methods at the time. An up to date of 5 years of progresses in the subject is presented in [3] after the first survey. Another actualization is presented 5 years later [5], embracing together the principal methods of segmentation evaluation available until 2007.

In [4] a way to design synthetic images and a corresponding GT for evaluating segmentation algorithms is presented. They introduce a general framework and general design considerations. Also, an evaluation system for generating synthetic gray level images taking into account their design considerations is presented.

## 3 Design of Synthetic Images for Benchmark Testing

In [4] the authors present three important design considerations for creating synthetic images: 1. Synthetic images should be appropriate for a quantitative study and should allow objective evaluations of their properties; 2. The synthetic images should reflect the main features of real images, i.e. corruption factors, such as noise and blurring, variation of parameters such as size, shape, etc.; 3. The system should allow the generation of images with progressive variations of each parameter. In this way the study of the influence of each individual parameter is possible.

Comparative tests between an adaptive color similarity function [6] and the Euclidean metric in the L*a*b* color space [8] were performed. The manner in which the tests were implemented is as follows:

In the case of the L*a*b* color space, the RGB image was previously transformed to L*a*b* color space discarding in all cases the luminance L* in order to calculate the Euclidean distance on the planes a*b* (color information) independently of the illumination. Then, the centroids (average of the values a* and b*) representing the colors of the figure and the background in the color space L*a*b* were calculated. Details are shown in [8].

For the adaptive similarity function [6] the following steps were performed:

1. Samples of both background and figure were taken, from which centroid and standard color dispersion were calculated. Details can be consulted in [6]. 2. The 24-bit RGB image (true color) was transformed into a modified HSI color space. 3. For each pixel, the similarity function for the centroids of object and background was calculated creating two color similarity images (CSI) [6]. 4. Each pixel of the RGB image was classified by calculating the maximum value for each pixel position between the CSI images of the object and that of the background.

The base shape of the synthetic test image was created with the following features: 1. Concave and convex sections were included in order to make it more representative of real images, such as natural flowers. 2. Extreme omnidirectional curvature in the entire image was selected to hinder edge detection by mask edge detectors. 3. The object was centered in the image.

The resulting flower-shaped object in the image is considered as the object of interest and as the ground truth GT in all subsequent tests (Figure 1 left).

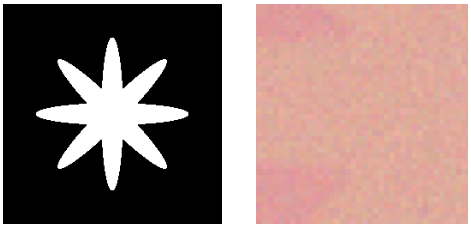


**Fig. 1.** Flower-shaped ground truth (left) and an image zoomed showing the Gaussian noise introduced (right).

In addition to this object of interest, several features were imposed in order to hinder its color-based segmentation:

1. Low contrast. The contrast between the object and the background in all images was very low for an observer, including some in which at a first glance the user cannot see the difference (e.g. Flower 5 in Figure 2). The difference between the color characteristics of the object of interest and the background we call “Delta” and it occurs at different directions of the HSI space. The tests were performed in color quadrants 0, 60, 120, 180, 240 and 300 degrees.

2. Blurred edges by an average filter. A sliding mean filter of size 3 x 3 pixels is applied to the whole image in order to blur the corners and to make detection of the object more difficult; this averaging is done before the introduction of Gaussian noise.

3. Introduction of Gaussian noise with SNR value = 1. The noise was applied to each RGB channel separately, and later we assembled the channels to create the noisy RGB color image with noise. Figure 1 right shows an example of such a noisy image.

The basic colors selected for both object and background were based on maintaining constant intensity at 0.5 and saturation at 0.3 and only varying the hue. Hue was selected as the parameter because its change integrates three RGB color channels together, making it more difficult to be processed by extending grayscale techniques on each color channel, thus forcing the segmentation algorithms in evaluation to use the color information holistically.

Samples of pixels corresponding to the figure were obtained by two squares of 2 x 2 pixels starting at the pixel (84, 84) and (150, 150). Samples for background pixels were obtained by two squares of 2 x 2 pixels starting at pixel (15, 15) and (150, 180).

The images were generated in the sectors 0, 60, 120, 180, 240 and 300 degrees corresponding to the images flower_0, flower_1 … flower_5 (Figure 2). To each of these test images we later applied a faded shadow in increments of 10% at each step.

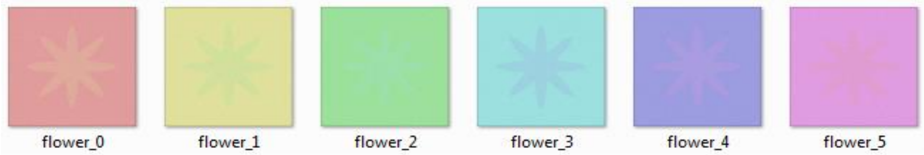



**Fig. 2.** Testing with Low Saturation with Delta in HUE.

A shadow fading was applied to all noisy blurred images with the light center in the fixed coordinates (150,150) in images of 256 x 256 pixels. It was applied gradually with 10% increments at each step. Figure 3 shows this for Flower 0.


**Fig. 3.** Example in color quadrants with a faded shadow applied at 0 degrees.

## 4 Results and Discussion

In this section we show the results in terms of TP (true positives) and FP (false positives) plotted against the level of shadow fading, representing by 10% of increment at each step. The first position means no shadow and position 11 means 100% shadow fading. All the images had the same post-processing: elimination of areas smaller than 30 pixels and a morphological closing with a circular structuring element of radius equal to two pixels.

The results of the application with the solution given by [6] of the color image segmentation with a different level of shadow fading (shown in every even row) compared with those obtained with the Euclidean metric in the L*a*b* rejecting L* (shown in every odd row) are included in Figure 4 for each color quadrant (0°, 60°, 120°, 180°, 240° and 300°) and at 10% increments of the shadow fading.

GT 0° 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

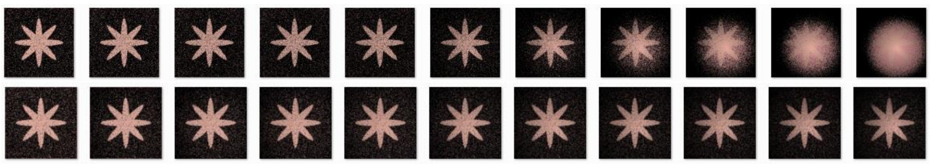



**Fig. 4**. Results of the color segmentation achieved between the Euclidean metric of a* and b* (top rows of each color) and the adaptive color similarity function (bottom rows of each color), for each color quadrant (0°, 60°, 120°, 180°, 240° and 300°) and at 10% increments of shadow fading in each step.

As it is shown in the graphs of Figure 5 and in coincidence with the visual analysis of the corresponding flower (Figure 4), segmentation failures in the L*a*b* space (right) start at different levels of faded shadow, whereas the adaptive color similarity function [6] is practically immune to the faded shadow (left).

We can see three general trends in the FP behavior in Figure 5 right (See Table 2): 1. Increase in an angle of approximately 45º in cases of Flower 0 and Flower 3 (with diamond marker); 2. Slowly and progressively increases in cases of Flower 1 and Flower 4 (with square marker) and 3. Sharply increases in cases of Flower 2 and

Flower 5 (with circular marker). The behavior is repeated every 180 degrees and coincides with the opponent color positions (yellow-blue for example).

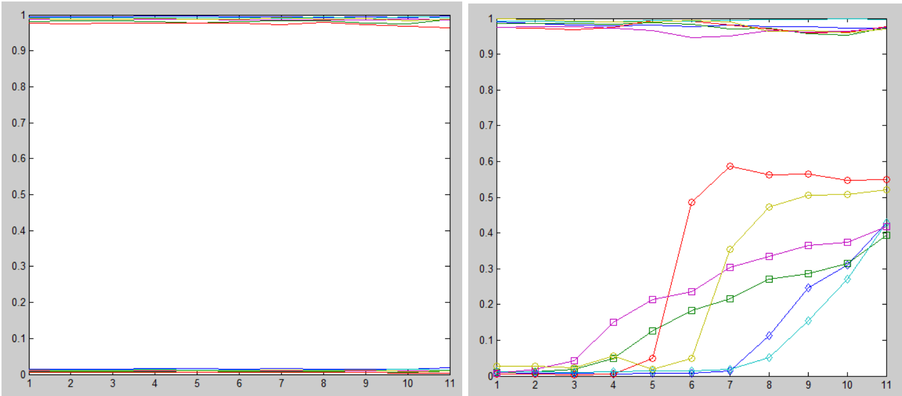



**Fig. 5.** Plots of the color similarity function [6] (left) and the Euclidean metric in L*a*b* rejecting L* (right).

Figure 6 shows details of the curves related to TP and FP of the color similarity function [6], with the following color code: Flower 0 (blue), Flower 1 (green), Flower 2 (red), Flower 3 (cyan), Flower 4 (purple) and Flower 5 (yellow). Variations in curves are lower than 1%.

To obtain a representative ROC curve illustrating behavior of the Euclidean metric in the L*a*b* space (rejecting L*) compared to the color similarity function [6] in all color sectors under study, we calculated the average TP and FP for all color flowers, obtaining the results shown in Figure 7.

**Table 2.** Observations concerning the behavior of the plot curves of the two colour metrics.

| Flower | Line Color | Euclidean metric in L*a*b* rejecting L* | Color similarity function [6] |
|---|---|---|---|
| 0 | Blue | 60% (position 7) Increases at 45° | Immune |
| 1 | Green | 30% (position 4) Increases slowly and progressively | Immune |
| 2 | Red | 40% (position 5) Sharply increases | Immune |
| 3 | Cyan | 70% (position 8) Increases at 45° | Immune |
| 4 | Purple | 20% (position 3) Increases slowly and progressively | Immune |
| 5 | Yellow | 50% (position 6) Sharply increases | Immune |

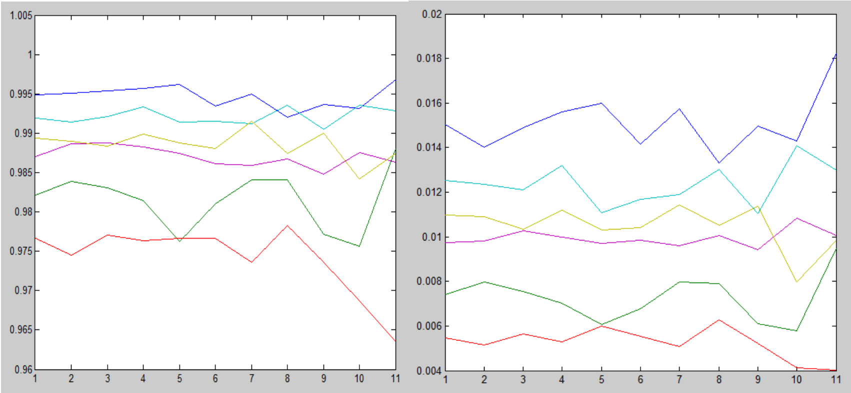



**Fig. 6.** Details of TP (left) and FP (right) of the color similarity function [6].

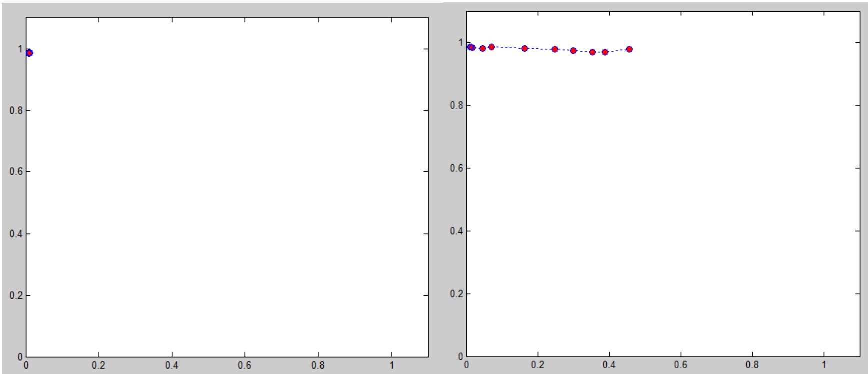



**Fig. 7.** ROC curve of the color similarity function [6] (left) and the Euclidean metric in the L*a*b* rejecting L.*

In the ROC curve corresponding to the average of TP and FP of all flowers, it can be seen that the results of the adaptive similarity function are maintained in the high efficiency area (coordinate 0, 1) while the color segmentation in L*a*b* space progressively moves away from the high efficiency area.

The L*a*b* results keep stable initially and later slowly and progressively moves to the upper right area of the ROC curve that can be thought of as the "liberal" side (coordinate 1, 1) as they make positive classifications, and, although there is weak evidence that almost all positives were classified properly, they have a high rate of false positives.

## 5 Conclusions

Regarding the evaluation of the color segmentation method with really difficult conditions, we can notice that the adaptive color similarity function performed well in all tests and remained close to the high efficiency zone of the ROC curves (coordinates 0,1) without noticeable changes when the level of faded shadow increases as shown in the corresponding PLOT curves.

The segmentation algorithm using the L*a*b* color space and discarding L* in calculating the Euclidean distance, suffered errors in all cases. It manifested in different degrees and at different levels of faded shadow (20% to 80%). Three types of trends or recurring symmetries can be noticed in sectors with 180 degrees of difference: 1. Rise of the curve gradually (Flowers 1 and 4); 2. Rise abruptly (Flowers 2 and 5), and 3. Increase near at 45° angle (Flowers 0 and 3).

As it can be seen from the results of both direct segmentation, and from PLOT & ROC curves, that the adaptive color similarity function in all cases exceeded the Euclidean distance in color space L*a*b* and discarding L*. The similarity function segmentation method performed well in all cases with rates higher than 95% of true positives (TP) and false positive (FP) rate less than 3% on average.

According to the experiment results we believe that keeping high values of TP increased only from the FP is due to the position of the center of the shadow fading in (150, 150). If this position is moved away from the object of interest, we can reduce the quantity of TP.

For future work we wish to evaluate different color zones like of different saturations, gray images, and with delta saturation among others. Our testing system can be used either to explore the behavior of a similarity function (or metric) in different color spaces or to explore different metrics (or similarity functions) in the same color space. Instead of exchanging color spaces in the experiments, it would only be necessary to exchange the metric or the similarity function.

It can be noticed that the non-consideration of the luminance parameter L* in calculating Euclidean distance (in each pixel of the object or of the background) did not made it immune to changes in lighting; so simple shadow can alter the quality of the results, concluding from them that the parameters a*b* from the color space L*a*b* are not independent of the L* parameter as one might suppose.

**Acknowledgements.** The authors of this paper wish to thank the Centro de Investigaciones Teóricas, Facultad de Estudios Superiores Cuautitlan (FES-C); Universidad Nacional Autónoma de México (UNAM), México; PAPIIT IN112913 and PIAPIVC06, UNAM; Centro de Investigación en Computación (CIC); Secretaría de Investigación y Posgrado (SIP); Instituto Politécnico Nacional (IPN), México, and CONACyT, México, for their economic support to this work.

## References

1. Plataniotis, K.N., Venetsanopoulos, A.N.: Color Image Processing and Applications. First Edition, Springer, Berlin Heidelberg Germany, 354 p. (2000)
2. Zhang, Y.J.: A Survey on Evaluation Methods for Image Segmentation. Pattern Recognition, Vol. 29, No 8, pp. 1335–1346 (1996)

3. Zhang, Y.J.: A review of recent evaluation methods for image segmentation. Proceedings of the 6th International Symposium on Signal Processing and Its Applications, pp. 148–151 (2001)
4. Zhang, Y.J., Gerbrands, J.J.: On the Design of Test Images for Segmentation Evaluation. Proceedings EUSIPCO, 1, pp. 551–554 (1992)
5. Zhang, Y.J.: A Summary of Recent Progresses for Segmentation Evaluation. In: Zhang Y.J., Advances in Image and Video Segmentation. IGI Global Research Collection, Idea Group Inc (IGI), pp. 423–439 (2006)
6. Alvarado-Cervantes, R., Felipe-Riveron, E.M., Sanchez-Fernandez, L.P.: Color Image Segmentation by means of a Similarity Function. In: 15th Iberoamerican Conference on Pattern Recognition, CIARP 2010, Rodolfo Alvarado-Cervantes, Edgardo M. Felipe-Riveron y Luis P. Sánchez-Fernandez, I. Bloch and R.M. Cesar, Jr. (Eds.): Sao Paulo, Brazil, LNCS 6419, pp. 319–328, Springer, Heidelberg (2010)
7. Fawcett, T.: An introduction to ROC analysis. Pattern recognition Letters, pp. 861–874 (2006)
8. http://www.mathworks.com/help/images/examples/color-based-segmentation-using-the-l-a-b-color-space.html (Revised on April 27, 2015)

# Instance Selection in the Performance of Gamma Associative Classifier

Jarvin A. Antón Vargas[1], Yenny Villuendas-Rey[1, 2], Itzamá López-Yáñez[2], Abril V. Uriarte-García[3]

[1] Universidad de Ciego de Avila, Departamento de Ciencias Informáticas, Cuba

[2] Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Mexico

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

janton@unica.cu, {yenny.villuendas, itzama}@gmail.com, auriarteb10@sagitario.cic.ipn.mx

**Abstract.** The Gamma associative classifier is among the most used classifiers of the alpha-beta associative approach. It had been used successfully to solve many Pattern Recognition tasks, including environmental applications. However, as most classifiers, Gamma suffers with the presence of noisy or mislabeled instances in the training sets. This paper evaluates the impact of using instance selection techniques in the performance of Gamma classifier. The numerical experiments carried out over well-known repository datasets allows to conclude that instance selection may increase the testing accuracy of the Gamma classifier.

**Keywords:** Gamma classifier, instance selection, supervised classification.

## 1 Introduction

The Gamma associative classifier [1, 2] was proposed recently to address supervised classification tasks, including regression. It belongs to the alpha-beta associative approach to Pattern Recognitions, due to its mathematical foundations. Gamma classifier had been used effectively to solve many recognition and prediction tasks, such as the prediction of development effort of software projects [3], estimation of pollutant contamination trough time [4] and determination of air quality in Mexico City [5]. However, as Gamma classifier stores a training set and uses it to assign class labels, it is affected by the presence of noisy or mislabeled instances.

Instance selection algorithms of the error-based (or editing) approach aims at removing the instances considered as outliers or misclassified [6, 7], smoothing decision boundaries and improving classifier accuracy. The first instance selection algorithm from the editing approach was proposed by Wilson in 1972 and named Edited Nearest Neighbor (ENN) [8]. It consist in the elimination of the instances misclassified by a k-NN classifier. Despite its simplicity, ENN have maintain a competitive performance with respect to recently proposed methods [6].

The GGE algorithm [9] is another well studied instance selection method. It was proposed by Toussaint in 2000 [9]. The GGE algorithm consist in deleting the instances connected to others of different class labels in a Gabriel Graph. It removes frontier instances, and keeps significant ones. A Gabriel graph is a directed graph such that two instances $x \in U$ and $y \in U$ form an arc if and only if $\forall z \in U \; (d((x + y/2), z) > d(x, y)/2)$, where d is a dissimilarity function. That is, two instances x and y are related in a Gabriel graph if there is no object in the hypersphere centered in the middle point of x and y, and with radius the distance between x and y.

We also considered in our study the MSEditB algorithm [10] for instance selection, which is a recently proposed algorithm, also graph-based. MSEditB was proposed in 2009 by García-Borroto et al. [10] and constructs a Maximum Similarity Graph (MSE) to determine the instances to delete.

A Maximum similarity graph is a directed graph such that each instance is connected to its most similar instances. Formally, let be S a similarity function, an instance $x \in U$ form an arc in a Maximum similarity graph with an instance $y \in U$ if and only if $d(x, y) = \max_{z \in U} d(x, z)$.

The MSEditB algorithms removes the instances having a majority of linked instances (successors and predecessors) not belonging to its class.

Most instance selection algorithms are proposed for improving the performance of the Nearest Neighbor (NN) classifier [11]. The approaches to instance selection that considered another classifiers such as Neural Networks [12] and ALVOT [13, 14] are quite specific and there are not appropriate for improving the Gamma classifier.

Instance selection algorithm proposed for Nearest Neighbor classifier [11] need a similarity function to determine neighborhood instances and to construct graph structures. To overcome this problem, Antón-Vargas et al. [15] proposed a novel similarity function based in the foundations of the Gamma classifier. However, that pioneer study does not considered the influence of feature weighting in the Gamma classifier.

This paper includes feature weighting in the performance of the Gamma associative classifier and explores the impact of instance selection in this scenario. The thorough experimental study carried out shows the significant performance gains of the proposed approach.

## 2 Gamma Classifier

The Gamma associative classifier belong to the alpha-beta approach of associative Pattern Recognition. That is due to it has its foundations on the Alpha and Beta operators of Alpha-Beta associative memories [16]. The Alpha and Beta operators are defined in a tabular form considering the sets $A = \{0, 1\}$ and $B = \{0, 1, 2\}$, as shown in figure 1.

| $\alpha : A \times A \rightarrow B$ | | |
|---|---|---|
| $x$ | $y$ | $\alpha(x, y)$ |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 2 |
| 1 | 1 | 1 |

| $\beta : B \times A \rightarrow A$ | | |
|---|---|---|
| $x$ | $y$ | $\beta(x, y)$ |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 0 | 1 |
| 2 | 1 | 1 |

**Fig. 1.** Operators Alpha and Beta.

To perform classification tasks, the Gamma associative classifier incorporates the unary operator $u_\beta$ and the generalized gamma similarity operator, $\gamma_g$, both based on the Alpha and Beta operators. The unary operator $u_\beta$ receives as an input a binary n-dimensional vector, and returns a number $p \in \mathbb{Z}^+$ according to the following expression:

$$u_\beta = \sum_{i=1}^{n} \beta\,(x_i, x_i), \tag{1}$$

the generalized gamma similarity operator receives as input two binary vectors $\mathbf{x} \in A^n$ y $\mathbf{y} \in A^m$, with $n, m \in \mathbb{Z}^+, n \leq m$, and also a non-negative integer θ, and returns a binary digit, as follows:

$$\gamma_g(x, y, \theta) = \begin{cases} 1 & if\ m - u_\beta[\alpha(x, y) mod 2] \leq \theta \\ 0 & otherwise \end{cases}. \tag{2}$$

That is, the $\gamma_g$ operator returns 1 if the input vectors differentiates at most in θ bits, and returns zero otherwise.

The Gamma classifier uses the modified Johnson-Möbius code [1] to codify numeric instances, due to the generalized gamma similarity operator receives as input two binary vectors. In the following, we explain the parameters considered in the Gamma classifier.

w.- Is the vector of feature weights, which indicates the relative importance of each variable for the classification process.

initial θ.- Denotes the initial value of θ (typically zero). It indicates the maximum allowed difference between two patterns for the generalized similarity operator.

ρ.- Is the stopping parameter, referred as the maximum value allowed to θ, that permits to continue the search for a unique maximum. When ρ=θ, the Gamma classifier stops the iterations and will assign an arbitrary label. In reference [1] are proposed suggested values for this parameter.

$\rho_0$.- Is the pause parameter. In the pause, the Gamma classifier performs an evaluation of the pattern to classify, to determine or not its pertinence to the unknown class. In reference [1] are proposed suggested values for this parameter.

d.- Is the variable that is evaluated to decide if the pattern to classify belongs to the unknown class, or if it belongs to any of the known classes.

u.- Is the comparison threshold to determine if the pattern to classify belongs to the unknown class, or if it belongs to any of the known classes. In reference [1] are proposed suggested values for this parameter.

The steps of the functioning of Gamma classifier are shown in Figure 2.

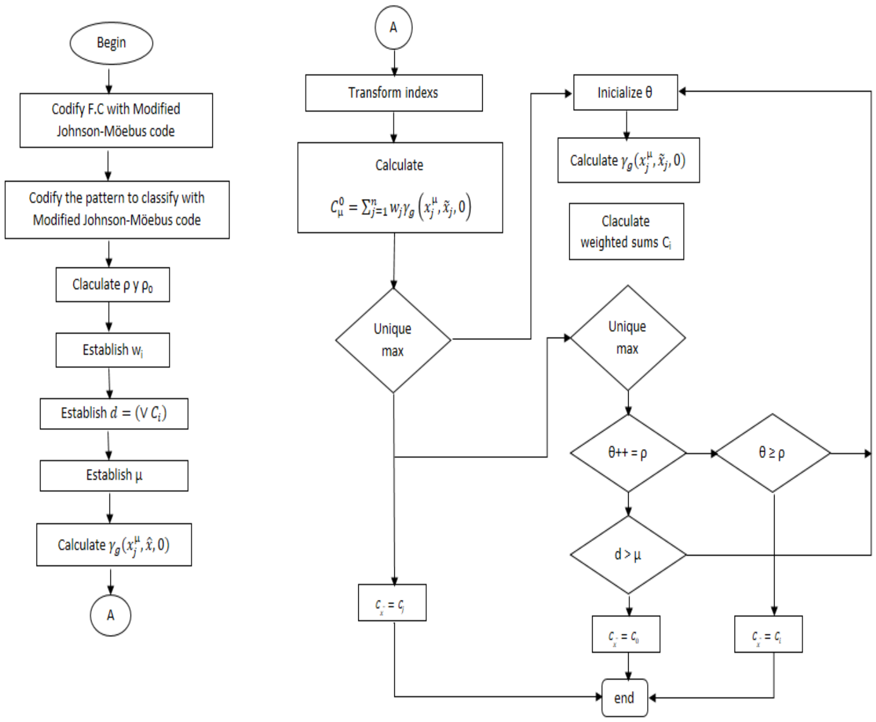



**Fig. 2.** Schema of the classification process with the Gamma classifier.

To automatically determine the values of the w vector of features weights, Ramírez et al. [17] proposed the use of Differential Evolution metaheuristic. They use a real-valued codification strategy and classifier accuracy over the training set as heuristic evaluation function, to evolve the features weights vector.

According to the classification strategy of the Gamma classifier, Antón-Vargas et al. [15] proposed a similarity function named GBS to compare pairs of instances, regarding the θ parameter.

The Gamma based similarity (GBS) uses the generalized gamma operator, but it considers the standard deviation of the feature instead of the θ parameter. Let be X and Y to instances, the Gamma based similarity between them is computed as [15]:

$$GBS(X,Y) = \sum_{i=1}^{p} \gamma_g \left(x_i, y_i, \sigma_i\right), \tag{3}$$

where p is the amount of features describing the instances, $\sigma_i$ is the standard deviation of the i-th feature, and $x_i$ and $y_i$ are the binary vectors associated with the i-th feature in instances X and Y, respectively.

## 3 Experimental Results

We select some of the most representative instance selection algorithms and perform the test over six databases from the Machine Learning repository of the University of California at Irvine [18]. Table 1 shows the characteristics of the selected databases.

**Table 1.** Databases used in the experiments.

| Databases | Objects | Attributes | Classes |
|---|---|---|---|
| balance-scale | 625 | 4 | 3 |
| ecoli | 336 | 7 | 8 |
| heart-statlog | 270 | 13 | 2 |
| ionosphere | 351 | 34 | 2 |
| iris | 150 | 4 | 3 |
| vehicle | 846 | 18 | 4 |

We selected error-based editing methods due to their ability of smoothing decision boundaries and to improve classifier accuracy. The selected methods are the Edited Nearest Neighbor (ENN) proposed by Wilson [8], the Gabriel Graph Editing method (GGE) proposed by Toussaint [9] and the MSEditB method, proposed by García-Borroto et al. [10].

For the application of the mentioned instance selection algorithms, we used the Gamma Based Similarity (GBS) function proposed in [15].

We also used the Differential Evolution approach to compute features weights for the Gamma classifier, as proposed in [17].

All algorithms were implemented in C# language, and the experiments were carried out in a laptop with 3.0GB of RAM and Intel Core i5 processor with 2.67HZ. We cannot evaluate the computational time of the algorithms, because the computer was not exclusively dedicated to the execution of the experiments.

To compare the performance of the instance selection algorithms, it was used the classifier accuracy. The classifier accuracy is measure as the ratio of correctly classified instances. It was also computed the Instance retention ratio (IRR) for every algorithm, in order to determine the amount of selected instances. Table 2 and 4 show the results according to classifier accuracy and instance retention ratio, respectively. Best results are highlighted in bold.

In table 2, we show the accuracy of the weighted Gamma classifier without selecting instances (Gamma) and the accuracy of the weighted Gamma classifier trained using the instances selected by ENN, GGE and MSEditB, respectively.

**Table 2.** Accuracy of the weighted gamma classifier before and after the selection of instances.

| Databases | Gamma | Instances selected by | | |
|---|---|---|---|---|
| | | ENN | GGE | MSEditB |
| balance-scale | 0.760 | 0.685 | 0.770 | **0.811** |
| ecoli | 0.486 | **0.504** | 0.447 | 0.439 |
| heart-statlog | 0.837 | 0.837 | 0.833 | **0.844** |
| ionosphere | **0.761** | 0.641 | 0.359 | 0.000* |
| iris | **0.933** | **0.933** | **0.933** | 0.927 |
| vehicle | **0.584** | 0.579 | 0.551 | 0.564 |

* the MSEditB method deletes all the instances.

As shown, the instance selection algorithms were able to improve the Gamma classifier accuracy in two of the compared databases, and to obtain the same accuracy with fewer instances in one database. Still, for the ionosphere and vehicle datasets, no improvement were obtained.

In addition, it is important to mention that for the ionosphere dataset, the MSEditB algorithm delete all the instances, considering that the entire dataset was mislabeled or noisy.

However, to determine the existence or not of significant differences in algorithm´s performance it was used the Wilcoxon test [19]. It was set as null hypothesis no difference in performance between the gamma classifier without instance selection (Gamma) and the gamma classifier with instance selection algorithms, and as alternative hypothesis that latter had better performance. It was set a significant value of 0.05, for a 95% of confidence. Table 3 summarizes the results of the Wilcoxon test, according to classifier accuracy.

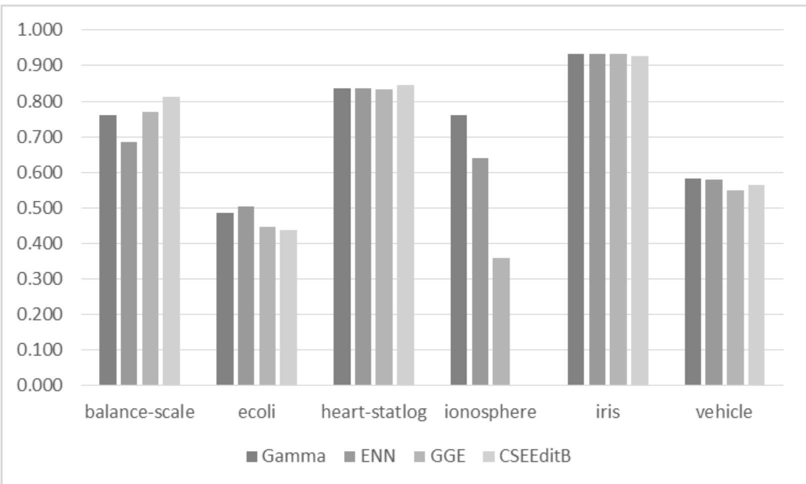



**Fig. 3.** Accuracy of the Gamma classifier using selected instances.

**Table 3.** Wilcoxon test comparing classifier accuracy.

| Original Gamma vs | ENN | GGE | MSEditB |
|---|---|---|---|
| wins-looses-ties | 3-1-2 | 4-1-1 | 4-2-0 |
| probability | 0.273 | 0.138 | 0.463 |

The Wilcoxon test obtains probability values greater than the significance level, and thus, we do not reject the null hypothesis. These results confirm the instance selection approach is able to preserve classifier accuracy, using a small amount of instances.

**Table 4.** Instance retention ratio obtained by the selection of instances.

| Databases | ENN | GGE | MSEEditB |
|---|---|---|---|
| balance-scale | 0.912 | 0.847 | **0.777** |
| ecoli | 0.844 | 0.888 | **0.675** |
| heart-statlog | 0.887 | 0.772 | **0.763** |
| ionosphere | 0.641 | **0.359** | 0.000* |
| iris | 0.955 | 0.973 | **0.934** |
| vehicle | 0.838 | 0.851 | **0.675** |

*the MSEditB algorithm deletes all the instances.

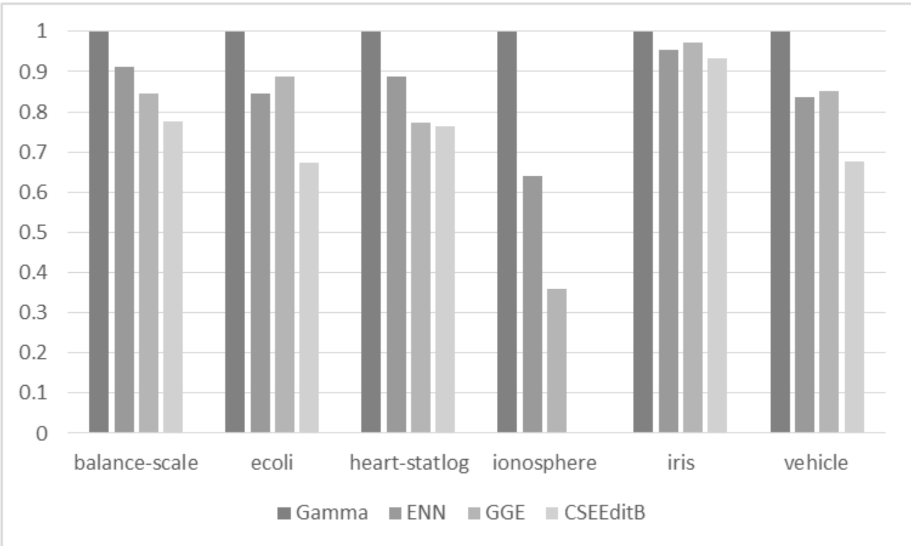



**Fig. 4.** Instance retention ratio obtained by the algorithms.

As shown in table 4, all instance selection methods are able to delete among the 60% and 4% of the data, without decreasing the classifier accuracy. These results confirm the proposed approach is able to obtain an adequate training set for the Gamma classifier, without losing representative objects.

**Table 5.** Wilcoxon test comparing instance retention ratio.

| Original Gamma vs | ENN | GGE | MSEditB |
|---|---|---|---|
| wins-looses-ties | 0-6-0 | 0-6-0 | 0-6-0 |
| probability | **0.028** | **0.028** | **0.027** |

According to instance retention ratio, the Wilcoxon test rejects the null hypothesis in all cases. That is, the number of selected objects using ENN, GGE and MSEditB with the proposed gamma based similarity function, was significantly lower than the original amount of instances in the training set.

The experimental results carried out in our research show that using automatic weigh computation, as well as a similarity function based on the Gamma operator, allows to successfully apply instance selection algorithms to improve the performance of the Gamma associative classifier. The statistical tests show that instance selection algorithms are able to maintain classifier accuracy, and also to reduce the cardinality of the training sets, diminishing the computational cost of the Gamma classifier.

## 4 Conclusions

In this paper is explored the impact of instance selection algorithms in conjunction with automatic feature weight in the performance of the Gamma associative classifier. The numerical experiments were carried out over well- known repository data. The obtained results confirm the hypothesis that instance selection algorithms may decrease the computational cost of the Gamma classifier, while preserve the classifier accuracy. In addition, the study conclude that automatic feature weighting procedures may increase the performance of the Gamma classifier.

## References


1. López Yáñez, I.: Clasificador automático de alto desempeño. MS dissertation, Instituto Politécnico Nacional-Centro de Investigación en Computación (2007)
2. López-Yáñez, I., Sheremetov, L., Yáñez-Márquez, C.: A novel associative model for time series data mining. Pattern recognition Letters, 41, pp. 23–33 (2014)
3. López-Martin, C., López-Yánez, I., Yánez-Márquez, C.: Application of Gamma Classifier to Development Effort Prediction of Software Projects. Appl. Math., 6, pp. 411–418 (2012)
4. Lopez-Yanez, I., Argüelles-Cruz, A.J., Camacho-Nieto, O., Yanez-Marquez, C.: Pollutants time-series prediction using the Gamma classifier. International Journal of Computational Intelligence Systems, 4, 680–711 (2012)
5. Yánez-Márquez, C., López-Yánez, I. Morales, G.D.: Analysis and prediction of air quality data with the gamma classifier. Progress in Pattern Recognition, Image Analysis and Applications, pp. 651–658 (2008)
6. García, S., Derrac, J., Cano, J.R., Herrera, F.: Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34, pp. 417–435 (2012)

7. Triguero, I., Derrac, J., Garcia, S., Herrera, F.: A taxonomy and experimental study on prototype generation for nearest neighbor classification. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions, 42, pp. 86–100 (2012)
8. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data, IEEE Transactions on Systems, Man and Cybernetics, SMC-2, pp. 408–421 (1972)
9. Toussaint, G.T.: Proximity Graphs for Nearest Neighbor Decision Rules: Recent Progress. 34 Symposium on Computing and Statistics INTERFACE-2002, Montreal, 1–20, Canada (2002)
10. García-Borroto, M., Villuendas-Rey, Y., Carrasco-Ochoa, J.A., Martinez-Trinidad, J.F.: Using Maximum Similarity Graphs to edit nearest neighbor classifiers. Lecture Notes on Computer Science, 5856, 489–496 (2009)
11. Cover, T.M., Hart, P.E.: Nearest Neighbor pattern classification. IEEE Transactions on Information Theory, 13, 21–27 (1967)
12. Ishibuchi, H., Nakashima, T., Nii, M.: Learning of neural networks with GA-based instance selection. IFSA World Congress and 20th NAFIPS International Conference, 4, pp. 2102–2107 (2001)
13. Medina-Pérez, M.A., García-Borroto, M., Villuendas-Rey, Y., Ruiz-Shulcloper, J.: Selecting objects for ALVOT. Progress in Pattern Recognition, Image Analysis and Applications, pp. 606–613 (2006)
14. Medina-Pérez, M.A., García-Borroto, M., Ruiz-Shulcloper, J.: Object selection based on subclass error correcting for ALVOT. Progress in Pattern Recognition, Image Analysis and Applications, pp. 496–505 (2007)
15. Antón-Vargas, J.A., Villuendas-Rey, Y., López-Yáñez, I.: Gamma classifier based instance selection. Research in Computer Science (Accepted paper) (2015)
16. Yáñez-Márquez, C., Díaz, L.: Memorias Asociativas basadas en relaciones de orden y operaciones binarias. Computación y Sistemas, Vol. 6, pp. 300–311 (2003)
17. Ramirez, A., Lopez, I., Villuendas, Y., Yanez, C.: Evolutive Improvement of Parameters in an Associative Classifier. Latin America Transactions, IEEE (Revista IEEE America Latina), Vol. 13, pp. 1550–1555 (2015)
18. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
19. Demsar, J.: Statistical comparison of classifiers over multiple datasets. The Journal of Machine Learning Research, Vol. 7, pp. 1–30 (2006)
20. Caballero, Y., Bello, R., Salgado, Y., García, M.M.: A method to edit training set based on rough sets. International Journal of Computational Intelligence Research, Vol. 3, pp. 219–229 (2007)

# Localización automática de placas de automóviles

Nery Daniel Tovar-Espinoza, Juan Carlos Rodríguez-Sánchez,
Victor Manuel Landassuri-Moreno, Saúl Lazcano Salas,
José Martín Flores Albino

Centro Universitario UAEM Valle de México,
Universidad Autónoma del Estado de México,
México

ndtovare@gmail.com, {jcrsylatp, jose_martin_70}@yahoo.com,
{vmlandassurim, slazcanos}@uaemex.mx

**Resumen.** El problema de localización de placas de automóvil ha sido ampliamente estudiado con anterioridad, dando resultados favorables. Sin embargo, hasta el momento de esta investigación no se han encontrado conjuntos de prueba extensos para validar los algoritmos. Así, en éste trabajo se presenta la primera etapa de un Sistema de Reconocimiento Automático de Placas de Automóviles, el cual es enfocado a encontrar únicamente la localización de la placa. Tres algoritmos similares fueron desarrollados en cuanto a las distintas técnicas que se implementan, como el erosionado, el dilatado o el filtro de Sobel, siendo probados en un conjunto de imágenes con distintas condiciones como iluminación, problemas ambientales, tamaño de imagen en píxeles, entre otros. Los resultados obtenidos muestran que el mejor de los algoritmos propuestos permite localizar más placas con más efectividad y con más robustez a las condiciones de la imagen en comparación de los otros dos, en donde uno de ellos no detecta la placa en la mayoría de las imágenes usadas.

**Palabras clave:** segmentación de imágenes, reconocimiento de placas, procesamiento de imágenes, filtro Sobel.

# Automatic Localization of Automobile Licence Plate

**Abstract**. License plate localization for vehicle has been widely studied before, giving favorable results. However, as far as this research has advanced, there is no indication of extensive benchmarks to test the algorithms. Thus, in this work is presented the first stage of a vehicle's license plate Automatic Recognition System, which is focused to only find the license plate. Three similar algorithms were developed using techniques as erosion, dilation, and Sobel filter, being tested in a benchmark with images from different illumination conditions, ambient problems, image size in pixels, among others. The results obtained shows that the best proposed algorithm allows successfully locate more license plates, supporting different image conditions, in comparison with the other algorithms, were one of them cannot detect the license plate in the majority of the images tested.

**Keywords:** Image segmentation, licence plate recognition, image processing, Sobel filter.

## 1. Introducción

El procesamiento de imágenes tiene como objetivo principal el análisis de los píxeles y las matrices que las componen para su manipulación y recomposición, con la finalidad de mejorar la imagen y de segmentar e interpretar los objetos de interés. Debido a la amplia gama de aplicaciones que requieren y utilizan el tratamiento digital de imágenes, su relevancia ha aumentado en varias áreas del conocimiento en los últimos años, por ejemplo, en medicina para la detección de tumores o el reconocimiento y extracción de células [6], así como en el reconocimiento de rostros [10], de huellas dactilares [1, 13], reconocimiento de patrones [14] y el reconocimiento de placas de automóviles [2, 3, 4, 5, 11, 15]. Algunos autores [8, 11] dividen el procesamiento de imágenes enfocado al reconocimiento de placas en las siguientes etapas [11]: Pre-procesamiento, Segmentación, Detección y Análisis. Cabe destacar que este trabajo se enfocará a la primera de ellas. Xiaofeng Zhang et. al. [15], exponen un algoritmo para el reconocimiento de placas de automóvil, en donde el pre-procesamiento consiste en eliminar el ruido de la imagen, con el objetivo de resaltar la información más importante, aunque cabe mencionar que se enfocan solamente en el reconocimiento de los caracteres de la placa. Kumar Parasuraman y P. Vasantha Kumar [11] presentan un método que realiza la extracción de la placa del automóvil y la segmentación de los caracteres, sin embargo, únicamente se presenta una imagen de un automóvil de frente para probar las técnicas, donde usan los algoritmos utilizados: de bordes horizontales y verticales [4]. Kaushik Deb, et. al. [4] presentan un sistema para la extracción de placas mediante la búsqueda de líneas verticales y horizontales, utilizando una técnica de segmentación llamada “Ventanas Corredizas Concéntricas”, sus siglas en inglés “SCW” (Sliding Concentric Windows); muestra resultados muy exactos, pero igualmente con automóviles que se encuentran de frente. En su contraparte, H. Erdinc Kocer y K. Kursat Cevik [5] presentan un método de reconocimiento de placas usando la detección de bordes, la eliminación de ruido y a partir del gradiente detectar la placa, mostrando resultados exactos, con la limitante de que usan una sola imagen de prueba.

Esto parece indicar, que los algoritmos desarrollados no funcionen adecuadamente en un conjunto de imágenes, debido a todas las posibles combinaciones que éstas pueden presentar. Así, éste artículo presenta una propuesta de localización de placas de automóviles, con tres algoritmos desarrollados usando filtros para la detección de bordes y de operadores morfológicos, todos éstos enfocados al pre-procesamiento de la imagen. Los cuales son evaluados en un conjunto de 193 imágenes con distintos factores de ruido, como lo es la iluminación (de día y de noche), problemas ambientales (lluvia), tamaño de imagen en píxeles (tomadas con diferentes cámaras, así como fotos tomadas de Internet), entre otros. El resto de este trabajo de investigación está organizado en cinco secciones. La segunda sección presenta las técnicas de pre-procesamiento de imágenes más utilizadas aquí, posteriormente se muestran los tres algoritmos implementados en la sección 3 así como una prueba preliminar que sirvió para ajustar diversos parámetros de los algoritmos. Los resultados son comentados en la sección 4 y las conclusiones y trabajos futuros en la sección 5.

## 2. Técnicas para el pre-procesamiento de imágenes

A continuación se describirán las técnicas utilizadas para lograr separar la placa del resto de la imagen, las cuales son combinadas más adelante en los tres algoritmos implementados.

El método de Sobel [9] es un operador que se utiliza en la técnica de detección de bordes en una imagen en escala de grises, ésta funciona a partir de dos máscaras, una horizontal en función de “x” y otra vertical en función de “y” (figura 1). Estas dos mascaras al combinarse pueden encontrar la magnitud absoluta de la pendiente de cada punto (píxel), ecuación 1, y la orientación del gradiente (ecuación 2).

| | | |
|---|---|---|
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

| | | |
|---|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

**Fig. 1.** Gx - Máscara horizontal (matriz de la derecha) y Gy - vertical (izquierda).

Para más detalles acerca de éste y otros operadores similares para detectar bordes, se recomienda consultar [5].

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (1)$$

$$\vartheta = \arctan(G_y / G_x) \quad (2)$$

Otras herramientas para el procesamiento de imágenes son los denominados operadores morfológicos [7], los cuales incluyen la erosión y la dilatación, que se aplican en imágenes binarias (formadas por 1’s y 0’s), donde erosionado (ecuación 3) consiste en remover píxeles de las fronteras de los objetos de una imagen, como se muestra en la figura 2. El dilatado (ecuación 4) hace totalmente lo opuesto al erosionado, en lugar de remover píxeles, éste los agrega (figura 3).

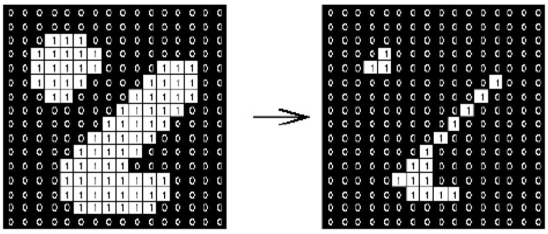


**Fig. 2.** Erosionado.

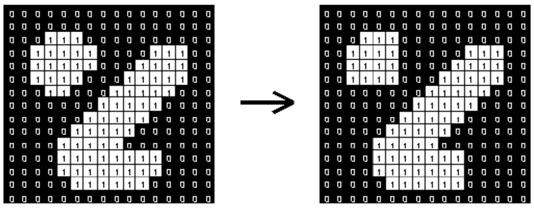


**Fig. 3.** Dilatado.

$$\alpha \ominus \beta = \bigcap_{b \in \beta} \alpha_{-b} = \{x : \beta_x \subset \alpha\} \tag{3}$$

$$\alpha \oplus \beta = \bigcup_{b \in \beta} \alpha_b = \left\{x : \check{\beta}_x \cap \alpha \neq 0\right\} \tag{4}$$

## 3. Métodos y pruebas preliminares

En una primera etapa se realizaron pruebas preliminares, utilizando operaciones morfológicas implementadas en un primer algoritmo (Algoritmo A) para llevar a cabo la segmentación de placas de automóviles utilizando MATLAB. Las pruebas se realizaron en placas que tienen impresos caracteres rojos, o donde la placa es de color rojo, separando los canales de la imagen en RGB de forma similar a Cañadas Betancourt y Haro Figueroa en [2], quienes trabajan con placas anaranjadas. Los resultados se muestran a continuación:


**Fig. 4.** Resultado previo con Algoritmo A.

Éste algoritmo posee la limitante de que sólo funciona para segmentar el canal Rojo de las imágenes y trabaja con imágenes con poco ruido alrededor de la placa, donde la imagen fue previamente recortada como el realizado por Cárdenas Hidalgo et al. [3]. Éste fue modificado para segmentar otras fotografías con mayor cantidad de ruido y que además incluyen placas que tienen caracteres en color negro. Primero se comenzó con el redimensionamiento, donde después de una serie de experimentos se obtuvo que 493x700 pixeles eran adecuados para obtener mejores resultados al encontrar la placa. Posteriormente se convierte a formato de 8 bits, el cual es un formato más compacto que RGB, reduciendo el tiempo de procesamiento. La imagen de 8 bits se separa en los canales Rojo, Verde y Azul guardando cada valor de RGB en una variable para cada caso.

Es importante señalar que se puede utilizar cualquiera de los tres canales para convertir a formato binario y posteriormente después se pueda eliminar los bordes de la imagen e iniciar con la eliminación el fondo. Esto reduce considerablemente el ruido, pero aún es necesario quitar ciertas regiones que están formadas por diferentes grupos de pixeles, para lo cual se eliminaron regiones menores a 2167 pixeles (mejor valor obtenido al probar diferentes combinaciones). El algoritmo A fue probado en 193 imágenes. Con autos cuya vista es frontal, o con una ligera inclinación se obtuvo una efectividad del 38%. Pruebas con imágenes aún más inclinadas indicaron 12% de efectividad, donde se descubrieron mayores errores en imágenes alejadas o con iluminación muy intensa. Algunos resultados se muestran en la figura 5.

En la segunda parte del trabajo, se proponen tres métodos distintos, nombrados como Algoritmo B, C y D respectivamente. Los primeros dos consisten en encontrar la placa del automóvil sin importar el ángulo donde se haya tomado la foto, su única diferencia es que el segundo es recursivo, como se muestra en las figuras 6 y 7. El tercero se enfoca en encontrar las placas de automóviles que están solamente de frente (figura 8). Cabe destacar que los 4 algoritmos aquí presentados fueron evaluados con el mismo conjunto de imágenes de automóviles, el cual está dividido en tres categorías. En la Categoría “1” se encuentran 119 imágenes que van de 180x110 píxeles a 971x717 píxeles. En la Categoría “2” están 37 imágenes entre las dimensiones 1024x432 píxeles y 1920x1440 píxeles y la Categoría “3” tiene 37 imágenes entre las dimensiones 2048x1536 píxeles a 3264x2448 píxeles. Fueron separadas por dimensiones, asumiendo que a mayor cantidad de pixeles, mas información y detalle en la imagen, lo cual puede ayudar o no a encontrar la placa del automóvil.


**Fig. 5.** Resultados parciales de la primera etapa con Algoritmo A.

## 4. Resultados

Los resultados se dividieron en 3 partes: las imágenes que detectó solamente la placa (sin mayor ruido alrededor), las imágenes que detecta la placa con ruido y las imágenes que no detectan la placa. En la tabla 1, 2 y 3 se muestran los resultados de la categorías respectivas con los algoritmos B, C y D.

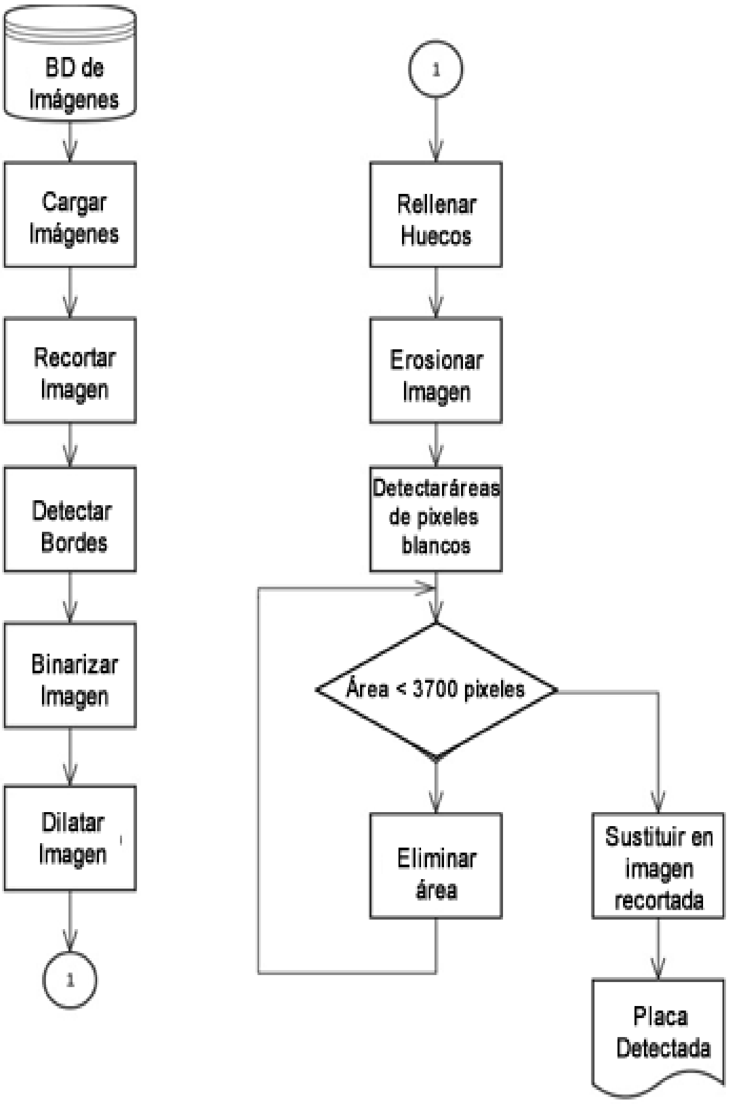



**Fig. 6.** Algoritmo B.

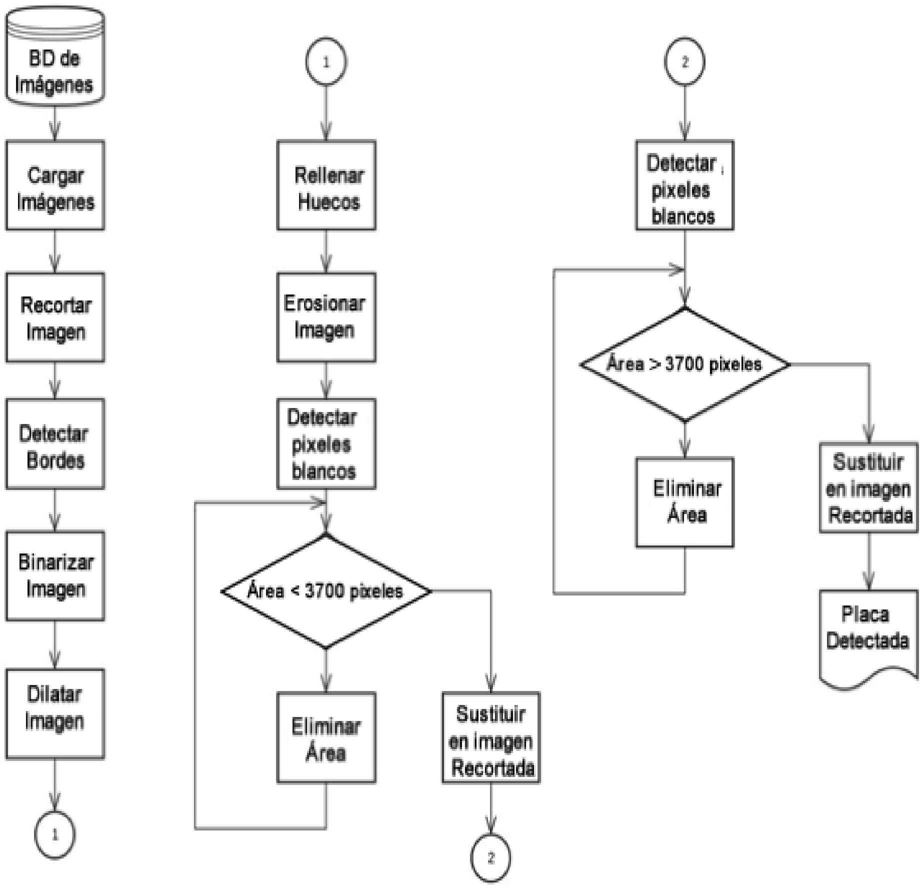



**Fig. 7.** Algoritmo C.

**Tabla 1.** Resultados de la Categoría 1 de imágenes con los tres algoritmos.

| | Detecta la placa | Detecta la placa con ruido | No detecta la placa |
|---|---|---|---|
| Alg. B | 9 | 41 | 69 |
| Alg. C | 10 | 40 | 69 |
| Alg. D | 4 | 73 | 42 |

**Tabla 2.** Resultados de la Categoría 2 en los tres algoritmos.

| | Detecta la placa | Detecta la placa con ruido | No detecta la placa |
|---|---|---|---|
| Alg. B | 2 | 25 | 10 |
| Alg. C | 2 | 10 | 25 |
| Alg. D | 1 | 17 | 19 |

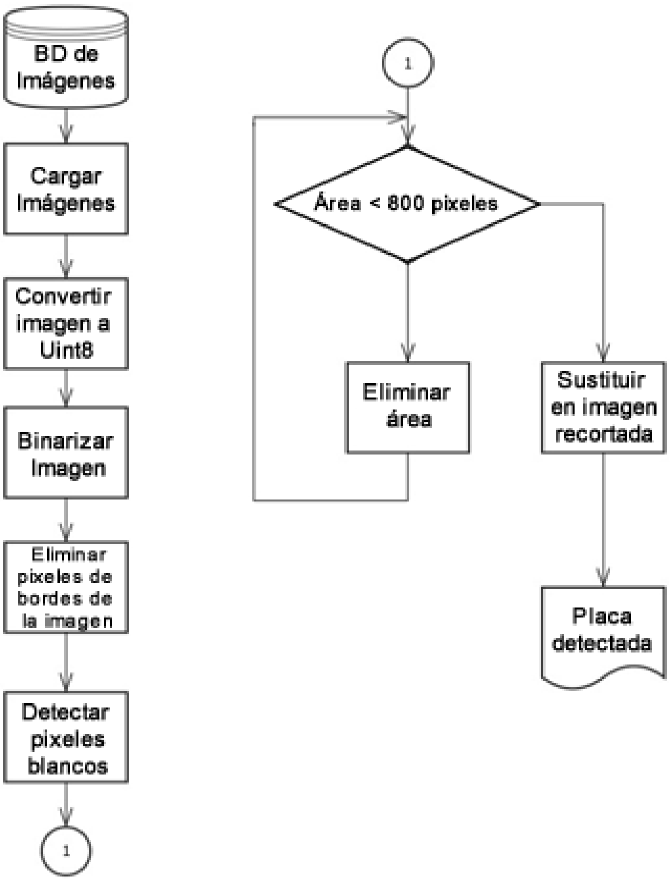



**Fig. 8.** Algoritmo D.

**Tabla 3.** Resultados de la Categoría 3 en los tres algoritmos.

| | Detecta la placa | Detecta la placa con ruido | No detecta la placa |
|---|---|---|---|
| Alg. B | 2 | 24 | 11 |
| Alg. C | 0 | 10 | 27 |
| Alg. D | 1 | 21 | 15 |

En la tabla 1 se puede notar que el algoritmo D detecta la placa con ruido en la mayoría de las imágenes, pero cabe mencionar que ésta categoría abarca un rango de píxeles muy pequeño, pero en la tabla 2 y 3 los resultados son inferiores a los del algoritmo B, que tiene mayor eficacia en la detección de la placa con ruido. El algoritmo C al igual que el algoritmo D sólo fue algo eficiente con las imágenes de la Categoría 1. Se debe especificar, que identificar la placa con ruido, significa que la placa permanece al final, como más regiones de la misma, es decir, si "Detecta la placa", se asume que se eliminó todo el ruido/información de alrededor de la placa, Con ruido, permanece la placa y más información, y cuando no se detecta, significa que o hay demasiada información o se eliminó la placa en el procesamiento de la imagen. A continuación se presentan los porcentajes totales de los algoritmos B, C y D del conjunto de imágenes que se utilizó en la tabla 4.

**Tabla 4.** Porcentajes totales de los 3 algoritmos.

| | Detecta la placa | Detecta la placa con ruido | No detecta la placa |
|---|---|---|---|
| Alg. B | 7% | 46% | 47% |
| Alg. C | 6% | 31% | 63% |
| Alg. D | 3% | 58% | 39% |

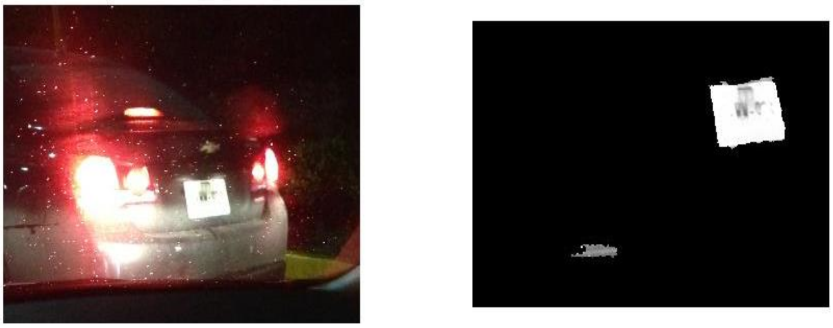


**Fig. 9.** Resultado del Algoritmo B.

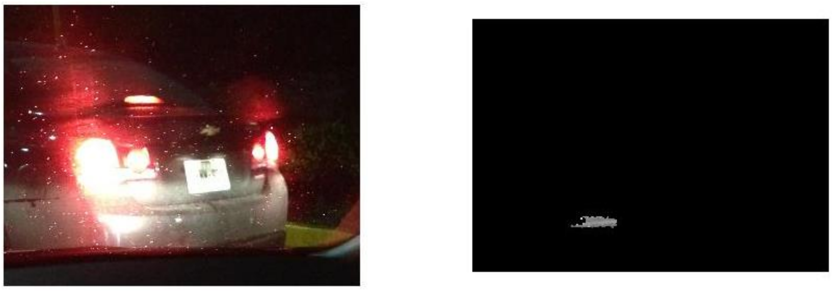


**Fig. 10.** Resultado del Algoritmo C.

**Fig. 11.** Resultado del Algoritmo D.

La tabla 4 muestra que el algoritmo D tiene un mayor porcentaje sobre los otros dos, pero eso es debido a que sólo es eficaz con las imágenes de la categoría 1 que en proporción a las otras dos categorías es más grande en cuanto al número de muestras (imágenes), sin embargo, dadas las especificaciones de las imágenes, el algoritmo B es el más eficiente de los tres.

Las figuras 9, 10 y 11 corresponden a los resultados de aplicar los algoritmos B, C y D respectivamente con una imagen de 3264x2448 píxeles.

Basándonos en los resultados mostrados en las figuras 9, 10 y 11, el algoritmo B es el que presenta una tasa de localización más alta exceptuando por el pequeño ruido que se encuentra en la parte inferior. En el algoritmo C, la placa desaparece, debido a que está implementando recursividad, y al aplicarse, desaparece la placa. El algoritmo D encuentra la placa pero aún tiene mucho ruido, lo cual puede dificultar la extracción de ésta. Con este ejemplo particular se muestra el comportamiento del Algoritmo B, el cual supera a los otros dos aquí mostrados.

## 5. Conclusiones y trabajo a futuro

En este trabajo se presentó la primera etapa de un proyecto de detección y reconocimiento de caracteres de placa de automóvil, donde una prueba preliminar sirvió como base para el desarrollo de otros tres algoritmos para la localización de la placa de automóvil, lo cual entra en la categoría de pre-procesamiento de imágenes. Se encontró que la configuración del Algoritmo B fue la mejor, aunque no tuvo un rango muy alto en el porcentaje de placas localizadas (53%) pero muestra mejores resultados al ser aplicado a imágenes donde las placas del auto se encuentran de frente o en diferentes ángulos, con diferentes niveles de iluminación o con condiciones particulares de la imagen.

A diferencia de otros trabajos encontrados en la literatura, donde únicamente tienen imágenes de frente para probar sus algoritmos, aquí se creó un conjunto de prueba con 193 imágenes clasificadas en tres categorías de acuerdo a su tamaño en pixeles. De ahí se obtuvo que las imágenes con menor cantidad de píxeles tuvieron los rangos más bajos de detección de placa, dada la poca información en la imagen. Cómo trabajo a futuro, se debe de ampliar el tamaño del conjunto de prueba de las imágenes, clasificándolas no solo por tamaño, sino por las condiciones ambientales de cada una. Así como mejorar los algoritmos aquí presentados, para una mejor detección, por ejemplo tomando en cuenta el anti reflejante que tienen las placas. Por último, segmentar la placa y los caracteres particulares de ella para aplicar técnicas de reconocimiento de patrones como Redes Neuronales Artificiales o Maquinas de soporte vectorial, y poder reconocer cada patrón particular.

**Agradecimientos.** Se agradece al laboratorio de Cómputo de Alto rendimiento del Centro Universitario UAEM Valle de México por las facilidades prestadas para la realización de este trabajo.

## Referencias


1. Aguilar, G., Sánchez, G., Toscano, K., Nakano, M., Pérez, H.: Reconocimiento de Huellas Dactilares Usando Características Locales. Revista de la Facultad de ingeniería de la Universidad de Antioquia, 46, 101–109 (2008)
2. Cañadas Betancourt, P.A., Haro Figueroa, G.M.: Prototipo de un sistema de adquisición de imágenes de vehículos, detección y reconocimiento automático de los caracteres de la placa en tiempo real por medio de visión artificial, aplicando al control vehicular. Tesis de Licenciatura, Escuela Politécnica Nacional, Quito (2011)
3. Cárdenas Hidalgo, P., Flores Vargas, J.A., López Zavaleta, J., Martínez Moreno, P.: Diseño de Sistema de Reconocimiento de placas utilizando MATLAB. Tesina de Licenciatura, Instituto Politécnico Nacional, México (2009)
4. Deb, K., Khan, I., Saha, A., Jo, K.: An Efficient Method of Vehicle License Plate Recognition Based on Sliding Concentric Windows and Artificial Neural Network. Procedia Technology, 3, 812–819 (2012)
5. Erdnic Kocer, H., Kursat Cevik, K.: Artificial neural networks based vehicle license plate recognition. Procedia Computer Science, 3, 1033–1037 (2011)
6. Gómez, O., Alvarado, P.: Segmentación de nematodos en imágenes digitales usando redes neuronales artificiales (2001)
7. Jawas, N., Suciati, N.: Image Inpaiting using Erosion and Dilation Operation. International Journal of Advanced Science and Technology, 51, 127–134 (2013)
8. La Serna Palomino, N., Román Concha, U.: Técnicas de Segmentación en Procesamiento Digital de Imágenes. Revista de Ingeniería de Sistemas e Informática, 6 (2), 9–16 (2009)
9. Maini, R., Aggarwal, H.: Study and Comparision of Various Image Edge Detection Techniques. International Journal of Image Processing, 3, 1–11 (2009)
10. Méndez-Vázquez, H., Chang, L., Rizo-Rodríguez, D., Morales-González, A.: Evaluación de la calidad de las imágenes de rostros utilizadas para la identificación de las personas. Computación y Sistemas, 16 (2), 147–165 (2012)
11. Parasuraman, K., Kumar, P.V., Blesa, P.: An Efficient Method for Indian Vehicle License Plate Extraction and Character Segmentation. In: IEEE International Conference on Computational Intelligence and Computing Research (2010)
12. Ramírez Q., J.A., Chacón M., M.I.: Redes neuronales artificiales para el procesamiento de imágenes, una revisión de la última década. Revista de Ingeniería Eléctrica, Electrónica y Computación 9 (1), 7–16 (2011)
13. Ruíz E., M.E., Morales S., M., Hernández M., Y.: Una estrategia de segmentación de imágenes digitales de huellas dactilares. Revista de Ingeniería Eléctrica, Electrónica y Computación, 9 (1), 1–6 (2011)
14. Tse, C., O., W.: Sistema de Reconocimiento de Patrones Visuales Basado en Técnicas de Procesamiento de Imágenes y Redes Neuronales. Telematique, Revista Electrónica de Estudios Telemáticos, 3 (2) 75–99 (2004)
15. Zhang, X., Fengchang, X., Su, Y.: Research on the License Plate Recognition based on MATLAB. Procedia Engineering, 15, 1330–1334 (2011)

# Metodología de clasificación de señales electromiográficas

J.R. Caro Vásquez[1], J.I. Chairez Oria[2], C. Yáñez Márquez[1]

[1] Instituto Politécnico Nacional, Centro de Investigación en Computación, México

[2] Instituto Politécnico Nacional, Unidad profesional multidisciplinaria de Biotecnología, México

**Resumen.** In recent years, there has been extensive use electromyographic signals (EMG´s) as the primary inputs on algorithms used in control of smart prosthesis; at the same time, the use of classification algorithms to identify patterns in upper limb movements is rising . Some of the main challenges that exist in classification are; EMG's feature extraction and high complexity of algorithms that have been used for that application. This paper present a proposal solution to the problematic mentioned before, a pattern classification method using energy as the only feature and an algorithm capable of running in parallel whose principal function is make the classifying task faster than another conventional algorithm.

**Palabras clave:** electromiografía, EMGs, patrones, características, prótesis, clasificadores dedicados, energía.

## 1. Introducción

La electromiografía superficial ha sido el área de investigación principal para el control de prótesis de miembro superior por más de seis décadas. [1], [2].

Con fin de mejorar de mejorar el control de prótesis mioeléctricas, en las últimas dos décadas se ha investigado las metodologías de extracción de características haciendo uso de electrodos superficiales. Estos esfuerzos han estado enfocados principalmente en el área de reconocimiento de patrones, específicamente atendiendo la tarea de clasificación. En la cual se asume que las EMGs tienen patrones que pueden ser identificados por un algoritmo de cómputo inteligente. [1, 3].

Hoy en día, las EMGs son las principales señales utilizadas para el control en prótesis mioeléctricas de miembro superior. En el estado del arte, varios métodos distintos de clasificación de patrones y diferentes características extraídas de la señal se han utilizado con anterioridad. Por citar solo algunos ejemplos, podemos mencionar que Xing et al [4] hacen uso de máquinas de soporte vectorial (SVM) para la clasificación de las EMGs, Yousefi et al [5] mencionan sus aplicaciones con Redes neuronales artificiales (ANN) con el mismo propósito. Por otro lado, la extracción de características de señales cuya función es extraer información de las EMGs es un problema abierto ya que existe un sinfín de posibles técnicas de procesamiento de señales que pueden ser usadas para esta aplicación. Este hecho queda evidenciado en los trabajos de Phinyomark et al [6] quienes enfatizan el uso de área bajo la curva de EMGs (IEMG), la media absoluta (MAV), la varianza (VAR), el valor efectivo de la señal (RMS), y doce características más extraídas década una de las señales, en cambio Gokgoz  et al [7] prefieren utilizar únicamente la transformada de ondícula (DWT).

Con objeto de mejorar el rendimiento de un algoritmo y tener un menor tiempo de procesamiento, se hacen uso de técnicas de procesamientos, entre ellas destaca principalmente las técnicas de selección de características las principales técnicas utilizadas son análisis de componentes principales (PCA), y análisis de componentes mutuos (MCA) [8].

En el presente trabajo se plantea y describe una metodología de clasificación de patrones, que se enfoca en la extracción de una característica única, para este caso se hace el uso de la energía de la EMGs. Se presenta, además, una solución para modificar algoritmos de clasificación de patrones capaz de realizar cómputos en paralelo, lo cual permite clasificar más rápidamente que algoritmos convencionales.

El resto del documento se encuentra organizado de la siguiente forma: en la Sección 2, se presenta la metodología y las características del experimento, en la Sección 3, se presentan los resultados obtenidos y una pequeña discusión sobre ellos, finalmente en la sección 4 se presentan algunas conclusiones.

## 2. Metodología

En esta sección se presenta de manera detallada la naturaleza de los experimentos y la descripción del banco de datos, se muestra también el pre procesamiento de la EMGs, los modelos de clasificación de señales y la arquitectura del modelo final.

Banco de datos

El banco de datos que se utiliza en la presente investigación lo realizaron los investigadores Rami N. Khusaba, Sarath Kodagoda, Dikai Liu y Gamini Dissanayake  miembros del grupo de investigación de la universidad tecnología de Sydney y está disponible para la comunidad científica en (www.rami-khushaba.com/electromyogram-emg-repository.html).

El banco está conformado por señales de EMG obtenidas de todo alrededor del antebrazo por 8 electrodos superficiales (DE 2.x series EMG sensors) y es procesado por (Bagnoli desktop EMG system from Delsys Inc). La señal fue amplificada con una ganancia de 1000,recuperada con un convertidor analógico-digital de 12 bits (National Instruments, BNC-2090) a una frecuencia de muestreo de 4000 Hz, y filtrada con un pasabanda de frecuencias entre 20Hz – 450 Hz y un filtro notch de 50 Hz para eliminar la frecuencia de la red eléctrica.

Los datos se conforman de pruebas de 5s. tomadas a 8 personas diferentes , 6 hombres y 2 mujeres con miembros totalmente sanos (sin ningún desorden muscular).

Durante este experimento fueron recolectados 15 diferentes posiciones del miembro i.e., Thumb (T), Index (I), Middle (M), Ring (R), Little (L) los combinados Thumb-Index (T-I), Thumb-Middle (T-M), Thumb-Ring (T-R), Thumb-Little (TL), Index-Middle (I-M), Middle-Ring (M-R), Ring-Little (RL), Index-Middle-Ring (I-M-R), Middle-Ring-Little (M-RL), y finalmente la clase Hand close (HC), se tomó prueba de cada una de ellas 6 veces a cada persona por un periodo de tiempo de 5. El autor recomienda utilizar 4 de ellas como entrenamiento y 2 de prueba para validación.

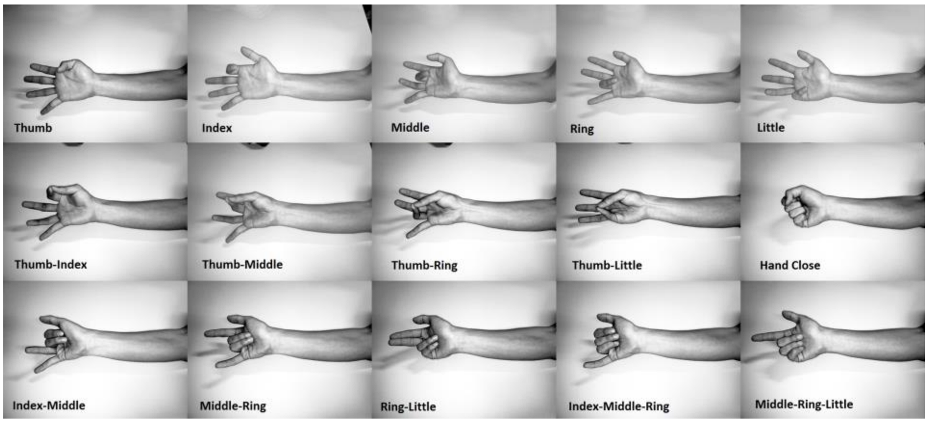



**Fig. 1.** Movimientos de clasificación.

La figura 1 fue recuperada del artículo original de Khushaba et al donde se introduce el banco de datos [8].

### 2.1. Extracción de características

La metodología propuesta en este artículo consiste en tomar una única característica de la señal, tal que represente información suficiente para representar la señal en un rasgo numérico real permitiéndonos omitir el pre procesamiento de selección de características.

En este caso en particular se toma la única característica de energía de la señal en tiempo discreto por un periodo de tiempo de 0.25s. generando un nuevo banco de datos cuyos patrones están conformados de la siguiente forma [9]:

$$x^i = \sum_{1000i}^{1000i+1000} n^2$$

Siendo n la muestra que de señal recuperada por el convertidor analógico- digital.

### 2.2. Clasificación

Se hace uso del software "weka" como apoyo a la clasificación con uso de los modelos más utilizados en el estado del arte [10]. Con ayuda del software se clasifica el banco de datos completo para una sola persona con características de energía para los clasificadores ANN [11, 12, 13, 14]. K-nearest neighbor (KNN) [11, 15]. Clasificadores bayesianos [11, 16]. SVM [17], C4.5 [11].

A partir de este punto el experimento se enfoca principalmente en el clasificador que presente un mayor rendimiento.

### 2.3. Algoritmo paralelo

Una vez que conocemos el algoritmo de mayor rendimiento se hace uso de técnicas de ensamble de clasificadores para hacer trabajar un algoritmo de clasificación en forma paralela [18]. En este caso se usarán específicamente los clasificadores dedicados o "one-class clasifier ensemble" [19], cuya función es generar un clasificador en paralelo agregando tantos clasificadores como clases haya y dedicando a cada uno de ellos a enfocarse en los patrones de su clase correspondiente.

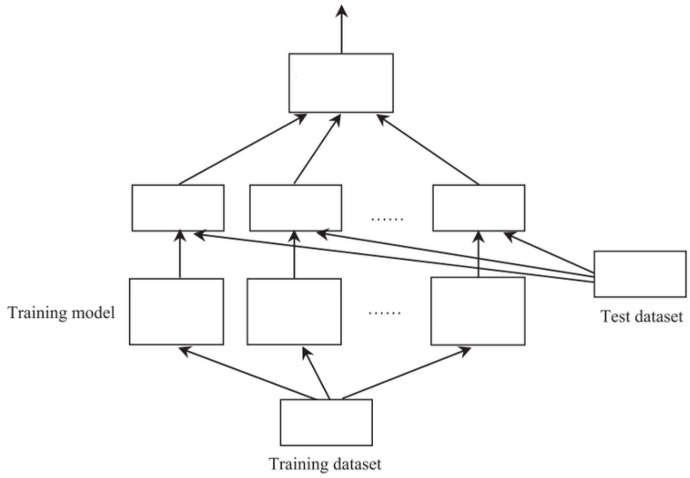



**Fig. 2.** Esquema de clasificadores dedicados.

La figura anterior muestra el esquema de clasificadores dedicados aplicado a SVM utilizado por Xiang-Yang Wang et al  en su trabajo [19].

## 3. Resultados

Una vez generado el banco de datos de energía con las características mencionadas anteriormente tal y como se ha mencionado en el capítulo anterior, el primer experimento del método hace uso del software "weka" para encontrar los resultados de los clasificadores más usados en el estado del arte.

Se prueba el algoritmo con el método Hold-Out tal y como lo recomienda el autor del banco de datos [8]; y además se utiliza el método de validación Leave-One-Out para validar estos modelos de clasificación. Los resultados se muestran en la tabla 1.

Se hace notorio en la tabla anterior que el algoritmo más competitivo para esta aplicación es el KNN. Sin importar que los algoritmos naive bayes y C4.5 hayan mostrado un rendimiento muy bueno con el método de validación leave one out, éstos muestran un rendimiento muy pequeño para el caso de hold out; en cambio, el KNN muestra un rendimiento adecuado tanto en Hold-Out y Leave-One-Out.

**Tabla 1.** Resultados de clasificación en modelos del estado del arte.

| Clasificador | Hold-Out | Leave-One-Out |
|---|---|---|
| KNN | .88 | .88 |
| Naive Bayes. | .80 | .80 |
| SVM | .81 | NA |
| C4.5 | .72 | .94 |
| ANN | .81 | NA |

Para este caso en particular, las máquinas de soporte vectorial y las redes neuronales pueden mejorar su desempeño con una arquitectura y configuración correcta, pero su tiempo de respuesta suele ser muy alto, por lo que no fueron considerados para el experimento.

Una vez elegido el clasificador base, se genera barrido clase por clase identificando el rendimiento del clasificador para diferentes números de clases, el cual nos indicará cuáles y cuántas clases pueden ser consideradas para control de una prótesis mioeléctrica. A partir de este experimento el método de validación utilizado es Hold-Out tomando en cuenta la recomendación del autor del banco de datos. La siguiente figura muestra los resultados de este experimento.

**Tabla 2.** Rendimiento del algoritmo para n clases.

| Número de clases | rendimiento | Número de clases | rendimiento | Número de clases | rendimiento |
|---|---|---|---|---|---|
| 1 | 1 | 6 | .97 | 11 | .88 |
| 2 | 1 | 7 | .92 | 12 | .88 |
| 3 | .98 | 8 | .89 | 13 | .87 |
| 4 | .98 | 9 | .89 | 14 | .84 |
| 5 | .98 | 10 | .88 | 15 | .84 |

El resultado del experimento anterior nos indica que sin problema podríamos generar un código para el control de una prótesis mioélectrica y su eficiencia dependerá directamente de su rendimiento, queda claro como para 2 clases distintas no existe discriminación alguna.

Una vez hechas estas pruebas, se genera un algoritmo paralelo con el esquema de clasificadores dedicados mostrado en la figura 2 y se evalúa buscando al menos un rendimiento similar al clasificador original para al menos 2 personas distintas como base para evaluar el algoritmo, ya que en cada uno de los casos el algoritmo se comporta de manera diferente. Los resultados del experimento anterior se muestran en la siguiente tabla.

**Tabla 3.** Resultados de algoritmo KNN dedicado.

| Persona | KNN | KNN dedicado |
|---|---|---|
| 1 | 84.33 | 84.33 |
| 2 | 79.00 | 79.00 |
| 3 | 79.67 | 79.67 |
| 4 | 79.00 | 79.00 |

Aparentemente, el algoritmo dedicado de clasificación no muestra ninguna ventaja; sin embargo, su mayor aportación se encuentra en el tiempo de ejecución, ya que si el proceso se realiza en paralelo, el tiempo de respuesta tanto como para su fase de entrenamiento como su fase de prueba se verá reducido de la siguiente manera.

$$t_d = \frac{t_c}{N},$$

siendo $t_c$ el tiempo de respuesta del algoritmo original y N el número de clasificadores dedicados.

## 4. Conclusión

En este artículo se introduce una metodología rápida para clasificación de EMG con optimización de recursos adecuados para generar algoritmos de control, que pueden ser usados en la implementación de un circuito haciendo uso del concepto de clasificadores dedicados totalmente aplicable en cómputo paralelo y FPGAs.

Se introduce el concepto de energía como característica única de la señal de EMG, y se verifica que este único rasgo proporciona información suficiente para aplicaciones concretas para la clasificación, sin necesidad de un pre-procesamiento.

## Referencias

1. Jiang, N., Vujaklija, I., Rehbaum, H., Graimann, B., Farina, D.: Is Accurate Mapping of EMG Signals on Kinematics Needed for Precise Online Myoelectric Control? IEEE Transactions on Neural Systems and Rehabilitation Engineering, 22, 549–558 (May 2014)
2. Englehart, K., Hugdins, B., Parker, P.: Multifunction Control of Prostheses Using the Myoelectric Signal. Boca Raton, FL: CRC Press (2000)
3. E. Scheme and K. Englehart.: Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use. J. Rehabil. Res. Develop. 48, 643 (2011)
4. Kexin Xing, Peipei Yang, Jian Huang, Yongji Wang, Quanmin Zhu.: A real-time EMG patter recognition method for virtual myoelectric hand control. Neurocomputing, 136, 345–355 (January 2014)
5. Yousefi, J., Hamilton-wright, A.: Charazterizing EMG data using machine-learning tools. Computers in Biology and Medicine, 51, 1–13 (April 2014)
6. Phinyomark, A., Quaine, F., Charbonnier, S., Serviere, C., Tarpin-Bernard, F., Laurillau, Y.: Feature extraction of the first difference of EMG time series for EMG pattern recognition. Computer Methods and Programs in Biomedicine, 117, 247–256 (November 2014)
7. Gokgoz, E., Subasi, A.: Comparison of decision tree algorithms for EMG signal classification using DWT. Biomedical Signal Processing and Control, 18, 138–144 (April 2015)

8. Khushaba, R.N., Kodagoda, S.: Electromyogram (EMG) Feature Reduction Using Mutual Components Analysis for Multifunction Prosthetic Fingers Control. ICARCV 2012 (December 2012)
9. Oppenheim, A.V., Schaffer, R.W., Jhon R.: BuckDiscrete-Time Signal Processing. Upper Saddle River, New Jersey: Prentice-Hall, Inc. (1999)
10. Witten, I.H., Frank, E., Hall, M.A.: Data Mining Practical Machine Learning Tools and Techniques. Burlington, MA 01803, USA: Elsevier Inc. (2011)
11. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. Wiley, New York (2001)
12. McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 5, 115–133 (1943)
13. Rosenblatt, F.: The Perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65, 386–408 (1958)
14. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representation by Backpropagating errors. Nature, 323, 533–536 (1986)
15. Cover, T.M., Hart, P.E.: Nearest Pattern Classification. IEEE Trans. on Information Theory, 13, 21–27 (1967)
16. Díaz-de-León, J.L., Yánez-Márquez, G., Sánchez-Garfias, F.A.: Reconocimiento de patrones. Enfoque probabilístico-estadístico. IT 83, Serie Verde, Centro de Investigación en Computación, IPN, México (2003)
17. Cortes, C., Vapnik, V.: Support Vector Network. Machine Learning, 20, 273–297 (1995)
18. Kuncheva, L.: Combining pattern classifiers. Hoboken, New Jersey. John Wiley & Sons, Inc. (2014)
19. Xiang-Yang Wang, Bei-Bei Zhang, Hong-Ying Yang.: Active SVM-based relevance feedback using multiple classifiers ensemble and features reweighting. Engineering Applications of Artificial Intelligence, 26, 368–381 (January 2013)

# Model for the Creation of Mobile Node Knowledge Networks

Chadwick Carreto A., Elena F. Ruiz, Marina Vicario

Instituto Politécnico Nacional, School of Computing, Mexico City, Mexico

{carretoa, eruizf, mvicario}@ipn.mx

**Abstract**. Knowledge, of any kind or source, is an intangible active, and in consequence, invisible and hard to value. One of mankind's characteristics is his ability to use his own experiences and turn them into actions, which are susceptible of being generalized so they can be transmitted to the following generations. Such natural quality of the individual has transformed into a gradual and systematic process of knowledge, whose purpose has been, amongst others, a higher accessibility to personal development so that it increase the human race's adaptation capacity to the demands of its environment and its social context. In this paper we show the development of a model for creating Knowledge Networks (KNM) networks based on mobile nodes, this model will share relevant and useful for different types of users anywhere, anytime (anytime, anywhere). The KNM aims to make available to users, developing tools for both synchronous and asynchronous communication and to develop the training process with the use of mobile devices with limited capacity Internet connection.

**Keywords:** Knowledge, mobile nodes, knowledge network.

## 1 Introduction

Today's society is facing and endless amount of challenges, which include defining better and more timely ways for it to communicate and collaborate. Knowledge is the most important variable in any organization, without leaving aside the importance of material, technological or financial resources, it is currently considered as one of the most benefiting or prejudicial factors.

This situation requires models and ways to communicate and above all, share and increment knowledge to face the changes demanded by organizations. Changes with a focus on integral development, considering that traditional and current administration models are in a critical phase, in which they cannot resolve problems or satisfy the needs of today's society; it's necessary to migrate to Knowledge Management solutions and create networks that allow the transfer and exchange of knowledge.

However, most specialized authors agree in defining knowledge management based on the different stages that make it up and/or in its objectives [2].

Knowledge Management has the capacity to generate knowledge and induce learning [4] define it as the ability that people have to understand and manage information using technology and knowledge sharing.

In the globalized society of the twenty first century, knowledge networks constitute the maximum expressions of man as a knowledge producer and his need to exchange and transfer what he learns and creates, based on the social interaction in a technological platform given a very particular context.

Knowledge production is closely related to the information needs and the formal organizations created for that purpose. This integration producing knowledge based on society's integral needs as its objective. The main challenges are: Communication, Collaboration and Sharing (CCS).

Given this issue, it is important for organizations to implement knowledge management models and networks to allow their staff development through schemas that allow sharing and generating knowledge in an easy, practical and cheap way.

The structure of the paper is defined in the following way: Section 2 describes the context and the Model proposal to create Knowledge Networks; section 3 describes the technology that was used for this, section 4 identifies the implementation of the model with its results and finally section 5 presents the conclusions and future work.

## 2 Knowledge Network and the Context

We shall define a Knowledge Network as a community of people, who in a formal or informal way, either part or full time, work for a common interest and base their actions in the formation, development and sharing of new knowledge [1].

In a more social and dynamic context, knowledge networks are the human interactions that produce, store, distribute, share, access and analyze knowledge produced by mankind in a systematic way or by personal or group interest of sharing any kind of data by any means, generally electronic; in order to develop their capacities for creating, understanding, power, study and transformation of reality surrounding territory in a given social and economic context [5].

As mentioned earlier, knowledge networks are the result of the human activity essentially formed by producing, managing and transferring knowledge and largely structured by organizations created for that purpose. In this regard, there are several models aimed at the technological and social.

In this fashion, some authors, such as Andreoli [2] propose a knowledge network model with a technological approach based on three fundamental elements: a central core, collaboration clusters and knowledge generation units. They state that a network is made up of an entity or groups of people who lead or coordinate activities aimed towards knowledge generation making use of previously defined topics, then generate groups or research facilities where and analogous network can group a network or more of them. Electronic communication media mainly gives support of this integration.

Also, Atwell propose a model of knowledge networks based of the same technological standpoint but directed to knowledge management in the so-called virtual communities [3]. This model is structured by three basic elements: the network members, the network tools made up by a pure information network and the discussion topics defined by the members (people or organizations).

These experiences, one being Latin American and another of the European Union, enrich the subject of knowledge networks through different schemas that share two fundamental variables: a group of people who live in a society and a technological platform which optimizes knowledge production and transfer made by the former.

In this proposal, knowledge networks are based in a knowledge node interconnection model, which can be translated into a mobile collaboration network that is more focused in its social and organizational character. In this sense, knowledge networks are made up by different kinds of networks, e.g. social networks, main networks, institutional networks, collaboration networks and transfer networks.

Social networks are inserted in the conception of the collaboration action theory, which defines that "social capital made by reciprocity, voluntary cooperation and commitment" [1] are part of the social system dynamics

Social networks share different types of information, data, knowledge and assets. In the same way, they involve mutual benefit, trust and coordinated work that are intimately mediatized by the globalized information world.

The basic primary networks are divided into social action network, which are "the sum of management, administration, participation or association relationships that span the plurality of people or micro-organizations" [1]

Institutional networks are all the organizations created or organized to produce knowledge by researching the needs of social networks and/or the problems of the primary networks, to help their development and social advancement. University, research institutes, and technological development facilities either private or public, among others, compose these networks. Social action networks also operate in this kind of networks.

This networks operate by the thousands in every country around the globe, so that there can be various institutions in the same country researching the same problem without being dynamically and effectively integrated in order to save time, money and effort. Because of this problem, collaboration networks were formed.

Collaboration networks are public or private organization in a local, regional and international level, which can cooperate technically and financially with the institutional networks to manage joint product under relevancy, efficiency, productivity and development parameters.

On the other hand, transfer networks are known for grouping entities, people or organizations that have as a purpose mediating or translating knowledge into social networks and are intimately bonded to the technological innovation and knowledge development processes.

The purpose of this paper is to contextualize how this kind of networks could work together making use of current technology and based primarily on information exchange media that are becoming more common, such as mobile devices.

Technology and telecommunications allow creating node networks that are no more that simple users exchanging information and which can be extended to networks of people sharing knowledge.

## 3 Model of Knowledge Network Creation

In the real world, individuals share information. According to their characteristics and their need to work with this information, they can be classified in groups of interest and knowledge. In this case, the defining characteristics are the individual needs and the way they interact with other Knowledge Network elements. Under this schema, the networks are characterized by the kind of users that interact among themselves. For example, an academic knowledge network will be composed by students, teachers and researchers with particular interest, each of them with restrictions and a way to interact with other entities of this network.

Along this line, as shown in Figure 1, networks are defined by the sets of entities that belong to different knowledge groups and the sets of media which they use to interconnect and make use of sharing, classification and knowledge sharing services or any kind of service. The set made up by all the elements that exist in a KN will be denoted as S and the set of all the types of interconnection media as A.

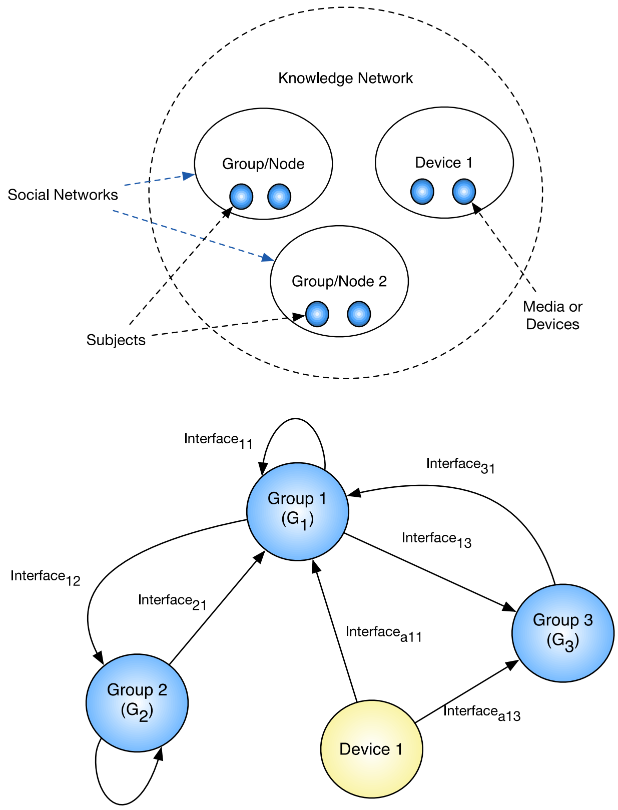



**Fig. 1.** Node Network Components where a user group collaboration schema.

When a group inside the KN is defined, their information sharing capacity along with the media the entities belonging to this group will possess, and the set of services that they can provide to other entities when forming an interaction must also be defined.

Then, each of this groups define in a partial form, the way in which it's going to interact with the rest of the groups, because it defines the interchange actions and the knowledge generation that it can do in favor of the entity with whom it interacts.

To define the interactions that can be done in an KN, we can use a directed graph:

$$G = (V, E) \text{ where } V = S \cup A \text{ y } E \subseteq S \cup A \times S. \quad (1)$$

The edges represent a service relationship, that is, as shown on Figure 1, if an edge gets out of a group G1 and goes into a group G2, indicates that and entity belonging to G1 will offer a service interface to an entity belonging to G2 at the moment a collaboration is done. Similarly and due to an edge existing from G2 to G1, an entity belonging to G2 will offer a service interface to an entity belonging to G2 at the moment of collaboration. We shall define the process of executing an action defined in the collaboration interface exposed by another individual as knowledge generation action.

Every edge of G defines a particular collaboration interface; this allows an individual to collaborate in a particular way depending with whom it interacts. Using the academic network example, a student will present different collaboration interfaces when interacting with another student or a teacher. These interfaces must be defined when designing a KN and as shown later, will be reflected in the methods of knowledge generation as in components that will be used to call the services of the same networks.

The definition of E, the devices have the particularity that the information exchange is direct among entities or other devices and allows them to update with valuable and timely information that can be characterized as knowledge. This is due to the fact that artifacts are conceptualized as reactive entities that execute activities in response to request from the entities, but that are also available and updatable anytime, anywhere. As the result of using the services exposed by a device, an entity can make use of knowledge and information in an automatic way, making it aware of the status of some other entities or group or complete network, relational databases with the groups, with another entity or with any aspect modeled in the KN.

As we said earlier, the relationships defined in the graph are translated into graphic interfaces that will be shown on the user's mobile devices that belong to a KN when the user triggers an information request event. A wide spectrum of collaboration events can be defined, from which the simplest of them gets triggered when two entities of the same group share all their information and filter it according to their needs.

In a graphical way, inside the interaction graph, the names of the entities that the entities will deploy to collaborate with their peers and the interfaces that must be implemented by the entities if they required modifying their status as a result of a knowledge generation action can be defined. For example, if a user I1 belonging to G1 gets integrated into a knowledge group or to another user I2 belonging to G2, this

will make that both individuals exchange their collaboration interface, so that I2 will send I2 an Interface21 and I1 will send I2 an Interface12. When the individuals do the requested interaction, it is assumed that collaboration has ended and the graphical interfaces representing Interface21 and Interface12 disappear from devices I1 and I2 respectively.

# 4 Implementation and Results

The model is designed for mobile systems and basically in node networks (Figure 2 shows the architecture), where it is defined that each user has access to one or more interconnection devices and that the groups that generate one or more knowledge networks are connected using interconnection domains. This is done using various Wireless Technologies that can operate between them, so that this is not an obstacle when communicating. For practical purposes we worked with devices connected to a IEEE 802.11g standard Wi-Fi network. The end-user interaction is fundamental for the model, because that's what makes the user comfortable and completely familiarized with the service management in addition to offering them in the time and manner in which they are requested.

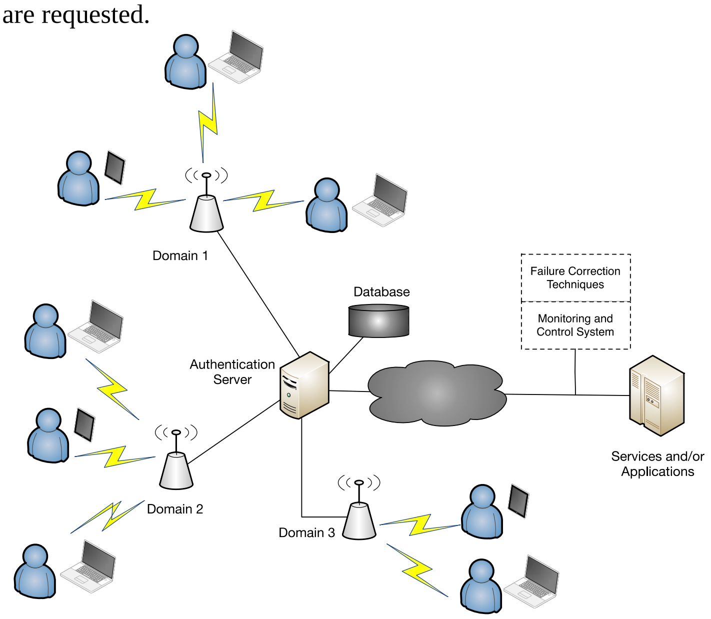



**Fig. 2.** Architecture of the Model Implementation.

We will briefly describe the process that a service request must follow so that the existing services in a Network can be assigned to the end-users.

- The user, through its device, and an interconnection device from the Network domain exchange information to establish an information request in the knowledge network
- Then the user and its interest groups are validated and verified with the purpose of integrating information of its database to the social database and filtering the user's preferences.
- The information that was found according to the defined criteria is once again filtered in a semantic context to integrate just the relevant information according to the needs of the requesting user, when the user needs it.
- Right away, a server will be in charge of offering a list of services and application associated to a user profile database.

- The user must select one of the available services and applications. For this, a connection with the servers where the filtered and defined information is established, so that it can be processed and delivered to the user.
- Software will monitor and control job status and the interaction between nodes based on the aforesaid work model.

The model implementation case study was developed in the ESCOM-IPN, which is an academic unit belonging to the IPN that trains Computer System professionals in undergraduate and postgraduate levels. It has a student body of 4000 undergraduate students and 50 postgraduate students. The educational offer in the postgraduate level is a Master of Science Degree on Mobile Computational Systems.

The case study implementation and tests of the model were done in the “Application Development for Mobile Devices” subject, with the help of 40 undergraduate students and in the postgraduate lab.

Testing was done to measure the ratio of participation and information, resource and service use of the “Application Development for Mobile Devices” students, specifically on the “Android Application Development” topic. We looked for a greater degree of development in programming competences using systems and the knowledge network implementation.

The group was examined on the concepts of Java development for Android version 2.0, the educational model was based on live lessons and practices guided and evaluated during a 4 week span, which generated a set of practices and a test where the academic performance of the 40 students was evaluated.

Using said results; students were given credentials to access the Academic Knowledge Network System (figure 3) so they could have information on the subject (Application Development for Mobile Devices) according to their requirements at the moment they needed it, using any mobile device with Internet access.

In the next test based over the same topic (Java development for Android OS), but now on version 2.2, which implies a change of libraries and programming logic for sensors and accelerometers; the educational model was based on counseling given by the Facilitator, the system was in charge of giving the students information on the libraries and technical bases for software development when they needed them and it was also in charge of following up with the proposed practices in a span of 4 weeks.

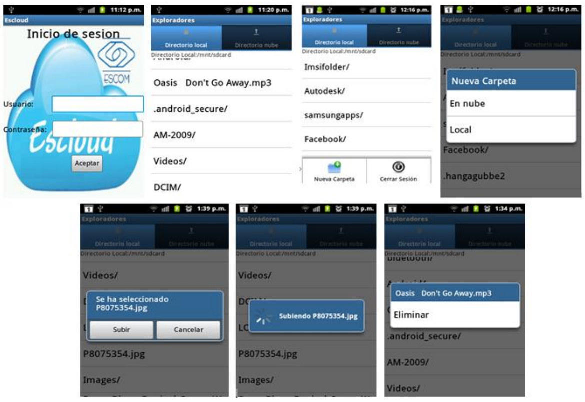



**Fig. 3.** Interface of the System used for the Model Implementation

The obtained results helped us observe how the students generated a series of statements between them based on the information contained in the core documentation and the updated information generated via the collaboration and interaction of the users within the system. The system used for this implementation could be tested on different platforms, due to it being developed for iOS, Android and having a responsive interface developed in HTML5, compatible with the majority of browsers and devices.

The model was also tested with different knowledge networks, made up by a group of entities and their devices. In these tests, service domains of a network were split and a network was created for each communication domain. Communication between entities was done by sharing their knowledge bases, but most importantly; by sharing experiences and information sharing and by the interaction between groups and entities.

The main test defined to work with different networks was designed to interconnect different areas of the Instituto Politécnico Nacional so that they could share information and services in a Knowledge Network composed by 3 Academic Units. Each Academic Unit had a set of users and each user had one or more interconnection devices. Each user registered his/her device on the system and could access to it, in a way that a group was form given the user's mobile location and domain. A node that shared information and services was formed by each of the users, which in turn generated a knowledge base.

Once the groups were defined, communication was established between them and the knowledge base generated by the community was shared.

According to the gathered results, we propose that the project can be applied to a greater amount of nodes. According to technological evaluations and studies, an average of 600 to 1000 simultaneous online nodes or 1000 to 5000 offline nodes can be supported by a simple and basic technological infrastructure. This due to the low hardware requirements, simple communication in which entities don't have to spend a lot of resources and the fact that every day the resources get cheaper.

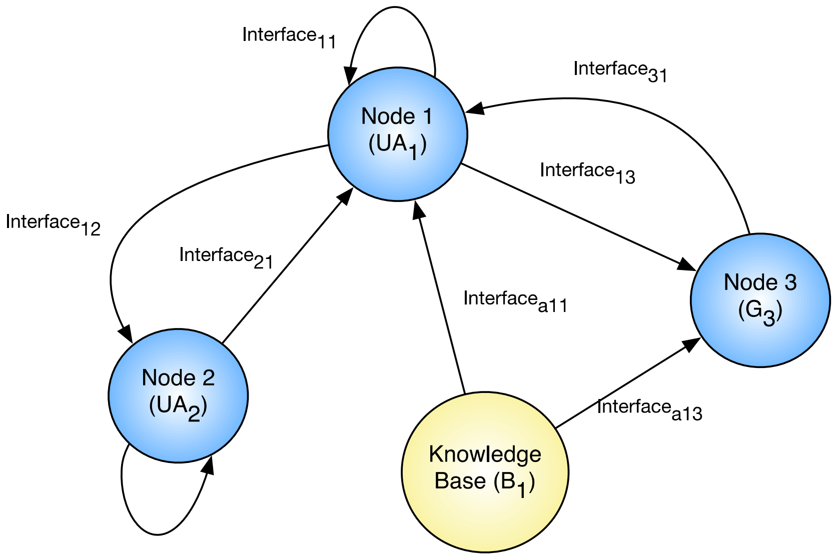



**Fig. 4.** Interconnection between Test Nodes.

## 5 Conclusions and Future Work

The main contribution of the proposed model is the mobility, which translates to saving time and effort in the knowledge network generations. The model itself is another way to help people do their work in any area they desire, not only because it eases work, but because it gives access to well defined, relevant and explicit information when it is required.

The gathering, synthesizing, reflecting and discussing of information is essential for knowledge management; technology has to support all four activities. Knowledge will never be definite. It's incubated and grows without end.

It is important to test the amount knowledge acquired by the use of the proposed model with different users.

The proposed model is a product of a research project that involves two important aspects of information technologies: Knowledge networks and service nodes communication networks.

In such an environment, users can be in constant interaction, which brings feedback, encourages discussions and makes reaching goals a more agile process. Computing technologies, on the other hand, provide us the capacity of bringing the environment to any physical place and in any moment; removing the need to be in front of a fixed computer to be able to access the system.

The kind of architecture that was developed can be useful for other purposes. Thereby, an important conclusion that we have reached is that the architecture if fully flexible and applicable to multiple areas and different user needs. Then, applicability of the architecture can be changed in a simple way.

One way that the architecture could be substantially improved would be creating a cloud that allows storing and sharing the knowledge base with the linked services to all users.

As future work, it is important to highlight that this proposal can be implemented over diverse research fields and education; but the possibility of it being implemented on information distribution areas like economics, politics, social and cultural areas is not dismissed.

**Acknowledgements.** The authors thank the support given to this paper, especially by IPN, ESCOM, CIC, UPIICSA, COFAA, SIP and ICyT DF for the facilities given to conduct our research.

## References

1. Adell, J.: Sobre Entornos Personales de Aprendizaje de Universitat Jaume I. http://files.competenciasbasicas.webnode.es/200000168-105691150b (2009)
2. Hewell, H.: eLearning now. eLearning Papers, 2 (1), Barcelona: elearningeuropa.info. Retrieved (December 2014)
3. Attwell, G.: Personal Learning Environments - the future of eLearning? eLearning Papers, 2 (1), Barcelona: elearningeuropa.info. Retrieved December 18, http://www.elearningeuropa.info/out/?doc_id=9758&rsr_id=11561 (2012)

4. Dutta, S., De Meyer, A.: Building Assets in Real Time and in Virtual Space. Knowledge Management INSEAD, Denmark (2007)
5. Kalz, M.: Building Eclectic Personal Learning Landscapes with Open Source Tools. In: Conference proceedings for the Open Source for Education in Europe, Research & Practise conference, Open University of the Netherlands, Heerlen, Retrieved December 18, 2008, http://www.openconference.net/viewpaper.php?id=16&cf=3 (2008)

# Model of Making Decisions during an Information Search Task

Francisco López-Orozco, Luis D. Rodríguez-Vega

Universidad Autónoma de Ciudad Juárez,
División Multidisciplinaria de Ciudad Universitaria, Cd. Juárez, Chih.,
Mexico

Francisco.Orozco@uacj.mx

**Abstract.** This paper presents a cognitive computational model of the way people read a paragraph with the task of quickly deciding whether it is related or not to a given goal. In particular, the model attempts to predict the time at which participants would decide to stop reading the paragraph because they have enough information to make their decision. Our model makes predictions at the level of words that are likely to be fixated before the paragraph is abandoned. Human semantic judgments are mimicked by computing the semantic similarities between sets of words using Latent Semantic Analysis. A two-variable linear threshold is proposed to account for that decision, based on the rank of the fixation and the semantic similarity between the paragraph and the goal. Model performance is compared to eyetracking data of 19 participants.

**Keywords:** Eyetracking, information search, LSA.

## 1 Introduction

Knowing what web users are doing while they search for information is essential. Several cognitive models have been proposed to account for some of the processes involved in this activity. Pirolli & Fu ([8]) proposed a model of navigation. Brumby & Howes ([2]) describes how people process information partially in order to select links related to an information goal. Chanceaux et al. ([3]) show how visual, semantic and memory processes interact in search tasks.

Information search can be made on any kind of documents, but we are here interested in textual documents, composed of several paragraphs. Information search is different from pure reading because people have a goal in mind while processing the document. They have to constantly keep in memory this additional information. For example, if the task is to decide whether a paragraph is related or not to a given goal, the paragraph and the goal are the pieces of information involved and both have to be together managed in order to make a correct decision.

This paper attempts to model that particular decision making. It focuses on a behavior that is specific to information search, which is stopping processing a paragraph before it is completely read.

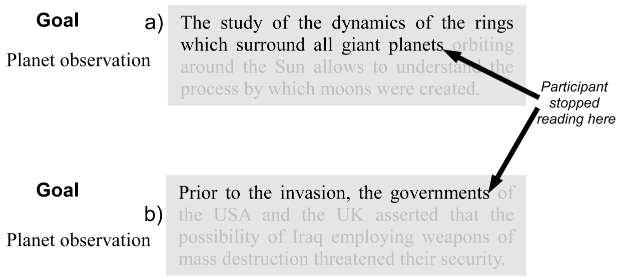



Fig. 1: Illustration of the 2 input data of the model: the goal and the paragraph. The paragraph is abandoned before its end because enough information has been gathered and maybe due to a) a high-relatedness b) a low-relatedness to the goal.

This particular problem has been studied by Lee and Corlett ([6]): participants were provided with a topic and a text, presented one word every second, and were asked to decide as quickly as possible if the text is about the given topic. However, we aim at studying a normal reading situation instead of presenting one word at a time. We will therefore rely on an eyetracker to identify the words processed. Figure 1 illustrates the situation we aim at modeling.

## 2 Experiment

In order to create and study a model, we designed an experiment to gather some data. This experiment was intended to emphasize the decision to stop reading a paragraph while one piece of information is stored in memory: the search goal. A set of 30 goals was created. Each one is expressed by a few words (e.g. *mountain tourism*). For each goal, 6 paragraphs were created (mean=30.1 words, $\sigma$=2.9), 2 of them being highly related to the goal, 2 of them being moderately related, and 2 of them being unrelated. We used Latent Semantic Analysis (LSA) (Landauer et al., [5]) to control the relatedness of a paragraph to the goal. We have not assured a uniform distribution among the frequency words. We sometimes had to manually revise the texts in order to keep an approximately constant length and a particular semantic similarity. The semantic similarity between two sequences of words such as a goal and a paragraph can be computed using the cosine function. The higher the cosine value, the more similar the two sequences of words. We trained LSA on a 24 million word general French corpus.

The experiment is composed of 30 trials, each one corresponding to a goal, in random order. Participants were asked to decide as quick as possible if a paragraph is related or not to a goal given beforehand. In each trial, one paragraph is presented to the participant. The participant should indicate when he is ready to make a decision. Then the paragraph disappears and the participant is asked to keep or reject the paragraph according its relatedness or not with the goal. A new paragraph is displayed and the participant should again decide if the paragraph it is related or not to the goal. This procedure is repeated until all


Fig. 2: Words processed during a fixation using our window-based approach.

6 paragraphs of the current goal were displayed. Each participant was therefore exposed to 30*6=180 paragraphs, and asked to decide if they are related or not to a goal. 19 students participated in the experiment. Eye movements were recorded using a SR Research EyeLink II eye tracker. From these coordinates, saccades and fixations were determined, leading to an experimental scanpath. The stimuli pages were generated with a software that stored the precise coordinates of each word on the screen. We wrote our experiment in Matlab, using the Psychophysics Toolbox (Brainard, [1]).

Before trying to mimic eye movements, we had to predict which words were actually processed by participants in each fixation. It is known that the area from which information can be extracted during a single fixation extends from about 3-4 characters to the left of fixation to 14-15 characters to the right of fixation (Rayner, [10]). This area is asymmetric to the right and corresponds to the global perceptual span. Therefore, more than one word may be processed for a given fixation. In order to determine which ones were processed for each fixation, we used a window, sized according to Rayner ([10]). He showed that the area from which a word can be *identified* extends to no more than 4 characters to the left and no more than 7-8 characters to the right of fixation and corresponds to the word identification span. Moreover, Pollatsek et al ([9]) show that even if information of the next line is processed during a reading task, participants are not capable of getting some semantic information. Therefore, the size of our window is 4 x 1 characters to the left plus 8x1 characters to the right of the fixation point. Since the initial fixations in the beginning part of a word facilitate its recognition more than initial fixations toward the end of the word (Farid & Grainger, [4]), we considered that a word is processed if at least the first third of it or the last two-thirds is inside the window. In the example of Fig. 2, two words were supposed to have been processed: *"des"* and *"données"*. The second one is considered because at least the first third of it is inside the window. The word *"traitement"* is not supposed to have been processed because at least the last two-thirds of the word are not inside the window.

## 3 Modeling

The model should be able to predict the way an average user would process a paragraph given a goal. Our method is therefore to consider the experimental scanpaths and for each participant's fixation to predict whether the paragraph would be abandoned or not. A very good model would predict an abandon at the same time the participant stopped reading. A bad model would abandon too early or too late.

Such a decision making model on paragraphs needs to be based on a model of semantic memory that would be able to mimic human judgments of semantic associations. We used LSA to dynamically compute the semantic similarities between the goal and each set of words that are supposed to have been fixated.

### 3.1 Relatedness Effect of the Paragraph

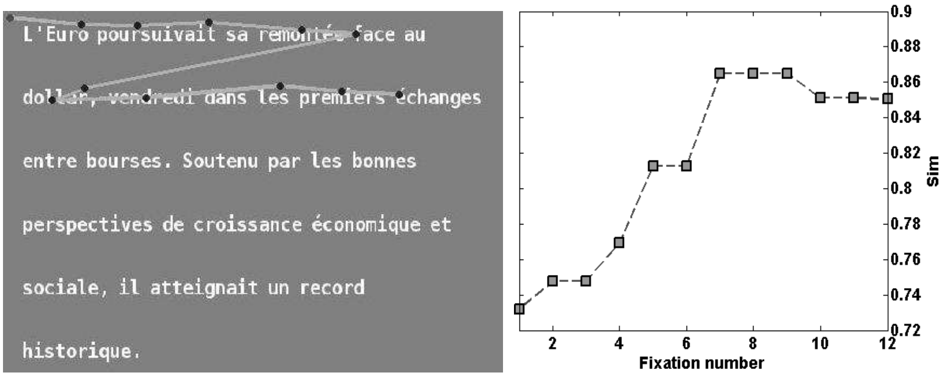
The relatedness of the paragraph to the goal may play a role in the way it is processed. We suspected that if the paragraph is not related to the goal at all, the paragraph would be processed just to verify that is not relevant for the given goal since that all its words are unrelated (with a low semantic association) with the goal. The number of fixations needed to confirm its not relatedness with the goal may depend of each participant and in this paper we are not interested in this particular case. However, if the whole paragraph is highly related to the goal, the paragraph is composed of unrelated, moderately and highly related words to the goal. Due to this richness of words, the paragraph should be processed with the idea of investigate whether it is relevant or not. Here the relatedness of the paragraph to the goal plays a role in the way it is processed. Our analysis is restricted when the paragraph is strongly related to the goal.

### 3.2 Relatedness Evolution of the Paragraph

All the paragraphs are composed of coherent text and with an approximately constant length. However the evolution of its similarity with each goal was not controlled. In some of the paragraphs, the similarity value could be going up or down from the beginning to the end of the paragraph and in other cases an alternating pattern of increasing/decreasing values could be observed. After a clustering procedure according to the cosine evolution of only the strongly related paragraphs we learned 3 different classes: *step*, *ramp* and *saw*.

The cosine evolution of step paragraphs is characterised by a constant value during some fixations followed by an abrupt change (positive jump) to a higher value which is set till to the *signal* end. In the case of the *ramp* paragraphs, a cosine value is maintained during some fixations but here it is followed by a progressive increasing of the cosine to a higher value which is set till to the *signal* end. The cosine evolution in the *saw* paragraphs is still more complicated. Here, the evolution of the cosine value follows a increasing/decreasing pattern along the duration of the *signal*. We called them *saw* because of its resemblance with a non-sinusoidal waveform. Some paragraphs were not clustered in any of these classes due its high complexity of its cosine evolution.

In order to simplify the posterior analyses, only the *ramp* paragraphs were chosen because they have a higher cosine variability than the *step* paragraphs but their variability is more predictable than the case of *saw* paragraphs.




(a) (b)

Fig. 3: a) Example of scanpath in the C|S condition. b) Its $Cos$ evolution.

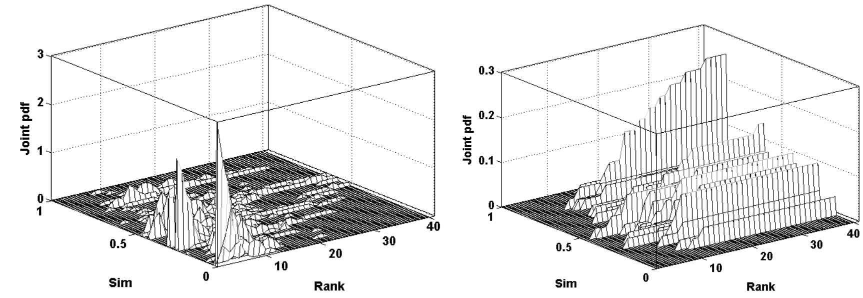
### 3.3 Modeling the Decision

**Two variables involved**. We first looked for the variables which could play a role in the decision to stop reading a paragraph $p$. Such a decision is made when the participant's perception about the relatedness of the paragraph with the goal is completed. The association to the goal $g$ is obviously involved in that perception. Therefore, we defined a variable called $Cos = |sim(\text{words of } p,\ g)|$ in which $sim$ is the LSA cosine between the two vectors. $Cos$ changes constantly while a paragraph is processed since it depends on the words actually processed. When the words seen are highly related with the goal that variable has a high value and it has a low value (close to zero) when the words are unrelated with the goal. It can be easily calculated dynamically, after each word of the paragraph has been processed. Consider for example Fig. 3a. The goal is *"faiblesse du dollar"* (*weak dollar*). In the first fixation on the paragraph, only the word *"L'Euro"* is supposed to has been processed according to our window-based prediction. This word is highly related with the goal. Therefore $Cos = |sim(\text{"L'Euro"}, \text{"faiblesse du dollar"})| = 0.731$.

During fixation 2, two words are processed, the word *"L'Euro"* and a new word *"poursuivait"* leading to a new value of $Cos = |sim(\text{"L'Euro poursuivait"}, \text{"faiblesse du dollar"})| = 0.747$.

During fixation 3, only the word *"poursuivait"* is processed, leading to the same value of $Cos = 0.747$.

In fixation number 4, $Cos = |sim(\text{"poursuivait sa remontée"}, \text{"faiblesse du dollar"})| = 0.769$. In fixation 5, the $Cos$ value goes up to 0.812 because of the words *"remontée face"* which makes the LSA vector much more similar to the goal vector. This value is maintained for the fixation 6. In fixation 7, the $Cos$ value goes up to 0.865 because of the word *"dollar"* makes still the LSA vector much more similar to the goal vector. During fixation number 9, the words *"vendredi dans"* makes the LSA vector less similar to the goal vector and this effect is showed by the decreasing value of $Cos = 0.850$. This value is maintained till




(a) (b)

Fig. 4: a) Empirical "no-abandon" distribution $\hat{p}_{CR}(c, r|\overline{Ab})$ and b) "abandon" distribution $\hat{p}_{CR}(c, r|Ab)$ in the $Cos \times Rank$ space.

to the 12th fixation (stop reading). Figure 3b shows the evolution of the $Cos$ value along the fixations in the scanpath. This example illustrates that a high value of $Cos$ may not directly induce the decision, in particular if it appears too early in the scanpath. We assume that the decision also depends on the number of words processed so far in the paragraph. The more words processed, the higher the confidence in the perception of the relatedness between the paragraph and the goal. If only two or three words have been processed, it is less likely that $Cos$ is accurate. Therefore, we assume that there should be a relationship between $Cos$ and the number of words processed. The second variable is then $Rank =$ number of words processed so far.

**Abandon and no-abandon distributions**. In order to study how the decision depends on these two variables, we computed two distributions in the $Cos \times Rank$ space of participant data: the distribution of the no-abandon cases and the distribution of the abandon cases. The goal is to learn the frontier between both cases in order to be able to predict if a sequence of words already processed is likely to lead to the abandon or the pursuance of the reading task. This work was done on two thirds of the data, in order to leave one third to test the model. Each participant fixation was associated to a point in the $Cos \times Rank$ space. $Rank$ is a discrete measure between 1 and the maximum number of fixations in the data (60 in our case). $Cos$ has been computed according to the previous formula, taking into account the words already processed in each paragraph as well as the goal and discretized into one of 100 bins, from 0 to 1.

The no-abandon distribution was computed by simply counting the number of fixations that did not lead to an abandon for each cell of the $Cos \times Rank$ grid. It concerns all fixations except the last one of each scanpath.

The abandon distribution was built from all very last fixations of all scanpaths, including also subsequent ranks. For example, if a given participant on a given stimulus made 13 fixations, the first 12 were counted in the no-abandon distribution and the 13th was counted in the abandon distribution. All virtual fixations from 14 to 60, with the same $Cos$ value as the 13th were also counted in

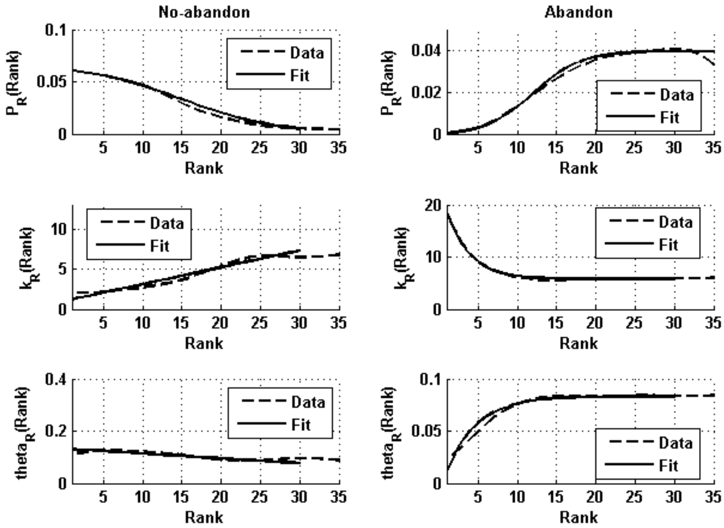



Fig. 5: Data and fitting of marginal distributions, shape and scale for the "no-abandon" and "abandon" distributions.

the abandon distribution, because if the participant stopped reading at fixation 13, he would have also stopped at fixation 14, 15, etc. The frontier between these two behaviors (continue or stop reading) is a curve in the $Cos \times Rank$ space. Depending on the location of any observation ($c$,$r$) above or under the curve, the reader's behavior can be predicted. To find this frontier, a methodology based on a Bayesian classifier is used. Let us consider a classification problem with two classes: Abandon ($Ab$) and No-abandon ($\overline{Ab}$). Given the posterior probabilities, which is the class of a two-dimensional observation ($c$,$r$) in the $Cos \times Rank$ space? The decision rule is then: $P(\overline{Ab}|c,r) \underset{Ab}{\overset{\overline{Ab}}{\gtrless}} P(Ab|c,r)$ with $P(\overline{Ab}|c,r) = \frac{P(\overline{Ab}) \times p_{CR}(c,r|\overline{Ab})}{p_{CR}(c,r)}$, and $P(Ab|c,r) = \frac{P(Ab) \times p_{CR}(c,r|Ab)}{p_{CR}(c,r)}$. Figures 4a and 4b represent the two empirical class-conditional probability density functions respectively $\hat{p}_{CR}(c,r|\overline{Ab})$ and $\hat{p}_{CR}(c,r|Ab)$. We adopt a statistical parametric approach. By this way, data will be regularized since they are obviously affected by the noise inherent to acquisition and pre-processing.

In the next sections, the statistical model to estimate the density functions and the prior probabilities are explained in order to use the Bayesian classifier:

$$P(\overline{Ab}) \times p_{CR}(c,r|\overline{Ab}) \underset{Ab}{\overset{\overline{Ab}}{\gtrless}} P(Ab) \times p_{CR}(c,r|Ab)\,.$$

**Parametric model for the "no-abandon" distribution**. The class-conditio– nal probability density function can be written as : $p_{CR}(g,r|\overline{Ab}) = p_{C|R}(c|R =$

$r, \overline{Ab}) \times p_R(r|\overline{Ab})$. Figure 5 (top, left) shows the empirical marginal distribution $\hat{p}_R(r|\overline{Ab})$. As the *Rank* increases, the probability of not abandoning the paragraph decreases. This evolution was modeled with a sigmoid function $\varphi(r) = \frac{P_{RMax} \times (1+e^{-\alpha r_0})}{1+e^{\alpha(r-r_0)}}$. There are actually only two parameters to fit because the integral is 1.

Concerning the probability density function $p_{C|R}(.)$, the natural model (Fig. 4a) is a Gamma one whose parameters depend on the *Rank* value. The shape $k(r)$ increases and the scale $\theta(r)$ decreases linearly (Fig. 5, left column). The linear regressions are only performed up to the $Rank30$ since that $\hat{p}_R(r > 30|\overline{Ab})$ is close to zero and there is no more enough data. Then we have:

$$p_{C|R}(c|R = r, \overline{Ab}) = \frac{A(r)c^{k(r)-1}}{(k(r)-1)!\theta(r)^{k(r)}} e^{-\frac{c}{\theta(r)}}, p_R(r, \overline{Ab}) = \varphi(r)\,.$$

As the *Cos* value is between 0 and 1, $A(r)$ is a normalization function to ensure that $p_{C|R}(c|R = r, \overline{Ab})$ is a probability density function: $A(r) = F_{k,\theta}(1) - F_{k,\theta}(0)$, with $F_{k,\theta}(.)$ being the repartition function of a Gamma distribution with a shape $k$ and a scale $\theta$. We then obtained six independent parameters to model the complete "no-abandon" joint distribution (offset and slope for the sigmoid, and the coefficients for the two linear functions).

**Parametric model for the "abandon" distribution**. Following a similar approach the class-conditional pdf is written as : $p_{CR}(c, r|Ab) = p_{C|R}(c|R = r, Ab) \times p_R(r|Ab)$. The marginal pdf $\hat{p}_R(r|Ab)$ was modeled with another sigmoid function $\varphi'(r)$ (Fig. 5, top right). But here, it is an increasing function. At rank 0, there is no abandon and at the maximal *Rank* value, all scanpaths have shown an abandon. The conditional distribution $\hat{p}_{C|R}(c|R = r, Ab)$ is a Gamma distribution with a shape $k'(r)$ and a scale $\theta'(r)$. The shape $k'(r)$ exponentially decreases while the scale $\theta'(r)$ exponentially increases (Fig. 5, right column). Equations of the pdf are the same as the previous case, but with a different set of functions $\{\varphi'(r), k'(r), \theta'(r)\}$ which gives us eight parameters (2 for the $\varphi'(r)$, 3 for $k'(r)$ and 3 for $\theta'(r)$):

$$p_{C|R}(c|R = r, \overline{Ab}) = \frac{A'(r)c^{k'(r)-1}}{(k'(r)-1)!\theta'(r)^{k'(r)}} e^{-\frac{c}{\theta'(r)}}, p_R(r, \overline{Ab}) = \varphi'(r)\,.$$

**Modeling the decision as the function of Rank and Cos**. As these two class-conditional probabilities were modeled, for each (*Rank*, *Cos*) values, the problem is to decide if there is enough information to stop reading ("abandon" class), or to continue reading ("no abandon" class). This binary problem is solved thanks to the Bayesian classifier. To find this decision rule, we have now to estimate the prior probabilities such as : $P(Ab) + P(\overline{Ab}) = 1$. $P(Ab)$ or $P(\overline{Ab})$ is another parameter to learn from the data. The total number of learning parameters is then 15 (6+8+1). The decision rule is then:

$$P(\overline{Ab}) \times p_{C|R}(c|R = r, \overline{Ab}) \times p_R(r|\overline{Ab}) \underset{Ab}{\overset{\overline{Ab}}{\gtrless}} P(Ab) \times p_{C|R}(c|R = r, Ab) \times p_R(r|Ab)\,.$$

## 4 Model Learning

After learning the two posterior probabilities $P(\overline{Ab}|c,r)$ and $P(Ab|c,r)$ to represent the decision frontier between the two classes, the two prior probabilities are $P(Ab) = 0.64$ and $P(\overline{Ab}) = 0.36$. The intersection between the posterior probabilities is oblique which is what was expected, from a cognitive point of view. $Rank$ and $Cos$ are dependent on each other: at the beginning of processing the paragraph, there should be a high relatedness between the paragraph and the goal to make the decision. However, after more fixations have been made, that relatedness could be lower to decide to abandon the paragraph.

For instance, at rank 10, a $Cos$ of 0.7 is necessary to stop reading, whereas at rank 15, a value of 0.3 is enough. The frontier is rather linear and can be approximated by the following equation in the $Cos \times Rank$ space: $Cos_0 = -0.0473 \times Rank + 0.9849$ . That equation was included in the computational model. That model constantly computes the $Cos$ value while it is moving forward in the text, increasing the $Rank$ value. As soon as the current $Cos$ value is greater than $Cos_0$, the decision is to stop reading the paragraph.

In order to test the model, we ran it on the remaining one third of the data. For each fixation in this testing set, the model decides either to leave or not to leave the paragraph. If the model did not leave at the time the participant stopped reading, simulation is pursued with the next rank and with the same value of the gap, and so on until the decision is made. The average difference between the ranks at which model and participant stopped reading was computed. We got a value of 4.2 (SE=0.5). To assess the significance of that value, we built a random model which stops reading after each fixation with probability $p$. The smallest average difference between participants' and model's ranks of abandoning was 9.9 (SE=0.9) and was obtained for $p = 0.22$. Our model therefore appears to be much better than the best random model.

## 5 Conclusions

We presented a binary model which predicts the sequence of words that are likely to be fixated before a paragraph is abandoned given a search goal. In spite of the drawbacks of LSA method, we got good model's performances but a further investigation should be done in order to know in more details the impact of these limits in our modeling work. For example, a natural comparison of performance would be by using LSA versus a simple Bag of words (BoW). In our model, two variables seem to play a role: the rank of the fixation and the semantic similarity between the paragraph and the search goal. We proposed a simple linear threshold to account for that binary decision. Our model will be improved in future work. In particular, we aim at considering a non linear way of scanning the paragraph, using another model of eye movements (Lemaire et al., [7]). We also plan to tackle more realistic stimuli as well to consider other decisions involved in Web search tasks.

**Acknowledgements.** Thanks to PROMEP for the financial support to the project "Modelización Cognitiva Computacional de Baja Complejidad para la Búsqueda de Información En Español basado en el Comportamiento Ocular de los Usuarios" (2015) with ID DSA/103.5/15/7004 under which this work has been finished. This work was part of the research project ANR Gaze-EEG "Joint synchronous EEG signal and eye tracking processing for spatio-temporal analysis and modeling of neural activities" (2009-2013).

## References

1. Brainard, D. H.: The Psychophysics Toolbox. Spatial Vision, 10, 433–436 (1997)
2. Brumby, D. P., Howes, A.: Good enough but I'll just check: Web-paged search as attentional refocusing. In: Proc of the 6th ICCM Conference, 46–51 (2004)
3. Chanceaux, M., Guérin-Dugué, A., Lemaire, B., Baccino, T.: A model to simulate Web users' eye movements. In: Proc of the 12th INTERACT Conference, LNCS 5726, Berlin: Springer Verlag, 288–300 (2009)
4. Farid, M., Grainger, J. How initial fixation position influences word recognition: A comparison of French and Arabic. Brain & Language, 53, 351–368 (1996)
5. Landauer, T., McNamara, D., Dennis, S., Kintsch, W.: Handbook of Latent Semantic Analysis. Lawrence Erlbaum Associates (2007)
6. Lee, M.D., Corlett, E.Y. Sequential sampling models of human text classification. Cognitive Science, 27(2), 159–193 (2003)
7. Lemaire, B., Guérin-Dugué, A., Baccino, T., Chanceaux, M., Pasqualotti, L.: A cognitive computational model of eye movements investigating visual strategies on textual material. In: L. Carlson, C. Hölscher, T. Shipley (Eds.) Proc of the Annual Meeting of the Cognitive Science Society, pp. 1146–1151 (2011)
8. Pirolli, P., Fu, W.: SNIF-ACT: a model of information foraging on the world wide web. In: P. Brusilovsky, A. Corbett, F. de Rosis (Eds.), 9th ICUM, pp. 45–54 (2003)
9. Pollatsek, A., Raney, G. E., LaGasse, L., Rayner, K.: The use of information below fixation in reading and in visual search. Can J Psychol, 47, 179–200 (1993)
10. Rayner, K.: Eye movements in reading and information processing: 20 years of research. Psychological Bulletin, 124(3), 372–422 (1998)

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Noviembre de 2015
Printing 500 / Edición 500 ejemplares