

Complexity

Intelligent Control Approaches for Modeling and Control of Complex Systems

Lead Guest Editor: José M. Andújar

Guest Editors: Eloy Irigoyen and Victor M. Becerra





Intelligent Control Approaches for Modeling and Control of Complex Systems

Complexity

Intelligent Control Approaches for Modeling and Control of Complex Systems

Lead Guest Editor: José M. Andújar

Guest Editors: Eloy Irigoyen and Victor M. Becerra



Copyright © 2018 Hindawi. All rights reserved.

This is a special issue published in “Complexity.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board

- José A. Acosta, Spain
Tarek Ahmed-Ali, France
Rodrigo Aldecoa, USA
Basil M. Al-Hadithi, Spain
Juan A. Almendral, Spain
Diego R. Amancio, Brazil
David Arroyo, Spain
Mohamed Boutayeb, France
Arturo Buscarino, Italy
Guido Caldarelli, Italy
Eric Campos-Canton, Mexico
Mohammed Chadli, France
Diyi Chen, China
Giulio Cimini, Italy
Danilo Comminiello, Italy
Sara Dadras, USA
Manlio De Domenico, Italy
Pietro De Lellis, Italy
Albert Diaz-Guilera, Spain
Thach Ngoc Dinh, France
Jordi Duch, Spain
Marcio Eisencraft, Brazil
Joshua Epstein, USA
Mondher Farza, France
Thierry Floquet, France
Mattia Frasca, Italy
Lucia Valentina Gambuzza, Italy
Bernhard C. Geiger, Austria
- Carlos Gershenson, Mexico
Peter Giesl, UK
Sergio Gómez, Spain
Wesley N. Gonçalves, Brazil
Lingzhong Guo, UK
Xianggui Guo, China
Sigurdur F. Hafstein, Iceland
Chittaranjan Hens, Israel
Giacomo Innocenti, Italy
Sarangapani Jagannathan, USA
Mahdi Jalili, Australia
Jeffrey H. Johnson, UK
M. Hassan Khooban, Denmark
Vincent Labatut, France
Lucas Lacasa, UK
Qingdu Li, Germany
Chongyang Liu, China
Xiaoping Liu, Canada
Rosa M. Lopez Gutierrez, Mexico
Vittorio Loreto, Italy
Didier Maquin, France
Eulalia Martínez, Spain
Marcelo Messias, Brazil
Ana Meštrović, Croatia
Ch. P. Monterola, Philippines
Marcin Mrugalski, Poland
Roberto Natella, Italy
Nam-Phong Nguyen, USA
- Beatrice M. Ombuki-Berman, Canada
Irene Otero-Muras, Spain
Yongping Pan, Singapore
Daniela Paolotti, Italy
Mahardhika Pratama, Singapore
Luis M. Rocha, USA
Miguel Romance, Spain
Avimanyu Sahoo, USA
Matilde Santos, Spain
Hiroki Sayama, USA
Michele Scarpiniti, Italy
Enzo Pasquale Scilingo, Italy
Dan Selișteanu, Romania
Dimitrios Stamovlasis, Greece
Samuel Stanton, USA
Roberto Tonelli, Italy
Shahadat Uddin, Australia
Gaetano Valenza, Italy
Dimitri Volchenkov, USA
Christos Volos, Greece
Zidong Wang, UK
Yan-Ling Wei, Singapore
Honglei Xu, Australia
Xinggang Yan, UK
Massimiliano Zanin, Spain
Hassan Zargarzadeh, USA
Rongqing Zhang, USA

Contents

Intelligent Control Approaches for Modeling and Control of Complex Systems

José M. Andújar , Eloy Irigoyen , and Victor M. Becerra 
Editorial (2 pages), Article ID 2090715, Volume 2018 (2018)

Finite-Time Switched Second-Order Sliding-Mode Control of Nonholonomic Wheeled Mobile Robot Systems

Hao Ce, Wang Hongbin , Cheng Xiaoyan , Zhou Zhen , Ge Shungang, and Hu Zhongquan
Research Article (10 pages), Article ID 1430989, Volume 2018 (2018)

Event-Based Time Varying Formation Control for Multiple Quadrotor UAVs with Markovian Switching Topologies

Zhen Zhou , Hongbin Wang , and Zhongquan Hu
Research Article (15 pages), Article ID 8124861, Volume 2018 (2018)

Intelligent UAV Map Generation and Discrete Path Planning for Search and Rescue Operations

Víctor San Juan, Matilde Santos , and José Manuel Andújar 
Research Article (17 pages), Article ID 6879419, Volume 2018 (2018)

Using Deep Learning to Predict Complex Systems: A Case Study in Wind Farm Generation

J. M. Torres , and R. M. Aguilar 
Research Article (10 pages), Article ID 9327536, Volume 2018 (2018)

Low Speed Longitudinal Control Algorithms for Automated Vehicles in Simulation and Real Platforms

Mauricio Marcano , José A. Matute, Ray Lattarulo, Enrique Martí, and Joshué Pérez
Research Article (12 pages), Article ID 7615123, Volume 2018 (2018)

Multi-GPU Development of a Neural Networks Based Reconstructor for Adaptive Optics

Carlos González-Gutiérrez , María Luisa Sánchez-Rodríguez, José Luis Calvo-Rolle ,
and Francisco Javier de Cos Juez 
Research Article (9 pages), Article ID 5348265, Volume 2018 (2018)

Optimal Modeling of Wireless LANs: A Decision-Making Multiobjective Approach

Tomás de Jesús Mateo Sanguino  and Jhon Carlos Mendoza Betancourt
Research Article (15 pages), Article ID 7560717, Volume 2018 (2018)

Design, Modelling, and Implementation of a Fuzzy Controller for an Intelligent Road Signaling System

José Manuel Lozano Domínguez  and Tomás de J. Mateo Sanguino 
Research Article (14 pages), Article ID 1849527, Volume 2018 (2018)

Neural Models for Imputation of Missing Ozone Data in Air-Quality Datasets

Ángel Arroyo , Álvaro Herrero, Verónica Tricio, Emilio Corchado, and Michał Woźniak
Research Article (14 pages), Article ID 7238015, Volume 2018 (2018)

Intelligent Torque Vectoring Approach for Electric Vehicles with Per-Wheel Motors

Alberto Parra , Asier Zubizarreta , Joshué Pérez, and Martín Dendaluze
Research Article (14 pages), Article ID 7030184, Volume 2018 (2018)

WH-EA: An Evolutionary Algorithm for Wiener-Hammerstein System Identification

J. Zambrano , J. Sanchis , J. M. Herrero , and M. Martínez 

Research Article (17 pages), Article ID 1753262, Volume 2018 (2018)

A Novel Fuzzy Algorithm to Introduce New Variables in the Drug Supply Decision-Making Process in Medicine

Jose M. Gonzalez-Cava , José Antonio Rebozo, José Luis Casteleiro-Roca , José Luis Calvo-Rolle ,
and Juan Albino Méndez Pérez 

Research Article (15 pages), Article ID 9012720, Volume 2018 (2018)

Editorial

Intelligent Control Approaches for Modeling and Control of Complex Systems

José M. Andújar ¹, **Eloy Irigoyen** ² and **Victor M. Becerra** ³

¹University of Huelva, Huelva, Spain

²University of Basque Country, Bilbao, Spain

³University of Portsmouth, Portsmouth, UK

Correspondence should be addressed to José M. Andújar; andujar@uhu.es

Received 10 May 2018; Accepted 10 May 2018; Published 31 May 2018

Copyright © 2018 José M. Andújar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A complex system can be defined as any system featuring a large number of interacting components whose aggregate activity is nonlinear and hence not derivable from the summations of the activity of individual components. As control methods are normally applied to dynamical systems, we are interested in complex dynamical systems. The rising complexity of man-made engineering systems (e.g., power networks, aircraft, autonomous vehicles, and communication networks) causes us increasing difficulties to understand their behaviour and to obtain accurate first-principle dynamical models to describe them, which reduces our capability to control those systems and make them behave in the way we want.

Intelligent control refers to approaches to control system design, modeling, identification, and operation that use artificial intelligence techniques, such as fuzzy logic, neural networks, machine learning, evolutionary computation, and genetic algorithms. Intelligent control techniques are often capable of controlling dynamical systems that, because of their complexity, are very difficult to control by other techniques. Intelligent control systems are characterized by attempts to emulate important aspects of biological intelligence. These aspects include, for instance, adaptation and learning, planning under uncertainty, and decision making. Intelligent control is interdisciplinary in nature as it seeks inspiration from biology, and it combines and extends theories and methods from areas such as control theory, mathematics, and computer science.

The aim of this special issue is to bring together a selection of the latest advances and trends in the application of

intelligent control to the modeling and control of complex systems. This special issue consists of 12 contributions describing recent work from active researchers in the area.

H. Ce et al. present a continuous finite-time robust control method for the trajectory tracking control of a nonholonomic wheeled mobile robot (NWMR). The approach involves an inner loop consisting of a conventional sliding mode controller and an outer loop that has a modified switched second-order sliding mode controller. Experiments are conducted on a Quanser QBot system. J. Zambrano et al. present an evolutionary approach for Wiener-Hammerstein system identification which can identify the linear and nonlinear sections of this nonlinear model structure in a single step, rather than two steps as is required in previous approaches. The proposed method is evaluated using well-known benchmark data for nonlinear modeling.

J. M. Lozano Domínguez and T. D. Mateo Sanguino describe their work on the development of a fuzzy controller for an intelligent road signalling system consisting of a set of autonomous, intelligent, and low-cost devices that are employed to alert drivers when approaching zebra crossings if a pedestrian is traversing. The results of successful tests of the proposed signalling system on laboratory and real environments are described. C. González-Gutiérrez et al. present a neural network-based reconstructor for adaptive optics that is specially designed for multi-GPU systems.

V. San Juan et al. present a fuzzy logic-based approach for intelligent unmanned aerial vehicle (UAV) map generation and path planning for search and rescue operations. The approach is tested on a simulated search and rescue scenario

based on real aerial images obtained during a forest fire in La Palma, Spain. A. Parra et al. provide an article on an intelligent torque vectoring approach for electric vehicles. The system consists of a neurofuzzy vertical tyre force estimator and a fuzzy yaw moment controller. A. Arroyo et al. present an approach for the development of neural models for imputation of missing ozone data in air quality datasets. Their approach is successfully tested using data from air quality stations based in Castilla y León, Spain.

T. D. Mateo Sanguino and J. C. Mendoza Betancourt describe an approach based on multiobjective genetic algorithms for the optimal modeling of wireless local area networks (WLANs). Their approach allows the investigation of various complex issues concerning the design of IEEE 802.11-based WLANs.

M. Marcano et al. present low speed longitudinal control algorithms for automated vehicles. They present a complete framework for automated driving, which includes the use of neurofuzzy techniques for longitudinal control and is implemented in virtual and real vehicles. Z. Zhou et al. describe an event-based time varying formation control for multiple UAVs that uses Markovian switching topologies. The effectiveness of the approach is assessed by means of simulations of six quadrotor UAVs flying in formation.

J. M. Gonzalez-Cava et al. present a fuzzy algorithm to introduce new variables in the drug supply decision-making process and test their proposed approach by using the analgesia nociception index (ANI) as a guiding variable for the analgesic process during surgical interventions. Finally, J. M. Torres and R. M. Aguilar describe an approach that uses deep learning to predict the future behaviour of complex systems and use power generation forecasting on a wind farm as a case study.

These authors highlight both the promise, the applications, the advantages brought about by the use of intelligent control approaches in modeling and control of complex systems, and challenges faced by engineers and scientists in developing and applying these techniques. In summary, this special issue provides a snapshot of very interesting and recent work on intelligent control approaches for modeling and control of complex systems.

*José M. Andújar
Eloy Irigoyen
Victor M. Becerra*

Research Article

Finite-Time Switched Second-Order Sliding-Mode Control of Nonholonomic Wheeled Mobile Robot Systems

Hao Ce, Wang Hongbin , Cheng Xiaoyan , Zhou Zhen ,
Ge Shungang, and Hu Zhongquan

Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China

Correspondence should be addressed to Wang Hongbin; hb_wang@ysu.edu.cn

Received 26 November 2017; Accepted 12 March 2018; Published 9 May 2018

Academic Editor: Eloy Irigoyen

Copyright © 2018 Hao Ce et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A continuous finite-time robust control method for the trajectory tracking control of a nonholonomic wheeled mobile robot (NWMR) is presented in this paper. The proposed approach is composed of conventional sliding-mode control (SMC) in the internal loop and modified switched second-order sliding-mode (S-SOSM) control in the external loop. Sliding-mode controller is equivalently represented as stabilization of the nominal system without uncertainties. An S-SOSM control algorithm is employed to counteract the impact of state-dependent unmodeled dynamics and time-varying external disturbances, and the unexpected chattering has been attenuated significantly. Particularly, state-space partitioning is constructed to obtain the bounds of uncertainty terms and accomplish different control objectives under different requirements. Simulation and experiment results are used to demonstrate the effectiveness and applicability of the proposed approach.

1. Introduction

In the past few decades, the design of a robust control of nonholonomic wheeled mobile robot (NWMR) has been a difficult task due to the nonlinearities, nonholonomic constraints, and uncertainties in the system [1]. So, it received wide attention and became a topic of great research interest due to the practicability of its application [2]. Basically, trajectory tracking is one of the important motion control problems, which not only requires a designed controller [3] but also has to robustly stabilize the nonlinear system against the system uncertainties. Depending on whether the NWMR is described by kinematics [4, 5] or dynamics [6, 7], the tracking problem can be classified as either a kinematic tracking problem or a dynamic tracking problem. However, it is hard to obtain excellent tracking performance if just a kinematic model is considered, because of the existence of errors between the actual velocity and the output of the controller. It is more realistic to consider the tracking problem of both dynamic and kinematic models [8, 9] than that of the kinematic model, where only two control signals are needed for the robot under the kinematic constraints.

Several approaches have been proposed and numerous valuable results for dynamic trajectory tracking control problems have been obtained. The main control approaches include backstepping approach [10], neural network control [11, 12], sliding-mode control (SMC) [13], fuzzy algorithm [14], and adaptive approach [15, 16]. The SMC, a robust control technique which has many attractive features such as robustness to parameter variation and insensitivity to disturbance, is a suitable control method for NWMR tracking control [17]. In comparison with other robust control approaches mentioned above, it has a simple control structure and can provide fast dynamic responses. A number of applications can be found in the literature [18]. Considering the system uncertainties and external disturbances, a hierarchical sliding-mode underactuated controller [19] is proposed for the system with completely and partially known frictions and uncertainties. Focusing on the fact that the SMC method has high robustness, the tracking control for a NWMR with unknown system uncertainties was solved by the method [20, 21]. However, conventional sliding-mode controllers feature an obvious drawback where control torque is discontinuous and

oscillates at a high frequency, which cannot be tolerated in some practical applications.

To avoid this phenomenon, some methods have been proposed, such as equivalent control, boundary layer method [22], and intelligent SMC [23]. A second-order sliding-mode (SOSM) controller [24–26], which is effective in extending the performance and robustness of the standard sliding-mode controller, was also proposed to reduce the chattering effect [27, 28]. However, the system uncertainties and external disturbance are not completely considered and the bounds of uncertainties are assumed to be known, which would limit the wide application of the designed control method. Aiming at the above problems, the PD-SOSM controller was proposed to deal with the NWMR [29] with system uncertainties and disturbance. An adaptive second-order terminal sliding-mode controller [30] was proposed, and the bounds of unknown parameters were estimated using the adaptation law, but the estimate progress is rather complicated. Meanwhile, it is also important to note that most of the tracking problems treated in the literature have been mainly concerned with NWMR whose trajectory converges to the desired trajectory asymptotically. However, the tracking control achieved in finite time is required in some practical situations. It can be observed from the literature that the finite time is mainly achieved by the sign function of states [31] or relative states [32] for the nonlinear systems. Unlike most of the methods mentioned above, state-space partitioning [33, 34] is embedded into the switched second-order sliding-mode (S-SOSM) controllers, such that the trajectories can be driven onto the desired trajectory in finite time and do not require the knowledge of the bounds of the uncertainties. But it is mainly applied to the systems whose trajectories converge to a constant value of zeros instead of time-varying desired trajectories, which is a big limitation of the application. In this paper, the SOSM controller is adopted in the external loop, overcoming the disturbance and uncertainties without obtaining the upper bounds in advance.

Motivated by the idea [21, 33, 34], we further propose a combination of an integral sliding-mode controller and an S-SOSM controller. At first, an integral sliding-mode controller is applied to the system without uncertainties. Then, an S-SOSM algorithm is employed to conquer the unmodeled dynamics and external disturbances and guarantee system's precision and robustness. In the end, comparative simulation and experimental studies demonstrate the effectiveness and applicability of the proposed approach.

The paper is organized as follows. Section 2 gives the NWMR's kinematic and dynamic models and the SMC algorithm. In Section 3, the improved S-SOSM algorithm is proposed. In Section 4, the closed-loop stability and convergence are given to demonstrate the efficiency and the correctness of the improvement algorithm. In Section 5, the simulation and experimental results are provided, and the conclusions are given in Section 6.

2. SMC for NWMR

2.1. Description of NWMR Systems. The NWMR shown in Figure 1 is an example of a nonholonomic mechanical system

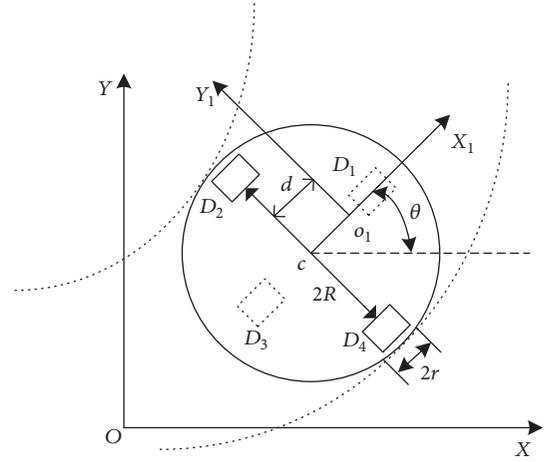


FIGURE 1: NWMR configuration.

consisting of a cart with two driving wheels (marked as D_2 and D_4) and two omnidirectional castors (marked as D_1 and D_3). It generally moves in the flat ground, so it is assumed that there is no gravitational term.

The dynamic and kinematic models of NWMR are adopted from [35]:

$$\overline{M}_0(q) \dot{\eta} + \overline{\tau}_d + \overline{F}(q, \dot{q}, \ddot{q}) = \overline{B}(q) \tau, \quad (1)$$

$$\dot{q} = S(q) \eta, \quad (2)$$

where $\eta(t) = [v_{o1}, w_{o1}]^T$ are the velocities of the robot, v_{o1} and w_{o1} denote the linear and angular velocities, $\overline{M}(q) = \overline{M}_0(q) + \Delta\overline{M}(q) \in \mathbb{R}^{2 \times 2}$ is a symmetric positive-definite inertia matrix, $\overline{\tau}_d \in \mathbb{R}^{2 \times 1}$ denotes bounded external disturbances, $\overline{F}(q, \dot{q}, \ddot{q}) = \Delta\overline{M}(q)\dot{\eta} \in \mathbb{R}^{2 \times 1}$ is the unmodeled dynamics, $\overline{B}(q) \in \mathbb{R}^{2 \times 2}$ is the vector of transformation matrix, and $q = [x, y, \theta]^T$ with associated velocities \dot{q} and acceleration \ddot{q} are controlled by applied torque $\tau \in \mathbb{R}^{2 \times 1}$. Moreover,

$$S(q) = \begin{bmatrix} \cos \theta & -d \sin \theta \\ \sin \theta & d \cos \theta \\ 0 & 1 \end{bmatrix},$$

$$\overline{M}_0(q) = \begin{bmatrix} m & 0 \\ 0 & I - md^2 \end{bmatrix}, \quad (3)$$

$$\overline{B}(q) = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ R & -R \end{bmatrix},$$

$$\tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix},$$

where m and I are the mass and inertia of the NWMR; d is the distance between points O_1 and c ; r and $2R$ represent the radius of the wheel and the distance of the two driving wheels, respectively.

2.2. Sliding-Mode Function and Controller Design. The nominal system is governed using the sliding-mode controller, which can effectively realize the tracking objective of the system without uncertainties.

The tracking errors are defined as

$$q_e = [e_1 \ e_2 \ e_3]^T = T\tilde{q}, \quad (4)$$

with

$$T = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

$$\tilde{q} = \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix},$$

where $q_r = [x_r, y_r, \theta_r]^T$ denotes the posture of the reference NWMR.

An auxiliary velocity control input η_c based on backstepping method [36] is given as

$$\eta_c(t) = \begin{bmatrix} v_c \\ w_c \end{bmatrix} = \begin{bmatrix} v_{o1r} \cos e_3 + k_1 e_1 \\ w_{o1r} + k_2 v_{o1r} e_2 + k_3 v_{o1r} \sin e_3 \end{bmatrix}, \quad (6)$$

where v_c is the speed tracking controller output of linear velocity and w_c is the angular velocity and $k_i > 0$, $i = 1, 2, 3$.

The integral sliding-mode surface is defined as

$$s(t) = e_c(t) + \lambda \int_0^t e_c(\tau) d\tau, \quad (7)$$

where $e_c(t) = \eta(t) - \eta_c(t)$ and λ is a positive surface integral constant.

If $e_c(t)$ moves to sliding-mode manifold and stays on it all the time, then it will be steered to the origin and the exponential stability is guaranteed. Considering $\dot{s}(t) = 0$ and using the model of the robot (1), the equivalent control law is designed as

$$\tau_{eq} = \left(\overline{M}_0^{-1}(q) \overline{B}(q) \right)^{-1} (\dot{\eta}_c(t) - \lambda e_c(t)). \quad (8)$$

The whole control input vector is assumed as follows:

$$\tau = \tau_{eq} + \tau_{sw}, \quad (9)$$

where τ_{sw} is designed by guaranteeing the global stability of the NWMR against the unmodeled dynamics and external disturbances.

Substituting (8) and (9) into system (1) results in the following closed-loop unknown subsystem dynamics equation:

$$\begin{aligned} \dot{\eta} &= \overline{M}_0^{-1}(q) \overline{B}(q) \tau_{sw} - \lambda e_c(t) \\ &\quad - \overline{M}_0^{-1}(q) (\overline{\tau}_d + \overline{F}(q, \dot{q}, \ddot{q})) + \dot{\eta}_c(t). \end{aligned} \quad (10)$$

3. S-SOSM Control

In order to effectively deal with the trajectory tracking control problem for NWMR with state-dependent unmodeled dynamics and time-varying external disturbances, the definition of intermediate variables z_1 on the sliding-mode surface is employed.

Consider the second-order uncertain nonlinear auxiliary system

$$\begin{aligned} \dot{z}_1 &= z_2, \\ \dot{z}_2 &= f(z(t)) + g(z(t))v + h(t), \end{aligned} \quad (11)$$

where $z_i = [z_{i1}, z_{i2}]$, $i = 1, 2$, $z = [z_1, z_2]^T \in \mathbb{R}^2$ are auxiliary system state variables. $z_1 = s(t)$ is the sliding variable. $f(z(t))$ and $g(z(t))$ are uncertain smooth functions that can be obtained from (7)–(10). v is the time derivative of τ_{sw} . $h(t)$ are time-varying external disturbances for which an upper bound H is known.

To achieve the tracking objective, we need the following assumptions.

Assumption 1. $|f(z(t))| \leq F$, $0 < G_1 \leq g(z(t)) \leq G_2$, where F , G_1 , and G_2 are state-dependent positive constants.

Assumption 2. The extremal value z_{Max} of z_1 can be detected.

Define the state-space partitioning as follows.

Assume that the system state space Z is divided into n regions R_i , $i = 1, 2, \dots, n$. The innermost region is defined as

$$R_i := \{(z_1, z_2) : |z_1| \leq \bar{z}_{1,i}, |z_2| \leq \bar{z}_{2,i}\}, \quad (12)$$

with $\bar{z}_{j,i+1} < \bar{z}_{j,i}$, $i = 1, 2, \dots, n-1$, $j = 1, 2$. The outermost region is defined as

$$R_1 := \{(z_1, z_2) : |z_1| > \bar{z}_{1,1}, |z_2| > \bar{z}_{2,1}\}, \quad (13)$$

where $Z_i = R_i/R_{i+1}$, $i = 1, 2, \dots, n-1$, $Z_n = R_n$. $W_i = \delta R_{i+1}$, $i = 1, 2, \dots, n-1$, denotes the edge of region R_{i+1} (see Figure 2).

Conclusion 3. Under Assumption 1 and the description of the state-space partitioning in (12) and (13), the bounds of the uncertainty terms $f(z(t))$ and $g(z(t))$ can be determined based on the fact that the state variables of each region are bounded. One can rewrite

$$|f(z(t))| \leq F_i, \quad (14)$$

$$0 < g_{1,i} \leq g(z(t)) \leq g_{2,i}, \quad i = 1, 2, \dots, n.$$

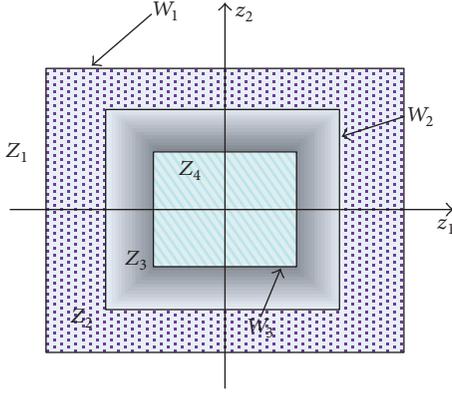


FIGURE 2: An example of the state-space partitioning.

Remark 4. In this paper, the state-space partitioning is employed to obtain the bounds of the uncertainty terms, so that the assumption of bound for the uncertain functions can be relaxed. Moreover, different from the traditional analysis methods such as Lyapunov functions [37, 38], it is difficult to build a Lyapunov function for the system, and it may be reconstructed if the system changes. It is noted that invariant sets are introduced in this paper, which is the key to prove that the trajectory converges to the desired trajectory. There is no need to build complex functions, and this will not change much with the system changes.

A set is expressed as

$$\bar{R}_i = Z_i \setminus \{\overline{P_{1,i}P_{2,i}} \cup \overline{P_{5,i}P_{6,i}}\}, \quad i = 1, 2, \dots, n, \quad (15)$$

where

$$\overline{P_{1,i}P_{2,i}} = \left\{ (z_1, z_2) \in W_i: z_1 \in \left[-z_{1,i} + \frac{z_2 |z_2|}{2\bar{V}_i}, z_2 = \bar{z}_{2,i} \right] \right\}. \quad (16)$$

$$\overline{P_{5,i}P_{6,i}} = \left\{ (z_1, z_2) \in W_i: z_1 \in \left[z_{1,i} - \frac{z_2 |z_2|}{2\bar{V}_i}, z_2 = -\bar{z}_{2,i} \right] \right\}.$$

If $z \in \bar{R}_i$, $i = 1, 2, \dots, n$, the control law v_i is defined as

$$v_i = -\alpha_i V_i \text{sign}(z_1 - \beta z_{\text{Max}}), \quad (17)$$

$$\alpha_i = \begin{cases} \alpha_i^* & (z_1 - \beta z_{\text{Max}})(z_{\text{Max}} - z_1) > 0 \\ 1 & \text{else,} \end{cases} \quad (18)$$

where V_i is the control gain for the region R_i , α_i is the modulation factor, and z_{Max} is a piecewise constant function ($z_{\text{Max}} = z_1(t_1)$, $z_2(t_1) = 0$), $\beta = 1/2$. The following constraints are satisfied to make the system trajectory converge to the desired trajectory in finite time:

$$\alpha_i^* \in (0 \ 1] \cap \left(0 \ \frac{3g_{1,i}}{g_{2,i}} \right),$$

$$V_i > \max \left\{ \frac{F_i + H}{\alpha_i^* g_{1,i}}, \frac{4(F_i + H)}{3g_{1,i} - \alpha_i^* g_{2,i}} \right\}, \quad (19)$$

$$V_i > V_{\text{Max}} = \max_{j=2,3,\dots,n} \left(\frac{F_j + H}{\alpha_j^* g_{1,j}} \right),$$

where $\bar{V}_i = \alpha_i^* g_{1,i} V_{\text{Max}} - F_i - H > 0$, $i = 1, 2, \dots, n$.

If $z \in \{\overline{P_{1,i}P_{2,i}} \cup \overline{P_{5,i}P_{6,i}}\}$, $i = 1, 2, \dots, n$, the control law v is defined as

$$v_i = -\alpha_i V_i \text{sign}(z_1 - \beta z_{\text{Max}}), \quad (20)$$

$$v_{i-1} = \alpha_i V_i \text{sign}(z_1 - \beta z_{\text{Max}}), \quad (21)$$

where α_i is chosen as (18).

Inserting (8) and (17)–(21) into (9), (9) can be rewritten as

$$\tau_i = \left(\bar{M}_0^{-1}(q) \bar{B}(q) \right)^{-1} (\dot{\eta}_c(t) - \lambda e_c(t)) + \int v_i dt. \quad (22)$$

Remark 5. The S-SOSM controller is applied into the design of the switching controller τ_{sw} . The discontinuous sign function is replaced by a continuous function, so the unexpected chattering can be attenuated significantly.

Remark 6. Different control objectives in the different regions could be accomplished, for the curvature of the trajectory can be adjusted, for example, a short settling time or a minimal overshoot. There is no need to design different control strategies for the various control objects in different environments; for example, the UAV needs a minimal overshoot before reaching the target position and needs a short settling time when hovering in the target position.

4. Convergence Analysis

Theorem 7. For the second-order uncertain nonlinear auxiliary system, under the proposed controller, the following contraction property is featured:

$$|z_{\text{Max},i+1}| < |z_{\text{Max},i}|, \quad i = 1, 2, \dots, n-1. \quad (23)$$

Proof. Assume that $z_1(0) = z_{\text{Max}}$, $z_2(0) = 0$, and the initial point lies on the positive direction of the abscissa with the control law $v_i = -\alpha_i V_i$. When the commutation takes place at time instant t_2 such that $z_1(t_2) = z_{\text{Max}}/2$, the value of $z_2(t_2)$ becomes within the interval

$$z_2(t_2) \in \left[-\sqrt{z_{\text{Max}}(\alpha_i^* g_{2,i} V_i + F_i + H)}, -\sqrt{z_{\text{Max}}(\alpha_i^* g_{1,i} V_i - F_i - H)} \right], \quad (24)$$

and when $t > t_2$, the auxiliary system with control law $v_i = V_i$ shows that the trajectories intersect with the axis, and the intersection point belongs to the interval

$$\left[-\frac{(\alpha_i^* g_{2,i} - g_{1,i}) V_i + 2(F_i + H)}{2(g_{1,i} V_i - F_i - H)} z_{\text{Max}}, \frac{(g_{2,i} - \alpha_i^* g_{1,i}) V_i + 2(F_i + H)}{2(g_{2,i} V_i + F_i + H)} z_{\text{Max}} \right]. \quad (25)$$

It is obvious that the right extreme of the interval is closer to the origin than the last intersection. Then, the sufficient condition for satisfying the contraction property is that the magnitude of the left extreme of the interval is less than z_{Max} , which can be denoted by the following system of inequalities:

$$\begin{aligned} \alpha_i^* &\leq 1, \\ \alpha_i^* g_{1,i} V_i &\geq F_i + H, \\ \frac{((\alpha_i^* g_{2,i} - g_{1,i}) V_i + 2(F_i + H))}{(g_{1,i} V_i - F_i - H)} &\leq 2. \end{aligned} \quad (26)$$

Finally, it can be obtained that

$$V_i > \max \left\{ \frac{F_i + H}{\alpha_i^* g_{1,i}}, \frac{4(F_i + H)}{3g_{1,i} - \alpha_i^* g_{2,i}} \right\}. \quad (27)$$

The proof process is the same when the initial point is at other locations.

For the second-order uncertain nonlinear auxiliary system, the control law v steers the state variables to the origin.

Define

$$\bar{R}_i = Z_i \setminus \{U_{1,i} \cup U_{2,i}\}, \quad i = 1, 2, \dots, n, \quad (28)$$

where

$$\begin{aligned} U_{1,i} &= \left\{ (z_1, z_2) : z_1 \in \left[z_{1,i} - \frac{z_2 |z_2|}{2\bar{V}_i}, z_2 > 0 \right] \right\}, \\ U_{2,i} &= \left\{ (z_1, z_2) : z_1 \in \left[-z_{1,i} - \frac{z_2 |z_2|}{2\bar{V}_i}, z_2 < 0 \right] \right\}. \end{aligned} \quad (29)$$

□

Proof. Region \bar{R}_i is the largest positively invariant set contained in R_i . The proof of this theorem can be found in [39]. □

Theorem 8. Consider the state-space partitioning (12) ad (13) and the region definition (17) and assume that the bounds (15) hold. Under the control of the proposed method, the auxiliary system states converge to the origin in finite time.

Proof. The time sequence $\{t_{\text{Max}_i}\}$ is the time instant at which an extremal value of z_1 occurs, where

$$\begin{aligned} t_{\text{Max}_{i+1}} &= t_{\text{Max}_i} \\ &+ \frac{(\alpha_i^* g_{2,i} + g_{1,i}) V_i}{(g_{1,i} V_i - F_i) \sqrt{(\alpha_i^* g_{2,i} V_i + F_i)}} \sqrt{z_{\text{Max}_i}}, \end{aligned} \quad (30)$$

$i = 1, 2, \dots, n.$

By means of simple computations, the following inequality holds:

$$\begin{aligned} t_{\text{Max}_{i+1}} &\leq t_{\text{Max}_i} \\ &+ \max_{j=2,3,\dots,n} \left(\frac{3\sqrt{2}}{2\sqrt{g_{1,i} V_i - F_i - H}} \right) \sum_{i=1}^n \sqrt{z_{\text{Max}_i}} \\ &= \eta \sum_{i=1}^n \sqrt{z_{\text{Max}_i}} + t_{\text{Max}_1}. \end{aligned} \quad (31)$$

Then, inserting (26) into (23) yields

$$|z_{\text{Max}_i}| \leq \left| -\frac{(\alpha_i^* g_{2,i} - g_{1,i}) V_i + 2(F_i + H)}{2(g_{1,i} V_i - F_i - H)} \right|^{i-1} |z_{\text{Max}_1}|. \quad (32)$$

From (31) and (32), it can be obtained that

$$\begin{aligned} t_{\text{Max}_{i+1}} &\leq \eta \sum_{i=1}^n \left| -\frac{(\alpha_i^* g_{2,i} - g_{1,i}) V_i + 2(F_i + H)}{2(g_{1,i} V_i - F_i - H)} \right|^{i-1} \sqrt{|z_{\text{Max}_1}|} \\ &+ t_{\text{Max}_1} \\ &= \eta' \sum_{i=1}^n \left| -\frac{(\alpha_i^* g_{2,i} - g_{1,i}) V_i + 2(F_i + H)}{2(g_{1,i} V_i - F_i - H)} \right|^{i-1} + t_{\text{Max}_1}. \end{aligned} \quad (33)$$

Obviously, $|-(\alpha_i^* g_{2,i} - g_{1,i}) V_i + 2(F_i + H)| / 2(g_{1,i} V_i - F_i - H) < 1$. Therefore, according to (31), there is

$$\lim_{i \rightarrow \infty} z_{\text{Max}_i} = 0. \quad (34)$$

Finally,

$$\begin{aligned} \lim_{i \rightarrow \infty} t_{\text{Max}_i} &< \frac{\eta'}{1 - |(\alpha_i^* g_{2,i} - g_{1,i}) V_i + 2(F_i + H)| / 2(g_{1,i} V_i - F_i - H)} \\ &+ t_{\text{Max}_1}. \end{aligned} \quad (35)$$

The convergence of the time sequence $\{z_{\text{Max}_i}\}$ in finite time implies the convergence to zero of the auxiliary system. Then, the NWMR system will be steered to the desired trajectory in finite time. Here, the proof is complete. □

5. Simulations and Experimental Results

In this section, simulation and experiment studies for tracking control of a NWMR system are conducted to verify the effectiveness of the proposed control laws. The numerical values of the parameters are listed in Table 1.

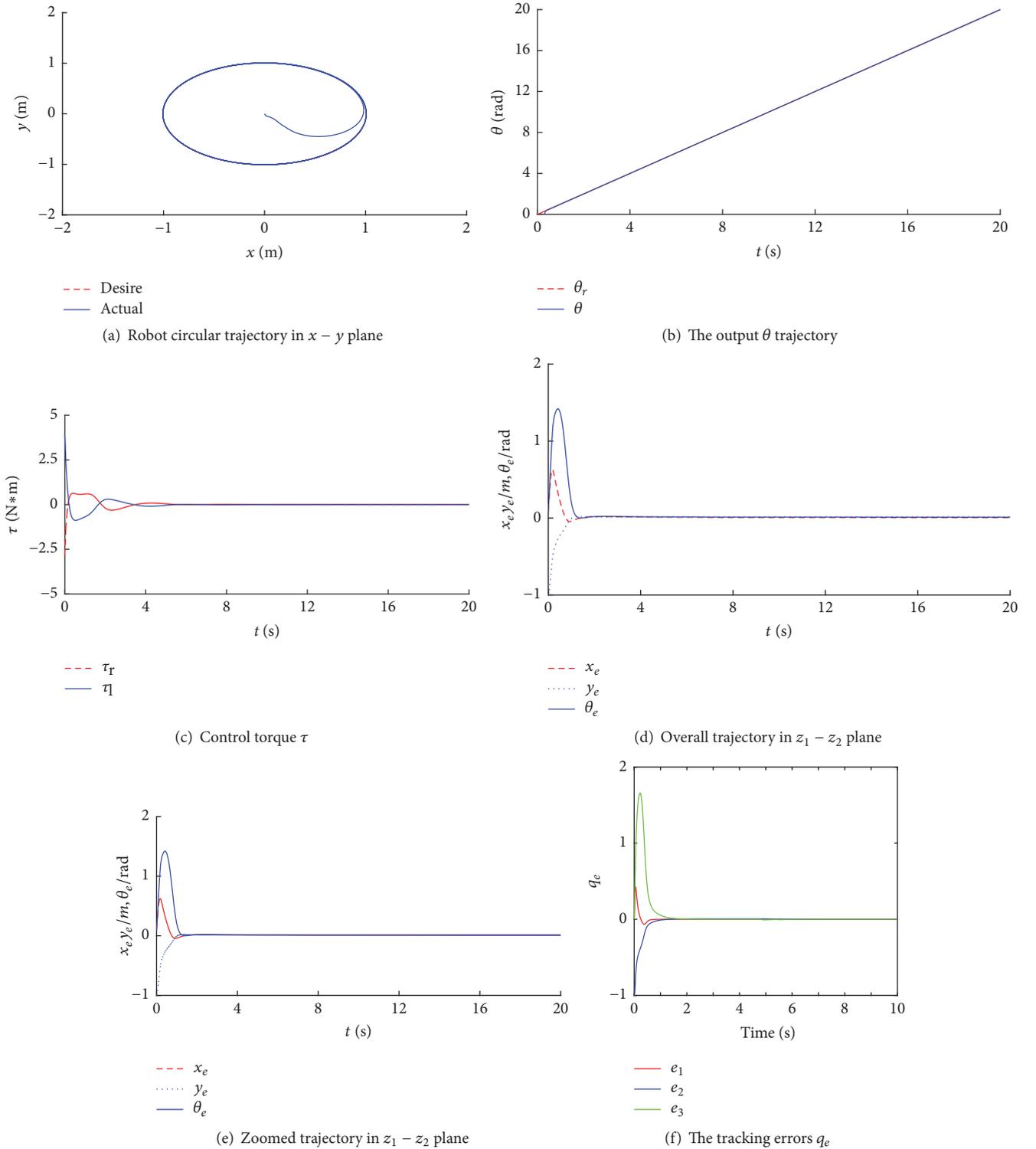


FIGURE 3: Simulation results of the mobile robot with the proposed controller.

5.1. Simulations Results. The tracking controller is compared with the traditional sliding-mode controller, and the adjustment of α_i in Remark 6 is verified.

Compared with Figures 3(a) and 3(b) and Figures 4(a) and 4(b), one can see that both trajectories can track the

desired trajectory. However, it can be observed from Figure 3(f) that the tracking errors decreased to zero, which cost 1.5s less than in Figure 4(d), which indicates that the proposed method has a better response with a faster convergence speed. In Figure 3(c), the torque control input

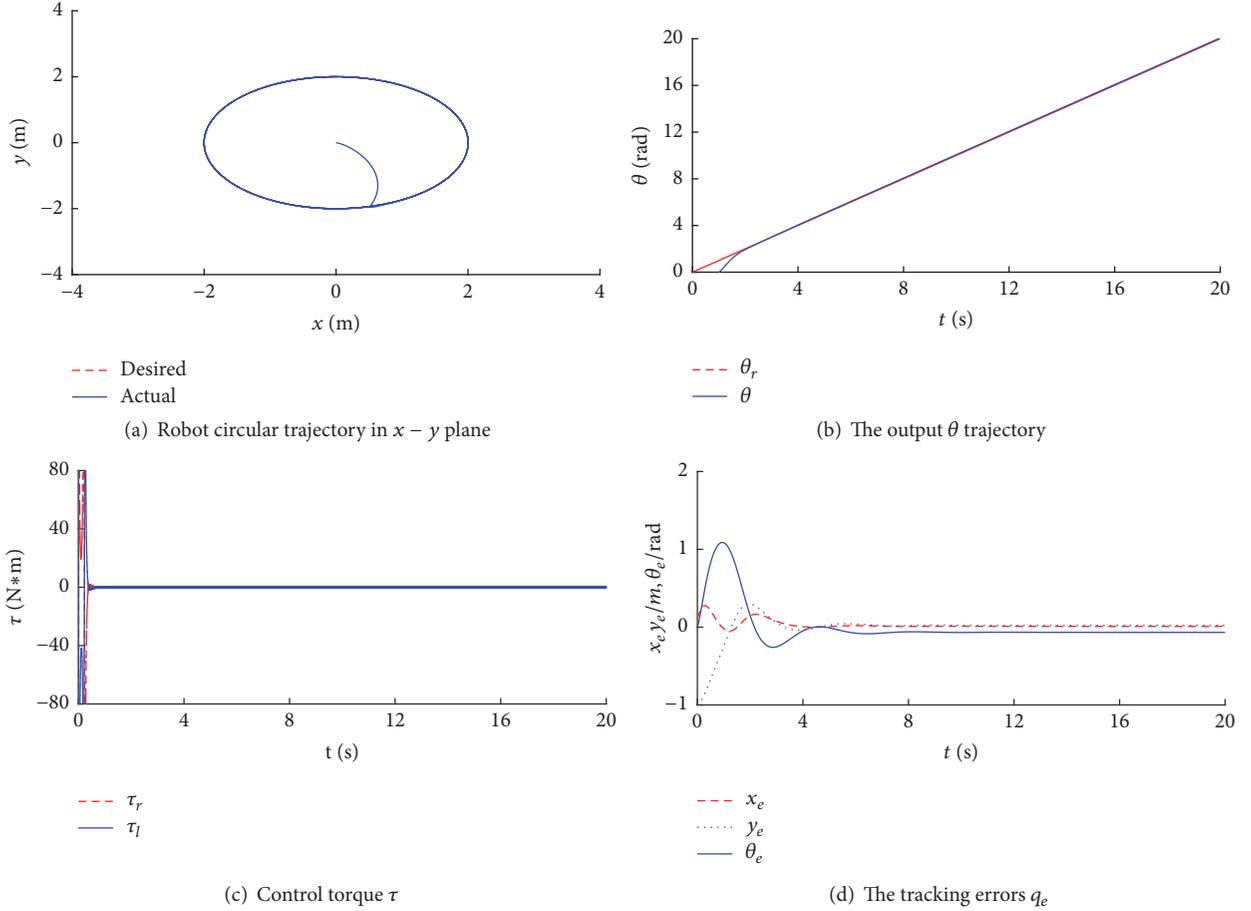


FIGURE 4: Simulation results of the mobile robot with the SMC controller.

TABLE 1: Parameters used for the simulation.

$m = 2.92$ kg	$\alpha^* = [0.65, 0.30, 0.8]^T$
$d = 0.11$ m	$[k_1, k_2, k_3] = [3, 120, 20]$
$I = 2.5$ kg m ²	$z_1 = [70, 50, 5; 70, 50, 5]^T$
$R = 0.17$ m	$z_2 = [70, 50, 5; 70, 50, 5]^T$
$r = 0.03$ m	$[v_{o1r}, \omega_{o1r}] = [1.0 \text{ m/s}, 1.0 \text{ rad/s}]$
$\lambda = 6.9$	$V = [5.0, 1.80, 1.0; 5.0, 0.81, 0.3]^T$
$H = 0.3$	$\bar{\tau}_d + \bar{F}(q, \dot{q}, \ddot{q}) = [0.1 * \eta + 0.1 \sin(t), 0.2 * \dot{\eta} + 0.1 \cos(t)]^T$

τ of the proposed control can be confined to an arbitrarily small repose, and the chattering is well attenuated compared with that in Figure 4(c).

5.2. Experimental Results. Experiments are conducted to test the effectiveness of the proposed control laws in a practical NWMR system. QUANSER QBot2 as shown in Figure 5(a) is used for the experiments. Figure 5(b) shows the diagram of experimental platform. The robots move on a 300×360 cm white surface. In the experiment, the robot starts at $x = 0$ m, $y = 0$ m, and $\theta = 0$ rad/s and should follow a circle trajectory of reference. The maximum velocity of QUANSER QBot2 is $v = 0.7$ m/s, so the reference velocities in experiment are

chosen as $v_{o1r} = 0.3$ m/s and $\omega_{o1r} = 0.25$ rad/s and the radius of the circle is 1.2 m.

The experimental results shown in Figure 6 illustrate the trajectory tracking result for a circle trajectory. The actual trajectory reaches the desired circle quickly. According to the experimental results for the circle trajectory tracking, the trajectory can remain stable and robust despite external disturbances and unmodeled dynamics in the robot.

In summary, both simulation studies and the experiment results demonstrate that the proposed control laws are effective.

6. Conclusion

An innovative robust controller has been investigated for dealing with the tracking problem of NWMR. Finite-time convergence of tracking error was achieved in the presence of external disturbances and unmodeled dynamics. Comparing simulation and experimental studies, the effectiveness and the superior performance of the proposed control method are verified. There are several features that are worth pointing out. (1) The assumption of bound for the uncertain functions can be relaxed. (2) The NWMR can accommodate different control objectives under different requirements. (3) The chattering problem is effectively alleviated and has a better

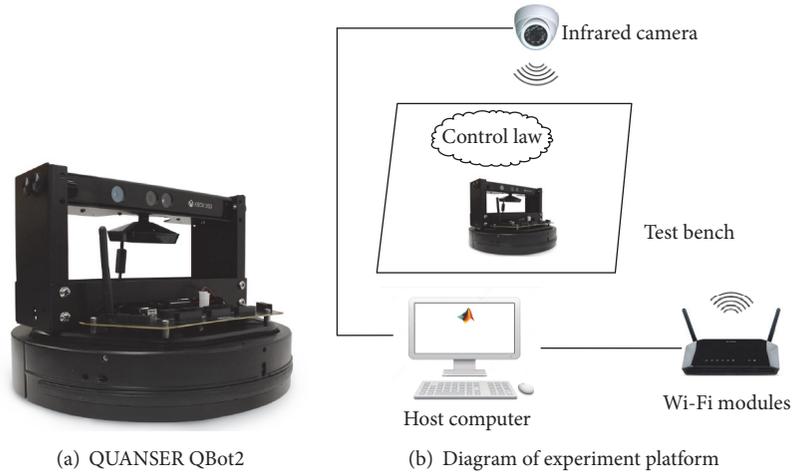


FIGURE 5: Experimental environment.

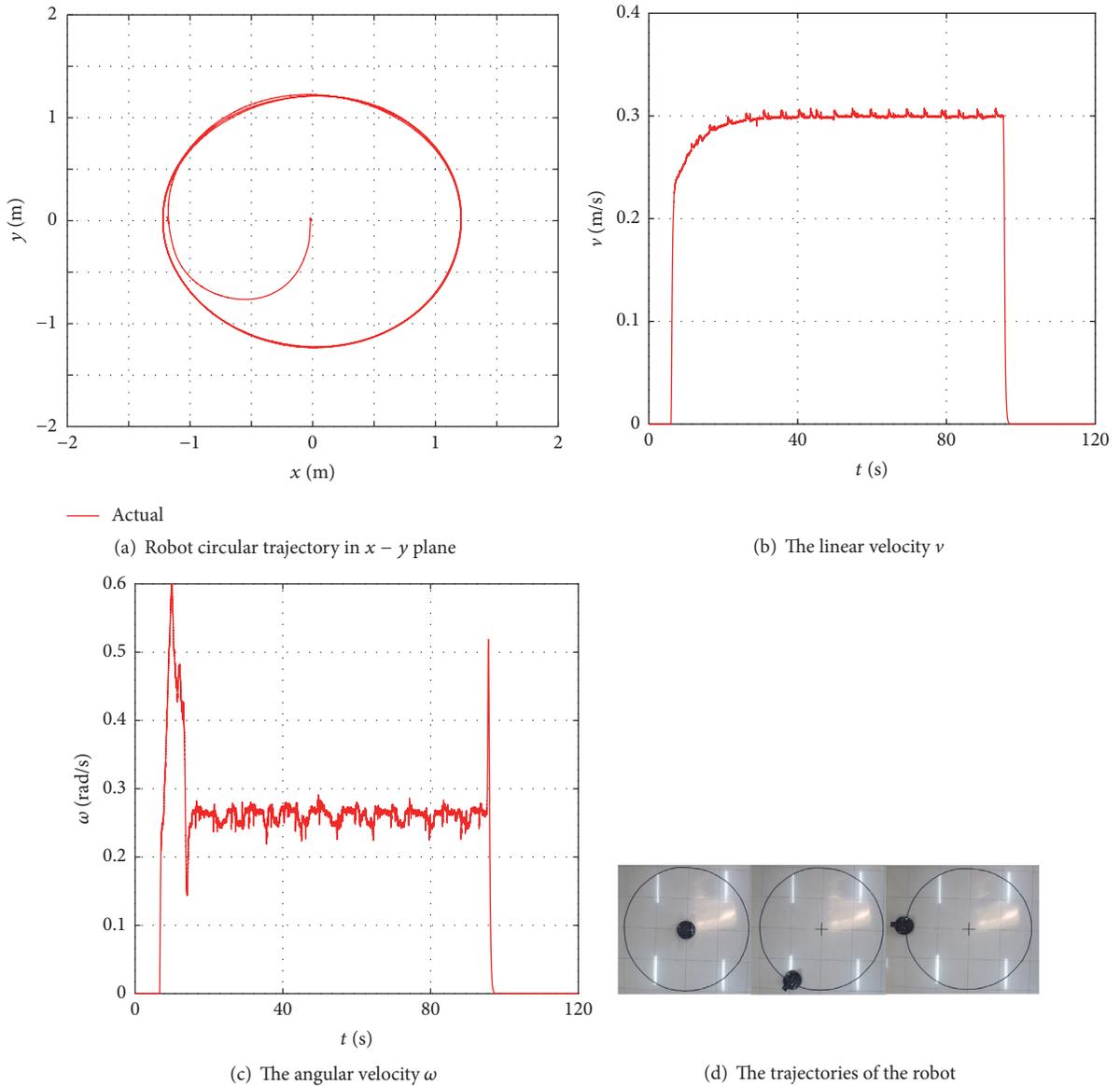


FIGURE 6: Experiment results of the mobile robot with the proposed controller.

response with a faster convergence speed. In future work, the assumption of the bound of disturbance should be relaxed so that this control algorithm can be applied to more nonlinear systems.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the NSFC under Grant 61473248 and by the Natural Science Foundation of Hebei Province under Grant F2016203496.

References

- [1] K.-D. Nguyen and H. Dankowicz, "Adaptive control of under-actuated robots with unmodeled dynamics," *Robotics and Autonomous Systems*, vol. 64, pp. 84–99, 2015.
- [2] T.-C. Lee, "Exponential stabilization for nonlinear systems with applications to nonholonomic systems," *Automatica*, vol. 39, no. 6, pp. 1045–1051, 2003.
- [3] K. Worthmann, M. W. Mehrez, M. Zanon, G. K. I. Mann, R. G. Gosine, and M. Diehl, "Model predictive control of nonholonomic mobile robots without stabilizing constraints and costs," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1394–1406, 2016.
- [4] S. Yildirim and S. Savas, "Tracking control of a nonholonomic mobile robot using neural network," in *Handbook of Research on Advancements in Robotics and Mechatronics*, pp. 631–661, 2014.
- [5] A. Al-Mayyahi, W. Wang, and P. Birch, "Design of fractional-order controller for trajectory tracking control of a non-holonomic autonomous ground vehicle," *Journal of Control, Automation and Electrical Systems*, vol. 27, no. 1, pp. 29–42, 2016.
- [6] Z. Peng, G. Wen, S. Yang, and A. Rahmani, "Distributed consensus-based formation control for nonholonomic wheeled mobile robots using adaptive neural network," *Nonlinear Dynamics*, vol. 86, no. 1, pp. 605–622, 2016.
- [7] S. Yang, Y. Cao, Z. Peng, G. Wen, and K. Guo, "Distributed formation control of nonholonomic autonomous vehicle via RBF neural network," *Mechanical Systems and Signal Processing*, vol. 87, part B, pp. 81–95, 2017.
- [8] M. Boukens and A. Boukabou, "Design of an intelligent optimal neural network-based tracking controller for nonholonomic mobile robot systems," *Neurocomputing*, vol. 226, pp. 46–57, 2017.
- [9] Z. Peng, S. Yang, G. Wen, A. Rahmani, and Y. Yu, "Adaptive distributed formation control for multiple nonholonomic wheeled mobile robots," *Neurocomputing*, vol. 173, pp. 1485–1494, 2016.
- [10] W. Liu, C. Lim, P. Shi, and S. Xu, "Backstepping fuzzy adaptive control for a class of quantized nonlinear systems," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1090–1101, 2017.
- [11] J. Ye, "Tracking control for nonholonomic mobile robots: Integrating the analog neural network into the backstepping technique," *Neurocomputing*, vol. 71, no. 16–18, pp. 3373–3378, 2008.
- [12] B. Chen, H. Zhang, and C. Lin, "Observer-based adaptive neural network control for nonlinear systems in nonstrict-feedback form," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 1, pp. 89–98, 2016.
- [13] V. Goyal, V. Deolia K, and T. Sharma, "Neural network based sliding mode control for uncertain discrete-time nonlinear systems with time-varying delay," *International Journal of Computational Intelligence Research*, vol. 12, no. 2, pp. 125–138, 2016.
- [14] G. Lai, Z. Liu, Y. Zhang, X. Chen, and C. L. Chen, "Robust adaptive fuzzy control of nonlinear systems with unknown and time-varying saturation," *Asian Journal of Control*, vol. 17, no. 3, pp. 791–805, 2015.
- [15] H. Gao, X. Song, L. Ding, K. Xia, N. Li, and Z. Deng, "Adaptive motion control of wheeled mobile robot with unknown slippage," *International Journal of Control*, vol. 87, no. 8, pp. 1513–1522, 2014.
- [16] J. Huang, C. Wen, W. Wang, and Z.-P. Jiang, "Adaptive output feedback tracking control of a nonholonomic mobile robot," *Automatica*, vol. 50, no. 3, pp. 821–831, 2014.
- [17] M. Sharifi and H. Moradi, "Nonlinear robust adaptive sliding mode control of influenza epidemic in the presence of uncertainty," *Journal of Process Control*, vol. 56, pp. 48–57, 2017.
- [18] Q. H. Ngo, N. P. Nguyen, C. N. Nguyen, T. H. Tran, and Q. P. Ha, "Fuzzy sliding mode control of an offshore container crane," *Ocean Engineering*, vol. 140, pp. 125–134, 2017.
- [19] C.-L. Hwang and H.-M. Wu, "Trajectory tracking of a mobile robot with frictions and uncertainties using hierarchical sliding-mode under-actuated control," *IET Control Theory & Applications*, vol. 7, no. 7, pp. 952–965, 2013.
- [20] J. X. Xu, Z. Q. Guo, and T. H. Lee, "Design and implementation of integral sliding-mode control on an underactuated two-wheeled mobile robot," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3671–3681, 2014.
- [21] L. Xin, Q. Wang, J. She, and Y. Li, "Robust adaptive tracking control of wheeled mobile robot," *Robotics and Autonomous Systems*, vol. 78, pp. 36–48, 2016.
- [22] Y.-J. Huang, T.-C. Kuo, and S.-H. Chang, "Adaptive sliding-mode control for nonlinear systems with uncertain parameters," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 2, pp. 534–539, 2008.
- [23] I. F. Jasim, P. W. Plapper, and H. Voos, "Adaptive sliding mode fuzzy control for unknown robots with arbitrarily-switched constraints," *Mechatronics*, vol. 30, pp. 174–186, 2015.
- [24] D. Shen, Z. Sun, and Y. Qiao, "Second-order sliding mode control for nonholonomic mobile robots formation," in *Proceedings of the 30th Chinese Control Conference, CCC 2011*, pp. 4860–4864, chn, July 2011.
- [25] Y. Kali, K. Benjelloun, A. Fatemi, M. Saad, and M. Benbrahim, "Second order sliding mode with time delay control for uncertain robot manipulators," in *Proceedings of the 2016 IEEE International Multidisciplinary Conference on Engineering Technology, IMCET 2016*, pp. 160–165, Lebanon, November 2016.
- [26] X. Huang, Y. Yan, and Y. Zhou, "Neural network-based adaptive second order sliding mode control of Lorentz-augmented spacecraft formation," *Neurocomputing*, vol. 222, pp. 191–203, 2017.
- [27] Y. Huangfu, R. Ma, and A. Miraoui, "Avoidance high-frequency chattering second-order sliding mode controller design: buck converter in wind power system," *International Journal of Antennas and Propagation*, vol. 2012, Article ID 176830, 5 pages, 2012.
- [28] G. Bartolini, A. Ferrara, L. Giacomini, and E. Usai, "Properties of a combined adaptive/second-order sliding mode control algorithm for some classes of uncertain nonlinear systems," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 45, no. 7, pp. 1334–1341, 2000.

- [29] E. S. El Youssef, N. A. Martins, E. R. De Pieri, and U. F. Moreno, "PD-super-twisting second order sliding mode tracking control for a nonholonomic wheeled mobile robot," in *Proceedings of the 19th IFAC World Congress on International Federation of Automatic Control, IFAC 2014*, pp. 3827–3832, zaf, August 2014.
- [30] S. Mondal and C. Mahanta, "Adaptive second order terminal sliding mode controller for robotic manipulators," *Journal of The Franklin Institute*, vol. 351, no. 4, pp. 2356–2377, 2014.
- [31] Y. Zhao, Y. Sheng, and X. Liu, "A novel finite time sliding mode control for robotic manipulators," in *Proceedings of the 19th IFAC World Congress on International Federation of Automatic Control, IFAC 2014*, vol. 47, pp. 7336–7341, zaf, August 2014.
- [32] S. Matoba, N. Nakamura, H. Nakamura, and H. Nishitani, "Robust finite-time control of robot manipulators," in *Proceedings of the 18th IFAC World Congress*, vol. 44, pp. 11863–11868, Italy, September 2011.
- [33] M. Tanelli and A. Ferrara, "Enhancing robustness and performance via switched second order sliding mode control," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 58, no. 4, pp. 962–974, 2013.
- [34] A. Pisano, M. Tanelli, and A. Ferrara, "Switched/time-based adaptation for second-order sliding mode control," *Automatica*, vol. 64, pp. 126–132, 2016.
- [35] A. Saradagi, V. Muralidharan, V. Krishnan, S. Menta, and A. D. Mahindrakar, "Formation control and trajectory tracking of nonholonomic mobile robots," *IEEE Transactions on Control Systems Technology*, vol. 99, pp. 1–9, 2017.
- [36] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 384–389, Washington, DC, USA, May 1990.
- [37] B. Sharma, J. Vanualailai, and S. Singh, "Tunnel passing maneuvers of prescribed formations," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 5, pp. 876–901, 2014.
- [38] B. Sharma, J. Vanualailai, and S. Singh, "Motion planning and posture control of multiple n-link doubly nonholonomic manipulators," *Robotica*, vol. 35, no. 1, pp. 1–25, 2017.
- [39] M. Rubagotti and A. Ferrara, "Second order sliding mode control of a perturbed double integrator with state constraints," in *Proceedings of the 2010 American Control Conference, ACC 2010*, pp. 985–990, usa, July 2010.

Research Article

Event-Based Time Varying Formation Control for Multiple Quadrotor UAVs with Markovian Switching Topologies

Zhen Zhou , Hongbin Wang , and Zhongquan Hu

Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China

Correspondence should be addressed to Zhen Zhou; zhouzhen0617@163.com

Received 22 December 2017; Accepted 20 February 2018; Published 23 April 2018

Academic Editor: Victor M. Becerra

Copyright © 2018 Zhen Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Time varying formation control problem for a group of quadrotor unmanned aerial vehicles (UAVs) under Markovian switching topologies is investigated through a modified dynamic event-triggered control protocol. The formation shape is specified by a time varying vector, which prescribes the relative positions and bearings among the whole agents. Instead of the general stochastic topology, the graph is governed by a set of Markov chains to the edges, which can recover the traditional Markovian switching topologies in line with the practical communication network. The stability proof for the state space origin of the overall closed-loop system is derived from the singular perturbation method and Lyapunov stability theory. An event-triggered formation control protocol in terms of a dynamically varying threshold parameter is delicately carried out, while acquiring satisfactory resource efficiency, and Zeno behavior of triggering time sequences is excluded. Finally, simulations on six quadrotor UAVs are given to verify the effectiveness of the theoretical results.

1. Introduction

Along with the increasing applications in various areas, such as aerial photography, express delivery, and disaster relief, formation control of multiagent systems has attracted considerable attention from many researchers [1, 2]. In particular, as a typical class of physical systems with practical interest, the quadrotor UAVs is widely used in the military and civilian fields [3, 4]. Actually, due to the strong nonlinear coupling and limited communication resources [5], the control problem of multiple quadrotor UAVs will be very challenging and difficult [6]. Therefore, how to design the formation control protocol for multiple quadrotor UAVs subject to limited communication resources becomes a significant research focus.

A defining feature of formation control problem is that multiple agents work together to accomplish a collaborative formation task [7]. Several classic formation control strategies, including leader-follower, virtual structure, and behavior based methods, were applied in the scientific community [8, 9]. For example, formation control of multiple quadrotor UAVs, based on position estimation [10], backstepping design technique [11], and finite time algorithms [12], respectively, was investigated so as to make a construct

and keep the formation shape during flying. It should be pointed out that time varying formation tracking problems arise in some scenarios, such as source seeking and target enclosing. For example, time varying formation analysis and design problems for multiagent systems with switching topologies were solved in [13, 14]. Based on the fact that multiagent systems subject to random abrupt variations could be modeled as the switching systems, then some results have been obtained on it [14]. Compared with the previous works, time varying formation control results for multiagent systems with switching topologies were provided in [15, 16]. Besides, due to random link failures, variation meeting the need and sudden environmental disturbances [17], some dynamical systems could be modeled as Markovian switching systems, which were governed by a set of Markov chains [18, 19]. By considering the complex network as Markovian switching topologies, it plays a crucial role in the field of networked control system [20].

In practice, under a limited bandwidth, it is necessary and important to consider the issues of energy waste and competition [21]. Therefore, event-triggered communication mechanism was born at the right moment [22, 23]. As a popular research topic, some latest event-triggered control

results were provided in [24–28]. In particular, compared with the general event-triggered controller with a fixed threshold parameter, the authors in [24, 29] developed the dynamic/adaptive event-triggered control protocol of multi-agent systems for acquiring satisfactory resource efficiency, respectively. Meanwhile, take the strong nonlinear coupling and underactuated of the quadrotor UAVs into account, time scaling based control method has also been recognized as a powerful tool in the analysis and design of controllers, which is with crucial importance in applications to the mobile inverted pendulum [30], the ball-beam system [31], and the quadrotor UAVs [32]. Therefore, it is of great importance to extend the event-triggered formation results to multiple quadrotor UAVs under Markovian switching topologies. In addition, it is difficult to obtain all the transition rates under the realistic communication environment [33]. So that randomly occurring control strategy is more realistic and meaningful to accomplish attitude stabilization and formation missions under the limited communication resources.

Motivated by these observations, the contributions of this paper are proposing a novel time varying formation control strategy and an event-triggered communication scheme to solve the formation problem of multiple quadrotor UAVs with Markovian switching topologies. The main highlights of this paper are summarized as follows. First, a modified graph of entire system is governed by a set of Markov chains to the edges, and the traditional Markovian switching topologies can be recovered through adjusting the modes of edges and the transition rates. Second, the dynamic event-triggered controller is derived from a time scaling based control strategy, which consists of two parts: the closed-loop system stability analysis based on the framework of singularly perturbed theory and the event-triggered control scheme in terms of a new dynamically varying threshold parameter to guarantee time varying formation shape. Third, Markovian switching topologies involve partly unknown transition rates, which are of great importance to be considered and thus closer to the realistic communication environment. In addition, Zeno behavior can be excluded during the whole running process. Finally, several simulations can illustrate the theoretical results.

The rest of this paper is organized as follows. The system dynamics and some preliminaries on graph theory are introduced in Section 2; Section 3 provides main results on event-triggered formation control for multiple quadrotor UAVs. In Section 4, simulation results are given and this paper is concluded in Section 5.

Notations 1. Throughout this paper, $\|\bullet\|$ denotes L_2 -vector norm and $\rho(\bullet)$ stands for the spectral radius for matrices. The notation $A \otimes B$ means the Kronecker product of matrices A and B , and $\lambda_{\max}(\bullet)$ and $\lambda_{\min}(\bullet)$ represent its maximum and minimum eigenvalues.

2. Preliminaries and System Formulation

2.1. Graph Theory. Define a time varying random undirected graph $G(\omega(t)) = (\nu, \xi(\omega_{ij}(t)), A(\omega(t)))$ with a nonempty finite vertex set $\nu = \{\nu_1, \dots, \nu_N\}$ and an edge set $\xi(\omega_{ij}(t)) \subseteq \nu \times \nu$.

Different with the general ones, it consists of a time sequence of random graphs in which the edge set ξ varies with t . Namely, each edge (ν_i, ν_j) evolves according to a homogeneous Markov process $\omega_{ij}(t)$, which takes values in $S = \{1, 2, \dots, s\}$ with the transition rate as

$$\Pr \left\{ \omega_{ij}(t + \Delta) = q \mid \omega_{ij}(t) = p \right\} = \begin{cases} \pi_{pq}\Delta + o(\Delta) & q \neq p \\ 1 + \pi_{pp}\Delta + o(\Delta) & q = p. \end{cases} \quad (1)$$

Assume that $\omega_{ij}(t)$ do not change infinitely fast; thus, $\omega_{ij}(t) = \omega_{ij}(t + \Delta_t)$ if $0 < \Delta_t < \Delta$. It means that the total number of system modes is $s^{(N+1)N/2}$ and the total transition rate is given by

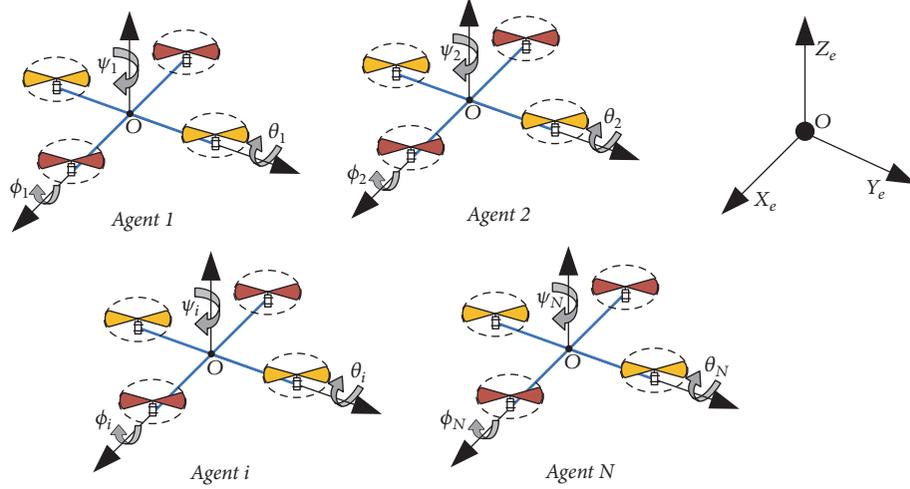
$$\Pr \{ \omega(t + \Delta) \mid \omega(t) \} = \prod_{i,j \in \nu} \Pr \left\{ \omega_{ij}(t + \Delta) = q \mid \omega_{ij}(t) = p \right\}. \quad (2)$$

The weighted adjacency matrix $A^\omega \triangleq A(\omega(t)) = [a_{ij}(\omega_{ij}(t))] \in \mathbb{R}^{N \times N}$ is associated with $G(\omega(t))$. Here $a_{ij}^\omega \triangleq a_{ij}(\omega_{ij}(t)) > 0$ if $(\nu_i, \nu_j) \in \xi(\omega_{ij}(t))$ and $a_{ij}^\omega = 0$ otherwise. Assumed that there is no self-loop in the graph, which implies that $a_{ii}^\omega = 0$. In this paper, the set of neighbors with respect to the agent ν_i is $\Omega_i^\omega = \{\nu_j \in \nu \mid (\nu_j, \nu_i) \in \xi(\omega_{ij}(t))\}$. A graph $G(\omega(t))$ is connected, if there is a path between any two vertices; otherwise, it is disconnected. A diagonal matrix $D^\omega = \text{diag}\{d_i\} \in \mathbb{R}^{N \times N}$ with $d_i = \sum_{j \in \Omega_i^\omega} a_{ij}^\omega$ being the i th row sum of A^ω . Then, the Laplacian of the graph is defined as $L^\omega = D^\omega - A^\omega$. Consider the formation with a leader-follower structure by introducing a diagonal matrix $L^l = \text{diag}\{a_{ii}^l\} \in \mathbb{R}^{N \times N}$, which evolves according to the Markov process $\omega_0(t)$ with a finite mode set $S_0 = \{1, 2, \dots, s_0\}$ and a time interval Δ_0 , where $a_{ii}^l > 0$ if ν_i is a leader and $a_{ii}^l = 0$ otherwise. Hence, the interaction matrix is given by $L^G = L^\omega + L^l$.

Assumption 2. The undirected graph $G(\omega(t))$ is connected.

Remark 3. For simplicity, we just consider the undirected graph in this paper, that is, $a_{ij}^\omega = a_{ji}^\omega$. Note that, if the graph is a general directed one, there will be some small differences. A possible approach to consider a directed graph is to introduce two Markov chains in each two agents; an alternative is to extend the state space set S , which could be defined according to the weight and direction of the graph. Both approaches will be addressed in our future work.

Remark 4. With a limited bandwidth, the switching of a_{ij}^ω is caused by the sensing/detecting failure and communication failure, which is passive. In fact, it is difficult to obtain all the elements of the transition rate matrix, or some of the elements are not necessary to guarantee the system stability. Since that, the transition rate matrices are assumed to be partly accessed;

FIGURE 1: Group of the N quadrotor UAVs.

even some of them are unknown completely, which could be described as follows:

$$(3) \quad \begin{pmatrix} P_{11} & P_{12} & \cdots & ? \\ ? & ? & \cdots & P_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ P_{s1} & ? & \cdots & P_{ss} \end{pmatrix},$$

$$\begin{pmatrix} ? & ? & \cdots & ? \\ ? & ? & \cdots & ? \\ \vdots & \vdots & \ddots & \vdots \\ ? & ? & \cdots & ? \end{pmatrix},$$

where “?” represents the unknown transition rate.

2.2. Problem Formulation. Consider a group of N quadrotor UAVs as shown in Figure 1; the dynamics of agent i are given as the following form [12]:

$$(4) \quad \begin{aligned} \ddot{x}_i &= \frac{(\sin \psi_i \sin \phi_i + \cos \psi_i \sin \theta_i \cos \phi_i) u_{1i}}{m_i} - \frac{k_{xi} \dot{x}_i}{m_i}, \\ \ddot{y}_i &= \frac{(-\cos \psi_i \sin \phi_i + \sin \psi_i \sin \theta_i \cos \phi_i) u_{1i}}{m_i} - \frac{k_{yi} \dot{y}_i}{m_i}, \\ \ddot{z}_i &= -g + \frac{(\cos \theta_i \cos \phi_i) u_{1i}}{m_i} - \frac{k_{zi} \dot{z}_i}{m_i}, \\ \ddot{\phi}_i &= \frac{l_i u_{2i}}{I_{xi}} - \frac{l_i k_{\phi i} \dot{\phi}_i}{I_{xi}}, \\ \ddot{\theta}_i &= \frac{l_i u_{3i}}{I_{yi}} - \frac{l_i k_{\theta i} \dot{\theta}_i}{I_{yi}}, \\ \ddot{\psi}_i &= \frac{K_{\psi i} u_{4i}}{I_{zi}} - \frac{k_{\psi i} \dot{\psi}_i}{I_{zi}}, \end{aligned}$$

where $x_i, y_i, z_i \in \mathbb{R}$ are positions and $\phi_i, \theta_i, \psi_i \in (-\pi/2, \pi/2)$ denote the three Euler angles of rotation, representing pitch, roll, and yaw, respectively. $k_{\phi_i, \theta_i, \psi_i}$ are the aerodynamic friction coefficients, and k_{x_i, y_i, z_i} are the coefficients of the translation drag forces. I_{x_i, y_i, z_i} are the quadrotor moments of inertias, m_i denotes the mass of the quadrotor, l_i is half length of the helicopter, K_{ψ_i} is thrust to moment gain, and g is gravitational acceleration.

Each system dynamics consist of six degrees of freedom model, which can be separated into position dynamic and attitude dynamic. With the choice of state variables $x_{1i} = x_i$, $x_{2i} = \dot{x}_i$; $y_{1i} = y_i$, $y_{2i} = \dot{y}_i$; $z_{1i} = z_i$, $z_{2i} = \dot{z}_i$; $\phi_{1i} = \phi_i$, $\phi_{2i} = \dot{\phi}_i$; $\theta_{1i} = \theta_i$, $\theta_{2i} = \dot{\theta}_i$; $\psi_{1i} = \psi_i$, $\psi_{2i} = \dot{\psi}_i$. Then, it can be represented by

$$\begin{aligned} \dot{x}_{1i} &= x_{2i}, \\ \dot{y}_{1i} &= y_{2i}, \\ \dot{z}_{1i} &= z_{2i}, \\ \dot{x}_{2i} &= \frac{(\sin \psi_i \sin \phi_i + \cos \psi_i \sin \theta_i \cos \phi_i) u_{1i}}{m_i} - \frac{k_{xi} x_{2i}}{m_i}, \\ \dot{y}_{2i} &= \frac{(-\cos \psi_i \sin \phi_i + \sin \psi_i \sin \theta_i \cos \phi_i) u_{1i}}{m_i} \\ &\quad - \frac{k_{yi} y_{2i}}{m_i}, \\ \dot{z}_{2i} &= -g + \frac{(\cos \theta_i \cos \phi_i) u_{1i}}{m_i} - \frac{k_{zi} z_{2i}}{m_i}, \\ \dot{\phi}_{1i} &= \phi_{2i}, \\ \dot{\theta}_{1i} &= \theta_{2i}, \\ \dot{\psi}_{1i} &= \psi_{2i}, \\ \dot{\phi}_{2i} &= \frac{l_i u_{2i}}{I_{xi}} - \frac{l_i k_{\phi i} \phi_{2i}}{I_{xi}}, \end{aligned}$$

$$\begin{aligned}\dot{\theta}_{2i} &= \frac{l_i u_{3i}}{I_{yi}} - \frac{l_i k_{\theta_i} \theta_{2i}}{I_{yi}}, \\ \dot{\psi}_{2i} &= \frac{K_{\psi_i} u_{4i}}{I_{zi}} - \frac{k_{\psi_i} \psi_{2i}}{I_{zi}}.\end{aligned}\quad (5)$$

As is well known to us, the quadrotor is under-actuated and differentially flat. Accordingly, choose four variables and specify the desired trajectory as $R_d = [x_d(t), y_d(t), z_d(t), \psi_d(t)]^T$.

2.3. Formation Definition. The formation shape can be described by a vector $\delta(t) = [\delta_1^T(t), \dots, \delta_N^T(t)]^T \in \mathbb{R}^{4N}$, where $\delta_i(t) = [\delta_{ix}(t), \delta_{iy}(t), \delta_{iz}(t), \delta_{i\psi}(t)]^T$ is the continuously differentiable formation vector, such that

$$\begin{aligned}\lim_{t \rightarrow \infty} (x_{1i}(t) - \delta_{ix}(t) - x_d(t)) &= 0, \\ \lim_{t \rightarrow \infty} (y_{1i}(t) - \delta_{iy}(t) - y_d(t)) &= 0, \\ \lim_{t \rightarrow \infty} (z_{1i}(t) - \delta_{iz}(t) - z_d(t)) &= 0, \\ \lim_{t \rightarrow \infty} (\psi_{1i}(t) - \delta_{i\psi}(t) - \psi_d(t)) &= 0.\end{aligned}\quad (6)$$

Define the desired position of agent i as $[x_{id}, y_{id}, z_{id}, \psi_{id}]^T = [\delta_{ix} + x_d, \delta_{iy} + y_d, \delta_{iz} + z_d, \delta_{i\psi} + \psi_d]^T$. For $j \in \Omega_i^\omega$, there exists $\delta_{ij} = [\delta_{ijx}, \delta_{ijy}, \delta_{ijz}, \delta_{ij\psi}]^T = [\delta_{jx} - \delta_{ix}, \delta_{jy} - \delta_{iy}, \delta_{jz} - \delta_{iz}, \delta_{j\psi} - \delta_{i\psi}]^T$. Hence, there are $N(N-1)/2$ such shape vectors satisfying the following properties:

$$\begin{aligned}\delta_{ik} &= \delta_{ij} + \delta_{jk}, \\ \delta_{ii} &= [0, 0, 0, 0]^T, \\ \delta_{ji} &= -\delta_{ij}, \\ \forall i, j, k &\in 1, \dots, N.\end{aligned}\quad (7)$$

Remark 5. The formation reference vector R_d is not available to all agents, but the desired interdistances of agent i and its neighbors are known. Note that the time varying formation

vector $\delta(t)$ is not unique, and the relative position between the reference vector R_d and the formation can be adjusted.

Lemma 6 (see [34]). *For any $X, Y \in \mathbb{R}^n$ and $\rho_1 > 0$, it holds that $X^T Y \leq \rho_1 X^T X / 2 + Y^T Y / 2\rho_1$.*

3. Main Results

In this section, time varying formation control problem for multiple quadrotor UAVs is solved through an event-triggered control scheme. The system stability analysis and exclusion of Zeno behavior are also provided.

3.1. Singularly Perturbed System. Define the error vectors as

$$\begin{aligned}\sigma_1^2 e_{1\phi}^i &= \phi_{1i} - \phi_{id}, \\ \sigma_1 e_{2\phi}^i &= \phi_{2i} - \dot{\phi}_{id}, \\ \sigma_1^2 e_{1\theta}^i &= \theta_{1i} - \theta_{id}, \\ \sigma_1 e_{2\theta}^i &= \theta_{2i} - \dot{\theta}_{id}, \\ \sigma_1^2 \tilde{\psi}_{1i} &= \psi_{1i} - \psi_{id}, \\ \sigma_1 \tilde{\psi}_{2i} &= \psi_{2i} - \dot{\psi}_{id}, \\ \tilde{x}_{1i} &= x_{1i} - x_{id}, \\ \tilde{x}_{2i} &= x_{2i} - \dot{x}_{id}, \\ \tilde{y}_{1i} &= y_{1i} - y_{id}, \\ \tilde{y}_{2i} &= y_{2i} - \dot{y}_{id}, \\ \tilde{z}_{1i} &= z_{1i} - z_{id}, \\ \tilde{z}_{2i} &= z_{2i} - \dot{z}_{id},\end{aligned}\quad (8)$$

where σ_1 is positive constant denoted as perturbing parameter, which satisfies $\sigma_1 \ll 1$. The overall system can be written as

$$\sum_{\phi^i} := \begin{cases} \sigma_1 e_{1\phi}^i = e_{2\phi}^i \\ \sigma_1 e_{2\phi}^i = \frac{l_i u_{2i}}{I_{xi}} - \frac{l_i k_{\phi_i} \phi_{2i}}{I_{xi}} - \ddot{\phi}_{id}, \end{cases}\quad (9)$$

$$\sum_{\theta^i} := \begin{cases} \sigma_1 e_{1\theta}^i = e_{2\theta}^i \\ \sigma_1 e_{2\theta}^i = \frac{l_i u_{3i}}{I_{yi}} - \frac{l_i k_{\theta_i} \theta_{2i}}{I_{yi}} - \ddot{\theta}_{id}, \end{cases}\quad (10)$$

$$\sum_{\psi^i} := \begin{cases} \sigma_1 \tilde{\psi}_{1i} = \tilde{\psi}_{2i} \\ \sigma_1 \tilde{\psi}_{2i} = \frac{K_{\psi_i} u_{4i}}{I_{zi}} - \frac{k_{\psi_i} \psi_{2i}}{I_{zi}} - \ddot{\delta}_{i\psi}, \end{cases}$$

$$\begin{aligned}
\sum_{xi} &:= \begin{cases} \dot{\tilde{x}}_{1i} = \tilde{x}_{2i} \\ \dot{\tilde{x}}_{2i} = \frac{(\sin \psi_i \sin \phi_i + \cos \psi_i \sin \theta_i \cos \phi_i) u_{1i}}{m_i} - \frac{k_{xi} x_{2i}}{m_i} - \delta_{ix}, \end{cases} \\
\sum_{yi} &:= \begin{cases} \dot{\tilde{y}}_{1i} = \tilde{y}_{2i} \\ \dot{\tilde{y}}_{2i} = \frac{(-\cos \psi_i \sin \phi_i + \sin \psi_i \sin \theta_i \cos \phi_i) u_{1i}}{m_i} - \frac{k_{yi} y_{2i}}{m_i} - \delta_{iy}, \end{cases} \\
\sum_{zi} &:= \begin{cases} \dot{\tilde{z}}_{1i} = \tilde{z}_{2i} \\ \dot{\tilde{z}}_{2i} = -g + \frac{(\cos \theta_i \cos \phi_i) u_{1i}}{m_i} - \frac{k_{zi} z_{2i}}{m_i} - \delta_{iz}. \end{cases}
\end{aligned} \tag{11}$$

System (9)–(11) has the standard form of a singularly perturbed system with a two-time scale; that is, $a_i = [e_{1\phi}^i, e_{2\phi}^i, e_{1\theta}^i, e_{2\theta}^i, \tilde{\psi}_{1i}, \tilde{\psi}_{2i}]^T$ represents the attitude states with fast time scale and $p_i = [\tilde{x}_{1i}, \tilde{x}_{2i}, \tilde{y}_{1i}, \tilde{y}_{2i}, \tilde{z}_{1i}, \tilde{z}_{2i}]^T$ represents the position states with slow time scale. Roughly speaking, in order to maintain a predefined formation shape for multiple quadrotor UAVs, the attitude stabilization should be guaranteed firstly.

3.2. Controller Design. Dynamic event-triggered control scheme is introduced to the multiple quadrotor UAVs. Define the following measurement errors:

$$\begin{aligned}
e_{1\psi}^i(t) &= \tilde{\psi}_{1i}(t_k^i) - \tilde{\psi}_{1i}(t), \\
e_{2\psi}^i(t) &= \tilde{\psi}_{2i}(t_k^i) - \tilde{\psi}_{2i}(t), \\
e_{1x}^i(t) &= \tilde{x}_{1i}(t_k^i) - \tilde{x}_{1i}(t), \\
e_{2x}^i(t) &= \tilde{x}_{2i}(t_k^i) - \tilde{x}_{2i}(t), \\
e_{1y}^i(t) &= \tilde{y}_{1i}(t_k^i) - \tilde{y}_{1i}(t), \\
e_{2y}^i(t) &= \tilde{y}_{2i}(t_k^i) - \tilde{y}_{2i}(t), \\
e_{1z}^i(t) &= \tilde{z}_{1i}(t_k^i) - \tilde{z}_{1i}(t), \\
e_{2z}^i(t) &= \tilde{z}_{2i}(t_k^i) - \tilde{z}_{2i}(t).
\end{aligned} \tag{12}$$

Let

$$\begin{aligned}
\psi_{pi} &= \sum_{j \in \Omega_i^o} a_{ij}^\omega (\tilde{\psi}_{1i}(t_k^i) - \tilde{\psi}_{1j}(t_{k'}^j)) + a_{ii}^l \tilde{\psi}_{1i}(t_k^i), \\
\psi_{vi} &= \sum_{j \in \Omega_i^o} a_{ij}^\omega (\tilde{\psi}_{2i}(t_k^i) - \tilde{\psi}_{2j}(t_{k'}^j)) + a_{ii}^l \tilde{\psi}_{2i}(t_k^i), \\
x_{pi} &= \sum_{j \in \Omega_i^o} a_{ij}^\omega (\tilde{x}_{1i}(t_k^i) - \tilde{x}_{1j}(t_{k'}^j)) + a_{ii}^l \tilde{x}_{1i}(t_k^i), \\
x_{vi} &= \sum_{j \in \Omega_i^o} a_{ij}^\omega (\tilde{x}_{2i}(t_k^i) - \tilde{x}_{2j}(t_{k'}^j)) + a_{ii}^l \tilde{x}_{2i}(t_k^i),
\end{aligned}$$

$$\begin{aligned}
y_{pi} &= \sum_{j \in \Omega_i^o} a_{ij}^\omega (\tilde{y}_{1i}(t_k^i) - \tilde{y}_{1j}(t_{k'}^j)) + a_{ii}^l \tilde{y}_{1i}(t_k^i), \\
y_{vi} &= \sum_{j \in \Omega_i^o} a_{ij}^\omega (\tilde{y}_{2i}(t_k^i) - \tilde{y}_{2j}(t_{k'}^j)) + a_{ii}^l \tilde{y}_{2i}(t_k^i), \\
z_{pi} &= \sum_{j \in \Omega_i^o} a_{ij}^\omega (\tilde{z}_{1i}(t_k^i) - \tilde{z}_{1j}(t_{k'}^j)) + a_{ii}^l \tilde{z}_{1i}(t_k^i), \\
z_{vi} &= \sum_{j \in \Omega_i^o} a_{ij}^\omega (\tilde{z}_{2i}(t_k^i) - \tilde{z}_{2j}(t_{k'}^j)) + a_{ii}^l \tilde{z}_{2i}(t_k^i).
\end{aligned} \tag{13}$$

Then, the auxiliary control variables are given by

$$\begin{aligned}
v_{\phi i} &= -\alpha_\phi e_{1\phi}^i - \beta_\phi e_{2\phi}^i, \\
v_{\theta i} &= -\alpha_\theta e_{1\theta}^i - \beta_\theta e_{2\theta}^i, \\
v_{\psi i} &= -\alpha_\psi \psi_{pi} - \beta_\psi \psi_{vi}, \\
v_{xi} &= -\alpha_x x_{pi} - \beta_x x_{vi}, \\
v_{yi} &= -\alpha_y y_{pi} - \beta_y y_{vi}, \\
v_{zi} &= -\alpha_z z_{pi} - \beta_z z_{vi}.
\end{aligned} \tag{14}$$

Therefore, the choice of control inputs

$$\begin{aligned}
u_{2i} &= \frac{I_{xi} v_{\phi i}}{l_i} + k_{\phi i} \phi_{2i} + \frac{I_{xi} \ddot{\phi}_{id}}{l_i}, \\
u_{3i} &= \frac{I_{yi} v_{\theta i}}{l_i} + k_{\theta i} \theta_{2i} + \frac{I_{yi} \ddot{\theta}_{id}}{l_i}, \\
u_{4i} &= \frac{I_{zi} v_{\psi i}}{K_{\psi i}} + \frac{k_{\psi i} \psi_{2i}}{K_{\psi i}} + \frac{I_{zi} \ddot{\delta}_{i\psi}}{K_{\psi i}}, \\
u_{1i} &= \frac{m_i v_{zi} + m \delta_{iz} + m_i g + k_{zi} z_{2i}}{\cos \theta_{id} \cos \phi_{id}},
\end{aligned} \tag{15}$$

where the reference angles ϕ_{id} and θ_{id} are addressed as

$$\theta_{id} = \tan^{-1} \frac{\cos \psi_{id} (v_{xi} + \ddot{\delta}_{ix} + k_{xi} x_{2i}/m_i) + \sin \psi_{id} (v_{yi} + \ddot{\delta}_{iy} + k_{yi} y_{2i}/m_i)}{v_{zi} + g + \ddot{\delta}_{iz} + k_{zi} z_{2i}/m_i}, \quad (17)$$

$$\phi_{id} = \tan^{-1} \frac{\cos \theta_{id} (\sin \psi_{id} (v_{xi} + \ddot{\delta}_{ix} + k_{xi} x_{2i}/m_i) - \cos \psi_{id} (v_{yi} + \ddot{\delta}_{iy} + k_{yi} y_{2i}/m_i))}{v_{zi} + g + \ddot{\delta}_{iz} + k_{zi} z_{2i}/m_i}. \quad (18)$$

Remark 7. It should be stressed that (17) and (18) are needed to be nonsingular; in other words, the denominator of (17) or (18) cannot be zero. In practice, the quadrotor has to take a certain thrust to overcome gravity in order to maintain hovering; otherwise, it would sink vertically. It means that the denominator of (17) or (18) is approximate to g , and during the flight, it will also be greater than zero based on the defined domain of z dynamics $[z_{1i}, z_{2i}, \dot{z}_{2i}]^T \in D_z = \mathbb{R} \times \{|z_{2i}| < d_{z2}\} \times \{|\dot{z}_{2i}| < d_{\dot{z}2}\} \subset \mathbb{R}^3, d_{z2} \ll g$.

3.3. Dynamic Event-Triggered Communication. Figure 2 depicts the control law running in the i th quadrotor. Generally speaking, each agent updates its controller whenever the designed trigger condition is reached, called the triggered event. Based on local information, it decides when to broadcast its current state over the network. In other words, the key problem is to find a triggering rule that determines when agent i has to broadcast the new state information to its neighbors.

In contrast to most of the existing works, a new dynamic event-triggered communication mechanism is developed to schedule interagent communication. The threshold parameter in the proposed event triggering condition will not be fixed permanently but vary with time by following a dynamic rule. The detailed dynamic rule is provided in the following section. The event-triggered control strategy works as follows: define a trigger function $f_i(t)$ for each agent, which depends on local information only; an event is triggered as soon as the trigger condition $f_i(t) > 0$ is fulfilled, while

each agent recomputes its control law in accordance with the measurement error, such that all the agents could reach and keep the predesigned formation shape. It will be shown numerically that the dynamic event-triggered communication mechanism can achieve a better tradeoff between reducing data transmissions and preserving favorable formation performance.

3.4. System Stability via Two-Time Scale. The stability analysis for each subsystem will be done by starting from the faster one to the slower one. The main results of this paper are presented next with the help of the following theorems.

Theorem 8. *Consider the i th quadrotor dynamics, which is also a singularly perturbed system. Then, there always exists $\sigma_1^* > 0$ such that the state space origin of the closed-loop system (9)–(11) is exponentially stable with $\sigma_1^* > \sigma_1 > 0$.*

Proof. The proof is set down in the following five items.

(1) System (9)–(11) has a unique equilibrium point at $[a_i^T, p_i^T]^T = 0$.

(2) The quasi-steady-state solution of closed-loop system (9)–(10) is $a_i = a^* = h(t, p_i)$. Hence, by substituting $\sigma_1 = 0$ into (9)–(10), the isolated root is given by $a^* = 0$. It is noteworthy that the isolated root evaluated in (16)–(18) gives the specific values ϕ_{id}^* , θ_{id}^* , and ψ_{id}^* .

(3) System (9)–(11) and the isolated root have bounded partial derivatives in compact sets.

(4) By considering $\sigma_1 = 0$ and using the isolated root into (11), the following slow dynamics system is obtained:

$$\begin{aligned} \sum_{xi} &:= \begin{cases} \dot{\tilde{x}}_{1i} = \tilde{x}_{2i} \\ \dot{\tilde{x}}_{2i} = \frac{(\sin \psi_{id}^* \sin \phi_{id}^* + \cos \psi_{id}^* \sin \theta_{id}^* \cos \phi_{id}^*) u_{1i}}{m_i} - \frac{k_{xi} x_{2i}}{m_i} - \ddot{\delta}_{ix}, \end{cases} \\ \sum_{yi} &:= \begin{cases} \dot{\tilde{y}}_{1i} = \tilde{y}_{2i} \\ \dot{\tilde{y}}_{2i} = \frac{(-\cos \psi_{id}^* \sin \phi_{id}^* + \sin \psi_{id}^* \sin \theta_{id}^* \cos \phi_{id}^*) u_{1i}}{m_i} - \frac{k_{yi} y_{2i}}{m_i} - \ddot{\delta}_{iy}, \end{cases} \\ \sum_{zi} &:= \begin{cases} \dot{\tilde{z}}_{1i} = \tilde{z}_{2i} \\ \dot{\tilde{z}}_{2i} = -g + \frac{(\cos \theta_{id}^* \cos \phi_{id}^*) u_{1i}}{m_i} - \frac{k_{zi} z_{2i}}{m_i} - \ddot{\delta}_{iz}. \end{cases} \end{aligned} \quad (19)$$

The local stability of the slow dynamics (11) can be derived by using event-triggered control method. For the sake of simplicity, it will be given in Theorem 12 later.

(5) To obtain the boundary layer system, by deriving $a-a^*$ with respect to scaled time $\kappa = t/\sigma_1$ and setting $\sigma_1 = 0$, the boundary layer system is obtained:

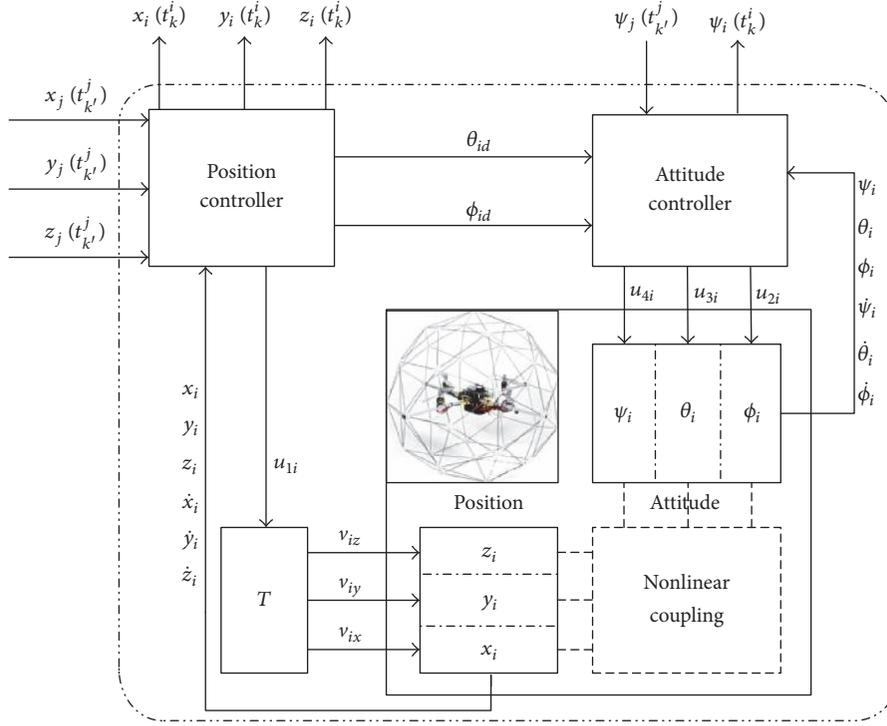


FIGURE 2: Block diagram of the event-based control strategy.

$$\begin{aligned} \frac{d}{d\kappa} e_{1\phi}^i &= e_{2\phi}^i, \\ \frac{d}{d\kappa} e_{2\phi}^i &= -\alpha_\phi e_{1\phi}^i - \beta_\phi e_{2\phi}^i, \end{aligned} \quad (20)$$

$$\begin{aligned} \frac{d}{d\kappa} e_{1\theta}^i &= e_{2\theta}^i, \\ \frac{d}{d\kappa} e_{2\theta}^i &= -\alpha_\theta e_{1\theta}^i - \beta_\theta e_{2\theta}^i, \\ \frac{d}{d\kappa} \tilde{\psi}_{1i} &= \tilde{\psi}_{2i}, \end{aligned} \quad (21)$$

$$\frac{d}{d\kappa} \tilde{\psi}_{2i} = -\alpha_\psi \psi_{pi} - \beta_\psi \psi_{vi}.$$

Thus, for the ϕ_i and θ_i subsystems

$$\begin{aligned} \frac{d}{d\kappa} e_\phi^i &= A_\phi e_\phi^i \implies \\ \frac{d}{d\kappa} \begin{bmatrix} e_{1\phi}^i \\ e_{2\phi}^i \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\alpha_\phi & -\beta_\phi \end{bmatrix} \begin{bmatrix} e_{1\phi}^i \\ e_{2\phi}^i \end{bmatrix}, \\ \frac{d}{d\kappa} e_\theta^i &= A_\theta e_\theta^i \implies \\ \frac{d}{d\kappa} \begin{bmatrix} e_{1\theta}^i \\ e_{2\theta}^i \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\alpha_\theta & -\beta_\theta \end{bmatrix} \begin{bmatrix} e_{1\theta}^i \\ e_{2\theta}^i \end{bmatrix}. \end{aligned} \quad (22)$$

It is obvious that A_ϕ and A_θ are Hurwitz, where system (22) is exponentially stable. Meanwhile, the stability result of ψ_i subsystem will be established in Theorem 10 as follows.

According to [32], there are sufficient conditions to claim that there exists σ_1^* such that $\sigma_1^* > \sigma_1 > 0$ guarantees that system (9)–(11) achieves the following limit $\lim_{t \rightarrow \infty} [a_i^T, p_i^T]^T = 0$. \square

Remark 9. Based on the system stability analysis by the singular perturbation method, instead of using the feedback angles ϕ_i, θ_i, ψ_i , the control input u_{1i} is related to the reference angles $\phi_{id}, \theta_{id}, \psi_{id}$, and the coupling between the attitude subsystem and the position subsystem is diminished.

Theorem 10. Given $\tilde{\psi}_{pi} = \sum_{j \in \Omega_i^\omega} a_{ij}^\omega (e_{1\psi}^i - e_{1\psi}^j) + a_{ii}^l e_{1\psi}^i$ and $\tilde{\psi}_{vi} = \sum_{j \in \Omega_i^\omega} a_{ij}^\omega (e_{2\psi}^i - e_{2\psi}^j) + a_{ii}^l e_{2\psi}^i$, then define $E_\psi^i = [\tilde{\psi}_{pi}, \tilde{\psi}_{vi}]^T$ and $\tilde{\Psi}_i = [\psi_{pi} - \tilde{\psi}_{pi}, \psi_{vi} - \tilde{\psi}_{vi}]^T$. Consider the yaw subsystem (21); the system stability could be obtained through the distributed control law (14) and (15) and the following event-triggered communication condition:

$$f_i(t) = \|B_2^T Q E_\psi^i\| - \sigma(t) \|B_2^T Q \tilde{\Psi}_i\|, \quad (23)$$

where $\sigma(t) = e^{-\mu e_s}$ is the dynamic threshold parameter with $e_s = \int_0^{t_k} e_\psi^i{}^T e_\psi^i ds$, $e_\psi^i = [e_{1\psi}^i, e_{2\psi}^i]^T$, $\mu > 0$, and the initial value $\sigma(0) = \sigma_0 \in (0, 1)$, which ensures that $f_i(t) \leq 0$ always holds.

Proof. Obviously, $\sigma(t)$ is a monotone nonincreasing function; thus, $\sigma(t) \leq \sigma_0$ always holds. Particularly, if the threshold

parameter is preset as a constant σ_0 before dynamic event-triggered controller is implemented, this is called the static event-triggered controller. Define $B_1 = [1, 0]^T$, $B_2 = [0, 1]^T$, and $\tilde{\psi}_i = [\tilde{\psi}_{1i}, \tilde{\psi}_{2i}]^T$. Based on (13), (21) becomes

$$\frac{d}{d\kappa} \tilde{\psi}_i = B_1 B_2^T \tilde{\psi}_i + B_2 (-\alpha_\psi \psi_{pi} - \beta_\psi \psi_{vi}). \quad (24)$$

Hence, it follows from [35] that there exist positive constants α_0 , α_ψ , and β_ψ such that

$$\frac{1}{2} (QB_1 B_2^T + B_2 B_1^T Q) - \rho_0 QB_2^T B_2 Q + 2I_2 \leq 0 \quad (25)$$

$$Q > 0,$$

where $Q = [\alpha_0, \alpha_\psi; \alpha_\psi, \beta_\psi]$, $B_2^T Q = [\alpha_\psi, \beta_\psi]$, and

$$0 < \rho_0 \leq \left(1 - \frac{\rho_1}{2} - \frac{\sigma_0^2}{2\rho_1}\right) \lambda_{\min}(L^G). \quad (26)$$

Let $\Psi = [\tilde{\psi}_1, \dots, \tilde{\psi}_N]^T$ and $E_\psi = [e_\psi^1, \dots, e_\psi^N]^T$. Consider the following Lyapunov function:

$$V_\psi = \frac{1}{2} \Psi^T (I_N \otimes Q) \Psi. \quad (27)$$

Then, its derivative is

$$\begin{aligned} \frac{d}{d\kappa} V_\psi &= \Psi^T \left(I_N \otimes \frac{QB_1 B_2^T + B_2 B_1^T Q}{2} \right) \Psi \\ &\quad - \Psi^T (L^G \otimes QB_2 B_2^T Q) \Psi \\ &\quad - \Psi^T (L^G \otimes QB_2 B_2^T Q) E_\psi. \end{aligned} \quad (28)$$

Based on Lemma 6, the following inequality holds:

$$\begin{aligned} &\Psi^T (L^G \otimes QB_2 B_2^T Q) E_\psi \\ &\leq \frac{\rho_1}{2} \Psi^T (L^G \otimes QB_2 B_2^T Q) \Psi \\ &\quad + \frac{1}{2\rho_1} E_\psi^T (L^G \otimes QB_2 B_2^T Q) E_\psi. \end{aligned} \quad (29)$$

Based on (23) and (25), (28) can be written as

$$\begin{aligned} \frac{d}{d\kappa} V_\psi &\leq \Psi^T \left(I_N \otimes \frac{QB_1 B_2^T + B_1 B_2^T Q^T}{2} \right) \Psi \\ &\quad - \lambda_{\min}(L^G) \Psi^T (I_N \otimes QB_2 B_2^T Q) \Psi + \frac{\rho_1}{2} \Psi^T (L^G \\ &\quad \otimes QB_2 B_2^T Q) \Psi + \frac{1}{2\rho_1} E_\psi^T (L^G \otimes QB_2 B_2^T Q) E_\psi \\ &\leq \Psi^T \left(I_N \otimes \left[\frac{QB_1 B_2^T + B_1 B_2^T Q^T}{2} \right. \right. \\ &\quad \left. \left. - \left(1 - \frac{\rho_1}{2} - \frac{\sigma^2(t)}{2\rho_1}\right) \lambda_{\min}(L^G) QB_2 B_2^T Q \right] \right) \Psi \\ &\leq -2\rho(Q) V_\psi. \end{aligned} \quad (30)$$

Hence, $V_\psi \leq \sigma_1 V_\psi(0) e^{-2\rho(Q)t}$, and $\lim_{t \rightarrow \infty} (\psi_{1i}(t) - \delta_{i\psi}(t) - \psi_{d_i}(t)) = 0$ exponentially.

Next, Zeno behavior is excluded. For any $t > t_k^i$, it follows from (23) that

$$\begin{aligned} \left\| \frac{d}{d\kappa} (L^G \otimes B_2^T Q) E_\psi \right\| &\leq \sqrt{\alpha_\psi^2 + \beta_\psi^2} \|L^G\| (\|E_\psi\| \\ &\quad + \sqrt{\alpha_\psi^2 + \beta_\psi^2} \|L^G\| \|\Psi\| + \sqrt{\alpha_\psi^2 + \beta_\psi^2} \|L^G\| \|E_\psi\|) \\ &\leq (\sigma(t) + (1 + \sigma(t)) \sqrt{\alpha_\psi^2 + \beta_\psi^2} \|L^G\|) \\ &\quad \cdot \sqrt{\alpha_\psi^2 + \beta_\psi^2} \|L^G\| \|\Psi\|, \end{aligned} \quad (31)$$

where $\|\Psi\| \leq \sqrt{2\sigma_1 \rho(Q) V_\psi(0)} e^{-\rho(Q)t}$. Hence,

$$\begin{aligned} \left\| \frac{d}{d\kappa} (L^G \otimes B_2^T Q) E_\psi \right\| &\leq (\sigma(t) + (1 + \sigma(t)) \sqrt{\alpha_\psi^2 + \beta_\psi^2} \|L^G\|) \\ &\quad \cdot \sqrt{\alpha_\psi^2 + \beta_\psi^2} \|L^G\| \sqrt{2\sigma_1 \rho(Q) V_\psi(0)} e^{-\rho(Q)t}. \end{aligned} \quad (32)$$

Thus,

$$\begin{aligned} &\|(L^G \otimes B_2^T Q) E_\psi\| \\ &\leq \sigma_1 (\sigma(t) + (1 + \sigma(t)) \sqrt{\alpha_\psi^2 + \beta_\psi^2} \|L^G\|) \\ &\quad \cdot \sqrt{\alpha_\psi^2 + \beta_\psi^2} \|L^G\| \sqrt{2\sigma_1 \rho(Q) V_\psi(0)} e^{-\rho(Q)t} (t - t_k). \end{aligned} \quad (33)$$

The next event t_{k+1} will not be triggered before $\|(L^G \otimes B_2^T Q) E_\psi\| = \sigma(t) \|(L^G \otimes B_2^T Q) \Psi\|$. Thus, a lower bound on the interevent intervals is given by $\tau = t - t_k$ that solves the following equation:

$$\begin{aligned} &\sigma(\tau) \|(L^G \otimes B_2^T Q) \Psi\| \\ &= \sigma_1 \tau (\sigma(\tau) + (1 + \sigma(\tau)) \sqrt{\alpha_\psi^2 + \beta_\psi^2} \|L^G\|) \\ &\quad \cdot \sqrt{\alpha_\psi^2 + \beta_\psi^2} \|L^G\| \sqrt{2\sigma_1 \rho(Q) V_\psi(0)} e^{-\rho(Q)t}. \end{aligned} \quad (34)$$

Therefore, for all $t_k \geq 0$ the solutions $\tau(t_k)$ are greater than or equal to τ given by

$$\sigma(\tau) = \sigma_1 \tau (\sigma(\tau) + (1 + \sigma(\tau)) \sqrt{\alpha_\psi^2 + \beta_\psi^2} \|L^G\|) \quad (35)$$

which is strictly positive, so Zeno behavior is excluded. This completes the proof. \square

Remark 11. It should be pointed out that the vectors ϕ_{id} and θ_{id} are obtained inside of the controller; as shown in Figure 2, the pitch and roll attitude dynamics are not event-triggered. In other words, only the yaw angle in the attitude subsystem is controlled through an event-triggered strategy. In addition, compared with some existing works, in which

assuming $\psi_{1i} = 0$ in the position control loop, the yaw angle is not always zero in practice. Since that the control method is more suitable in this paper.

After the attitude angles reach quasi-steady states, hierarchically, $\sigma_1 = 0$, $\phi_{1i} \rightarrow \phi_{id}$, $\theta_{1i} \rightarrow \theta_{id}$, $\psi_{1i} \rightarrow \psi_{id}$. Substituting (14), (16)–(18) into (19), it is possible to show that the closed-loop position subsystem can be written as

$$\begin{aligned}\dot{\tilde{x}}_{1i} &= \tilde{x}_{2i}, \\ \dot{\tilde{x}}_{2i} &= -\alpha_x x_{pi} - \beta_x x_{vi}, \\ \dot{\tilde{y}}_{1i} &= \tilde{y}_{2i}, \\ \dot{\tilde{y}}_{2i} &= -\alpha_y y_{pi} - \beta_y y_{vi}, \\ \dot{\tilde{z}}_{1i} &= \tilde{z}_{2i}, \\ \dot{\tilde{z}}_{2i} &= -\alpha_z z_{pi} - \beta_z z_{vi}.\end{aligned}\quad (36)$$

Without loss of generality, we only provide the proof along x -subsystem in the following.

Theorem 12. Given $\tilde{x}_{pi} = \sum_{j \in \Omega_i^o} a_{ij}^\omega (e_{1x}^i - e_{1x}^j) + a_{ii}^l e_{1x}^i$ and $\tilde{x}_{vi} = \sum_{j \in \Omega_i^o} a_{ij}^\omega (e_{2x}^i - e_{2x}^j) + a_{ii}^l e_{2x}^i$, then let $E_x^i = [\tilde{x}_{pi}, \tilde{x}_{vi}]^T$ and $\bar{X}_i = [x_{pi} - \tilde{x}_{pi}, x_{vi} - \tilde{x}_{vi}]^T$. Considering the position subsystem (19), the system stability could be obtained through the distributed control law (16)–(18) and the following event-triggered communication condition:

$$f_i(t) = \|B_2^T Q_0 E_x^i\| - \sigma(t) \|B_2^T Q_0 \bar{X}_i\|, \quad (37)$$

where $\sigma(t) = e^{-\mu e_s}$ is the dynamic threshold parameter with $e_s = \int_0^{t_k} e_x^i T e_x^i ds$, $e_x^i = [e_{1x}^i, e_{2x}^i]^T$, $\mu > 0$ and the initial value $\sigma(0) = \sigma_0 \in (0, 1)$.

Proof. Let $\tilde{x}_i = [\tilde{x}_{1i}, \tilde{x}_{2i}]^T$; the x -subsystem could be rewritten as

$$\dot{\tilde{x}}_i = B_1 B_2^T \tilde{x}_i + B_2 (-\alpha_x x_{pi} - \beta_x x_{vi}). \quad (38)$$

Similarly, there exist positive constants α_0 , α_x , and β_x such that

$$\begin{aligned}\frac{1}{2} (Q_0 B_1 B_2^T + B_2 B_1^T Q_0) - \rho_0 Q_0 B_2^T B_2 Q_0 + 2I_2 \leq 0 \\ Q_0 > 0,\end{aligned}\quad (39)$$

where $Q_0 = [\alpha_0, \alpha_x; \alpha_x, \beta_x]$. Note that $B_2^T Q_0 = [\alpha_x, \beta_x]$, and let $X = [\tilde{x}_1, \dots, \tilde{x}_N]^T$ and $E_X = [e_x^1, \dots, e_x^N]^T$. Design the following Lyapunov function:

$$V_x = \frac{1}{2} X^T (I_N \otimes Q_0) X. \quad (40)$$

Then, its derivative is

$$\begin{aligned}\dot{V}_x &= X^T \left(I_N \otimes \frac{Q_0 B_1 B_2^T + B_2 B_1^T Q_0}{2} \right) X \\ &\quad - X^T (L^G \otimes Q_0 B_2 B_2^T Q_0) X \\ &\quad - X^T (L^G \otimes Q_0 B_2 B_2^T Q_0) E_X.\end{aligned}\quad (41)$$

Based on Lemma 6, the following inequality holds:

$$\begin{aligned}X^T (L^G \otimes Q_0 B_2 B_2^T Q_0) E_X \\ \leq \frac{\rho_1}{2} X^T (L^G \otimes Q_0 B_2 B_2^T Q_0) X \\ + \frac{1}{2\rho_1} E_X^T (L^G \otimes Q_0 B_2 B_2^T Q_0) E_X.\end{aligned}\quad (42)$$

Based on (37) and (39), (41) can be written as

$$\begin{aligned}\dot{V}_x &\leq X^T \left(I_N \otimes \left[\frac{Q_0 B_1 B_2^T + B_1 B_2^T Q_0^T}{2} \right. \right. \\ &\quad \left. \left. - \left(1 - \frac{\rho_1}{2} - \frac{\sigma^2(t)}{2\rho_1} \right) \lambda_{\min}(L^G) Q_0 B_2 B_2^T Q_0 \right] \right) X \\ &\leq -2\rho(Q_0) V_x.\end{aligned}\quad (43)$$

Hence, $V_x \leq V_x(0)e^{-2\rho(Q_0)t}$, and $\lim_{t \rightarrow \infty} (x_{1i}(t) - \delta_{ix}(t) - x_{d}(t)) = 0$ exponentially.

Next, Zeno behavior is excluded. For any $t > t_k^i$, one has

$$\begin{aligned}\left\| \frac{d}{dt} (L^G \otimes B_2^T Q) E_X \right\| &\leq \sqrt{\alpha_x^2 + \beta_x^2} \|L^G\| (\|E_X\| \\ &\quad + \sqrt{\alpha_x^2 + \beta_x^2} \|L^G\| \|X\| + \sqrt{\alpha_x^2 + \beta_x^2} \|L^G\| \|E_X\|) \\ &\leq \left(\sigma(t) + (1 + \sigma(t)) \sqrt{\alpha_x^2 + \beta_x^2} \|L^G\| \right) \\ &\quad \cdot \sqrt{\alpha_x^2 + \beta_x^2} \|L^G\| \|X\|,\end{aligned}\quad (44)$$

where $\|X\| \leq \sqrt{2\rho(Q_0)V_x(0)}e^{-\rho(Q_0)t}$. Hence,

$$\begin{aligned}\left\| \frac{d}{dt} (L^G \otimes B_2^T Q) E_X \right\| \\ \leq \left(\sigma(t) + (1 + \sigma(t)) \sqrt{\alpha_x^2 + \beta_x^2} \|L^G\| \right) \\ \cdot \sqrt{\alpha_x^2 + \beta_x^2} \|L^G\| \sqrt{2\rho(Q_0)V_x(0)}e^{-\rho(Q_0)t}.\end{aligned}\quad (45)$$

Thus,

$$\begin{aligned}\left\| (L^G \otimes B_2^T Q) E_X \right\| \\ \leq \left(\sigma(t) + (1 + \sigma(t)) \sqrt{\alpha_x^2 + \beta_x^2} \|L^G\| \right) \\ \cdot \sqrt{\alpha_x^2 + \beta_x^2} \|L^G\| \sqrt{2\rho(Q_0)V_x(0)}e^{-\rho(Q_0)t} (t - t_k).\end{aligned}\quad (46)$$

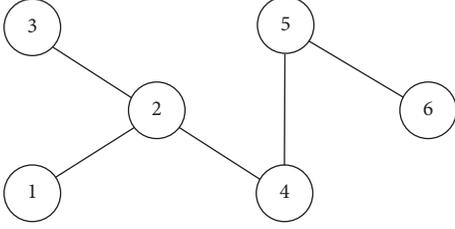


FIGURE 3: Initial mode of interaction topology.

The next event t_{k+1} will not be triggered before $\|(L^G \otimes B_2^T Q)E_X\| = \sigma(t)\|(L^G \otimes B_2^T Q)X\|$; it means

$$\begin{aligned}
 & \sigma(t) \|(L^G \otimes B_2^T Q)X\| \\
 & \leq \left(\sigma(t) + (1 + \sigma(t)) \sqrt{\alpha_x^2 + \beta_x^2} \|L^G\| \right) \\
 & \quad \cdot \sqrt{\alpha_x^2 + \beta_x^2} \|L^G\| \\
 & \quad \cdot \sqrt{2\rho(Q_0) V_x(0)} e^{-\rho(Q_0)t} (t_{k+1} - t_k).
 \end{aligned} \tag{47}$$

By the same graphical argument as in the proof of Theorem 10, it can be concluded that a lower bounded on the interevent times is given by τ , so Zeno behavior is excluded.

This completes the proof. \square

Remark 13. Compared with the existing works, the threshold parameter of event triggering condition will not be fixed permanently but vary with time by following the measurement errors. Since that it can achieve a better tradeoff between reducing event times and preserving favorable formation performance.

4. Simulation Results

In this section, simulations are given to demonstrate the effectiveness of the theoretical results. Consider a group of agents, which consists of six quadrotor UAVs, and the dynamics can be written as (4). The system parameters are shown in Table 1. Furthermore, the initial mode of undirected topology $G(\omega(t))$ is given in Figure 3. A special case is considered, where ω_0 takes the switch value in $S_0 = \{1, 2, 3, 4, 5, 6\}$ with an equal probability $1/6$ and the time interval $\Delta_0 = 5$ s. Meanwhile, assume that each Markov chain ω_{ij} takes values in a finite state space $S = \{0, 1\}$, and the time interval $\Delta = 1$ s. Otherwise, each undirected edge has a Markov chain, and the six agents have 15 chains totally. The transition rate matrices are considered as Table 2.

As an example, the time evolutions of some Markov chains are depicted in Figure 4.

Let the initial states of six quadrotor UAVs be

$$\begin{aligned}
 & [x_1(0), y_1(0), z_1(0), \phi_1(0), \theta_1(0), \psi_1(0)]^T \\
 & = [1, 1, 0, 0, 0, 0]^T,
 \end{aligned}$$

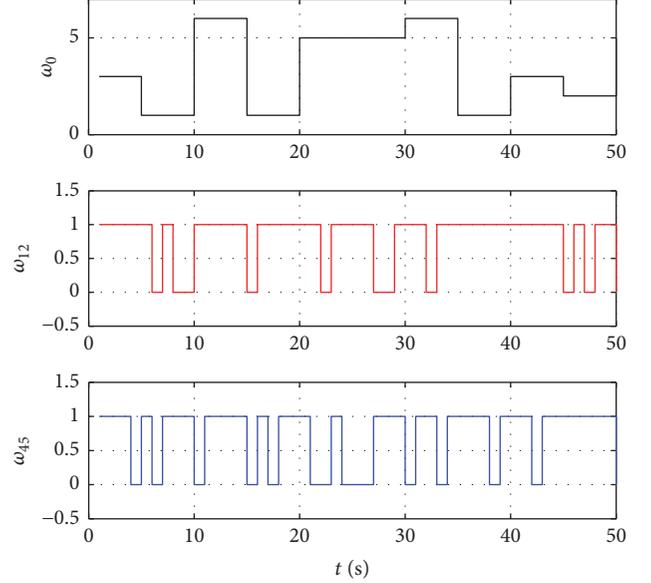


FIGURE 4: Evolutions of some Markov chains.

$$\begin{aligned}
 & [x_2(0), y_2(0), z_2(0), \phi_2(0), \theta_2(0), \psi_2(0)]^T \\
 & = [3, -1, 0, 0, 0, 0]^T, \\
 & [x_3(0), y_3(0), z_3(0), \phi_3(0), \theta_3(0), \psi_3(0)]^T \\
 & = [-3, 2, 0, 0, 0, 0]^T, \\
 & [x_4(0), y_4(0), z_4(0), \phi_4(0), \theta_4(0), \psi_4(0)]^T \\
 & = [-2, -2, 0, 0, 0, 0]^T, \\
 & [x_5(0), y_5(0), z_5(0), \phi_5(0), \theta_5(0), \psi_5(0)]^T \\
 & = [3.5, 2, 0, 0, 0, 0]^T, \\
 & [x_6(0), y_6(0), z_6(0), \phi_6(0), \theta_6(0), \psi_6(0)]^T \\
 & = [-3, 1, 0, 0, 0, 0]^T.
 \end{aligned} \tag{48}$$

The desired trajectory is given by $R_d = [0, 0, 0.05t, \pi/30]^T$, and the formation shape specified by the desired interagent distances is described as

$$\delta_i(t) = \begin{bmatrix} \delta_{ix}(t) \\ \delta_{iy}(t) \\ \delta_{iz}(t) \\ \delta_{i\psi}(t) \end{bmatrix} = \begin{bmatrix} 3 \sin\left(0.2t + \frac{(i-1)\pi}{3}\right) \\ 3 \cos\left(0.2t + \frac{(i-1)\pi}{3}\right) \\ 0 \\ 0 \end{bmatrix} \tag{49}$$

which means that if the quadrotor system achieves the desired time varying formation, then the six quadrotor UAVs will follow a circle while keeping a phase separation of $\pi/3$. In order to verify the effectiveness and advantage of the

TABLE 1: The system parameters of quadrotor UAVs.

Parameters	Nominal value	Parameters	Nominal value
l_i	0.20 m	k_{xi}	0.010 N/m·s ⁻¹
m_i	1.79 kg	k_{yi}	0.010 N/m·s ⁻¹
I_{xi}	0.03 kg·m ²	k_{zi}	0.010 N/m·s ⁻¹
I_{yi}	0.03 kg·m ²	$k_{\phi i}$	0.012 N/rad·s ⁻¹
I_{zi}	0.04 kg·m ²	$k_{\theta i}$	0.012 N/rad·s ⁻¹
g	9.81 m/s ²	$k_{\psi i}$	0.012 N/rad·s ⁻¹
$K_{\psi i}$	0.40 N·m		

TABLE 2: Transition rate matrices of the Markov chains in graph.

Edges	$(v_1, v_2); (v_2, v_4); (v_4, v_5)$ $(v_3, v_5); (v_3, v_6)$	$(v_1, v_3); (v_2, v_3); (v_2, v_6)$ $(v_5, v_6); (v_1, v_5)$	$(v_1, v_4); (v_1, v_6); (v_2, v_5)$ $(v_3, v_4); (v_4, v_6)$
Transition rate matrices	$\begin{bmatrix} 0.2 & 0.8 \\ 0.4 & 0.6 \end{bmatrix}$	$\begin{bmatrix} 0.5 & 0.5 \\ ? & ? \end{bmatrix}$	$\begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$

TABLE 3: Event times of all agents.

Agent	v_1	v_2	v_3	v_4	v_5	v_6
Case 1	131	123	109	126	143	115
Case 2	62	48	67	60	61	54

developed formation control protocol, the following two cases are evaluated:

Case 1. The distributed event-triggered function (23), (37) and the controller are implemented with the parameters as follows:

$$\begin{aligned}
\sigma_0 &= 0.6, \\
\sigma_1 &= 0.1, \\
\mu &= 0.3, \\
\alpha_\phi &= \alpha_\theta = \alpha_\psi = 4.0, \\
\beta_\phi &= \beta_\theta = \beta_\psi = 3.2, \\
\alpha_x &= \alpha_y = \alpha_z = 2.0, \\
\beta_x &= \beta_y = \beta_z = 1.8.
\end{aligned} \tag{50}$$

Simulation results are shown in Figures 5 and 6. The response curves of formation trajectory and attitude are given in Figure 5, which reveal that positions of six agents successfully achieve and maintain a predesigned formation shape. As depicted in Figure 6, it demonstrates the snapshots of position trajectories of the six robots at $t = 0$ s, $t = 5$ s, $t = 20$ s, and $t = 40$ s, respectively.

Case 2. The same formation task is carried out by implementing the event-triggered controller with a constant threshold parameter $\sigma(t) = \sigma_0 = 0.6$. The aforementioned results derived from theorems indicate that the formation control problem is feasible under the general event-triggered

TABLE 4: Comparison results in Cases 1 and 2.

Signals	Case 1	Case 2
Mean		
$e_{1\phi}^i$	-0.0281	-0.0856
$e_{1\theta}^i$	-0.0257	-0.0849
$\tilde{\psi}_{1i}$	-0.0319	-0.1147
\tilde{x}_{1i}	0.0156	0.0664
\tilde{y}_{1i}	0.0221	0.0702
\tilde{z}_{1i}	-0.0202	-0.0445
Standard deviation		
$e_{1\phi}^i$	0.1075	0.2490
$e_{1\theta}^i$	0.1196	0.2772
$\tilde{\psi}_{1i}$	0.1402	0.3128
\tilde{x}_{1i}	0.0809	0.1824
\tilde{y}_{1i}	0.0873	0.1905
\tilde{z}_{1i}	0.0894	0.1943

controller. Figure 7 shows the trajectories of positions and attitude angles within 50 s and Figure 8 shows the snapshots of position trajectories at $t = 0$ s, $t = 5$ s, $t = 20$ s, and $t = 40$ s, respectively.

The actual event release instants for all agents are shown in Figures 9 and 10. A quantitative comparison of how many events on each agent are actually transmitted is provided in Table 3, and Table 4 shows the mean and standard deviation for each error signal in $20 \text{ s} \leq t \leq 50 \text{ s}$.

Remark 14. Some conclusions could be summarized as follows:

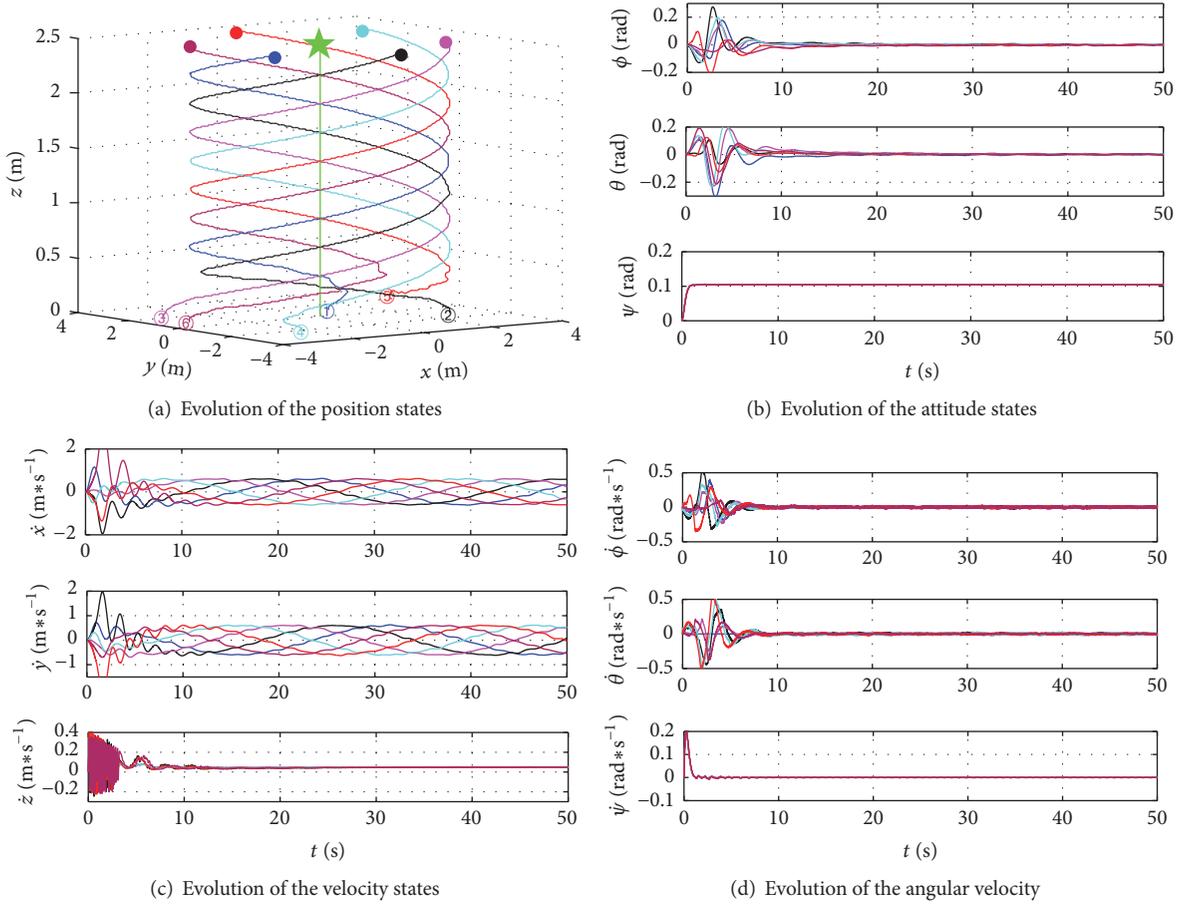


FIGURE 5: Trajectories of the agents within 50 s in Case 1.

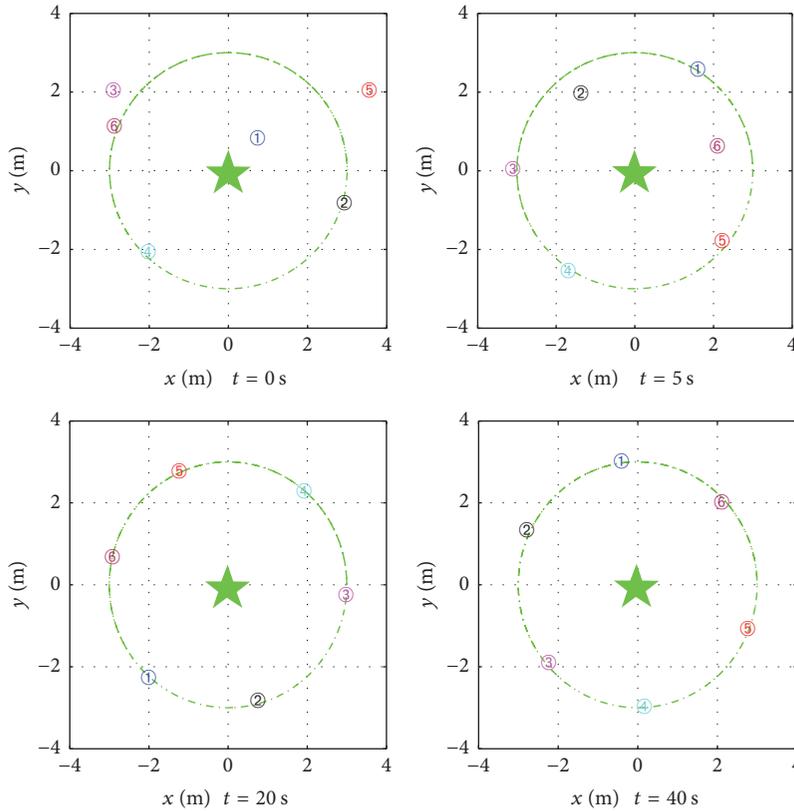


FIGURE 6: Position snapshots of the six agents in Case 1.

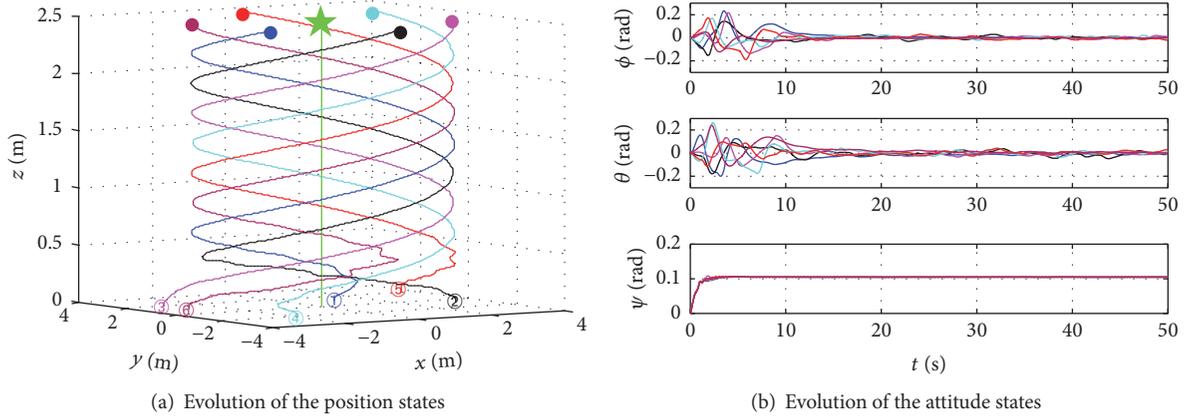


FIGURE 7: Trajectories of the agents within 50 s in Case 2.

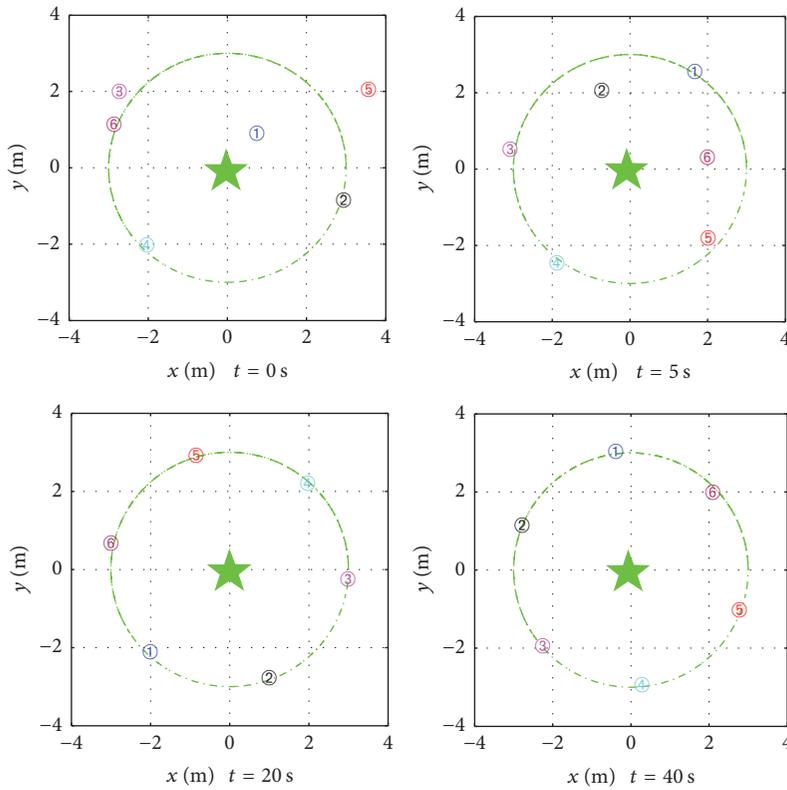


FIGURE 8: Position snapshots of the six agents in Case 2.

(1) Simulation results in both Cases 1 and 2 show that the formation shape could be guaranteed under Markovian switching topologies with partially unknown transition rates.

(2) In Figure 5, the convergence time of ψ and $\dot{\psi}$ (i.e., 0.80 s) is shorter than that of x , \dot{x} , y , \dot{y} , z , and \dot{z} (i.e., 7.20 s); these plots have different time horizons; it confirms the two-time-scale structure controller derived from the perturbing parameter σ_1 . Particularly, the reason why the convergence time of ϕ and θ in Figure 5 is as long as that of the position states is that ϕ and θ are tracking the reference ϕ_{id} and θ_{id} rather than zero.

(3) As shown in Figures 6 and 8, at $t = 0$ s, $t = 20$ s, and $t = 40$ s, the real shape of six agents is always uniform. When $t = 5$ s, there is a little difference because the graph is randomly governed by Markov chains.

(4) Compared with Case 2, there are more events for all agents in Case 1, which means that it has more potential to reduce the occupancy of limited communication resources in Case 2. That is because the threshold parameter $\sigma(t)$ is fixed as σ_0 all the time, while in Case 1, $\sigma(t)$ is monotonically nonincreasing (shown in Figure 11). Generally, the larger the value of the threshold parameter is, the less events are derived.

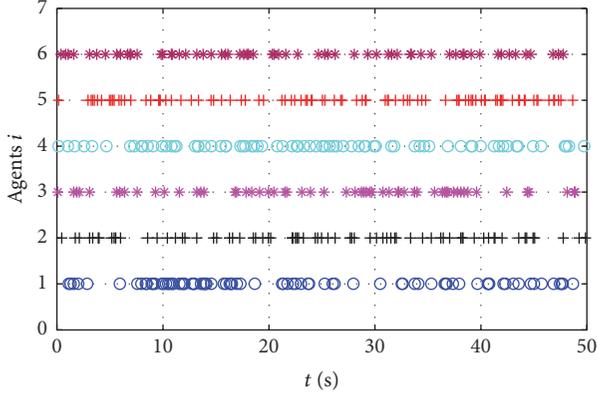


FIGURE 9: Events times of all agents in Case 1.

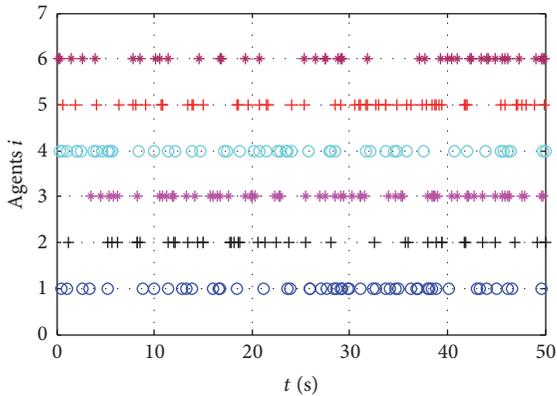


FIGURE 10: Events times of all agents in Case 2.

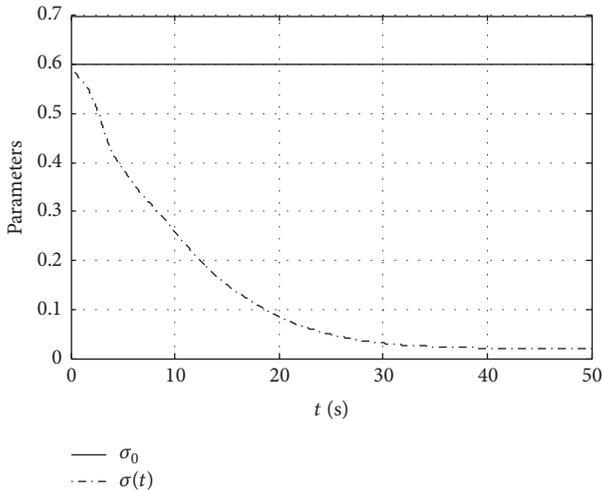


FIGURE 11: The threshold parameters in Cases 1 and 2.

(5) Based on most of the computed performance indexes in Table 4, the controller in Case 1 can achieve better formation performance than that in Case 2. Although less events are derived by using a fixed threshold parameter, the formation performance is compromised.

5. Conclusion

In this paper, formation control problem of multiple quadrotor UAVs under Markovian switching topologies is addressed through an event-triggered control strategy. Markovian switching topologies are redesigned through utilizing the Markov chains to the edge set, which could recover the traditional ones by adjusting the modes of edges and the transition rates. Then, a predesigned formation shape can be kept along with a distributed formation controller constructed with a reasonably event-triggered updated rule, even subject to the unknown transition rates. Rigorous analyses of the convergence results are obtained based on singular perturbations theory and Lyapunov stability theory. In addition, Zeno behavior is excluded for the triggered time sequences. Simulation results have been given to illustrate the effectiveness of the proposed control strategy. In some of the cases, the velocity states are unmeasurable. Since that, event-triggered formation control problem for multiple quadrotor UAVs without collision subject to output feedback will be considered in our future work.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by NSFC under Grant 61473248 and Natural Science Foundation of Hebei Province under Grant F2016203496.

References

- [1] J. Yu, J. Ji, Z. Miao, and J. Zhou, "Adaptive formation control of networked Lagrangian systems with a moving leader," *Nonlinear Dynamics*, pp. 1–12, 2017.
- [2] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.
- [3] A. Franchi, C. Secchi, M. Ryll, H. Büthoff, and P. R. Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 57–68, 2012.
- [4] W. Hao and B. Xian, "Nonlinear adaptive fault-tolerant control for a quadrotor UAV based on immersion and invariance methodology," *Nonlinear Dynamics*, pp. 1–14, 2017.
- [5] A. Abdessameud and A. Tayebi, "Formation control of VTOL Unmanned Aerial Vehicles with communication delays," *Automatica*, vol. 47, no. 11, pp. 2383–2394, 2011.
- [6] H. Qiu and H. Duan, "Receding horizon control for multiple UAV formation flight based on modified brain storm optimization," *Nonlinear Dynamics*, vol. 78, no. 3, pp. 1973–1988, 2014.
- [7] Y.-Y. Chen, Z.-Z. Wang, Y. Zhang, C.-L. Liu, and Q. Wang, "A geometric extension design for spherical formation tracking control of second-order agents in unknown spatiotemporal flowfields," *Nonlinear Dynamics*, vol. 88, no. 2, pp. 1173–1186, 2017.
- [8] Z. Hou, W. Wang, G. Zhang, and C. Han, "A survey on the formation control of multiple quadrotors," in *Proceedings of the 2017 14th International Conference on Ubiquitous Robots and*

- Ambient Intelligence (URAI)*, pp. 219–225, IEEE, Jeju, South Korea, 2017.
- [9] Y. Wang, Q. Wu, and Y. Wang, “Distributed cooperative control for multiple quadrotor systems via dynamic surface control,” *Nonlinear Dynamics*, vol. 75, no. 3, pp. 513–527, 2014.
- [10] Y. Xia, X. Na, Z. Sun, and J. Chen, “Formation control and collision avoidance for multi-agent systems based on position estimation,” *ISA Transactions*, vol. 61, pp. 287–296, 2016.
- [11] J. Ghommam, L. F. Luque-Vega, B. Castillo-Toledo, and M. Saad, “Three-dimensional distributed tracking control for multiple quadrotor helicopters,” *Journal of The Franklin Institute*, vol. 353, no. 10, pp. 2344–2372, 2016.
- [12] H. Du, W. Zhu, G. Wen, and D. Wu, “Finite-time formation control for a group of quadrotor aircraft,” *Aerospace Science and Technology*, vol. 69, pp. 609–616, 2017.
- [13] X. Dong, L. Han, Q. Li, and Z. Ren, “Time-varying formation control for double-integrator multi-agent systems with jointly connected topologies,” *International Journal of Systems Science*, vol. 47, no. 16, pp. 3829–3838, 2016.
- [14] X. Dong, Y. Zhou, Z. Ren, and Y. Zhong, “Time-Varying Formation Tracking for Second-Order Multi-Agent Systems Subjected to Switching Topologies With Application to Quadrotor Formation Flying,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 5014–5024, 2017.
- [15] F. Liao, R. Teo, J. L. Wang, X. Dong, F. Lin, and K. Peng, “Distributed Formation and Reconfiguration Control of VTOL UAVs,” *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 270–277, 2017.
- [16] X. Dong and G. Hu, “Time-varying formation control for general linear multi-agent systems with switching directed topologies,” *Automatica*, vol. 73, pp. 47–55, 2016.
- [17] Y. Shang and Y. Ye, “Leader-follower fixed-time group consensus control of multiagent systems under directed topology,” *Complexity*, vol. 2017, Article ID 3465076, 9 pages, 2017.
- [18] M. Faraji-Niri, M. R. Jahed-Motlagh, and M. Barkhordari-Yazdi, “Stabilization of active fault-tolerant control systems by uncertain nonhomogeneous Markovian jump models,” *Complexity*, vol. 21, no. S1, pp. 318–329, 2016.
- [19] X. Ge and Q.-L. Han, “Consensus of Multiagent Systems Subject to Partially Accessible and Overlapping Markovian Network Topologies,” *IEEE Transactions on Cybernetics*, vol. 47, no. 8, pp. 1807–1819, 2017.
- [20] A. Wang, T. Dong, and X. Liao, “Event-triggered synchronization strategy for complex dynamical networks with the Markovian switching topologies,” *Neural Networks*, vol. 74, pp. 52–57, 2016.
- [21] L. Rong and H. Shen, “Distributed containment control of second-order multiagent systems with input delays under general protocols,” *Complexity*, vol. 21, no. 6, pp. 112–120, 2016.
- [22] C. Viel, S. Bertrand, M. Kieffer, and H. Piet-Lahanier, “Distributed event-triggered control for multi-agent formation stabilization,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8025–8030, 2017.
- [23] X.-M. Zhang and Q.-L. Han, “Event-triggered dynamic output feedback control for networked control systems,” *IET Control Theory & Applications*, vol. 8, no. 4, pp. 226–234, 2014.
- [24] X. Ge and Q.-L. Han, “Distributed Formation Control of Networked Multi-Agent Systems Using a Dynamic Event-Triggered Communication Mechanism,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 8118–8127, 2017.
- [25] X. Yi, J. Wei, D. V. Dimarogonas, and K. H. Johansson, “Formation control for multi-agent systems with connectivity preservation and event-triggered controllers,” <https://arxiv.org/abs/1611.03105v4>, 2016.
- [26] X. Ge, Q.-L. Han, and F. Yang, “Event-Based Set-Membership Leader-Following Consensus of Networked Multi-Agent Systems Subject to Limited Communication Resources and Unknown-But-Bounded Noise,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 5045–5054, 2017.
- [27] C. Peng, Y. Song, P. Xie X et al., “Event-triggered output tracking control for wireless networked control systems with communication delays and data dropouts,” *IET Control Theory & Applications*, vol. 10, no. 17, pp. 2195–2203, 2016.
- [28] H. Zhang, R. Yang, H. Yan, and F. Yang, “ H_∞ consensus of event-based multi-agent systems with switching topology,” *Information Sciences*, vol. 370, pp. 623–635, 2016.
- [29] C. Peng, J. Zhang, and Y. C. Yan, “Adaptive event-triggering H_∞ load frequency control for network-based power systems,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1685–1694, 2018.
- [30] J. Lee, R. Mukherjee, and H. K. Khalil, “Output feedback stabilization of inverted pendulum on a cart in the presence of uncertainties,” *Automatica*, vol. 54, pp. 146–157, 2015.
- [31] M. T. Ravichandran and A. D. Mahindrakar, “Robust stabilization of a class of underactuated mechanical systems using time scaling and Lyapunov redesign,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 9, pp. 4299–4313, 2011.
- [32] S. González-Vázquez and J. Moreno-Valenzuela, “Motion control of a quadrotor aircraft via singular perturbations,” *International Journal of Advanced Robotic Systems*, vol. 10, 2013.
- [33] J. Zhou, H. Dong, and J. Feng, “Event-triggered communication for synchronization of Markovian jump delayed complex networks with partially unknown transition rates,” *Applied Mathematics and Computation*, vol. 293, pp. 617–629, 2017.
- [34] D. Xie, S. Xu, Z. Li, and Y. Zou, “Event-triggered consensus control for second-order multi-agent systems,” *IET Control Theory & Applications*, vol. 9, no. 5, pp. 667–680, 2015.
- [35] K. Vladimír, “A Contribution to Matrix Quadratic Equations,” *IEEE Transactions on Automatic Control*, vol. 17, no. 3, pp. 344–347, 1972.

Research Article

Intelligent UAV Map Generation and Discrete Path Planning for Search and Rescue Operations

Víctor San Juan,¹ Matilde Santos ,¹ and José Manuel Andújar ²

¹Department of Computer Architecture and Automatic Control, Faculty of Informatics, University Complutense of Madrid, 28040 Madrid, Spain

²Department of Electronic, Computer Science and Automatic Engineering, University of Huelva, 21819 Huelva, Spain

Correspondence should be addressed to Matilde Santos; msantos@ucm.es

Received 9 December 2017; Accepted 28 February 2018; Published 23 April 2018

Academic Editor: Dimitri Volchenkov

Copyright © 2018 Víctor San Juan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Search and Rescue Operations (SAR) take place in any emergency situation where people are involved and their lives are at risk. These operations are nowadays carried out with the help of advanced technology, such as Unmanned Aerial Vehicles (UAVs). In this work, several methods are proposed to calculate the UAV discrete path planning. Previously, an intelligent characterization of the searching area is performed to estimate a potential risk/occupancy degree of the gridding map. This estimation is mainly based on fuzzy logic, considering different factors. Then, four methods are applied to calculate the path planning: an original proposal called attraction, fuzzy logic, ANFIS, and a PSO algorithm. All of them calculate the location of the waypoints to be followed by the UAVs to minimize the distance and the risk the people is exposed to. Then, these strategies are adapted to the possibility of having more than one UAV searching at the same time, and the swarm formation is discussed. Finally, these four solutions for path planning, including different number of UAVs, are tested in a real simulation scenario, and then the performance of each method is analyzed and compared with the others.

1. Introduction

Unfortunately, every year there are some news of natural disasters, like fires or earthquakes which affect cities, towns, and natural environment. Besides, we also see human tragedies as maritime accidents or people who get lost in hazardous places. All these situations, and many others, have one important aspect in common. They subject the life of many people involved in these situations in one way or another to danger and a direct risk.

The most critical aspect in these cases, even before starting to mitigate the risk (i.e., extinguishing a fire), is the search and rescue (SAR) of potential survivors. The SAR process is carried out by specialized teams that even risk their own lives to save others.

Fortunately, these SAR teams are, in most of the cases, well equipped with advanced technology, which nowadays includes Unmanned Aerial Vehicles (UAVs) to improve people searching and also to prevent rescue team from

going through risky places with no real evidences of finding survivors [1].

These autonomous vehicles have experienced a huge development during the last ten years. UAVs are nowadays used in many different applications such as defence, surveillance, or package delivery, among others [2]. But the unmanned aerial vehicle path planning problem is an important issue in the UAV mission planning [3].

Many of these applications use automatically controlled aerial vehicles, which are able to flight through a preplanned trajectory or even to make a real-time decision on the trajectories. Nevertheless, most of the commercial UAVs still use an off-board pilot and a fixed preplanned trajectory [4].

The final motivation of this work is to find people at risk as fast as possible in SAR operations, using fully automatic UAVs equipped with on-board cameras. Therefore, we proposed some strategies to optimize this searching. The path planning takes into account the hazard of a particular scenario, where there will be some places with higher risk

than others. To do this, we have first developed a methodology based on fuzzy logic to characterize the surface to be explored and to estimate the degree of potential risk and the probability of people to be at that spot.

Then, we have developed a discrete path planner to find people as fast as possible using the map generated by the previous analysis. Four intelligent discrete path planning strategies have been implemented and tested on a real scenario. Finally, we will show how the number of UAVs in swarm formation affects the speed and effectiveness of people searching.

The following topics are not included in the scope of this work: UAV on-board control and obstacle avoidance and UAV path following and image processing. In fact, we are not working with a model of the dynamics of the UAV. The path planning is calculated as a trajectory along some discrete points, corresponding to different cells in which the map of the searching area is divided. The UAV moves from one cell to another (the adjacent one) with a constant speed. The number of cells visited gives the length of the trajectory and therefore the time spent.

The paper is organized as follows: Section 2 summarizes the related works. Section 3 describes the generation of the fuzzy occupancy/risk map. In Section 4, four approaches for the intelligent discrete path planning are proposed and results on a real simulation scenario are presented and analyzed. The outcome of using several UAVs in the search and rescue mission is discussed in Section 5. Section 6 shows a case studied and Section 7 summarizes and discusses the global results. The paper ends with the conclusions and future work.

2. Background

Search and Rescue (SAR) operations must be carried out by the competent authorities in case of emergency that involves people in danger. The nature of these emergencies can be very different; therefore, the hazards will vary depending on the kind of emergency and consequently the SAR mission has also to be adapted to the scenario. Some emergency situations are, for instance, people lost in uninhabitable places, fires, aircraft or vessel accidents, natural disasters like landslides, floods, and earthquakes [5].

Autonomous robots and vehicles have been used to carry out missions in hazardous environments, such as operations in nuclear power plants, exploration of Mars, and surveillance of enemy forces in the battle field [6]. Indeed, UAVs are often used in SAR operations. The common operation of these UAVs is to be commanded remotely by a SAR team member [7]. The UAV is moved by its pilot to the desired areas in order to collect the relevant information for the SAR mission [8]. The payload of the UAV may be of different nature, usually cameras to record video or to take pictures, but it is also possible to find other sensors such as gas analyzers to detect smoke composition.

But still planning a SAR mission in terms of obtaining the path or trajectory is a challenging issue that requires careful consideration. The autonomous UAVs flight can be based on a preplanned path or can even be calculated online [9]. The level of autonomy achieved by the UAVs depends on

the methodology used to control the vehicle and to generate its routes [10]. Therefore, autonomous UAVs must have an auto path planning calculation functionality to be really autonomous.

Discrete search is an interesting topic nowadays, with many applications. For example, in built infrastructure monitoring, an efficient path planning algorithm is essential for robotic inspection of large surfaces using computer vision [11, 12] or for controlling autonomous agents in interactive virtual worlds, given the growing demands on the size and complexity of these virtual scenarios, even for areas of crowd animation and whole-body motion planning for virtual characters [13].

Discrete path planning has different approaches. The book by Kallmann and Kapadia [13] reviews the evolution of several related techniques for achieving intelligent navigation in discrete spaces, starting from classical planning and computational geometry techniques and then gradually moving towards more advanced topics, although being applied to interactive virtual worlds, to address shortest paths and limited time budgets. Some of the path following algorithms more often used for UAVs are presented in [14], including a comparison between them in order to help in the selection of one or another. Geometrical methods such as spline interpolation are used in order to get smoother curves through a number of waypoints [15, 16]. In [7], path planning is defined using a Monte Carlo searching of the best path using online detected parameters.

The optimization of the path between waypoints is also addressed by several authors. Many of them use genetic algorithms (GA) [17–20]. Other proposals apply estimation of distribution algorithms [21] or A^* searching [22]. Generally, in these cases, the UAV needs to move in a known environment, and the optimal path is obtained to avoid obstacles, radar zones, buildings, battery run out, and so on.

3. Generation of the Risk/Occupancy Map

In this process of searching and rescue, it is desirable to exploit the increasing availability of data about a disaster that come from different sources. In particular, such information can be a valuable resource to drive the planning of UAV flight paths over a space in order to discover people who are in danger [8]. Therefore, the first step will be to estimate and define the searching area, including all the available data, and define the desired area coverage level, in order to generate the map that will be used for the discrete UAVs path planning.

This area characterization involves the analysis of the surface where the search will take place and the quantification of some characteristics. The input is a map of the area to be covered. This map is divided into a grid of cells. Each of these cells will have an assigned value of potential risk and occupancy that will be used to calculate the waypoints that the UAV will follow.

To make the grid partition, it is assumed that the UAV will be able to track the whole cell when located at the geometrical centre of it. That is, the size of the cells will not be bigger than the area covered by the on-board camera of the UAV. But this is difficult to be estimated since it depends on two

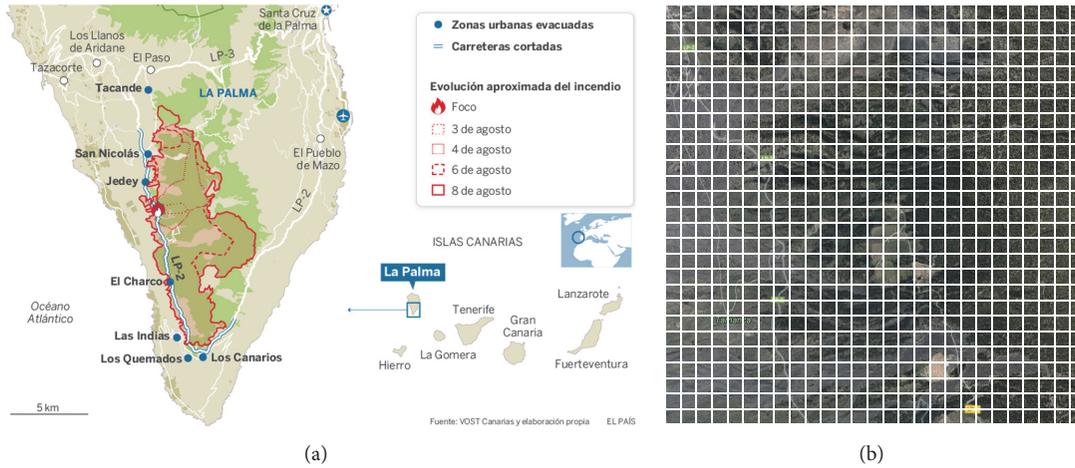


FIGURE 1: La Palma fire map (a) and gridding searching area (b).

main factors: the altitude of the UAV and the opening angle of the camera lens. At the same time, the altitude of the UAV, in case of visual tracking, will be limited for the camera characteristics. In fact, if the altitude is too high, the camera will not be able to focus correctly, and therefore the tracking will not work. Taking this into account and considering that we are working with a discrete scenario in a theoretical way, a limit of 100×100 m cell size has been established. This value can be scaled according to the technical specifications of the sensors used.

Once the size of the grid is determined, the number of cells is calculated straightforward for the particular zone we are exploring.

The application scenario of this work is a real case. In August 2016, a forest fire was declared in the island of La Palma, in the Canary Islands (Spain). The fire lasted several days affecting over 4000 hectares, that is, 6.8% of the island surface (Figure 1(a)). Fortunately, there were no civil victims, but sadly one person died during the extinguishing phase. More than 2500 people needed to be evacuated from their homes during the fire.

In Figure 1(a), we can see how the fire grew throughout the days. The simulation scenario corresponds to the third day of the fire (dotted red line). Three days is a reasonable time to search for a person. A rectangular area, larger than the fire area, was selected to be tracked and partitioned. The grid partition has 24×24 cells (Figure 1(b)).

Once the region to be covered is defined and the grid partition obtained, the potential risk/occupancy value for each cell is going to be estimated.

3.1. Potential Risk/Occupancy Estimation. The potential risk/occupancy map aims to give a value of the possibility of each cell to be occupied by people and, in that case, the potential hazard for the life of that person.

To calculate this value some information from different sources is necessary. To obtain the potential occupancy we need to know, for example, how often the area is visited by people. The potential risk can be given by the types

of emergencies. This information can be obtained from an analysis of the area by experts and/or from historical data. We will consider two main components to define the map, called terrain and emergency factors. They are defined by fuzzy inference systems (FIS).

The terrain FIS has as inputs the variables named *staying*, *hazard*, and *transit* (Table 1). The output is *Pterrain*, which is the contribution of this factor to the map.

Two of these input variables have been defined as fuzzy (*staying* and *hazard*), and the third one (*transit*) will be considered as a crisp weight to be applied to the output. Both the *staying* and the *hazard* linguistic variables are normalized between 0 and 1. The fuzzy sets are given by triangular and trapezoidal membership functions. The labels are Low, Mid, and High for *staying* (Figure 2(a)) and Low and High for *hazard* (Figure 2(b)).

The output is defined by three triangular fuzzy sets (Low, Mid, and High). The fuzzy rules, obtained by applying expert knowledge, are as follows:

Rule	Staying	Hazard	Output
1	high	high	high
2	high	low	low
3	medium	high	high
4	medium	low	medium
5	low	high	medium
6	low	low	low

The output of this fuzzy system is finally weighted by the *transit* variable. That is, the terrain FIS result is multiplied by it to obtain the final *Pterrain* value. The *transit* variable will be low, in general, in places where the access is easy and it is difficult that people get lost and stay there (low occupancy); if something happened, they will have been already found and evacuated. Besides, the risk will be also low as there will be help to solve that situation. Therefore, the transit value will lower the value of *Pterrain* in this case. And vice versa.

This *Pterrain* factor is part of the contribution to the final value that will be assigned to each cell in order to rank them to be visited (see equation (1)).

TABLE I: Terrain factor inputs.

Staying	It describes whether the area is often visited by people or not. High values increase potential occupancy. This information can be obtained from an analysis of the area and/or from historical data.
Hazard	It quantifies the hazards inherent to the area, affected by wild animals, weather conditions, terrain access, etc. Higher values increase potential risk.
Transit	It expresses whether the area is usually transited by people, meaning people could be more easily found by other people. The potential occupancy and the risk of these areas decrease.

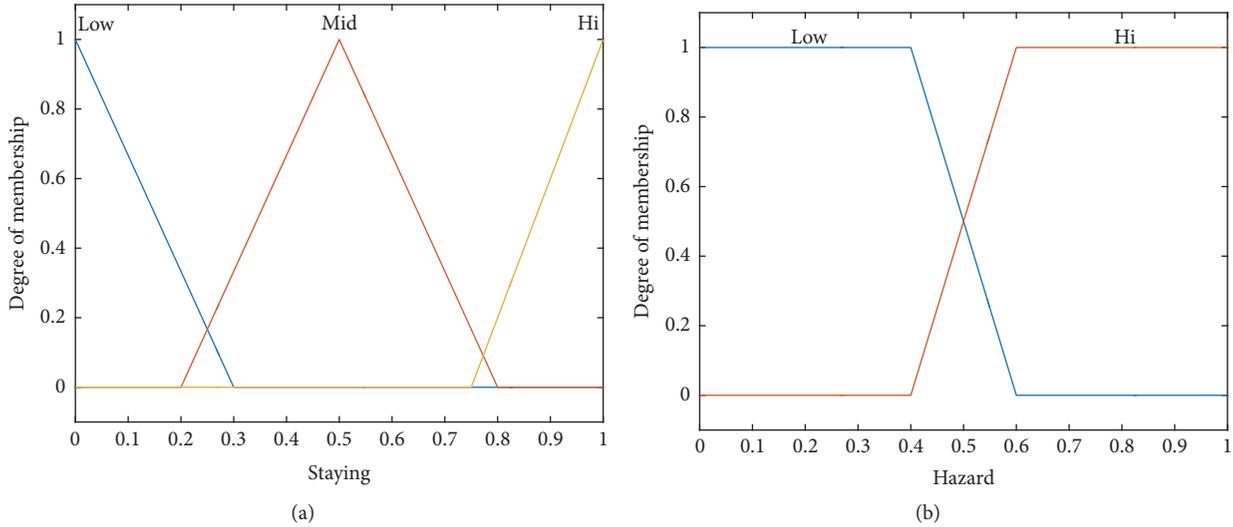
FIGURE 2: *Staying* (a) and *hazard* (b) fuzzy variables.

Table 3 shows an example of the values that these variables may have. The values given to the *staying* factor are estimated according to the terrain accessibility and orography. Inaccessible areas or areas with difficult access have smaller staying factor than others like roads or plains. The *hazard* has been estimated according to the terrain nature as well. The transit value depends on the presence of roads or buildings on that area.

To give an example, if the staying input is high, the contribution of the terrain factor to the risk/occupancy value of a cell should be high (there will be people there) but at the same time this value will be balanced for two other factors, transit and hazard. Anyway, if the staying is high it is very likely that the hazard of that place is low (no dangers) and in addition the transit factor will be low (roads, accessible places), decreasing the final contribution to the risk.

The same reasoning has been applied to determine the emergency FIS. The fuzzy input variables are *affected* and *injuries*, where *affected* is defined by three fuzzy sets with triangular membership functions and *injuries* has two trapezoidal ones. The third input, the *SAR team position*, is a binary variable with value equal to one except in the case the rescue team is already on the spot, in which case is 0. The definition of these variables is shown in Table 2. The output of this FIS is called *Pemergency* (see equation (1)).

For example, if a particular zone is strongly affected, and the type of accidents is dangerous for people lives, the value

of this FIS will be high. It will be pondered by whether the SAR team is near or even already there, decreasing the risk and the occupancy.

Finally, regarding the importance of *Pterrain* and *Pemergency* factors on the final decision about the path planning, the meaning of them is as follows. The first one is more important since it represents the zones that have to be visited because is very likely that people are there. Besides, they can also be hazardous areas. However, *Pemergency* represents the zones that should be visited because they are potentially dangerous for people due to some kind of emergency, in case people are there.

3.2. Potential Risk/Occupancy Map. We estimate the result of these fuzzy inference systems, *Terrain* and *Emergency*, for each cell of the map of the searching area of La Palma. A map is generated for both of them (Figure 3).

The colour represents the value of the corresponding factor for that cell, from dark blue (value 0) to yellow (value 1). Figure 3(a) shows the terrain contribution. A high staying factor can be observed in the middle of the map, and the secondary roads are clearly marked in blue. The emergency contribution is shown in Figure 3(b). There are three differentiated zones: the not affected one, the one in the middle that has been seriously affected by the fire and a pale green one.

TABLE 2: Emergency factor inputs.

Affected	It represents how the area is affected by an emergency. High values increase potential risk and low values the opposite.
Injuries	It describes the possibility of people being injured. High values increase potential risk.
SAR team position	It describes the location of the SAR team, e.g., how close to the emergency area they are. Indeed if the spot is already covered, the necessity of going to that point is cancelled.

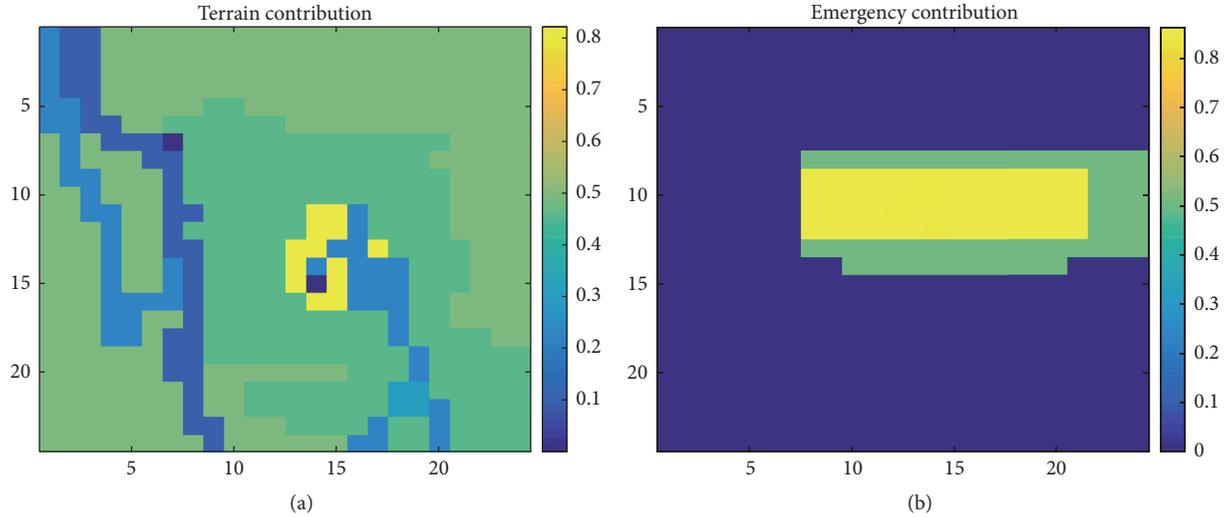


FIGURE 3: La Palma terrain (a) and emergency (b) maps.

TABLE 3: Terrain input values.

Description	<i>Transit</i>	<i>Staying</i>	<i>Hazard</i>
Main roads	0.2	0.8	0.1
Sec roads 1	0.4	0.6	0.2
Sec roads 2	0.6	0.4	0.2
Forest nearby	0.4	0.4	0.3
Plain	0.3	0.7	0.2
Forest type 1	0.6	0.4	0.4
Forest type 2	0.8	0.3	0.5
Forest type 3	0.9	0.1	0.7
Hills	0.5	0.5	0.3
Desert	0.9	0.2	0.6

Then both contributions are combined according to the relation given by (1) to obtain the potential risk/occupancy value of the zone caught on fire of La Palma (Figure 4), which will be the input for the discrete path planning:

$$P = 0.5 \cdot P_{\text{TERRAIN}} + 0.4 \cdot P_{\text{EMERGENCY}} + 0.1 \cdot P_{\text{HISTORICAL}} \quad (1)$$

The third term of (1), the historical contribution, quantifies the possibility of an event to happen again in the same place. It will be 1 in the positive case and 0 otherwise.

It is possible to see the different elements of the area; for example, some roads can be clearly identified and the yellow

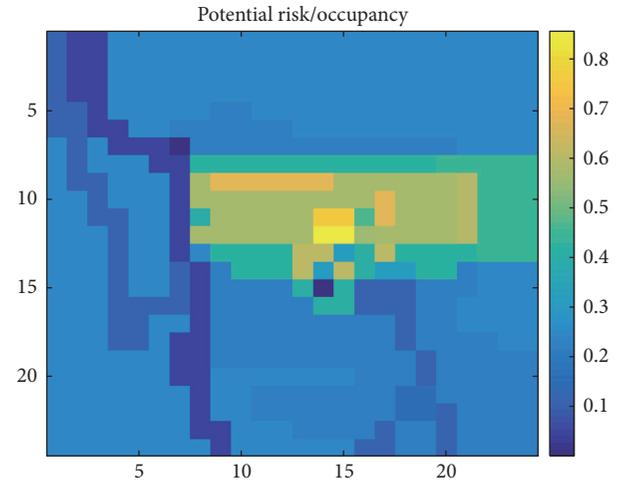


FIGURE 4: La Palma potential risk/occupancy map.

areas correspond to the forest, with higher risk values and where the fire started.

4. Discrete Path Planning

Four approaches for discrete path planning have been develop and applied, three of them using techniques that come from the artificial intelligence field. The final aim is to define the waypoints the UAV must follow, in order to find

as quickly as possible the people involved in a SAR operation and to minimize the risk of their lives.

To find people in the fastest way, it is necessary to explore the whole area minimizing the travelled distance. Moreover, not only the time but also the risk has to be minimized, so the UAVs should first go to those zones which are riskier or more hazardous. As it is difficult to solve both problems at the same time, firstly the areas with higher potential risk/occupancy will be visited and then the distance will be minimized as well, looking for a good trade-off between both aims.

As it has been said, the on-board camera is supposed to be able to track a whole cell of the grid partition and the waypoints will be located at the centre of these cells. The path planning consists then in setting the order the cells should be gone through.

To check the quality of the solution, the results are quantified by two figures:

- (i) Distance (d) is the total distance travelled by the UAV.
- (ii) Weight (w) is a factor that is calculated as

$$w = \sum_i^n \sum_j^m \text{order}(i, j) \cdot P(i, j), \quad (2)$$

where order is the correlative position of the cell (i, j) that is tracked and P is the potential risk/occupancy value of that cell. The meaning of this weight value is to quantify whether the most important cells, those with high P values, are tracked earlier than the other ones, so it is interesting to minimize it.

The scenario for the four approaches is the 24×24 cell map shown in Figure 4 that corresponds to the fire affected area in La Palma.

4.1. First Proposal: Attraction Approach. This so-called *attraction approach* is based on the potential fields' theory. In general, potential fields are used when the final point is known. A vector is calculated and its direction and magnitude depend on the distance to this final point, taking into account the obstacles or forbidden zones that could affect the trajectory.

In our case, the final point is unknown. To define it, an attraction value is estimated for each cell so that the cell with the highest attraction value will be the next waypoint. The attraction value depends directly on the potential risk/occupancy of the cell and the distance to the current position (3). In this expression (3), $A(i, j)$ represents the ij th cell of the attraction matrix A and $P(i, j)$ the potential risk/occupancy associated with that cell. Covered matrix contains the historical information about whether the cell has been already tracked. That is, covered is a matrix with the same dimension as P and A , with value 1 for nontracked cells and 0 for traced ones. The denominator is the Euclidean distance from the ij th cell to the current position

$$A(i, j) = \frac{P(i, j) \cdot \text{Covered}(i, j)}{\sqrt{(x_{ij} - x)^2 + (y_{ij} - y)^2}}. \quad (3)$$

The algorithm tries to go to the points with high P values but near the current position, so further cells with high

P values have less attraction than nearer points. Therefore, the trajectory is consistent and the UAV tracks low and high P cells minimizing the distance, but always being more attracted by higher P cells. If the distance was not taken into account, only P values would decide the waypoints, so the trajectory could have jumps from one waypoint to another and it may be bumpy.

Nevertheless, according to this proposal, the attraction by the distance is bigger than by the potential risk/occupancy value. So, in fact, the trajectory tries to first minimize the distance and then the potential risk/occupancy. While minimizing the distance is interesting, to minimize the weight is even more important because it is crucial to go first to the cells where the probability of finding someone is higher. Thus, (3) is changed to (4) in order to have a more balanced relation between those factors:

$$A(i, j) = \frac{P(i, j) \cdot \text{Covered}(i, j)}{\exp\left[\sqrt{(x_{ij} - x)^2 + (y_{ij} - y)^2}\right]}. \quad (4)$$

In (4), the exponential function of the distance has been chosen because of its asymptotic behaviour. The very close cells will take advantage of the distance, but in a short term all the cells will be similarly affected, in opposition to previous linear relationship, where further cells had less chance to be explored than nearer ones.

Two more improvements are made in order to get a better solution. The first one is about the continuity of the trajectory. To avoid jumps between consecutive cells a sequence of conditional rules is implemented. According to this, if two consecutive waypoints are located on the same vertical or horizontal line, all cells between these two points become automatically waypoints. This way these cells are considered already tracked and the UAV will not have to go backwards to explore them.

The second enhancement of the algorithm is about the waypoint selection. The cell with the maximum attraction is chosen as waypoint, but sometimes more than one cell has the same maximum value. Initially the first maximum found was selected, so the path depended on the starting point. A control strategy has been applied to avoid this bias. The first waypoint will be the cell that, having the maximum attraction value, is surrounded by cells with higher P values than the other candidates; that is, it is located in areas with more P density. A matrix with this density, Pg , is generated, according to the following formula:

$$Pg(i, j) = \sum_{n=i-2}^{i+2} \sum_{m=j-2}^{j+2} \frac{P(m, n)}{\sqrt{(x_{ij} - x_{nm})^2 + (y_{ij} - y_{nm})^2}}, \quad (5)$$

$\forall n \neq i, m \neq j.$

Then, after finding the maximum values of attraction of matrix A (4), a searching for the maximum Pg values (5) is carried out. The cell with highest density among the maximum attractions cells is selected as the next waypoint.

With all these modifications, a simulation is run on La Palma map to show this approach (Figure 5).

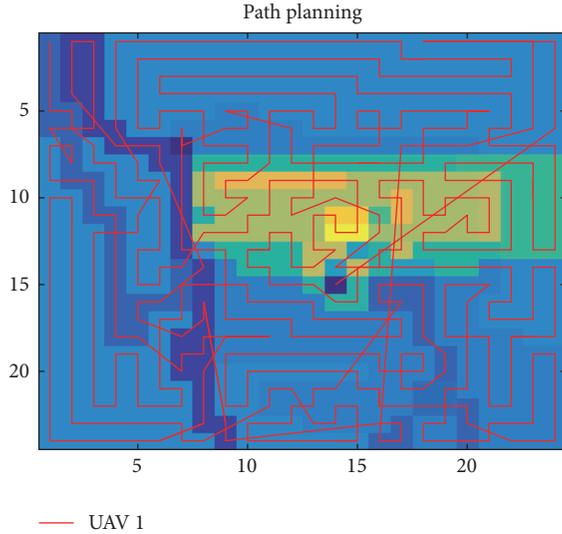


FIGURE 5: La Palma attraction path planning.

The trajectory starts at the top left corner and covers the entire surface. It seems to be consistent, with good continuity. The value of the distance is 680 units and the weight is 37392. As expected, the trajectory starts going across the light blue cells to the green ones, to quickly cover the yellow part of the map (higher P values). When the UAV gets near yellow cells, it goes directly to it and then covers all the nearer low P cells, without wasting time exploring low P cells that are automatically considered visited. Then it goes to the other yellow zones and so on.

To summarize, this original proposal works well and it is quite simple, but still the behaviour could be improved because there are some loops and paths cross each other. Other strategies will be tested.

4.2. Fuzzy Logic Approach. A different approach using fuzzy logic has been also implemented. A Mamdani Fuzzy Inference System (FIS) is designed with two inputs, P (potential risk/occupancy) and d (distance), and one output, attraction matrix (A). This way the contribution of the potential risk/occupancy value and the distance to the current position are separated and a nonlinear relationship between them is generated. For each input, three triangular and trapezoidal fuzzy sets have been assigned (high, medium, and low) and two for the output (high and low). The nine fuzzy rules are given in Table 4.

Simulations were carried out on different scenarios to test this initial fuzzy approach. The results were worse than with the previous solution. When analyzing the evolution of the trajectory, we found that the UAV prefers to go to an adjacent cell with a lower P value instead of going to further cells with high P values. In fact, the yellow cells are first surrounded by the UAV but not tracked until later.

Although a FIS is usually designed and adjusted by an expert who knows the behaviour of the system, in this case the knowledge was quite poor. So, as the target is to minimize distance and risk, an optimization method can be used to tune

TABLE 4: Rules of the fuzzy approach.

Rule	P	D	A
1	High	High	High
2	Medium	High	Low
3	Low	High	Low
4	High	Medium	High
5	Medium	Medium	Low
6	Low	Medium	Low
7	High	Low	High
8	Medium	Low	High
9	Low	Low	High

TABLE 5: GA parameters.

Generations	300
Initial population	200
Stall generations limit	100
Average relative change limit (TolFun)	10^{-10}
Crossover fraction	0.8
Mutation distribution	Gaussian, scale, and shrink equal to 1

the membership functions. That is, an evolutionary FIS is implemented, which consists of the same FIS proposed above but adjusted with genetic algorithms.

The configuration of the GA must include the 32 parameters of the trapezoidal membership functions of the FIS. Each trapezoidal membership function is defined by 4 points: x_1 , x_2 , x_3 , and x_4 . The value of these parameters must be correlative; that is, x_1 must be smaller than x_2 , x_2 than x_3 , and so on. These restrictions are defined by the linear constraints $Ax < b$, being $b = 0.001$ in our case.

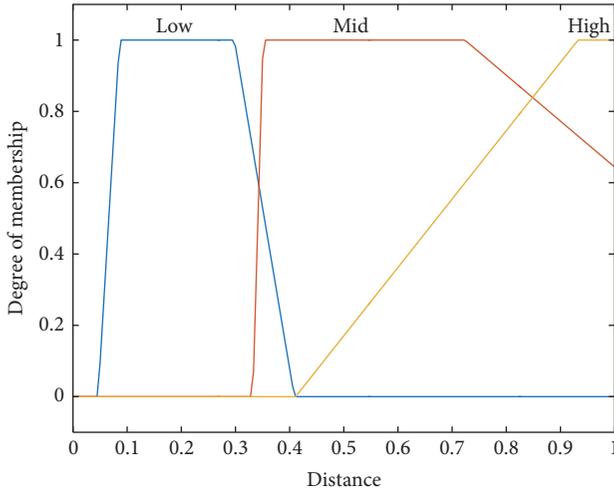
The fitness criterion is a function of the weight and the distance, both normalized in the range $[0, 1]$ and to be minimized as follows:

$$f = \left[1 - \frac{\sum_{i=1}^{\max(i)} \sum_{j=1}^{\max(j)} P(i, j)}{w} \right] + \left[1 - \frac{\max(i) \cdot \max(j)}{d} \right]. \quad (6)$$

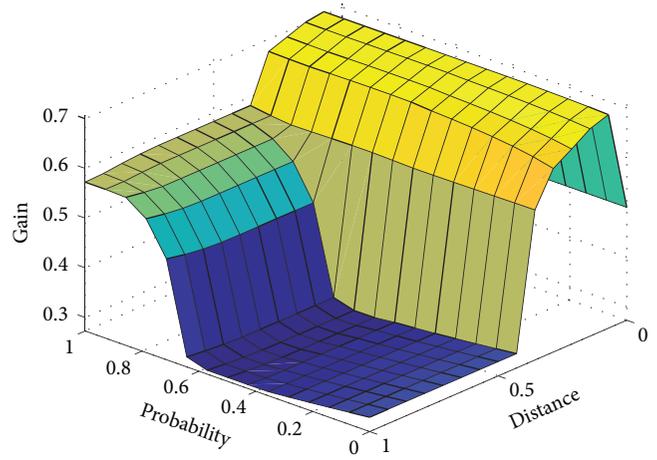
The rest of the GA parameters are shown in Table 5.

Initially, the process was so computationally slow that makes the problem unaffordable for this work. In order to reduce the computational time, we consider a smaller P map, and once the membership functions have been adjusted, it is scaled to a bigger one to see how it fits. Therefore, the GA is executed with the same configuration described but for a 5×5 P matrix.

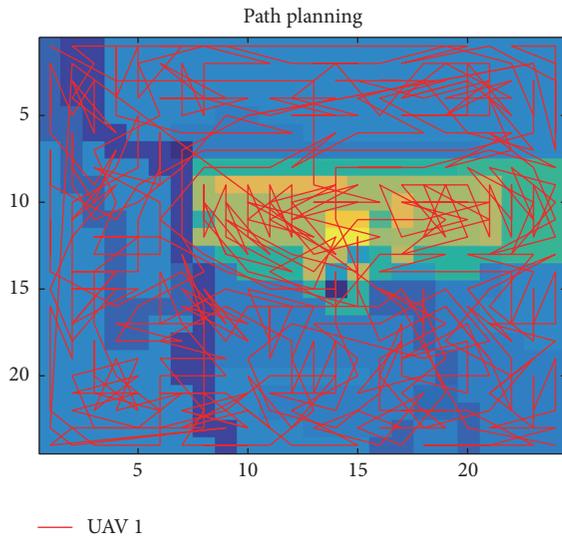
As an example, Figure 6 shows the membership functions of the input variable distance (a) and the output surface (b) after the GA adjustment. In the surface it is possible to see high output values for short distances and also for high P values in long distances.



(a)



(b)

FIGURE 6: Membership function of input *distance* (a) and fuzzy output *surface* (b).

— UAV 1

FIGURE 7: La Palma fuzzy logic path planning.

A test is now carried out with the 25×25 P map of La Palma (Figure 7), using the membership functions calculated for the 5×5 map.

The results are $d = 1497$ and $w = 35951$. Although the distance is much longer than in the attraction solution, weight is lower. Therefore we can say this solution also meets the target but with a stronger tendency to minimize risk/occupancy versus distance. However, the small angles that represent the transitions between two consecutive waypoints are not realistic and make the trajectory unaffordable for a UAV.

The strength of this approach is the fast convergence to the high P value cells, visiting later the lower P cells. This means that, even covering all the cells in a longer time, since the weight is lower and it also quantifies the occupancy, the probability of finding people in a short time is higher.

4.3. ANFIS Approach. Another solution is implemented using an Adaptive-Network-based Fuzzy Inference System (ANFIS) [23]. The FIS is the same as the previous one but the membership functions are now modified by the ANFIS algorithm.

Besides, a new input variable is included, the density matrix Pg (5). Now the system is able to give higher attraction values to higher density areas, instead of only using this factor for the selection of the maximum.

Therefore, the inputs are P , d , and Pg . The first two are normalized and the range of Pg is $[0.5, 5]$. Three Gaussian membership functions are calculated for each variable. The output is again attraction matrix A . But ANFIS requires samples for training. Hence, we have generated 80 synthetic data sets manually. The training is performed during 50 epochs (Figure 8(a)). An example of the membership functions generated for the new variable Pg is shown in Figure 8(b).

The new generated control surfaces are shown in Figure 9. It is possible to observe the difference with the one shown in Figure 6(b). In this case they are smoother although clearly nonlinear.

Once the ANFIS has been generated, simulations are run on the real scenario of La Palma. The trajectory is shown in Figure 10. The results obtained are $d = 808$ and $w = 36284$.

In this case, compared with the fuzzy logic solution ($d = 1497$ and $w = 35951$), the weight is higher but distance is much shorter, so it is not so easy to say which is better. However, what is really interesting of this approach is how the UAV goes over the highest P cells first, finishing the searching on low P cells.

4.4. Using Particle Swarm Optimization (PSO). Although the previous solutions seem to meet the main goal, they can be improved. In this case, an evolutive approach, a Particle Swarm Optimization (PSO) based solution is now proposed.

The GA was used to obtain the order of the cells that should be visited. GA worked well in small size maps but,

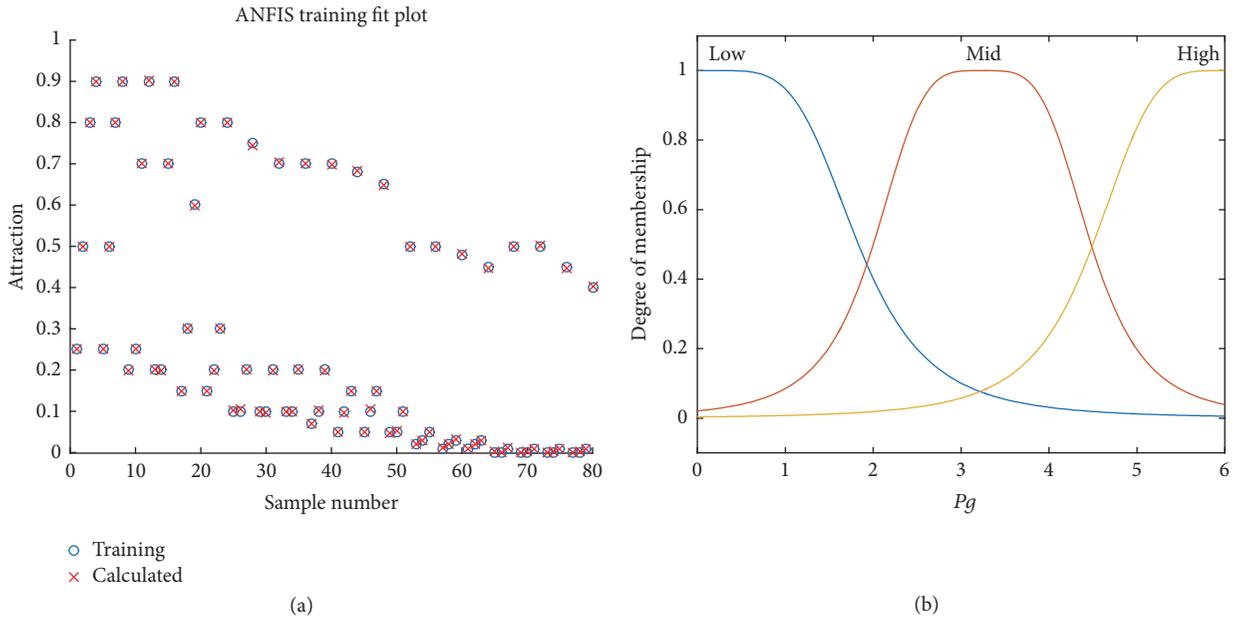


FIGURE 8: ANFIS training (a) and Pg membership functions (b).

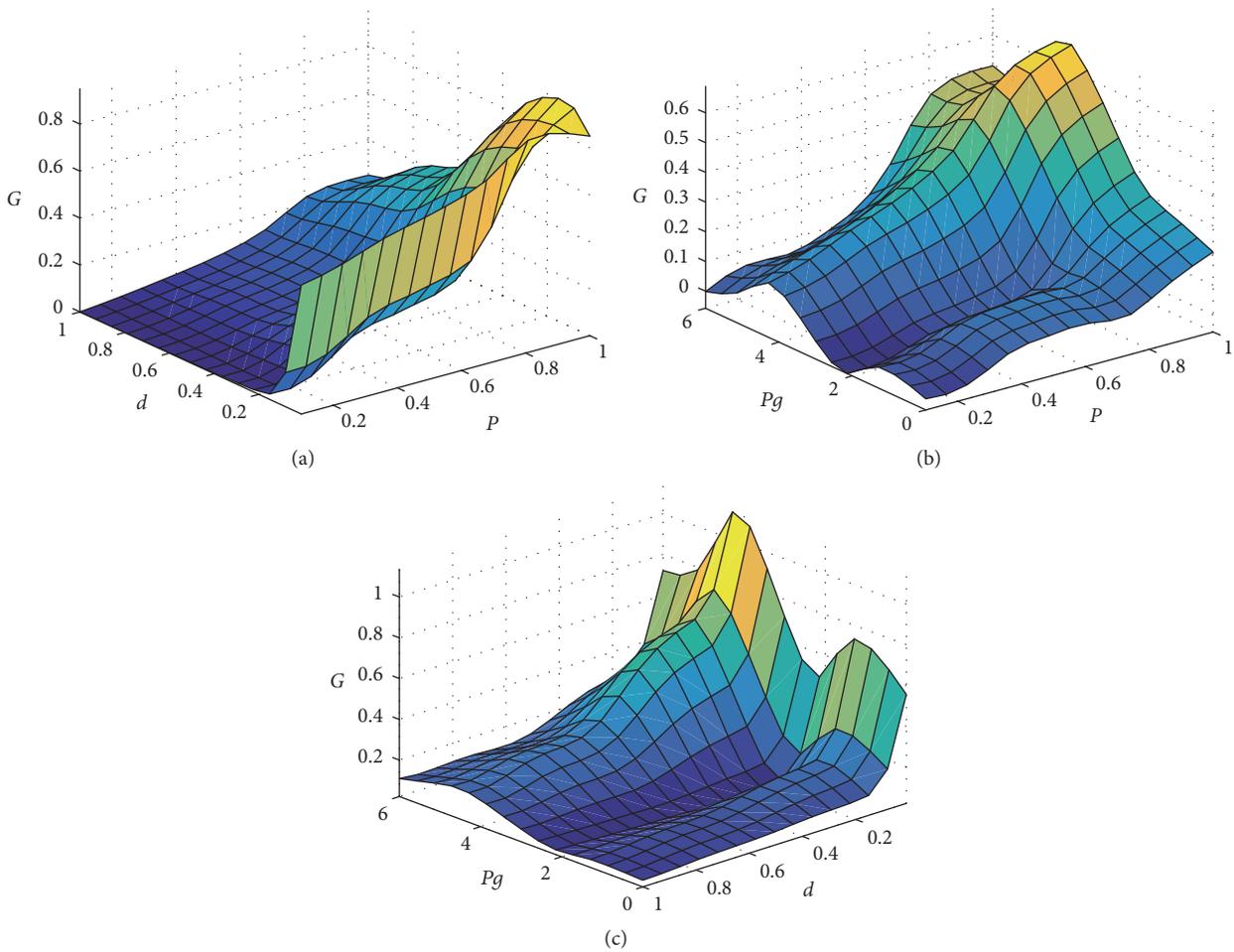


FIGURE 9: ANFIS fuzzy output surfaces: P - d , P - Pg , and Pg - d , respectively, from (a)-(c).

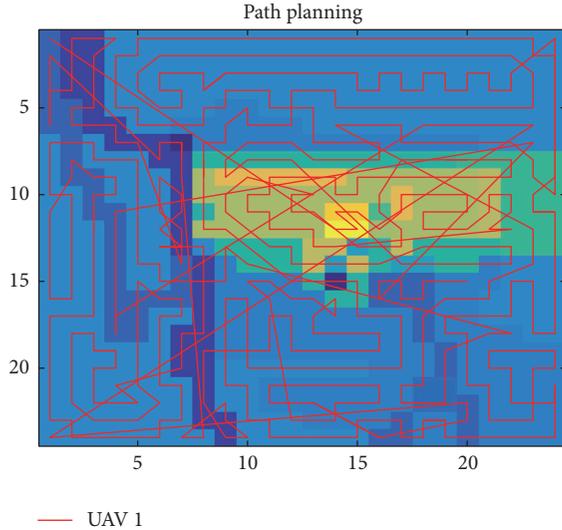


FIGURE 10: La Palma ANFIS path planning.

when scaled to the 25×25 map, some problems raised. In addition, in all the previous solutions, the waypoint calculation was very static, all the cells being tracked from its central point. Despite the assumptions made at the beginning regarding this way of tracking each cell, with this PSO solution, we try to find a continuous strategy.

The approach is quite different. We start from the same P map (24×24 cells), but every cell is divided by 10 to obtain a better discretization. Each new smaller cell has the same P value than the bigger cell that contains it. Once the map is so divided, the waypoints are searched applying the PSO algorithm.

The variables of the PSO fitness function are the coordinates of the waypoints. The calculation is structured in groups of 10 waypoints, so there will be 20 variables: 10 “ x ” coordinates and 10 “ y ” coordinates. Then, the distance, the weight, and the number of tracked cells are calculated. It is clear that the distance has to be minimized, and the tracked cells need to be maximized, what implies that the UAV is travelling over the maximum number of new cells, since the tracked ones will not be taken into account more than once.

To obtain the tracked cells, a straight path will be traced between two consecutive waypoints, and all the cells inside a square of 10×10 along the trajectory will be considered as tracked. That means that the UAV is continually taking images using the same area as in previous solutions. This is the first relevant modification.

Another important modification is that in this case the weight will be maximized. In the previous approaches, a minimization of the weight was interesting because it meant that the UAV first went to the areas with high P cells. However, with this approach, it is more interesting the opposite because, as said, the calculation is made in bunches of ten waypoints. So, if the weight is maximized for the first ten points, the weight of the second ten points will be lower. That is, the UAV is going over the high P cells first.

TABLE 6: PSO configuration parameters.

Creation distribution	Uniform
Maximum iterations	4000
Stall Iterations limit	20
Average relative change limit (TolFun)	10^{-6}
Swarm size	100
Inertia range	[0.1, 1.1]

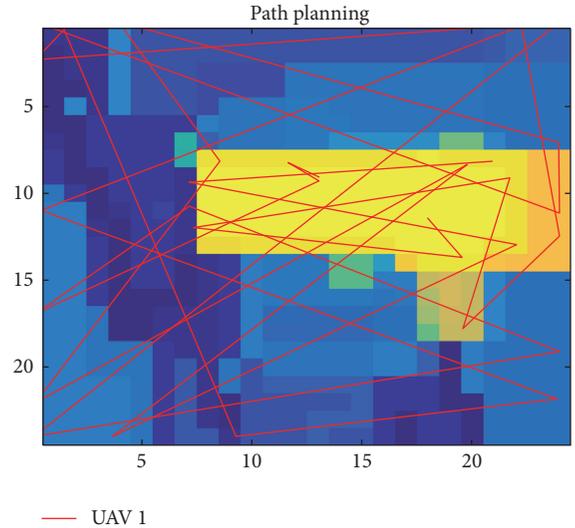


FIGURE 11: La Palma PSO path planning.

The relation between these factors is shown in the following:

$$f = \frac{1}{\left(\sum_{i=1}^{\max(i)} \sum_{j=1}^{\max(j)} \text{covered}(i, j)\right)^{0.1}} + \frac{1}{w^{0.1}} + \left[1 - \frac{1}{d}\right]. \quad (7)$$

The rest of the configuration of the PSO algorithm is shown in Table 6.

Once the setup is finished, the PSO algorithm is applied. The trajectory is shown in Figure 11.

The values of d and w are not now calculated because they are not relevant since the nature of the approach is totally different from the previous solutions. But it is worth seeing (Figure 11) how the trajectory has many straight lines covering all the possible space, especially between high P areas. Many more iterations could have been performed in order to cover the 100% of the map. However, we can also notice that the UAV goes forward and backwards to the same points, which means that this is not such an optimal solution.

5. Multiple UAVs Approach

In SAR operations, more than one UAV is usually used. Swarm systems take advantage of simple local behaviours to collectively solve complex problems, where the capability of the group is greater than the sum of its parts [24, 25]. In this

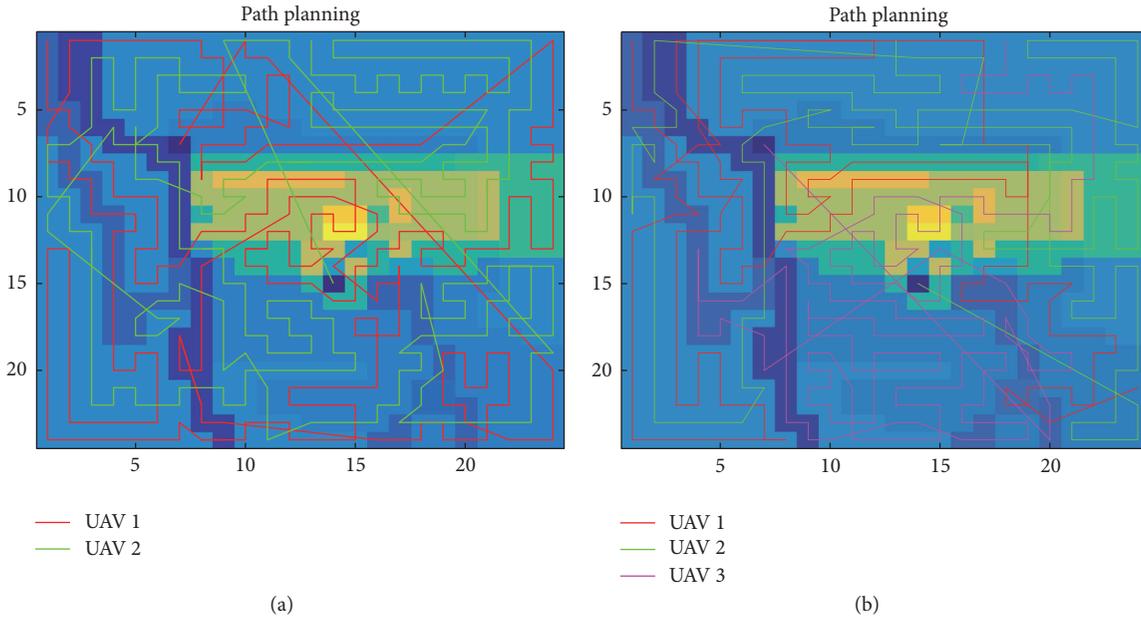


FIGURE 12: La Palma attraction path planning with 2 UAVs (a) and 3 UAVs (b), free formation.

work, two different swarm formations of UAVs for discrete path planning are studied: free and distributed. The UAVs are identical but start from different positions at the map and among them they track all the desired area.

The discrete path planning for several UAVs has been obtained by the attraction, fuzzy logic, and ANFIS approaches. The strategy is the same as the ones presented before for one UAV but now, for every iteration, the number of waypoints calculated is the same as the number of available vehicles. For simplicity, only 1, 2, or 3 UAVs are considered, but the solution could be extended to as many as needed.

In the gridding map of La Palma, the starting point for a 2-UAV configuration is (13, 1) and (18, 1) for 3 UAVs.

5.1. Free Formation. Free swarm formation consists of running the path planning algorithm for two or more UAVs at the same time in parallel; that is, each UAV follows an independent path along the coverage area. When calculating new waypoints, the information about the cells tracked by other UAVs is supposedly available, so the rest of the UAVs do not have to visit any already tracked cell.

As an example, Figure 12 shows the trajectories for 2 and 3 UAVs (a and b, resp.), when the attraction approach is used to obtain the discrete path planning.

The results obtained with the attraction path planning approach are $d = 680$, $w = 37393$ for 1 UAV; $d = 713$, $w = 19638$ for 2 UAVs; and $d = 720$, $w = 12812$ for 3 UAVs in free formation. The distance is quite similar in all the cases, but the weight is much lower with more than one UAV.

This effect on the weight is due to the parallel searching. As explained for a single UAV, the weight is calculated multiplying the order each cell is tracked by its P value, and getting the sum for all cells. In this case, there are two cells

with order 1, two with order 2 and so on, until the UAVs reach half of the number of cells.

The searching time is supposed to be reduced because although the distance is similar, it is shared by several UAVs, so the maximum distance travelled is now about a half or even a third of the distance travelled by only one UAV.

Regarding the path, it is interesting to see how the two UAVs (Figure 12(a)) converge to the green area in the middle of the map at the beginning. Then they separate each other and search in different high P value areas. At the end, they converge again in low P value areas. With 3 UAVs, the convergence is not as evident as in this case, meaning a more distributed tracking of every zone.

In all the cases, the trajectories of the different UAVs cross each other. This does not mean collision because the UAVs are not at the same point at the same time. However, the path planning must be completed with a collision avoidance algorithm, which is out of the scope of this work.

The free formation cannot be implemented with the PSO approach due to the nature of the solution, as explained before. For the ANFIS approach, the results are similar, the worst case being the fuzzy one.

5.2. Distributed Swarm Formation. A distributed swarm formation of UAVs means that the map is divided into several parts, as many as vehicles, and each UAV is assigned an area which it must track.

The path planning for 2 and 3 UAVs in a distributed swarm formation with the attraction path planning is shown in Figure 13. It is possible to see how each UAV tracks an independent zone and their paths do not cross. The distances are now $d = 673$ for 2 UAVs and $d = 678$ for 3 UAVs, and the weights $w = 20411$ and $w = 13712$, respectively. The distance

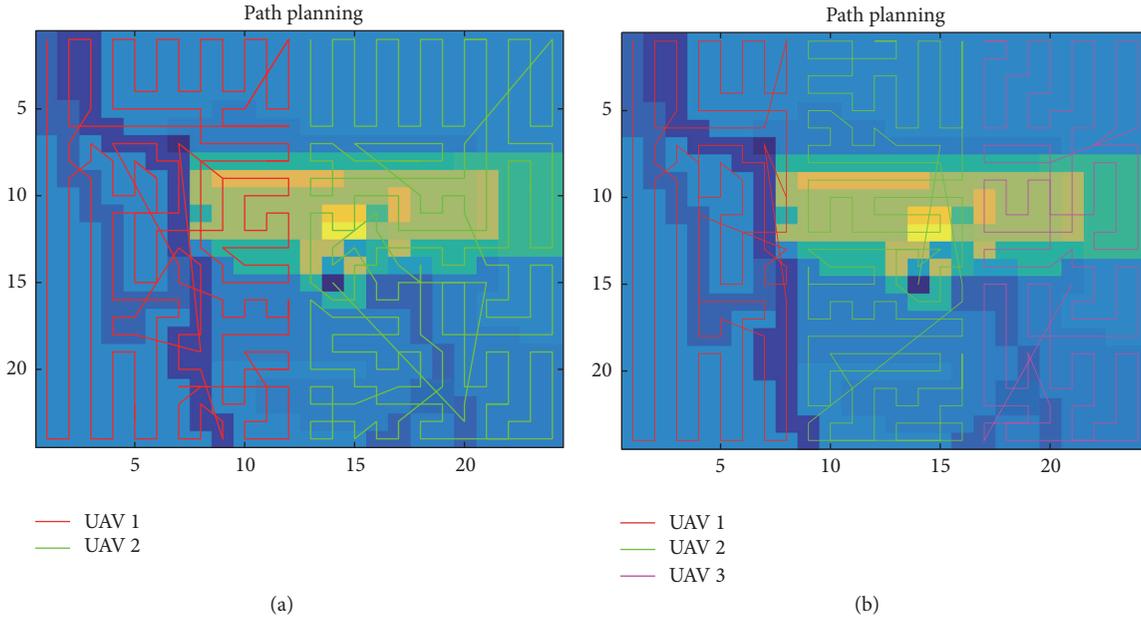


FIGURE 13: La Palma attraction path planning with 2 UAVs (a) and 3 UAVs (b), distributed formation.

TABLE 7: Comparison between the different approaches.

Formation	1 UAV		2 UAV				3 UAV			
			Free		Distributed		Free		Distributed	
	w	d	w	d	w	d	w	d	w	d
Attraction	37393	680	19638	713	20411	673	12812	720	13712	678
Fuzzy logic	35951	1497	17931	1550	19219	1212	12137	1596	13119	1155
ANFIS	36284	808	18579	827	19474	716	12520	876	13386	732

is similar but slightly longer than with only one UAV (680) and slightly better than the travelled distance obtained with UAVs in free formation (713 and 720). Regarding the weight, it is much smaller than the case of one UAV (37392) and much better for 3 UAVs than with 2 UAVs.

The same experiments have been carried out with different number of UAVs for the different approaches proposed (fuzzy logic and ANFIS).

5.3. Comparison of the Approaches with Different Number of UAVs. Finally, Table 7 summarizes the results of the different approaches for the two configurations. As expected, the results improve when the number of UAVs is increased. This is mainly due to the way the weight is calculated and proves the validity of the proposed strategies.

In general terms, it can be seen how the different methods have strong and weak points. The attraction method is the one with shortest paths, followed by the ANFIS one. However its weights are slightly higher than the other methods. Indeed, the fuzzy logic one has the lowest weight values although the difference is not big, but with much longer distances than the other two methods. In that sense, the ANFIS method has the most balanced results, with medium weights and low distance values.

To summarize, it is difficult to say which is better. It depends on the priority of the goals. The quickest is the attraction strategy, but the fuzzy approach is the best one finding people, at least visiting the areas where it is most probable to find someone. We will check these results on a real case in the next section.

6. A Case Studied

To analyze and prove the validity and utility of the discrete path planning proposals presented in this paper, they are going to be compared on a SAR real scenario.

Thirty missing people have been randomly located on the potential risk/occupancy map of La Palma. A uniformly distributed random generator gives a value that is associated with each cell of the map. Then, this value is multiplied by half of the potential risk/occupancy value of that cell. The resulting 25 positions with the highest value are selected. In this way we obtain a random distribution but weighted towards areas with high P values. The remaining 5 cases are manually distributed on areas with low P values (Figure 14). These last ones are set as the first five cases to be found so that they can be easily located in the graphs showing the time results of the searching.

TABLE 8: Time results with attraction strategy.

	1 UAV	2 UAVs		3 UAVs	
		Free	Distributed	Free	Distributed
Mean	217.62	113.88	115.60	88.84	87.60
Wins	1.00	2.00	2.00	3.00	1.00
Losses	6.00	0.00	5.00	0.00	0.00
Total wins			9.00		
Total losses			11.00		

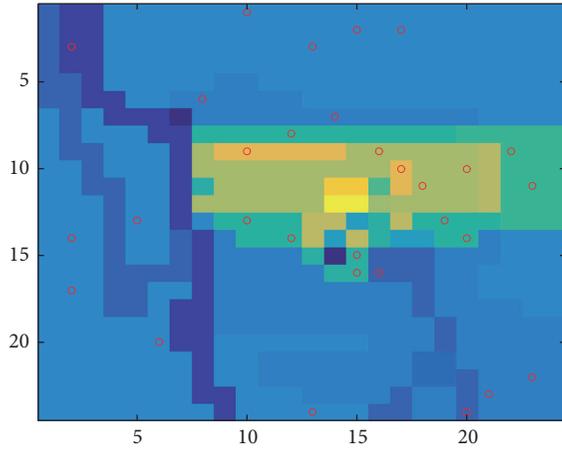


FIGURE 14: La Palma missing people.

The main criterion used to test the strategies is the time the UAV needs to find each of the 30 people individually. It is supposed to be proportional to the distance. All the UAVs are considered to have the same technical characteristics and to fly at the same speed (set to 3). Another assumption has been considered to make the simulation more realistic. When the UAV is tracking a cell, its speed is lower than when it is moving between different areas (tracking speed = 1).

Each case (finding someone) is evaluated estimating the time it takes the UAV to get there from the initial position with a particular configuration (strategy and formation).

When the cases are evaluated, the time that each method and its training require is calculated. For each case there is a method with a particular formation that has been the fastest, it is a winner; and there has been another one that has been the slowest, that is, a loose one. There should be a total of 30 wins and 30 losses adding all the possible configurations and strategies results, but there are some more because since the space and the speed are discretized, in some cases, there are ties.

In this way, we can compare the different methods, to see which are the best one exploring low P zones or high P cells and so on.

6.1. Finding People with the Attraction Strategy. The time results for attraction path planning are shown in Table 8. The number of wins represents the number of times that a specific configuration was the best one among the others. The number of losses represents when it was the worst one.

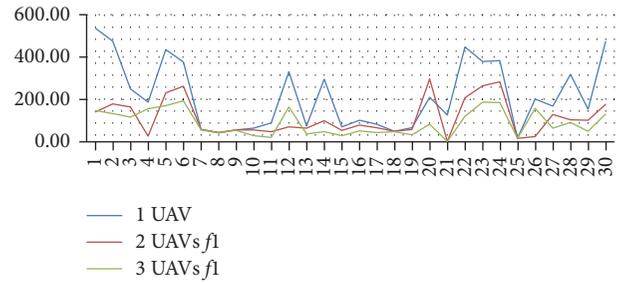


FIGURE 15: La Palma attraction path planning results for free swarm.

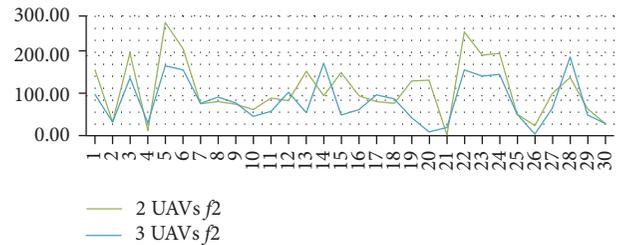


FIGURE 16: La Palma attraction path planning results for distributed swarm.

Figures 15 and 16 show the comparison between different number of UAV for this strategy. Time unit is represented in the vertical axis and the case in the horizontal one.

As expected, the time is reduced when the number of UAVs increases. Besides, the first five points need more time because they have low P values, so the UAV goes there after going to the higher P ones.

A comparison between the two swarm configurations with the same number of UAVs is shown in Figure 17 (free, $f1$; distributed, $f2$). We can observe that with two UAVs the difference is not significant. Nevertheless, with three UAVs, even having a similar tendency, there are big differences in some of the cases, which indicates that the searching is done with a very different philosophy.

6.2. Fuzzy Logic Path Planning. The results obtained with the fuzzy logic path planning are shown in Table 9.

In this case, there are more winning configurations. The worst ones are almost exclusively associated with the use of only one UAV. That means that this method is inefficient for one UAV, but its efficiency is highly improved when

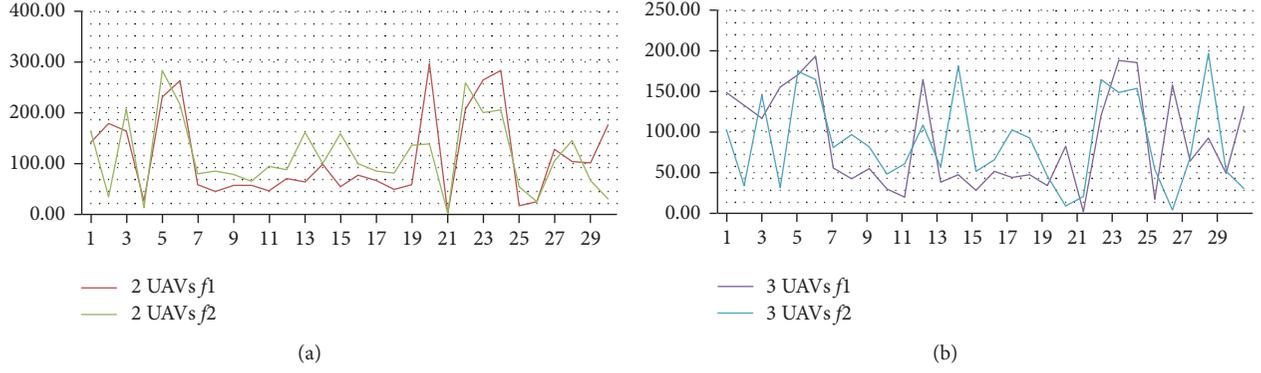


FIGURE 17: La Palma attraction path planning results for 2 UAVs (a) and 3 UAVs (b).

TABLE 9: Time results with fuzzy logic path planning.

	1 UAV	2 UAVs		3 UAVs	
		Free	Distributed	Free	Distributed
Mean	257.90	120.02	109.71	84.95	59.05
Wins	0.00	0.00	2.00	3.00	9.00
Losses	11.00	0.00	1.00	0.00	0.00
Total wins			14.00		
Total losses			12.00		

TABLE 10: Time results with ANFIS path planning.

	1 UAV	2 UAVs		3 UAVs	
		Free	Distributed	Free	Distributed
Mean	212.70	117.33	107.94	74.99	86.23
Wins	0.00	3.00	2.00	8.00	3.00
Losses	8.00	0.00	0.00	0.00	0.00
Total wins			16.00		
Total losses			8.00		

increasing the number of UAVs, especially in distributed formation.

6.3. ANFIS Path Planning Results. The results obtained for the ANFIS path planning are shown in Table 10.

Although this method seems to be the best one in terms of finding people, it has the same problem as the fuzzy logic approach. That is, better results are obtained with free swarm formation than with the distributed one; the worst case is again with an only UAV. Anyway, this strategy gives the best average values for this scenario.

6.4. PSO Path Planning Results in a SAR Mission. PSO path planning obtains the worst results of all. This method does not work well because the UAV travels long distances between waypoints in order to cover the surface with a distance/speed relation of one. Due to the continuous tracking, the UAV is very slow in comparison to the other methods. All results obtained are worse than with the other solutions, with many losses, meaning that the UAV did not find that people (Figure 18).

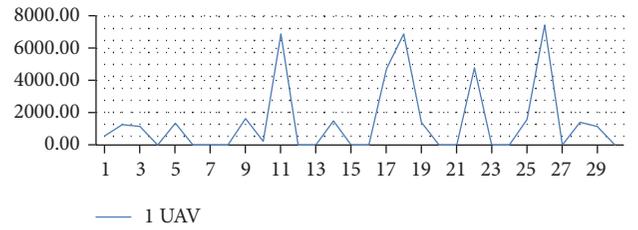


FIGURE 18: La Palma PSO path planning results.

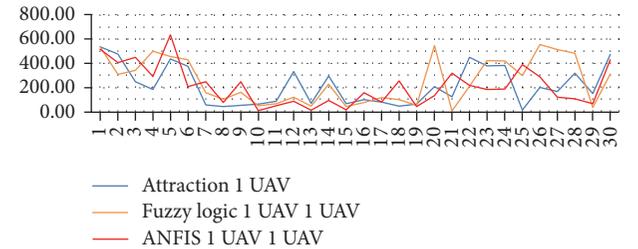


FIGURE 19: La Palma path planning results comparison for 1 UAV.

In view of these results, PSO is not taken into account for the global comparison between the different methods.

7. Discussion of the Global Results

A comparison between the different methods and configurations is carried out and the global results are discussed. To have a good overview, every strategy has been applied to the same 30 cases, with the two possible formations when more than one UAV is used in the searching.

First, Figure 19 shows the three different searching strategies with one UAV. The time results are very similar but, in general, ANFIS path planning is slightly better. The three methods work well with low P cells, but attraction gets shorter time than the others finding those worse cases.

In Figures 20 and 21, the path planning results for two and three UAVs, respectively, in free swarm formation are shown. Results are again quite similar. Fuzzy logic path planning seems to be somewhat worse than the others.

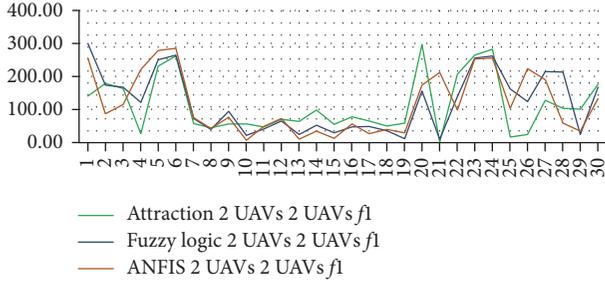


FIGURE 20: La Palma results comparison for 2 UAVs in free formation.

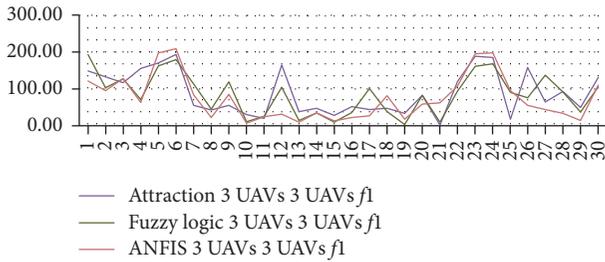


FIGURE 21: La Palma results comparison for 3 UAVs in free formation.

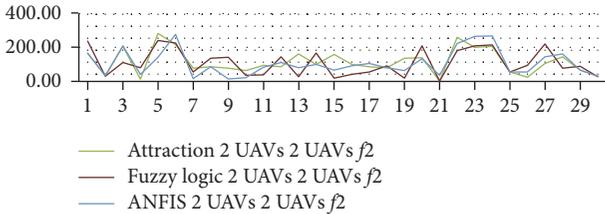


FIGURE 22: La Palma results comparison for 2 UAVs in distributed formation.

For distributed swarm configuration, Figures 22 and 23 show the results for 2 and 3 UAVs, respectively. The results for 2 UAVs are once again similar for all methods, having the ANFIS approach a slightly better performance.

However, in Figure 23, it is possible to see that there is a big difference between the fuzzy logic approach and the others. This method works better in distributed swarm maybe because it was designed for smaller maps, so when the searching area is smaller, the performance of this approach improves.

Finally, Figure 24 shows the average time of each configuration for every approach. According to these results, the fastest configuration for searching is to have 3 UAVs in distributed swarm formation ($f2$), as expected, and fuzzy logic approach gives the best results for this particular configuration, having ANFIS and attraction similar searching time.

Nevertheless, regarding the number of losses and wins (Figure 25), which is a critical factor, the ANFIS method is in general the best one. The fuzzy logic has also a very good

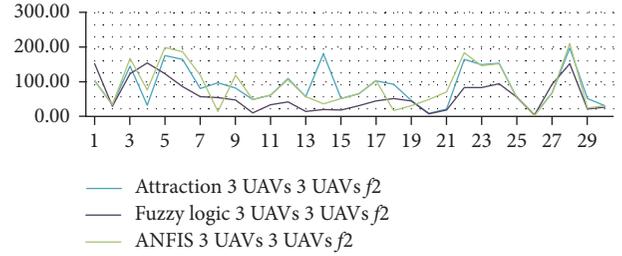


FIGURE 23: La Palma results comparison for 3 UAVs in distributed formation.

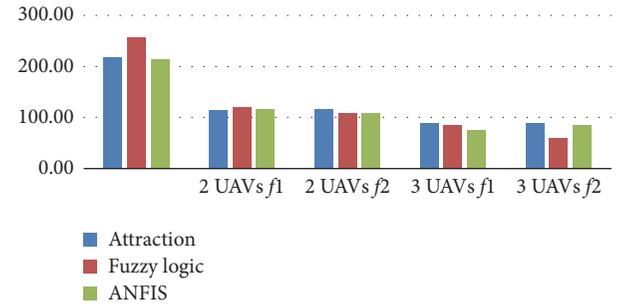


FIGURE 24: La Palma path planning mean time for all the configurations and every approach.

performance, especially for distributed configurations, as it has been said. However, even having less number of wins, attraction method is the most consistent due to its small number of losses, which indicates that its results are quite balanced.

As a summary, we can say that the attraction method gets medium positions regarding efficiency. It performs a regular and extensive search, which implies that it does not take it a lot of time finding people in low P areas, but it does not go as fast as other methods to high P values zones. This method works very well for homogenous P maps.

Fuzzy logic presents a quite fast convergence but it is very irregular. However, it works well in maps with large high P areas due to its type of searching, which performs like a random search in all dangerous areas.

The ANFIS approach has been proven to be the best one in general terms. It has a fast convergence to high P values and also performs a quite regular searching on the rest of the cells. It works worse on low P areas that are far from high P ones, where other methods have obtained better results.

8. Conclusions and Future Work

In this work, two main contributions are presented. The first one is a fuzzy map generation of a searching area according to a risk/occupancy factor that aims at search and rescue (SAR) missions. The second one is the application of different intelligent strategies for discrete path planning to cover a gridding searching area.

Regarding the map characterization, the conclusion is that it helps to focus the path planning on promising zones.

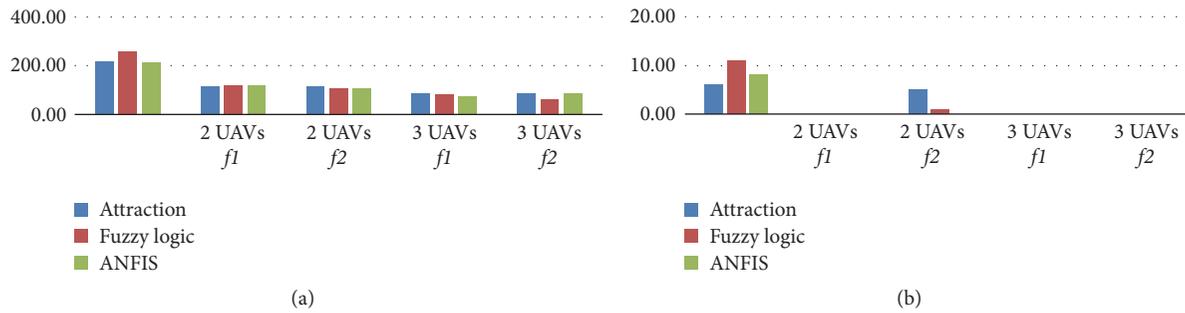


FIGURE 25: La Palma wins (a) and losses (b) cases.

The path planner finds trajectories that minimize the risk the people are exposed to. Indeed, in the simulations results, a fast tracking of the higher probability risk/occupied areas can be observed that results in a quicker finding of the people involved in a dangerous situation.

Four intelligent strategies have been applied to the discrete path planning, namely, attraction, an original one, and fuzzy, ANFIS, and PSO. Except this last one, the rest have obtained good results in terms of searching time and distance. Another conclusion is that the performance of the method strongly depends on the associated probability map. This remarks the importance of the generation of this map, taking into account all the possible variables and factors that can influence the searching.

Using multiple UAVs in swarm formation for the searching, a dependence on the type of path planning method has been observed. In general, distributed formation works better, but it should be chosen in association with the P map and the path planning method.

Future works related to several aspects of this work could be the following:

- (i) Studying other factors for the map characterization and their influence on the potential risk/occupancy value, which has been proved important for the SAR mission
- (ii) Working with a more realistic model of the UAVs, including the dynamics [26], and also avoidance and visual tracking algorithms [27]
- (iii) Testing the most successful proposals on real systems.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work is supported by the European Union EP-INTERREG V-A Spain-Portugal (POCTEP), Project 0115_TECNOLIVO_6_E, which provided funds for covering the costs to publish the paper in open access.

References

- [1] L. Novaro Mascarello and F. Quagliotti, "The civil use of small unmanned aerial systems (sUASs): operational and safety challenges," *Aircraft Engineering and Aerospace Technology*, vol. 89, no. 5, pp. 703–708, 2017.
- [2] G. Pajares, J. J. Ruz, P. Lanillos, M. Guijarro, J. M. De La Cruz, and M. Santos, "Trajectory generation and decision making for UAVs," *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 5, no. 1, pp. 83–92, 2008.
- [3] P. García-Auñón, M. Santos-Peñas, and J. M. de la Cruz García, "Parameter selection based on fuzzy logic to improve UAV path-following algorithms," *Journal of Applied Logic*, vol. 24, no. part B, pp. 62–75, 2017.
- [4] M. Silvagni, A. Tonoli, E. Zenerino, and M. Chiaberge, "Multipurpose UAV for search and rescue operations in mountain avalanche events," *Geomatics, Natural Hazards and Risk*, vol. 8, no. 1, pp. 18–33, 2017.
- [5] R. Dubé, A. Gawel, C. Cadena, R. Siegwart, L. Freda, and M. Gianni, "3D localization, mapping and path planning for search and rescue operations," in *Proceedings of the 14th International Symposium on Safety, Security and Rescue Robotics, SSRR 2016*, pp. 272–273, Switzerland, October 2016.
- [6] M. Jun and R. D'Andrea, "Path Planning for Unmanned Aerial Vehicles in Uncertain and Adversarial Environments," in *Cooperative Control: Models, Applications and Algorithms*, vol. 1 of *Cooperative Systems*, pp. 95–110, Springer US, Boston, MA, 2003.
- [7] F. Kamrani and R. Ayani, *UAV path planning in search operations*, Aerial Vehicles, InTech, 2009.
- [8] C. Baker, G. Ramchurn, L. Teacy, and N. Jennings, "Planning search and rescue missions for UAV teams," in *Proceedings of the At 2016: Conference on Prestigious Applications of Intelligent Systems at ECAI, 2016*, p. 6, 2016.
- [9] K. P. Valavanis and G. J. Vachtsevanos, "Uav mission and path planning: Introduction," *Handbook of Unmanned Aerial Vehicles*, pp. 1443–1446, 2015.
- [10] E. Besada-Portas, L. de La Torre, J. M. de La Cruz, and B. de Andrés-Toro, "Evolutionary trajectory planner for multiple UAVs in realistic scenarios," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 619–634, 2010.
- [11] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25–33, 2017.

- [12] G. E. Anaya Fuentes, E. S. Hernández Gress, J. C. Seck Tuoh Mora, and J. Medina Marín, "Solution of the job-shop scheduling problem through the traveling salesman problem," *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 13, no. 4, pp. 430–437, 2016.
- [13] M. Kallmann and M. Kapadia, "Geometric and Discrete Path Planning for Interactive Virtual Worlds," *Synthesis Lectures on Visual Computing*, vol. 8, no. 1, pp. 1–201, 2016.
- [14] P. B. Sujit, S. Saripalli, and J. B. Sousa, "An evaluation of UAV path following algorithms," in *Proceedings of the 2013 12th European Control Conference, ECC 2013*, pp. 3332–3337, che, July 2013.
- [15] A. Barrientos, P. Gutiérrez, and J. Colorado, *Advanced UAV trajectory generation: Planning and guidance*, Aerial Vehicles, InTech, 2009.
- [16] E. G. Hernandez-Martinez, E. D. Ferreira-Vazquez, G. Fernandez-Anaya, and J. J. Flores-Godoy, "Formation tracking of heterogeneous mobile agents using distance and area constraints," *Complexity*, vol. 2017, Article ID 9404193, 2017.
- [17] E. Besada-Portas, L. De La Torre, A. Moreno, and J. L. Risco-Martín, "On the performance comparison of multi-objective evolutionary UAV path planners," *Information Sciences*, vol. 238, pp. 111–125, 2013.
- [18] P. García Auñón and M. Santos, "Use of genetic algorithms for unmanned aerial systems path planning in. Decision Making and Soft Computing," *World Scientific*, vol. 9, pp. 430–435, 2014.
- [19] M. Davoodi, F. Panahi, A. Mohades, and S. N. Hashemi, "Multi-objective path planning in discrete space," *Applied Soft Computing*, vol. 13, no. 1, pp. 709–720, 2013.
- [20] F. Alonso Zotes and M. Santos Peñas, "Heuristic Optimization of Interplanetary Trajectories in Aerospace Missions," *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 14, no. 1, pp. 1–15, 2017.
- [21] P. Yang, K. Tang, and J. A. Lozano, "Estimation of Distribution Algorithms based Unmanned Aerial Vehicle path planner using a new coordinate system," in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, pp. 1469–1476, China, July 2014.
- [22] -J. Ruz, O. Arévalo, G. Pajares, and M. J. De Cruz, *UAV trajectory planning for static and dynamic environments*, Aerial Vehicles, InTech, 2009.
- [23] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [24] T. Tomic, K. Schmid, P. Lutz et al., "Toward a fully autonomous UAV: research platform for indoor and outdoor urban search and rescue," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 46–56, 2012.
- [25] J. B. Seco, J. Dormido Canto, M. Montalvo Martinez, J. López-Orozco, E. Besada Portas, and J. De la Cruz Garcia, "Centro de Control de Tierra para Colaboración de Vehículos Autónomos Marinos," *Revista Iberoamericana de Automática e Informática industrial*, vol. 15, no. 1, p. 1, 2017.
- [26] J. E. Sierra and M. Santos, "Modelling engineering systems using analytical and neural techniques: Hybridization," *Neurocomputing*, 2017.
- [27] Z. Wu, Z. Guan, C. Yang, and J. Li, "Terminal guidance law for UAV based on receding horizon control strategy," *Complexity*, vol. 2017, pp. 1–19, 2017.

Research Article

Using Deep Learning to Predict Complex Systems: A Case Study in Wind Farm Generation

J. M. Torres  and R. M. Aguilar 

Department of Computer and Systems Engineering, Universidad de La Laguna, La Laguna 38200 Tenerife, Spain

Correspondence should be addressed to R. M. Aguilar; raguilar@ull.edu.es

Received 30 November 2017; Accepted 25 February 2018; Published 3 April 2018

Academic Editor: José Manuel Andújar

Copyright © 2018 J. M. Torres and R. M. Aguilar. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Making every component of an electrical system work in unison is being made more challenging by the increasing number of renewable energies used, the electrical output of which is difficult to determine beforehand. In Spain, the daily electricity market opens with a 12-hour lead time, where the supply and demand expected for the following 24 hours are presented. When estimating the generation, energy sources like nuclear are highly stable, while peaking power plants can be run as necessary. Renewable energies, however, which should eventually replace peakers insofar as possible, are reliant on meteorological conditions. In this paper we propose using different deep-learning techniques and architectures to solve the problem of predicting wind generation in order to participate in the daily market, by making predictions 12 and 36 hours in advance. We develop and compare various estimators based on feedforward, convolutional, and recurrent neural networks. These estimators were trained and validated with data from a wind farm located on the island of Tenerife. We show that the best candidates for each type are more precise than the reference estimator and the polynomial regression currently used at the wind farm. We also conduct a sensitivity analysis to determine which estimator type is most robust to perturbations. An analysis of our findings shows that the most accurate and robust estimators are those based on feedforward neural networks with a SELU activation function and convolutional neural networks.

1. Introduction

A region's electricity grid consists of a series of components that have to work together to achieve a balance between generation and demand, while at the same time ensuring the security of the electricity supply and providing a certain level of quality and service. The structure of the power system may be divided into four key activities: generation, transport, distribution, and marketing. The start of the electricity supply process takes place at power plants, where the electricity is generated. Depending on the type of facility, different types of primary energy sources are used to drive a turbine or motor, thus converting the primary energy into mechanical energy. The turbine is connected to a generator, which turns the mechanical energy into electrical energy. The process of supplying electricity continues via the transport network, which links the various production plants to consumption centres. This process takes place at high voltages to lower the currents and thus the losses. The distribution process comes next, in which the electricity is relayed from substations to the

transport network for the various consumption points. These substations reduce the voltage from that of the transport network to values that are suitable for use by consumers. The electricity supply process concludes with the marketing activity, in which the electricity is sold to consumers based on their contracted power.

In Spain, Law 54/1997 went into effect in 1998. This law is notable because, as in the rest of Europe, it deregulated the generation and marketing activities, while continuing to regulate the transport and distribution aspects. Ever since, two primary operators have been charged with managing the technical and economic aspects of Spain's electricity market. One is Red Eléctrica de España (REE) and the other is Operador del Mercado Ibérico de Energía (OMIE), which are required to coordinate their efforts. The latter is charged with handling the bids for selling and buying energy. With this power system model, the price of energy became defined by the matching processes that started to be used in the various market sessions: daily, intraday, and ancillary services.

TABLE I: Hours of operation for intraday market sessions.

	1st session	2nd session	3rd session	4th session	5th session	6th session
Session opens	17:00	21:00	01:00	04:00	08:00	12:00
Session closes	18:45	21:45	01:45	04:45	08:45	12:45
Matching	19:30	22:30	02:30	05:30	09:30	13:30
Reception of scheduling disaggregations	19:50	22:50	02:50	05:50	09:50	13:50
Release PHF	20:45	23:45	03:45	06:45	10:45	14:45
Scheduling horizon	27 h	24 h	20 h	17 h	13 h	9 h
Time periods	22–24	1–24	5–24	8–24	12–24	16–24

The daily session, or daily market, takes place at 12:00, during which the bids for the 24 hours following the session close are placed. This is the main market and therefore the session in which much of the energy is negotiated.

The intraday markets are convened over the course of the previous day and the delivery day. Once the new offers are matched, they are added to the daily schedule to yield what is known as the final hourly schedule (PHF in Spanish). Obviously, less energy is traded in these markets since their time horizons are gradually reduced. They are designed to accommodate potential changes to trading forecasts. Table 1 shows the different time spans for the six intraday sessions, which are only open to those buyers or sellers that have taken part in the daily sessions.

Finally, the ancillary services are used when needed to resolve the imbalance between demand and generation, regulate the frequency/power, and control the voltage in the transport network. Their purpose, then, is to guarantee the balance, security, quality, and reliability of the electrical system.

For each hour, the producers and consumers that want to produce or consume electricity must place a bid in the various markets depending on their needs. Hourly in each session, the bids received are arranged from the highest to the lowest sale price and the highest to the lowest purchase price, with the lowest price being 0 and the highest being 180.3 €/MWh. Graphically, the result would be two aggregate curves, where the x -axis is the energy and the y -axis is the price. The matching method is “marginalist,” meaning that the matching price for that hour and session is set at the intersection of the two aggregate curves. Any units remaining below and above that value will be sold and bought, respectively, at that price. In other words, all of the power contracted will be sold at that price.

To illustrate the matching process, Figure 1 shows an example for a case with five power plants and six large consumers placing a bid in the market for hour H.

Since most of the demand is not manageable, it offers to buy at the maximum of 180.3 €/MWh. But it is worth asking what criteria producers use to craft their sale offers to cover this demand. Nuclear and renewable plants tend to sell at 0 €/MWh to ensure that all of the energy they produce is consumed. This is due to their technical limitations, such as the inability to halt production in the case of nuclear and the inability to store primary energy in the case of renewables. The difference between the total system demand and the

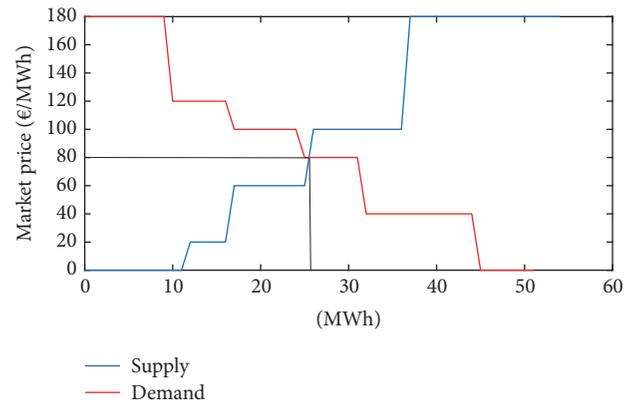


FIGURE 1: Aggregated bid-offer curves and market splitting.

energy produced by the above technologies and hydropower (in which the water flow can be regulated by using reservoirs) is known as the thermal gap. This difference is the energy demand that must be met using thermal technologies (such as gas and coal), the variable costs of which are higher than for renewables and nuclear. Therefore, the competition in the electricity market lies between the thermal generating plants, as it is on these plants that the intersection between the demand and supply curves during the market matching process, and therefore the final sale price, depends.

As Figure 1 shows, a lower demand entails lower prices by requiring fewer plants to be in operation and excluding the most expensive plants from the matching process. An increase in renewable production can result in a sharp drop in the matching price, leading to cases in which all of the demand is covered by the production priced at 0 €/MWh, as has already happened on numerous occasions. As the proportion of renewables in the energy mix grows, the average price in the electricity market drops. Specifically, the average energy price in Spain’s electricity market in 2016 (48.4 €/MWh) fell by 23% with respect to 2015 to the lowest price since 2010. This was due primarily to the large response by hydro and wind power to cover the demand in the first few months of the year.

Reducing the use of thermal technologies requires incorporating more wind power into the energy mix. To do so, producers have to know how many units to supply to the daily market. A wind farm’s generation must be forecast one day in

TABLE 2: Features in data examples.

Description	Units	Used as
Timestamp	ISO 8601	Input
Wind speed	m/s	Input
Wind direction	(°)	Input
Relative humidity	(1)	Input
Temperature	°C	Input
Atmospheric pressure	Pa.	Input
Issues	-	Input
Power generated	KWh.	Output
Power forecast	KWh.	-

advance, and, if necessary, these supply units can be corrected in the six intraday market sessions with time horizons of 27, 24, 20, 17, 13, and 9 hours. Unfortunately, the main drawback of wind energy is how difficult it is to make accurate forecasts due to this energy source's heavy reliance on atmospheric conditions, and how difficult, in turn, these are to predict.

In this paper we present a series of regression estimators that rely on deep-learning techniques to predict the generation of a wind farm based on an estimate of atmospheric conditions. These predictions are intended for the daily market, meaning they must offer sufficiently accurate results 12 to 36 hours in advance.

2. Materials and Methods

Below we describe the estimators implemented, the data used, and the procedures employed to train, evaluate, and compare the estimators.

2.1. Data Sets. The Institute of Technology and Renewable Energies (ITER) is the agency that runs the largest wind farm on the island of Tenerife. It also provided us with the data for this study. The ITER runs the MADE farm, which has a rated power of 4800 kW, supplied by eight MADE AE-46 generators. A weather forecast for the following 48 hours in one-hour periods is generated twice a day. Once a day the wind speed for the following 12 to 36 hours is forecast and a polynomial regression is carried out that is used to estimate the generation for each hour of said interval. This estimate is sent to the OMIE to be used in the daily market.

The ITER gave us a data set with an hourly sample with the results of the Numerical Weather Prediction (NWP) for different meteorological variables, the generation forecast made by the ITER using a polynomial regression, and the actual wind generation measured and a free-text field containing problems involving the operation of the generators. Table 2 summarises the features in each sample of the data set provided by the ITER and shows which were used as inputs and outputs for the estimators.

To prepare the data, we used certain feature engineering techniques. The timestamp was broken down into day of year and time of day, each of which was represented using the pair of values for their sine and cosine in an effort to capture their periodic nature and their effects on the local daily wind cycles.

The wind direction was also encoded using this method for the same reason. The following shows an example of this using the wind direction:

$$h \rightarrow \left(\sin \frac{h}{360}, \cos \frac{h}{360} \right). \quad (1)$$

The text for the issues was manually converted into the fraction of generators not in service at a given time, since some of the samples for the total farm output, which were used to adjust the predictor's output, were obtained when some of the generators were out of service. When the validation and test sets were configured, however, only those samples taken when all of the generators were in service during the period measured were used.

All of the inputs were normalised by min-max scaling between 0 and 1, the goal being to achieve maximum efficiency during training. For the training, we had data sampled each hour from January 2014 to April 2016, which were randomly divided into three sets: 60% comprised the data training set, 20% the validation set, and the remaining 20% the test data. The data were stored in TensorFlow TFRecords files for efficiency purposes for use with the TensorFlow framework [1], which was the tool used to develop, train, and evaluate the various predictors.

2.2. Feedforward Neural Networks. The first architecture evaluated was the Feedforward Neural Network (FNN). In a FNN [2] every neuron in one layer receives as its input all of the outputs from the neurons in the previous layer. The output a_j^l of the j th neuron in the l th layer can be expressed as indicated in the following:

$$a_j^l = \sigma^l \left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l \right), \quad (2)$$

where w_{jk}^l is used to denote the weight for the output of the k th neuron in the $(l-1)$ th layer as the input to the j th neuron in the l th layer, b_j^l is the bias of the j th neuron in the l th layer, and σ^l is the activation function of the neurons in the l th layer. The most common activation function is the sigmoid function, which is expressed as shown in the following:

$$\sigma^l(x) = \frac{e^x}{e^x + 1}. \quad (3)$$

In a regression problem like the one at hand, the sigmoid activation function is used in every layer except the last one, the output layer, which uses the identity function $\sigma^l(x) = x$ as the activation function.

The learning was done using standard deep learning techniques, like minibatch gradient descent [3] and Adagrad (for adaptive gradient algorithm) optimiser [4]. The latter allows different step sizes for different features, so it does not require a learning rate to be specified for it. The adjusted model was validated every 5000 trained batches. The stopping criterion employed was for the evaluation's Mean Square Error (MSE) not to decrease during three consecutive iterations. But in order to compare the accuracy of different predictors, we use the Mean Absolute Error (MAE) [5] and the Mean Absolute Scaled Error (MASE) [6]. The MAE and MASE are common measures of forecast error in time series analysis.

To avoid overadjusting the models when training them, the cost function includes a component to L_2 -norm to regularise all the weight w_{jk}^l and the bias b_j^l of the entire model. In some specific cases we used dropout [7] to check its effects when attempting to further generalise the trained models, but it did not help to improve the results. When dropout is used during training, only a selection of neurons chosen with probability P_{keep} can be activated. The following shows the generalisation of the output expression for neuron a_j^l when dropout is used:

$$a_j^l = \begin{cases} \frac{\sigma^l(\sum_k w_{jk}^l a_k^{l-1} + b_j^l)}{P_{\text{keep}}}, & \text{if } p \leq P_{\text{keep}} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $p \sim U(0, 1)$.

2.2.1. ReLU Activation Function. When training using minibatch gradient descent, the backward propagation undergoes a phenomenon called vanishing gradient [8], which considerably hampers the training of networks with a large number of layers. In these cases the ReLU activation function is very practical because it does not suffer from the vanishing gradient problem. Moreover, in regression problems it has the advantage of not being limited to outputs between 0 and 1, thus favouring the dispersity of the solution in the hidden neurons. The following shows the expression for the activation function for the l th layer of a neural network with N layers.

$$\sigma_{\text{relu}}^l(x) = \begin{cases} \max(0, x), & \text{if } l \neq N \\ x, & \text{if } l = N, \end{cases} \quad (5)$$

where x is used to indicate the input to the activation function, that is, the weighted sum of the inputs to the neuron, as shown in (2). The FNNs with the ReLU activation function that we trained use this function for the neurons in every layer except the output layer.

2.2.2. SELU Activation Function. Even using the ReLU activation functions, truly deep FNNs are difficult to train,

which hampers their ability to handle high-level abstract relationships in the input samples. The Scaled Exponential Linear Unit (SELU) activation function induces self-normalising properties that make the neuron activations automatically converge toward an average of 0 and a variance of 1 [9]. This property propagates throughout the network even in the presence of noise and perturbations. This allows training networks with more layers and the use of strong regularisation and it makes the training more robust.

The following shows the expression for this type of activation function:

$$\sigma_{\text{selu}}^l(x) = \lambda \begin{cases} x, & \text{if } x > 0 \\ \alpha e^x - \alpha, & \text{if } x \leq 0, \end{cases} \quad (6)$$

where x is used to indicate the input to the activation function. Klambauer et al. [9] justify why α and λ must have the values shown in

$$\begin{aligned} \alpha &= 1.6732632423543772848170429916717 \\ \lambda &= 1.0507009873554804934193349852946 \end{aligned} \quad (7)$$

in order to ensure that the neuron activations converge automatically toward an average of 0 and a variance of 1.

2.3. Convolutional Neural Networks. It is possible for the atmospheric conditions in previous hours to contain information that can be used to improve the forecast at any given time. This information was introduced into predictors based on FNNs to prepare samples that contained the weather forecast features for every hour t and for the $N - 1$ previous hours, with the input layer for said models being suitably expanded.

In order to check the results when the neural network is forced to exploit the time-local correlation between the features by forcing a connection pattern between adjacent neurons in each layer, we implemented some predictors using models based on Convolutional Neural Networks (CNNs). CNNs are biologically inspired variants of FNNs used primarily in computer vision problems [10], although their ability to exploit spatially local correlation in images can also be used in time-series forecasting.

In these models, the output of each neuron a_j^l is not generated based on the output of every neuron in the previous stage, as shown in (2); rather, it is generated from a subset of spatially adjacent neurons. So to improve the learning efficiency, every neuron in the same layer shares the same weight and bias, meaning the layer can be expressed in terms of a filter that is convoluted with the output of the previous layer. The following shows the output a_j^{lk} of the j th neuron of the l th convolutional layer:

$$a_j^{lk} = \sigma^{lk} \left((W^{lk} * a^{l-1})_j + b^{lk} \right), \quad (8)$$

where $(W^{lk} * a^{l-1})_j$ is the j th element resulting from the convolution of the filter defined by W^{lk} with the output of the previous layer a^{l-1} , σ^{lk} is the activation function for the

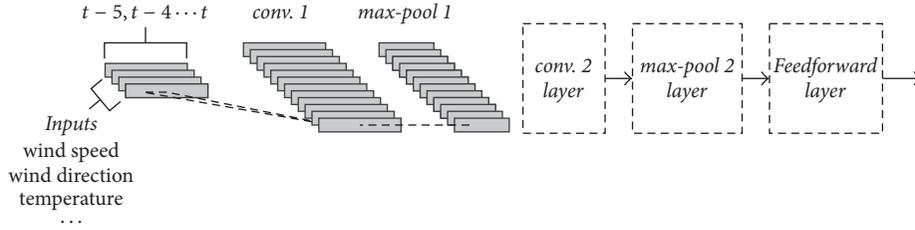


FIGURE 2: General diagram of the CNNs used.

convolutional layer, and $k \in [0 \dots K]$ indicates that it is the output of the k th channel of the layer. In each convolutional layer, different outputs can be applied to the output of the previous layer to generate different representations or channels, thus yielding a fuller representation of the data. In order to apply a CNN to the problem of time-series forecasting, we arranged the samples in such a way that the time series of each characteristic is an input channel to the network and thus to the first convolution later.

Figure 2 shows a general diagram of the CNN developed in this paper to forecast time series. Behind each convolutional layer with a ReLU activation function is a max-pooling layer, which partitions the input into a set of nonoverlapping ranges and, for each range, outputs the maximum value. Behind several convolutional exchange layers and max-pooling layers there is a feedforward layer (as described in (2)) to yield the output of the entire network.

2.4. Recurrent Neural Networks. As we have shown, FNNs and CNNs can use historical data series but they lack the memory to store information over the long term. They also cannot use the information contained in the output of the network at previous instants. Recurrent Neural Networks (RNNs) [11] solve this problem by making the output y_t at timestamp t depend on previous computations through a hidden state s_t that acts as a memory for the network, as shown in the following:

$$\begin{aligned} s_t &= \sigma(Wx_t + Us_{t-1}) \\ y_t &= f(s_t), \end{aligned} \quad (9)$$

where x_t is the input to the network, $\sigma(x)$ is the state activation function, W and U are the weights by which the inputs and the state for the previous instant are multiplied, respectively, to generate the new state s_t , and $f(x)$ is the function that generates the network's output based on the state. On occasion this function $f(x)$ will be the identity function $f(x) = x$, but it can also be a feedforward layer like the one described in (2).

Figure 3 shows the RNN we used, unfolded into a full network. By unfolded we simply mean that we write out the network for the complete sequence of inputs and take the output at t as the network's prediction for fitting. Instead of the basic RNN cell explained previously, we used two, more advanced cells: long short-term memory (LSTM) [11] and Gated Recurrent Unit (GRU) [12] cells.

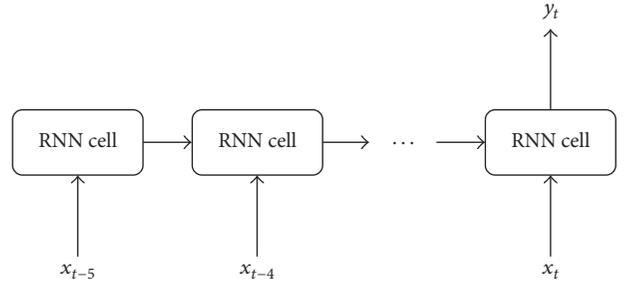


FIGURE 3: RNN unfolded into a full network.

LSTM recurrent neural networks are capable of learning and remembering over long input sequences and tend to work very well for time-series forecasting problems [13]. As (10) well shows, the output y_t depends on the state s_t of the LSTM cell through the activation function $\sigma_y(x)$ (which is generally $\tanh(x)$). The output gate o_t controls the extent to which the state s_t is used to compute the output y_t by means of the Hadamard product (\circ):

$$\begin{aligned} f_t &= \sigma_g(W_f x_t + U_f y_{t-1} + b_f), \\ i_t &= \sigma_g(W_i x_t + U_i y_{t-1} + b_i), \\ o_t &= \sigma_g(W_o x_t + U_o y_{t-1} + b_o), \\ \tilde{s}_t &= \sigma_s(W_s x_t + U_s y_{t-1} + b_s), \\ s_t &= f_t \circ s_{t-1} + i_t \circ \tilde{s}_t, \\ y_t &= o_t \circ \sigma_y(s_t). \end{aligned} \quad (10)$$

The state s_t depends on the state of the previous instant s_{t-1} and on the candidate for the new value of the state \tilde{s}_t . The input gate i_t controls the extent to which \tilde{s}_t flows into the memory and the forget gate f_t controls the extent to which s_{t-1} remains in memory. The o_t , f_t , and i_t gates and the candidate for the new value of the state of the cell \tilde{s}_t can be interpreted as the outputs of conventional artificial neurons whose inputs are the input to cell x_t at t and the output of cell y_{t-1} at $t-1$. The activation function for the gates σ_g is the sigmoid function, while for σ_y it is $\tanh(x)$.

GRU recurrent neural networks use a simpler cell without a forget gate and with fewer parameters, meaning they can generally be trained with fewer samples. Chung et al. [14] shows experimentally its superiority over LSTM for simple networks, but cannot conclude that GRU is better in different

cases. The following shows the equations that govern the behaviour of these cells:

$$\begin{aligned}
 z_t &= \sigma_g(W_z x_t + U_z y_{t-1} + b_z), \\
 r_t &= \sigma_g(W_r x_t + U_r y_{t-1} + b_r), \\
 \tilde{y}_t &= \sigma_y(W_y x_t + U_y (r_t \circ y_{t-1}) + b_y), \\
 y_t &= z_t \circ y_{t-1} + (1 - z_t) \circ \tilde{y}_t,
 \end{aligned} \tag{11}$$

where \tilde{y}_t is the cell's output candidate, z_t is the update gate, which controls the extent to which y_{t-1} or \tilde{y}_t is used to compute the output, and r_t is the reset gate, which controls the extent to which y_{t-1} flows into the cell's output candidate \tilde{y}_t . In GRU the activation functions $\sigma_g(x)$ and $\sigma_y(x)$ are the sigmoid function and $\tanh(x)$, respectively.

2.5. Sensitivity to Disturbances. An important aspect in this paper is to analyse the behaviour of our models in the presence of input disturbances. In the following equation we show the expression for the input x_j assuming that it undergoes a small incremental change Δx_j :

$$x_j = x_j^* + \Delta x_j, \tag{12}$$

where x_j^* is the j th value of the input sample without disturbance. It is important to note that the input x_j to the model is the j th input to the first layer. Similarly, y is the output of the last layer.

A perturbation Δx_j in the input x_j induces a disturbance in the output y of the neural network. When there is no perturbation in any of the model's inputs, the output of the neural network is y^* . In order to determine if the model is robust against perturbations in the j th input, the sensitivity s_j has to be calculated [15]. We show its expression in the following:

$$s_j = \frac{\Delta y / y^*}{\Delta x_j / x_j^*} = \frac{\Delta y / y^*}{\delta_j}, \tag{13}$$

where Δy is the corresponding change in the value of the output variable y and $\delta_j = \Delta x_j / x_j^*$ is the input perturbation ratio.

If the sensitivity s_j is lower than 1.0, it means that the network attenuates the input disturbances, whereas if it is equal to 1.0, it means that the network neither attenuates nor amplifies disturbances.

3. Results and Discussion

As noted earlier, the wind farm currently uses a polynomial regression to predict the farm's generation, as required to participate in the daily market. To give an idea of its accuracy, Table 3 shows the MAE and MASE for some estimators using the historical data available. The MASE indicates the absolute error relative to the error in the one-hour naive forecast reference estimator. Therefore, a MASE greater than 1.0 indicates the predictor works worse than the reference

TABLE 3: Accuracy of current estimators.

Estimator	MAE (kWh)	MASE
Naive 1 h	261	1.00
Naive 24 h–48 h	780	2.99
Polynomial	552	2.11

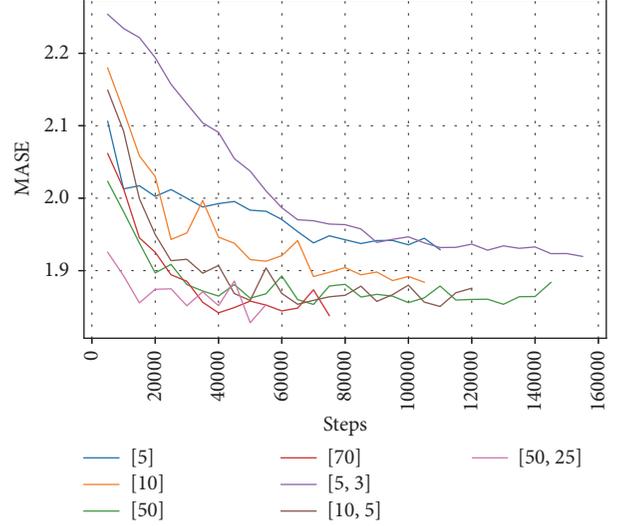


FIGURE 4: Trend in MASE for different FNNs with ReLU.

estimator, while a MASE lower than 1.0 indicates that it works better.

As Table 3 shows, the polynomial estimator is a little over two times worse than the one-hour naive forecast reference estimator; however, this naive estimator would never be able to be used because the prediction has to be made and sent to the grid operator 24 hours in advance.

To obtain a more realistic comparison, it was compared with another naive estimator that uses the actual generation measured 24 or 48 hours earlier for its prediction at a given time. This naive estimator could be used at the farm, though Table 3 shows that the polynomial regression is considerably better than this second naive estimator.

3.1. Feedforward Neural Networks. Figures 4, 5, and 6 show how the MASE evolved for the validation data set over the course of the iterations for networks with different numbers of hidden layers, neurons and ReLU, sigmoid, and SELU activation functions, respectively.

In every case the final result is similar and slightly better than for the polynomial regression, whose MASE is 2.11. Figure 4 shows that a ReLU activation function yields good results with around 20 neurons between all the hidden layers. If this size is increased, the number of overfitting cases rises gradually and the MASE is not reduced by either adding more layers or making the layers larger.

Figure 5 shows the results for FNNs with a sigmoid activation function. In this case we clearly see that two hidden layers yield better results than one, but after that no further improvements are obtained by expanding the network. The

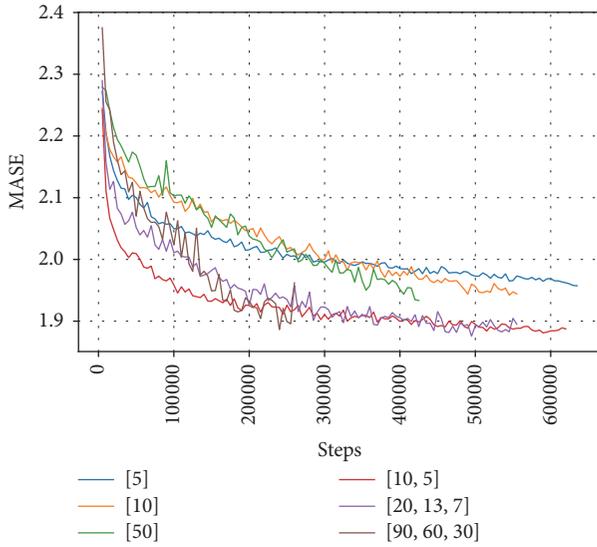


FIGURE 5: Trend in MASE for different FNNs with a sigmoid activation function.

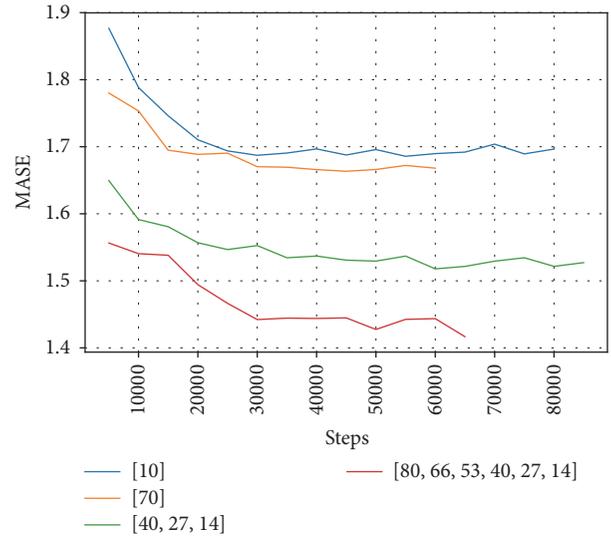


FIGURE 7: Trend in MASE for different FNNs with ReLU activation function and 6 h historical input data.

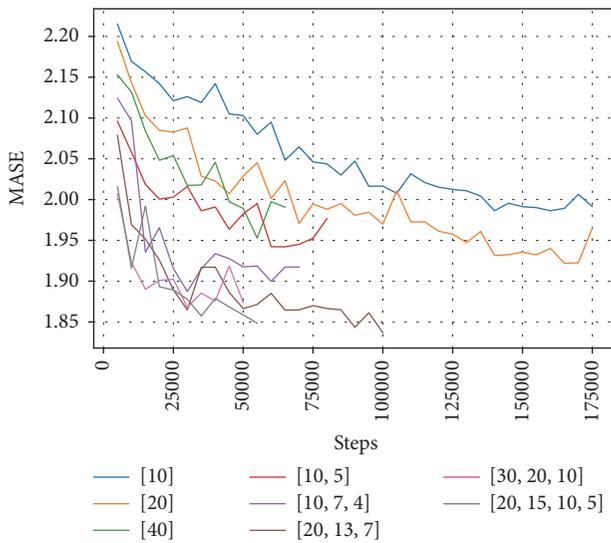


FIGURE 6: Trend in MASE for different FNNs with SELU activation function.

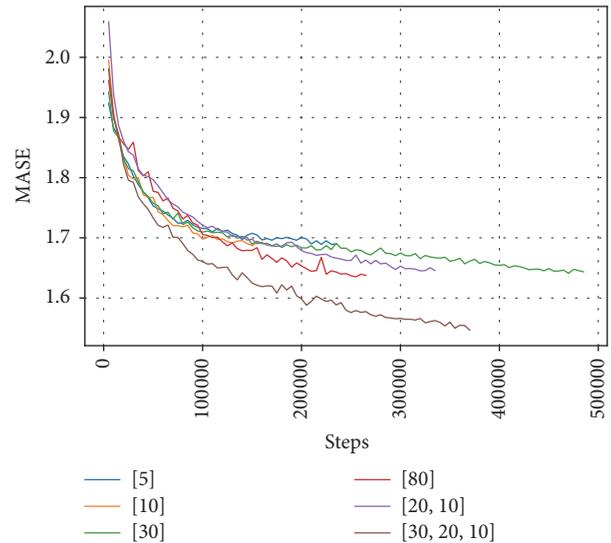


FIGURE 8: Trend in MASE for different FNNs with sigmoid activation function and 6 h historical input data.

FNNs with a sigmoid activation function need three or four times more steps than a ReLU network to converge. In fact, of all the network types studied, they required the highest number of steps to converge. This problem grows with the number of layers due to the vanishing gradient problem.

Figure 6 shows the same curves but for FNNs with a SELU activation function. The improvement resulting from increasing the size of the first layer has a bound that can be overcome by increasing the number of layers. In general, it does not yield better results than the ReLU activation function for our problem, but this is to be expected since the benefits of this function are evident when used in problems that require a large number of layers Klambauer et al. [9].

We then trained estimators similar to the above, but with samples that contained the weather forecast features for every hour t and for the previous 5 hours. As before, Figures 7, 8, and 9 show how the MASE evolved for this new data set. The final result is similar in every case, but better than for the polynomial regression and for the previous FNNs. This shows that these estimators are capable of making good use of time information.

Figures 8 and 9 also exhibit behaviour similar to Figures 5 and 6 respectively, converging to solutions with a smaller MASE. It is interesting to note that although the networks with a SELU activation function behave similarly to those with the ReLU function, the former exhibit fewer overfitting problems as the size of the network grows. In other words,

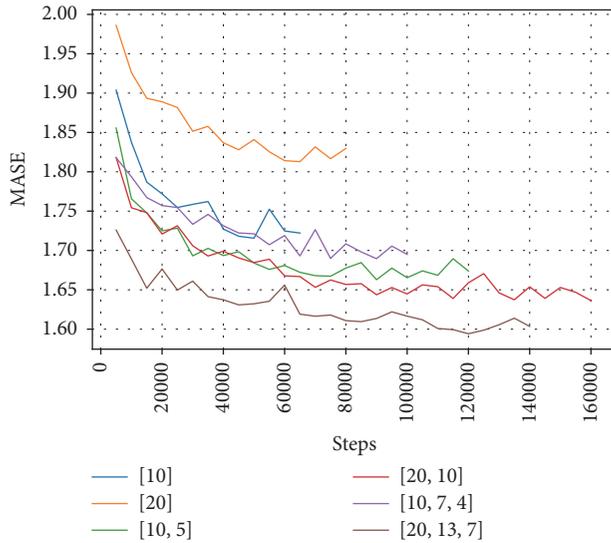


FIGURE 9: Trend in MASE for different FNNs with SELU activation function and 6 h historical input data.

with the SELU activation function, increasing the number of layers does not improve the results but the estimators are trained equally well.

3.2. Convolutional Neural Networks. We trained an CNN-based estimator using the architecture shown in Figure 2 in order to compare its performance with that of previous FNNs and to try to improve the results obtained by the latter. The size of the convolution filters was set at five, the ReLU activation function was selected for the convolutional layers, as is usual for these networks, and the size of the max-pooling window was set at three. Figure 10 shows how the MASE evolved for the validation data set over the course of the iterations for networks with different numbers of channels at the output of each convolution: between 32 and 8. The MASE for networks with at least 32 to 16 channels is very similar to that obtained for FNNs, but the CNNs converge much faster, in approximately half the time. The advantage is that the training for this type of network can be speeded up considerably by using graphics processing units (GPU), although this possibility was not explored for this paper.

Figure 11 uses an image with 256 grey levels to show a representation of the coefficients of the filters trained for the first convolutional layer of the CNN with eight channels. Each filter has one row per input characteristic and shows how the filter uses said characteristic to contribute to the layer's output.

3.3. Recurrent Neural Networks. A similar procedure was used to train the RNNs with LSTM and GRU cells of various sizes for the output y_t . Figures 12 and 13 show the trend in the MASE when training the RNN LSTM and GRU, respectively. Both RNN types exhibit an error that is very slightly larger than that of the FNNs and CNNs. The number of steps needed to complete the training is also very similar, though in reality each step consumes much more time. Although not shown

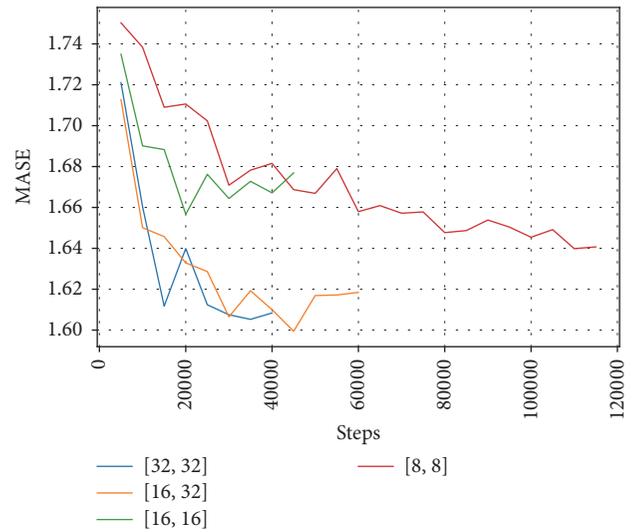


FIGURE 10: Trend in MASE for different CNNs.

in the graph, the amount of time needed by the RNNs was approximately five times that needed by the equivalent FNNs.

It should be noted that no large differences are evident between the RNN LSTM and the RNN GRU. It is also surprising how well both types of RNNs work, even with size 1 cells, which store considerably less information in their status than larger cells.

In light of these results, the problem of predicting wind generation for the daily market is better resolved by providing as input a time series with the forecast for previous hours and using estimators based on FNNs with a ReLU or SELU activation function, or CNNs. Specifically, the latter can be trained with a lower number of iterations and, presumably, in less time with the use of GPUs. As concerns ReLU FNNs and SELU FNNs, it is easier to avoid overfitting in the latter, though, if trained correctly, both are equally accurate.

3.4. Sensitivity to Disturbances. Finally, we selected the best models of each type in order to analyse their behaviour in response to disturbances in the input. In every case, we used models trained with 6 h time series. As we discussed in Section 2.5, we ran the models with and without perturbations in the inputs in order to calculate their sensitivity. The perturbations were applied into the matching inputs for the forecasts of wind velocity and direction for the time when the generation is to be predicted, because these are the inputs with the greatest influence on the model's output.

In this paper the input perturbation ratio and the sensitivity values obtained are shown in percentages. Sensitivities below 100% indicate that the estimators are capable of attenuating the perturbations, while sensitivities in excess of 100% indicate that they are amplified.

Figure 14 compares the average sensitivity of the best estimators for various network classes and sizes for different perturbation levels in wind speed. The figure shows that all of the estimators attenuate the input perturbations, reducing their influence on the output. In every case, the best

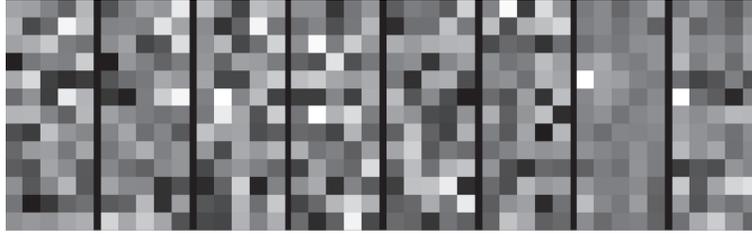


FIGURE 11: Coefficients of the filters in the first convolutional layer with 8 channels.

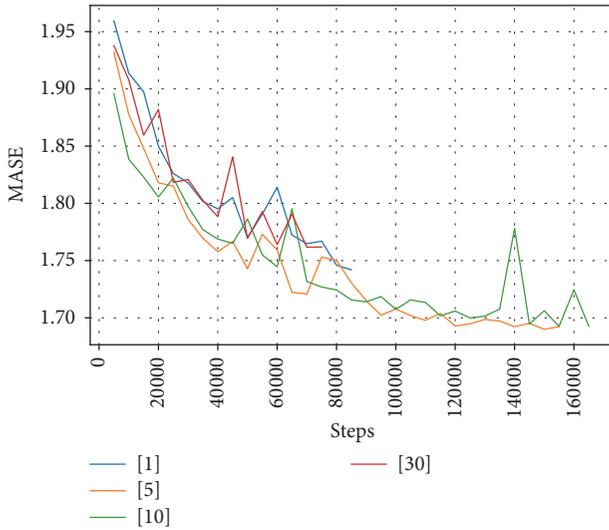


FIGURE 12: Trend in MASE for different LSTM RNNs.

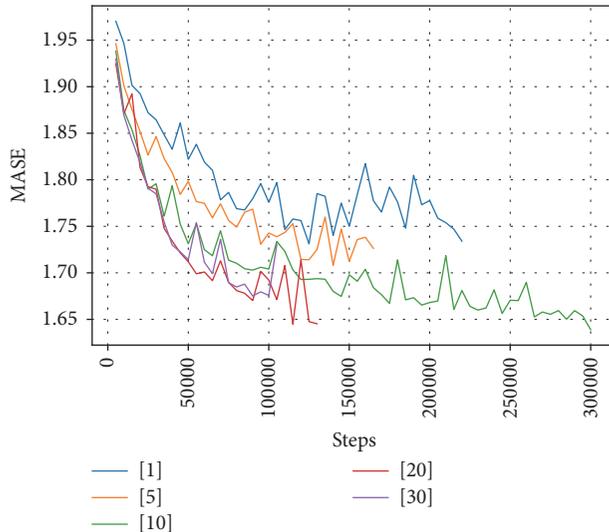


FIGURE 13: Trend in MASE for different GRU RNNs.

performance is exhibited by the estimators based on RNNs, followed by SELU FNNs, sigmoid FNNs, and CNNs.

Figure 15 shows the average sensitivity of the same models for different perturbation ratio values in the wind direction input. In this case the results are much more

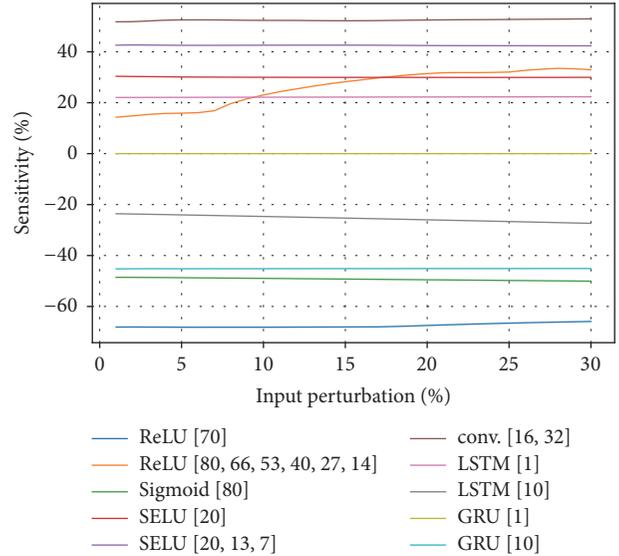


FIGURE 14: Trend in sensitivity for different disturbances in the wind speed.

dissimilar, undoubtedly because the relationship between wind direction and power output is highly nonlinear and much more complicated to model adequately during training. As Figure 15 shows, only the estimators based on SELU FNNs, CNNs, and RNNs cells of size 1 are capable of attenuating this type of perturbations.

In light of these results and considering those obtained previously involving the accuracy of the different estimator types evaluated, the problem of predicting wind power generation for the daily market is better resolved by using estimators based on SELU FNNs or CNNs.

4. Conclusions

In this paper we considered the problem of predicting wind power generation in order to take part in the daily market that regulates the supply and demand in the Spanish electric system. We used deep-learning techniques to develop different predictors based on neural networks that were trained using data provided by the MADE wind farm, operated by the ITER on the island of Tenerife.

The predictors evaluated are based on feedforward neural networks of varying sizes and with different activation functions, convolutional neural networks, and recurrent neural

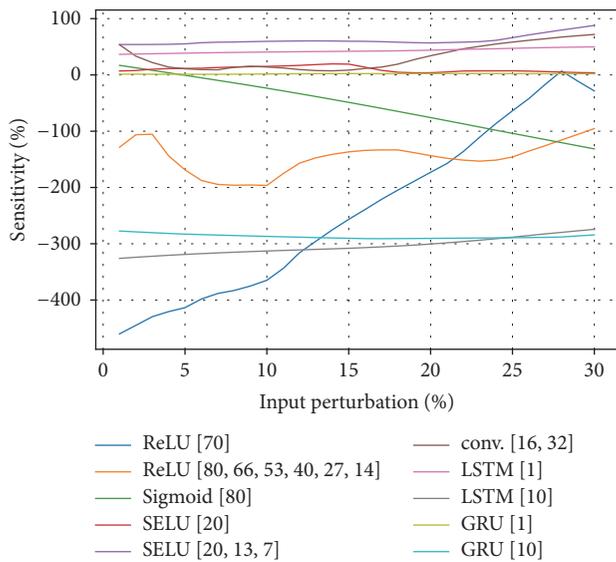


FIGURE 15: Trend in sensitivity for different disturbances in wind direction.

networks. The conditions were the same as those employed for the farm with the polynomial model now in use, namely, relying on the weather forecast at least 24 hours in advance to output a predicted generation for the farm. The methodology was checked by training and validating the model with samples taken every hour during the past three years.

The results were adequate, yielding better results than the 1-hour reference naive forecast estimator and the polynomial model used at the wind farm. Specifically, the use of time series for the input samples proved to be the best way to minimise the error. Moreover, of the different types of neural networks evaluated, the CNNs and FNNs with the ReLU or SELU activation function were shown to be the most accurate, although the differences between the best candidates from the various network types were not significant. The traditional sigmoid FNNs are on a par with the other types trained, though they converge much more slowly during training.

Finally, we conducted a sensitivity analysis of the models, which revealed that trained neural networks are able to attenuate some input disturbances. For disturbances in the wind speed input, the best candidates from every network type were able to attenuate the disturbances, though this is much more difficult to achieve with perturbations in the wind direction input, which even caused some network types to amplify the perturbations. In this case, the CNN, SELU FNNs, and the various RNN types exhibit the best performance. Taking all the results into consideration, the best neural network estimators, from the standpoint of offering the lowest absolute error and being the least sensitive to perturbations, are those based on SELU FNNs and CNNs.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors wish to thank the Institute of Technology and Renewable Energies (ITER), and especially Guillermo Galván García, Head of the Wind Department, and Jonatán Felipe García, Manager of Engineering Department R&D Projects, for their collaboration.

References

- [1] M. Abadi, A. Agarwal, P. Barham et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015.
- [2] H. Yañez-Badillo, R. Tapia-Olvera, O. Aguilar-Mejía, and F. Beltran-Carbajal, "On Line Adaptive Neurocontroller for Regulating Angular Position and Trajectory of Quadrotor System," *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 14, no. 2, pp. 141–151, 2017.
- [3] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, pp. 1139–1147, PMLR, 2013.
- [4] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research (JMLR)*, vol. 12, pp. 2121–2159, 2011.
- [5] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, no. 1, pp. 79–82, 2005.
- [6] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.
- [7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [8] X. Glorot, X. Glorot, and Y. Bengio, in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10)*, Society for Artificial Intelligence and Statistics, 2010.
- [9] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-Normalizing Neural Networks," 2017, <http://arxiv.org/abs/1706.02515>.
- [10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.
- [11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [12] K. Cho, B. van Merriënboer, C. Gulcehre et al., "Learning phrase representations using rnn encoder-decoder for statistical machine translation," <http://arxiv.org/abs/1406.1078>.
- [13] D. L. Marino, K. Amarasighe, and M. Manic, "Simultaneous generation-classification using LSTM," in *Proceedings of the IEEE Symposium Series on Computational Intelligence, SSCI '16*, Greece, December 2016.
- [14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, <http://arxiv.org/abs/1412.3555>.
- [15] J. Quan, X. Feng, G. Su, and G. Chen, *Chinese Journal of Rock Mechanics and Engineering*, vol. 26, article 2654, 2007 (Chinese).

Research Article

Low Speed Longitudinal Control Algorithms for Automated Vehicles in Simulation and Real Platforms

Mauricio Marcano , José A. Matute, Ray Lattarulo, Enrique Martí, and Joshué Pérez

Tecnalia Research & Innovation, Derio, Spain

Correspondence should be addressed to Mauricio Marcano; mauricio.marcano@tecnalia.com

Received 24 November 2017; Accepted 19 February 2018; Published 29 March 2018

Academic Editor: Victor M. Becerra

Copyright © 2018 Mauricio Marcano et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Advanced Driver Assistance Systems (ADAS) acting over throttle and brake are already available in level 2 automated vehicles. In order to increase the level of automation new systems need to be tested in an extensive set of complex scenarios, ensuring safety under all circumstances. Validation of these systems using real vehicles presents important drawbacks: the time needed to drive millions of kilometers, the risk associated with some situations, and the high cost involved. Simulation platforms emerge as a feasible solution. Therefore, robust and reliable virtual environments to test automated driving maneuvers and control techniques are needed. In that sense, this paper presents a use case where three longitudinal low speed control techniques are designed, tuned, and validated using an in-house simulation framework and later applied in a real vehicle. Control algorithms include a classical PID, an adaptive network fuzzy inference system (ANFIS), and a Model Predictive Control (MPC). The simulated dynamics are calculated using a multibody vehicle model. In addition, longitudinal actuators of a Renault Twizy are characterized through empirical tests. A comparative analysis of results between simulated and real platform shows the effectiveness of the proposed framework for designing and validating longitudinal controllers for real automated vehicles.

1. Introduction

Nowadays, traffic jam is a well-known cause of the increase in travel times, accidents, and fuel consumption in urban areas. In Europe, more than 200,000 traffic jams have been identified across 123 cities in September 2016 [1]. This study shows that the annual cost associated is more than €4.9 billion. This problem is being currently tackled by authorities, industry, and research centers.

Automated vehicles and Advanced Driver Assistance Systems (ADAS), as part of the Intelligent Transportation Systems (ITS), are considered as short and medium term solutions for the previous problem. Based on the last ERTRAC (European Road Transport Research Advisory Council) report [2], traffic jam assist systems are partially available in the market. However, most of these solutions are restricted to high speed (i.e., more than 30 Km/h), with automation level 2, based on the SAE J3016 standard [3].

The most important advances in this area include Cruise Control (CC) in high speed [4], dynamic stability control

[5], pedestrian detection systems combined with collision avoidance [6], and semiautonomous parking [7].

In spite of the aforementioned advances, validation of some functionalities is a complex and tough task. Based on the classical testing proposed in [8], automated vehicles need to drive around 275 million Km to demonstrate a performance superior to humans. This would take around 12.5 years considering a fleet of 100 vehicles driving nonstop. In contrast, the self-driving car project [9] from Waymo has only reached 5.6 million miles driven in real conditions [10]. However, they have recently announced a simulation environment able to drive 12.8 million Km per day, where they have drove 4 billion Km in total. The previous is aligned with the suggestions made in [8], encouraging simulation testing to validate automated vehicles functionalities.

Currently, most simulation tools are focused on computational perception [11], test benches for hardware in the loop testing [12], and evaluation of human acceptance regarding different ADAS [13]. However, from the control point of view, few solutions have been implemented.

In [14] authors present a comparative study for low speed longitudinal controllers, based on a gasoline-propelled vehicle, using classical and intelligent control techniques; nonetheless, the controllers were designed and validated directly in the real vehicle and no comparison against simulations was presented. Other authors worked in throttle and brake controllers in vehicle following applications presenting real and simulated results [15] based on fixed and variable gains PID controllers, but without an extensive comparison between the platforms.

This paper presents a use case for Cruise Control where three longitudinal low speed control techniques are designed, tuned, and validated using an in-house simulation framework and later applied in a real vehicle. Control algorithms include a classical PID, an adaptive network fuzzy inference system (ANFIS), and a Model Predictive Control (MPC). The experimental procedure aims to validate the simulation platform as an effective tool for ADAS functionalities development, able to save time and cut costs by reducing the development effort that involves working with the real vehicle.

The rest of the paper is organized as follows: Section 2 explains the vehicle model and its parameters; additionally, an empirical characterization of longitudinal actuators in real test platform is described. Section 3 describes the algorithms applied on low speed control. Tests and results of the three controllers implemented in both simulation and real vehicle are presented in Section 4. Lastly, conclusions and recommendations for future works are given.

2. Description of Platforms

Longitudinal speed controllers are implemented in both simulation and real vehicle. This section offers a description of both platforms based on an automated driving general framework. First, a brief summary of the global architecture is presented, followed by an explanation of the instrumentation and automation of Renault Twizy. Lastly, the simulation environment used to emulate the real vehicle through a mathematical model will be detailed.

2.1. Automated Driving General Framework. The control algorithms were developed under the automated driving framework presented by David González and Joshué Pérez in [16]. This architecture has been presented in previous works, including a validation of lateral controllers in simulation [17]. The global frame is composed of six blocks that abstract a generic processing pipeline for automated driving applications, as detailed below:

- (1) *Acquisition*: this module gathers information coming from on-board sensors and does preprocessing tasks such as time stamping, outlier rejection, or reference system transformations. Current implementation accepts data from a variety of sources, including differential GPS, Inertial Measurement Unit (IMU), vision sensors, and others. Low level CAN communication with vehicle internal system is also considered within this module.
- (2) *Perception*: perception module is in charge of mixing raw data received from acquisition block to generate

information useful for automated driving tasks. Generated data includes ego-vehicle estimation, obstacle detection/classification, lane detection, and traffic signal recognition. This module features machine learning, deep learning, and sensor fusion algorithms that provide an accurate, robust, and consistent view of the world. Those techniques allow reducing the uncertainty of original measure and characterize its associated error.

- (3) *Communication*: communication module provides information coming from external entities (vehicles and infrastructure). From a wider perspective, this module establishes a bidirectional information flow to manage the integration of the vehicle in a cooperative driving environment.
- (4) *Decision*: decision module determines the trajectory and speed plan to be followed by the vehicle. This is done in three consecutive steps: the global planner generates a feasible trajectory from the available road map. Local planner refines it to optimize different parameters (e.g., comfort, safety), through minimization of curvature and its derivatives, acceleration, and jerk. This is achieved using different types of curves, such as Joshué Pérez et al. [18]. The last step is behavioral planner, which adapts the route to face dynamic events as obstacles in the road, other vehicles' maneuvers, or unexpected traffic restrictions.
- (5) *Control*: it generates normalized control actions over longitudinal and lateral vehicle actuators, to ensure that the plan produced by decision module is effectively accomplished. This work contributes to this specific block, along with the actuation module presented below.
- (6) *Actuators*: abstraction layer transforms normalized control outputs to real action signals over the actuators: throttle, brake, and steering wheel. It enables transparent switch between real and simulated environments.

As shown in Figure 1, the general framework is independent of the test platform, making this approach a versatile tool for testing control algorithms for automated driving applications in both real and simulated vehicles. This work is based in Matlab/Simulink.

2.2. Real Test Platform. The experimental platform is a Renault Twizy 80 (see Figure 1), a 4-wheel electric vehicle capable of reaching a maximum speed of 80 [Km/h] with vehicle and motor parameters depicted in Table 1. It has been instrumented to control the steering and the pedals through CAN bus network.

As shown in Figure 2, the three systems are managed by different electronic devices and actuators. First, the steering wheel is rotated by a stepper motor controlled through PWM signals. Throttle command is communicated to vehicle's Engine Control Unit (ECU) through an analog voltage signal between 0 and 10 VDC. The brake pedal is coupled with a mechanical linear actuator. All the instrumentation is

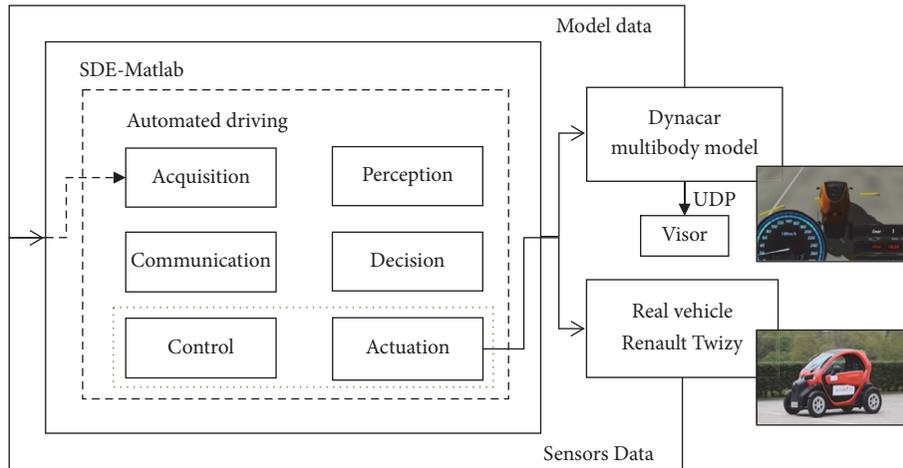


FIGURE 1: Test platforms and general framework.

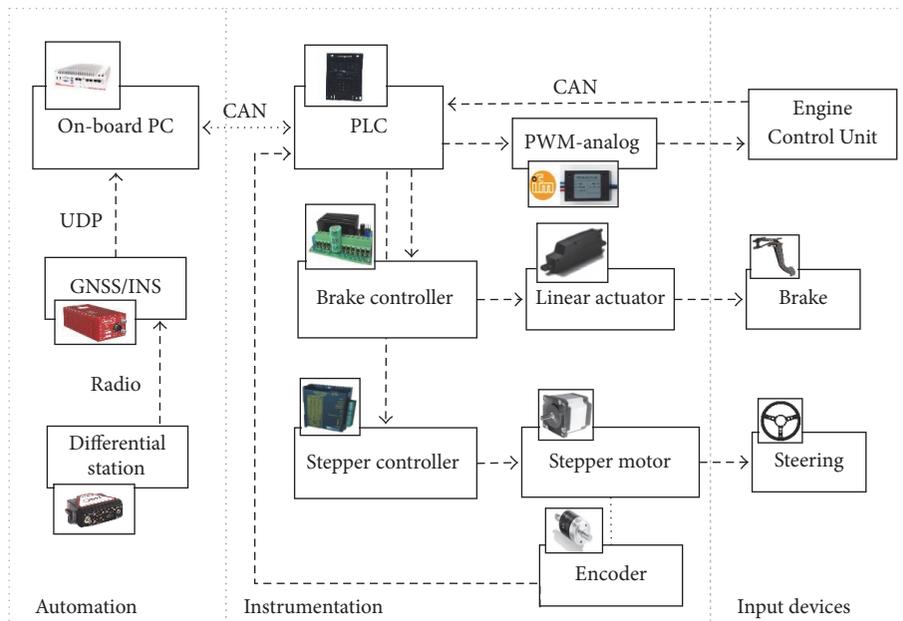


FIGURE 2: Instrumentation and automation of a Renault Twizy.

controlled by a Programmable Logic Controller (PLC) that acts as an interface between the actuators and the central computer. The former executes the automated driving general framework and the control algorithms. It is also in charge of collecting the data from the external sensors such as GPS and IMU. More details on sensors and devices are listed in Table 2.

2.3. Simulation Test Platform. The virtual platform is based on Dyncar, an integrated solution for design of electric and hybrid vehicles that features a vehicle physical model based on multibody formulation, making use of relative coordinates and semirecursive equations of motion based on velocity transformation. Suspensions are considered as macro-joints, and the behavior is modeled using lookup tables. As shown in Figure 3, local Cartesian coordinates of the chassis frame are located at the middle of the frontal track

width (C), cardan angles providing wheels orientation with respect to the chassis frame are located at the knuckles (K), and kinematic expressions for the macro-joints consider the position, velocity, and acceleration levels of the wheels (W) [19, 20].

The powertrain system includes different engine and transmission models. The electric motor is modeled defining values of motor torque [N-m], motor speed [rpm], and throttle pedal position [in%] through a lookup table and interpolating linearly during simulations. It is possible to introduce regeneration providing a constant braking torque as long as the throttle pedal is fully released and the motor speed is over certain value. A clutch is modeled using a linear interpolation within a predefined table between clutch's displacement and torque transmitted; however a fixed clutch was selected due the real platform characteristics. A

TABLE 1: Real platform parameters.

Mass (kg)	611.500
Dimensions (l)	611.500
CG location x, y, z (m)	-0.928, 0.000, 0.488
Wheelbase (m)	1.686
Track width (m)	1.094
Inertia I_x , inertia I_y , inertia I_z (kg-m ²)	243.175, 430.166, 430.166
Front wheel radius (m)	0.265
Rear wheel radius (m)	0.281
Motor type	Three-phase asynchronous w/regeneration
Power	11 [hp] between 220 and 785 [rad/sec]
Maximum torque	57 [N-m] between 0 and 220 [rad/sec]
Transmission reduction	1:9.23

TABLE 2: Sensors and electronic devices.

Position and inertial data	GNSS-aided IMS + base station
On-board computer	Fanless, i7-6700TE, 32 GB RAM
Brake system	Linear actuator, 750 N 20 A DC motors driver
Steering system	High torque hybrid stepping motor Magnetic encoder with CANopen 20-80 V stepping motor driver

gearbox is modeled calculating the output torque multiplying the input torque by transmission ratio and gear efficiency coefficient. The final traction moment is calculated after considering a fixed differential sending the propulsion flow to the rear wheels [21].

The brake system consists in a simplified model (see (1)) which combines the brake pedal travel (E_b) [mm], maximum pressure on the circuit (P_{max}) [MPa], percentage of braking pressure ($K_{i,j}$) [%], and brake pad capacity ($C_{i,j}$) [N-m/MPa] to obtain the braking torque on the wheels ($M_{i,j}$) [N-m]. Subindex i refers to the location of each wheel (i = front/rear, j = left/right) [21].

$$M_{i,j} = E_b P_{max} K_{i,j} C_{i,j}. \quad (1)$$

2.4. Characterization of Longitudinal Actuators. Throttle and brake pedal positions specify the capacity of acceleration and deceleration of the vehicle. Both simulation and real test platform have a signal control range that goes from 0 to 100% of pedals location. In the real platform, the intermediate instrumentation devices (acceleration potentiometer signal and brake linear actuator) introduce undesired effects such as biases, scaling, or delays that play an important role in low level of control framework, as well as the way longitudinal actuators act over the pedals (Figure 2).

In order to approximate the actuator behavior in simulations, moving tests on real test platform were evaluated at different throttle and brake pedals positions. Each test starts from a stopped position with a 9 s period of constant acceleration, followed by 4 s with no action over pedals and

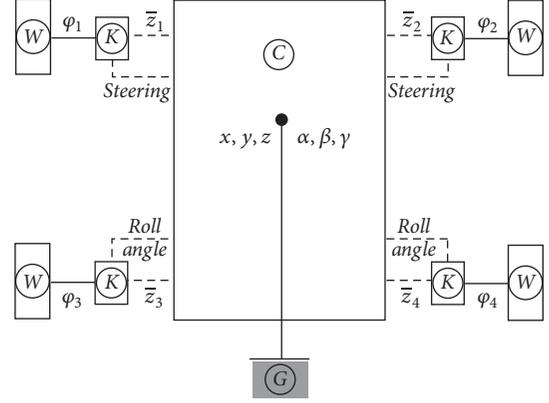


FIGURE 3: Dynacar multibody formulation [20].

finishing with 2 s of constant braking. A total of 7 different tests were performed, varying both pedals position from 40% to 100% of their maximum displacement in 10% steps (Figure 4(a)). These three periods were useful for evaluating actuator's influence on traction, regeneration, and braking.

The inertia of physical devices as actuators or the brake hydraulic system strongly affects the longitudinal behavior of the vehicle. Additionally, an electrical lumped lag can be found after pushing the throttle pedal due a combination of devices within the control framework and the vehicle. Therefore, mechanical and electrical lags (τ) during traction and braking observed in moving tests (Figure 4(b)) can be modeled in simulations as a discrete transfer function considering a sampling time (t_s):

$$H(z) = \frac{(t_s)}{(\tau z + (t_s - \tau))}. \quad (2)$$

Average acceleration values during traction and braking periods were normalized in $[-1; 1]$, obtaining curves that represent the real influence of actuators over pedals for different signal control values sent from on-board computer (Figure 5). Throttle response has a linear behavior due a direct connection with the potentiometer. In contrast, braking response has a polynomial behavior determined mostly by the orientation of the linear actuator with respect to brake pedal. Additionally, both pedals are also affected by range and scales of voltages previously programmed in the PLC. These results are very useful for resembling truthfully the longitudinal behavior of the vehicle during simulations, making an equivalence between the signal control and the actuators influence over the vehicle.

3. Design of Control Algorithms

This section presents the design of three longitudinal controllers for low speed CC; these are based on fixed gains (PID), trained data (ANFIS), and a predictive model (MPC). The algorithms depend on the speed error calculated as the difference between controller reference and the actual speed. In simulation inertial data is provided by the virtual model, while the experimental test uses the GNS/INSS sensor information.

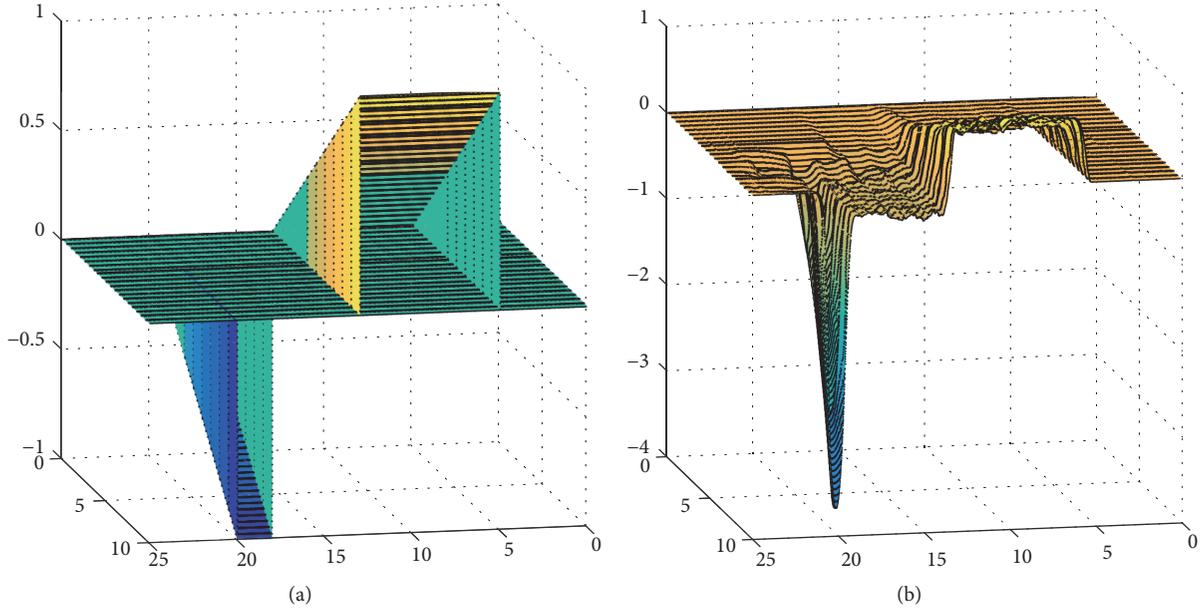


FIGURE 4: (a) Signal control references from on-board computer to longitudinal control actuators and (b) longitudinal acceleration in real test platform in 10 different tests.

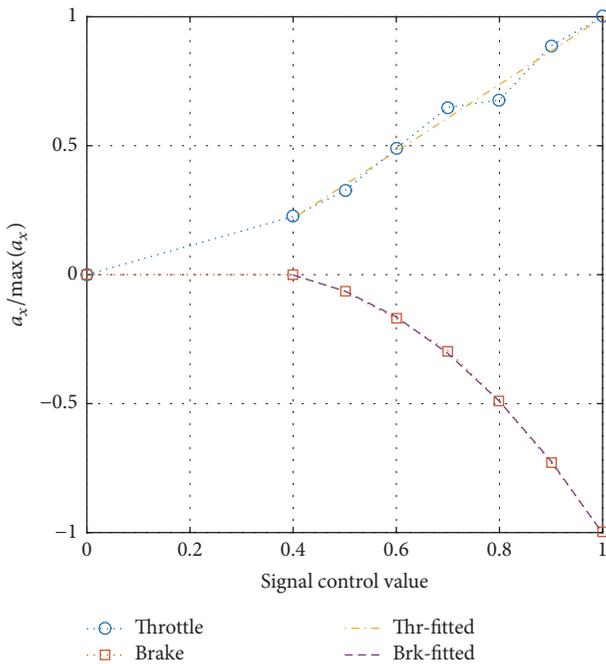


FIGURE 5: Longitudinal behavior of real test platform.

Vehicle speed depends on both accelerator and brake. However, since the dynamics of both actuators present considerable differences (as discussed in Section 2), an individual controller for each mechanism is needed. Figure 6 describes the control block within the automated driving framework. Accelerator and brake controllers are never active at the same time: they are switched depending on the sign of the speed error. Additionally, a lateral controller based on fuzzy

logic [22] has been previously adjusted in order to test the longitudinal algorithms along a specific route.

3.1. PID. Proportional-Integral-Derivative controller, known as PID, is a widely used technique in feedback control applications. According to [23], around 95% of the implemented control loop processes are based on this technique. PID output depends on an input error signal and three tunable gains changing the response of the system. A major advantage is that it reduces the steady state error through the integral action; in addition, derivative gain allows predicting future actions. Equation (3) shows the general formula for a discrete time PID controller using the forward Euler method.

$$u(z) = K_p + K_i T_s \frac{1}{z-1} + \frac{K_d(z-1)}{T_s z}, \quad (3)$$

where $u(z)$ is the control output and K_p , K_i , K_d represent the proportional, integral, and derivative gains, respectively; T_s corresponds to system sample time (1 ms in simulation and 10 ms in real platform). PID controllers have been used for different automated driving longitudinal applications and appear as a practical solution for CC applications where control signals can be modified according to the speed error and selecting the proper gains. Table 3 shows the PID parameters used in this work. These values were tuned in the simulator environment and then tested in the real vehicle.

3.2. ANFIS. Integrating human experience into automated driving algorithms is an idea that has received great interest over the last years. A commonly used technique is fuzzy inference system (FIS), which offers a solution for including human knowledge into controllers design, through a set of membership functions and linguistic if-then rules. Fuzzy controllers have proven its usefulness in different automation

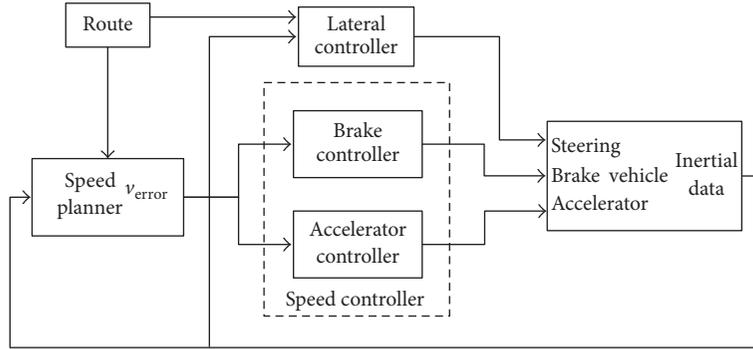


FIGURE 6: Decision and control block of general automated driving framework.

TABLE 3: PID gains.

	K_p	K_i	K_d
Accelerator	0.5	0.005	0
Brake	0.15	0.01	0.05

use cases, such as lateral control [24] and Adaptive Cruise Control (ACC) [25].

However, manual translation of human knowledge into a fuzzy logic database can be a daunting task, and it is complex to know if the designed membership functions and rules are the best choice. This drawback is addressed by ANFIS, combining Takagi and Sugeno fuzzy systems [26] with the learning capacity of adaptive neural network based on human-trained data.

This work implements a two-input-one-output neuro-fuzzy controller designed to depend on two kinds of speed error, absolute (s_{ea}) and relative (s_{er}), as shown in the following equations:

$$\begin{aligned}
 s_{ea} &= s_{ref} - s, \\
 s_{er} &= \frac{s_{ea}}{\Delta ref} \cdot 100, \\
 \Delta ref &= \begin{cases} s_{ref}(i) - s_{ref}(i-1), & \text{if } s_{ref}(i) \neq s_{ref}(i-1) \\ s_{ref}(i), & \text{otherwise,} \end{cases}
 \end{aligned} \tag{4}$$

where s_{ref} is the speed reference and Δref is the step size in a reference change (in Figure 9, Δref values are 15, -5, -5, 5, 5, -7, and -8 [Km/h], resp.). Adding relative error aims to capture the behavior over pedals for the same absolute error, depending on whether it occurs when acceleration is started, maintained, or finished. This strategy for adjusting speed to a reference resembles human-centered driving style at low speeds: pedal is pressed softly at the beginning, then held, and progressively relieved when getting close to reference.

The data for training the ANFIS longitudinal speed controller was gathered from a series of tests done in the simulator. The scenario consisted of a 1 Km highway with a lateral controller adjusted to maintain the vehicle in the center line. Then, USB pedals for brake and accelerator were used to drive the virtual vehicle while maintaining a reference

TABLE 4: ANFIS parameters.

	Accelerator	Brake
Number of inputs	2	2
Number of outputs	1	1
Training algorithm	Hybrid	Hybrid
Training data pair	118994	74912
Input type	Gaussian bell	Gaussian bell
Output type	Linear	Constant

TABLE 5: ANFIS output parameters.

Abs/Rel	Accelerator			Brake		
	None	Low	High	None	Low	High
None	0.25	-0.03	0.00	-0.52	-0.40	-0.08
Low	0.01	0.00	-0.02	-0.40	-0.54	0.00
High	0.15	0.94	2.37	-0.45	-1.24	0.00

speed shown on the screen. For the accelerator controller six tests were performed; the reference was increased in steps of 5, 10, 15, 20, 25, and 30 [Km/h]. Same strategy was used for training the brake, starting at a maximum speed of 30 [Km/h] and repeating the same six trials but decreasing the speed either.

The network structure is composed of two inputs described through Gaussian membership functions and three descriptors (No Error, Low Error, and High Error) in units of [Km/h]. The output of the system is a normalized control action in $[-1; 1]$, where positive values correspond to accelerator controller and negative values are assigned to the brake. Additional information on the neurofuzzy controller is shown in Tables 4 and 5. The former indicates the output values of the controller based on the inputs description. As a basic design rule, both controllers cannot be activated at the same time. Additionally, the controller was trained using a hybrid algorithm [27].

A representation of the control surface for both accelerator and brake is depicted in Figures 7(a) and 7(b), respectively, proving that outputs are within the limits of the values accepted by the actuators. Moreover, control outputs show being smooth and without sudden changes, guaranteeing that the controller will not output sharp commands.

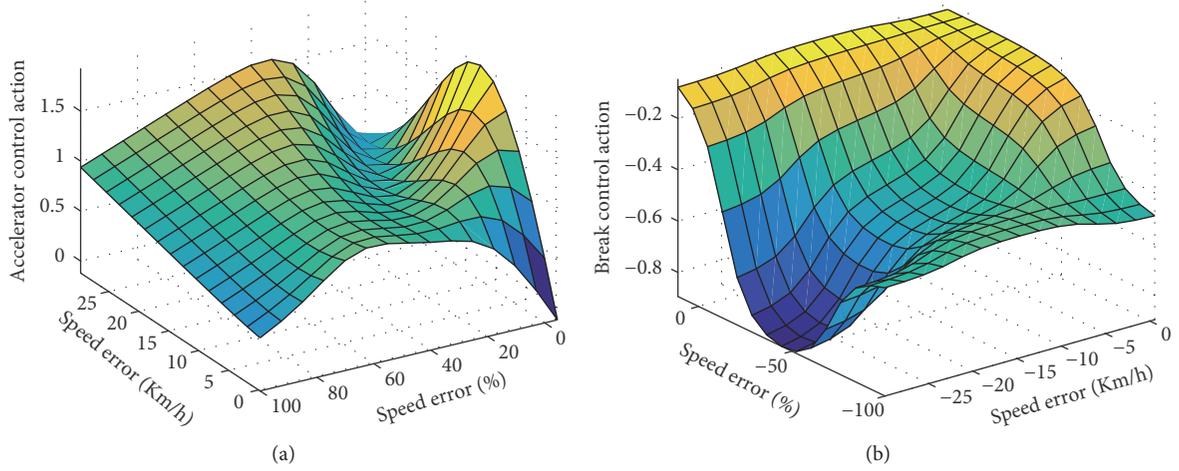


FIGURE 7: (a) Control surface of ANFIS for accelerator; (b) control surface of ANFIS for brake.

The aforementioned algorithm has been designed and trained with data obtained from the simulator, based on the vehicle model explained in Section 3. This responds to the objective of validating the simulation platform as a tool for ADAS design. The benefits include a lower dependence on real vehicle availability and a considerable reduction in trial times.

3.3. MPC. The third algorithm is a model based controller. Model Predictive Control (MPC) [28], also known as receding horizon control, is a concept based on optimal control theory where a plant model and a set of expected inputs are used to predict future system states. It considers a constrained response of the plant and control inputs, optimizing the control values based on an objective function. There are different methods for designing the MPC. For the purpose of this work a real time formulation of a linear model is used.

This method has been largely used in the last decades for applications including high level decision making module for motion planning, along with a secondary feedback control loop [29]; other approaches use open-loop control as a feedforward strategy. One of the major drawbacks of MPC is that it requires a plant model that can describe the real system very accurately. This problem affects especially open-loop solutions. By contrast, feedback based formulations can compensate model errors by updating the states each time the optimization problem is solved.

Next section explains the design of an MPC for CC using a high fidelity vehicle dynamics simulator. Problem formulation and a brief description of the optimization solver are given.

3.3.1. MPC Problem Formulation. The proposed kinematic linear MPC is based on a triple integration chain constructed under three requirements: (i) ability to track reference speed smoothly with low overshoots, (ii) generating acceleration commands feasible for the target vehicle, and (iii) considering the comfort of the passengers. In this sense, the equations that describe the plant are

$$\begin{aligned} \dot{d}_{\text{lon}} &= v_{\text{lon}}, \\ \dot{v}_{\text{lon}} &= a_{\text{lon}}, \\ \dot{a}_{\text{lon}} &= J_{\text{lon}}. \end{aligned} \quad (5)$$

And those can be represented on the classical state-space description, as

$$\begin{bmatrix} \dot{d}_{\text{lon}} \\ \dot{v}_{\text{lon}} \\ \dot{a}_{\text{lon}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} d_{\text{lon}} \\ v_{\text{lon}} \\ a_{\text{lon}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} J_{\text{lon}}. \quad (6)$$

d_{lon} , v_{lon} , a_{lon} , J_{lon} are the longitudinal distance, velocity, acceleration, and jerk, respectively. In this formulation the control variable is jerk, although requirements and constraints in this work are related to future acceleration values. Additionally, the constraints given by the three requirements are

$$\begin{aligned} \min \{v_i, v_{\text{ref}} - 0.5\} &\leq \mathbf{v}(t+k | t) \leq v_{\text{ref}} + 0.5, \\ -a_{\text{dec}} &\leq \mathbf{a}(t+k) \leq a_{\text{acc}}, \\ -j_{\text{dec}} &\leq \mathbf{j}(t+k | t) \leq j_{\text{acc}}, \end{aligned} \quad (7)$$

where $k = k_s t_s$ for $k_s = 0, 1, 2, \dots, N_s$ and t_s is the sample time and N_s is the number of samples used for the MPC prediction horizon. Additionally, v_i is the current speed of the vehicle, v_{ref} is the reference speed, and the factor 0.5 [m/s] is used to limit the overshoot in controller response. Maximum vehicle deceleration is represented by a_{dec} , while a_{acc} is the magnitude of the maximum acceleration; both parameters are related to the dynamic requirements of the vehicle. Lastly, j_{dec} is the maximum deceleration jerk and j_{acc} is the maximum acceleration jerk, which are set according to a passenger comfort criterion as explained in [31].

The objective function used in the optimization problem is the least squared error of the speed:

$$J = \min \left\{ \int_{t_i}^{t_i+t_H} (v - v_{\text{ref}})^2 dt \right\}, \quad (8)$$

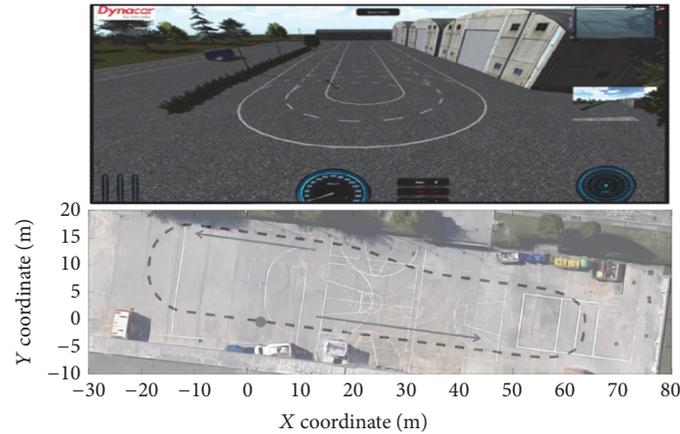


FIGURE 8: Derio 700 circuit: simulation and real one [30].

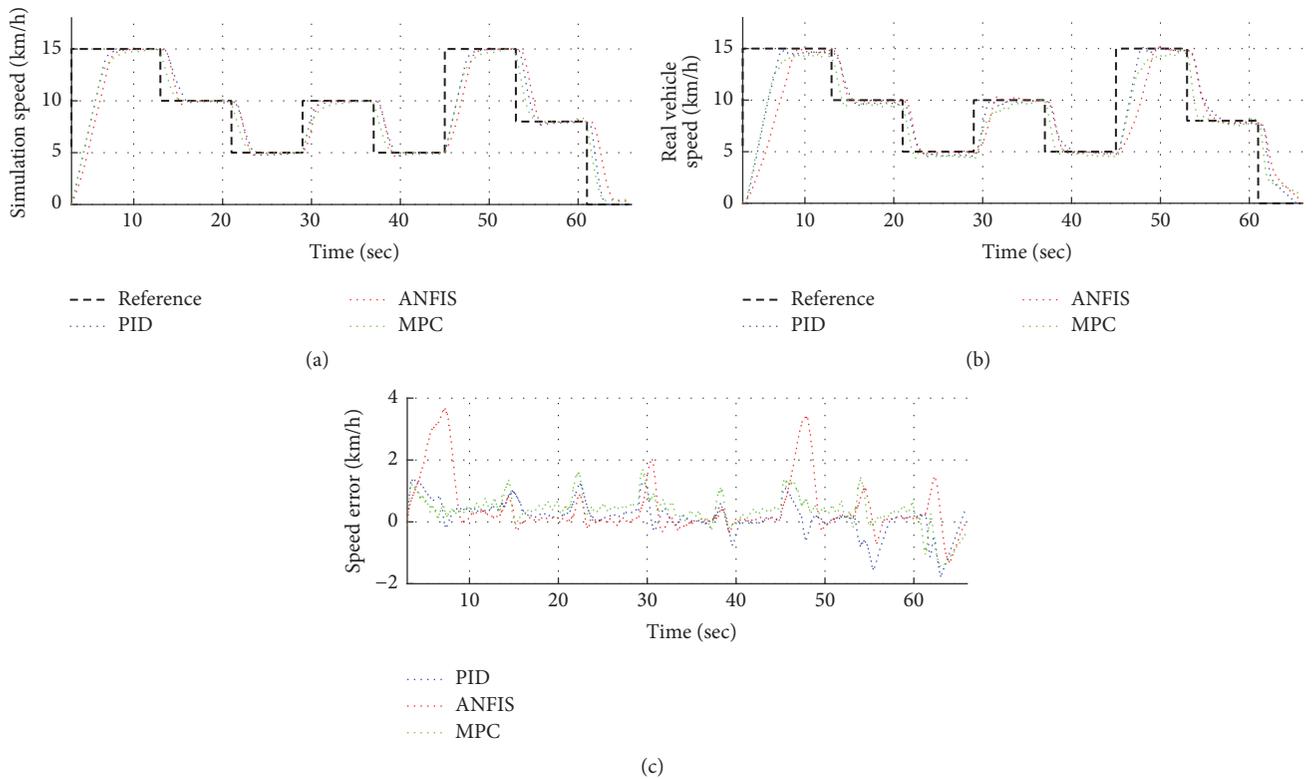


FIGURE 9: Speed response in simulation and real vehicle.

where t_i is the time at which control actions are optimized, and t_H is the prediction horizon; that is, $t_H = t_s \cdot N_s$. In addition, Table 6 summarizes the parameters used in the optimization problem for the MPC.

3.3.2. MPC Solver. The solver used to calculate the solution of the MPC is the ACADO toolkit [32]. It is an open-source and self-contained library based on C++ code, designed to solve linear and nonlinear dynamic problems under multiobjective optimization functions. Moreover, ACADO toolkit functionality supports interfacing with Matlab/Simulink environment [33], which makes it an ideal

option for integration with the presented automated driving framework.

4. Test and Results

An evaluation of three CC algorithms is presented, based on a comparative analysis between controllers in both simulation and real vehicle. First, speed response is analyzed using the tracking error as a performance indicator. Additionally, mean and maximum absolute acceleration are discussed.

Experimental tests were carried out within the facilities of Tecnalia Research & Innovation, in a closed circuit of 180

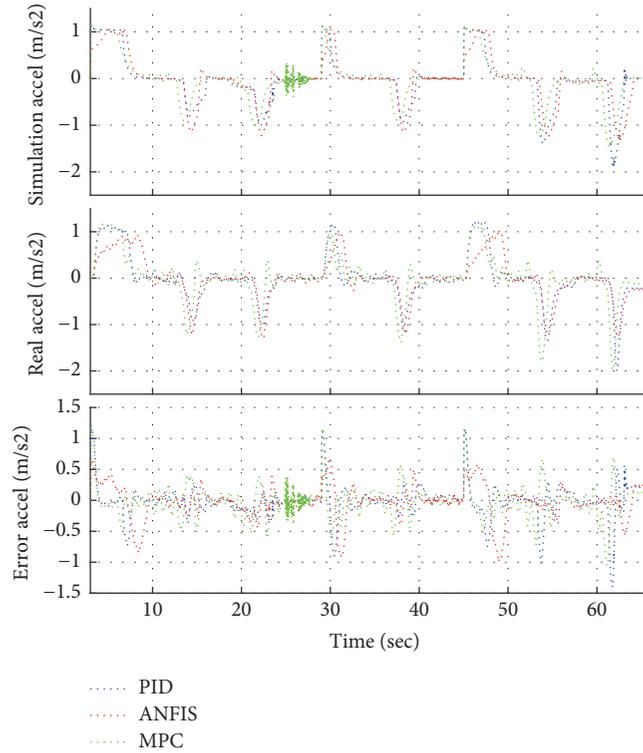


FIGURE 10: Acceleration response in simulation and real vehicle.

TABLE 6: MPC parameters.

Parameter	Value
v_i [m/s]	Dynamic
v_{ref} [m/s]	Dynamic
a_{dec} [m/s ²]	3.15
a_{acc} [m/s ²]	1.15
j_{dec} [m/s ³]	2
j_{acc} [m/s ³]	2
t_s [s]	0.5
N_s	10

[m] length. It is comprised of a straight path, two turns, and one lane change. Same scenario was replicated in the virtual environment as shown in Figure 8. Vehicle maximum speed was set to 15 [Km/h] with multiple reference changes delivered by the speed planner.

4.1. Speed Control Analysis. The speed response of the three controllers is depicted in Figure 9. It includes results from simulation and experimental tests, along with the error between both platforms. All controllers stick to the reference, showing similar response in simulated and real vehicle. The PID shows the faster settling time for the three acceleration maneuvers ($t \in [3-13] \cup [29-37] \cup [45-53]$). Meanwhile, MPC is the slower in reaching the steady state when accelerating; however, it is the faster in braking maneuvers. Neurofuzzy controller is the less aggressive; nonetheless, far from being a disadvantage, it is a good indicator of acceleration comfort.

Moreover, when analyzing the tracking error once the controller has reached 95% of the reference, the neurofuzzy technique shows smaller tracking error.

Previous observations can be complemented with Table 7, using average, median, and RMS of the tracking errors. In simulation, the best controller is the MPC with an overall average error of 1.27 [Km/h], followed by the PID and the ANFIS. Same order is obtained when evaluating this variable in the real vehicle. An important result is that analyzing median and RMS, the order is the same in the two platforms. This information suggests that the simulation tool represents the behavior of the real vehicle correctly.

In addition, Figure 9(c) shows the error comparing the controllers in simulation and experimentally. The neurofuzzy controller has the highest error peak; the controllers output plot (Figure 12) explains this behavior: the neurofuzzy system command softens initial acceleration that is progressively increased. This situation makes real vehicle reaction slower, and although the controller shows similar acceleration in both cases (≈ 1 [m/s²]), a delay in the acceleration action produces a temporary error peak. Moreover, the average, mean, and RMS errors are below 0.5 [Km/h] for all the controllers; based on smaller reference change (5 [Km/h]), it represents an approximated overall error of 10% between the simulated and the real vehicle.

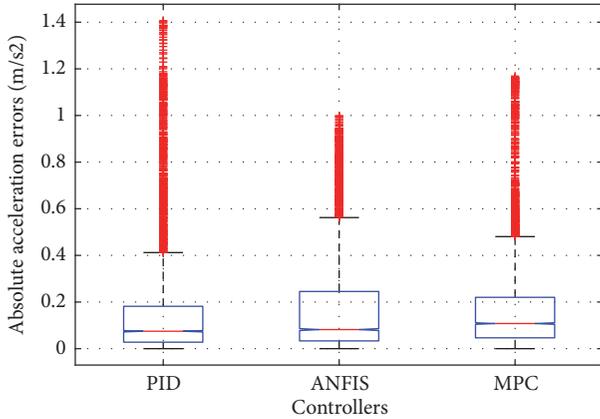
4.2. Acceleration Response Analysis. Acceleration performance is evaluated based on Figure 10. The test around the circuit produced 8 acceleration peaks, one per reference change, with a similar response in both simulated and real platforms. Results are consistent with the conclusions

TABLE 7: Tracking errors [Km/h].

	Track error avg			Track error median			Track error RMS		
	Sim	Real	Diff	Sim	Real	Diff	Sim	Real	Diff
PID	1.45	1.65	0.20	0.16	0.36	0.2	3.04	3.28	0.24
ANFIS	1.70	2.05	0.35	0.07	0.18	0.11	3.49	3.93	0.44
MPC	1.27	1.56	0.29	0.19	0.50	0.31	2.80	3.08	0.28

TABLE 8: Acceleration errors [m/s^2].

	Acceleration avg			Acceleration median			Acceleration RMS		
	Sim	Real	Error	Sim	Real	Error	Sim	Real	Error
PID	0.279	0.276	0.003	0.038	0.070	0.032	0.502	0.480	0.022
ANFIS	0.282	0.273	0.009	0.049	0.070	0.021	0.486	0.442	0.044
MPC	0.278	0.286	0.08	0.083	0.100	0.017	0.471	0.495	0.024

FIGURE 11: Boxplot of absolute acceleration errors between simulation and real platform [m/s^2].

extracted in the speed analysis: ANFIS controller softness can be observed while both PID and MPC techniques produce faster reactions.

Table 8 shows a small error when comparing the mean of the absolute acceleration values in the two platforms (below $0.1 [m/s^2]$). Median and RMS results present errors below $0.05 [m/s^2]$. These results are combined with the quartile analysis in Figure 11, showing that the difference between real and simulated vehicle acceleration is smaller than $0.2 [m/s^2]$ of 75% of the time. Red markers represent outlier values (around 9% of the collected data); most of them are caused by the variable delay between the simulated signals and the real ones.

Results discussed above can be extended by considering the errors during the eight zones of peak acceleration (Table 9). Data shows an average error of $0.09 [m/s^2]$ for the PID controller, $0.07 [m/s^2]$ corresponding to the ANFIS technique, and $0.28 [m/s^2]$ for the MPC. In conclusion, the neurofuzzy controller presents smaller values for acceleration averages and errors between the simulated and the real vehicle. Moreover, overall errors between both vehicles are below 10% based on a maximum absolute acceleration of

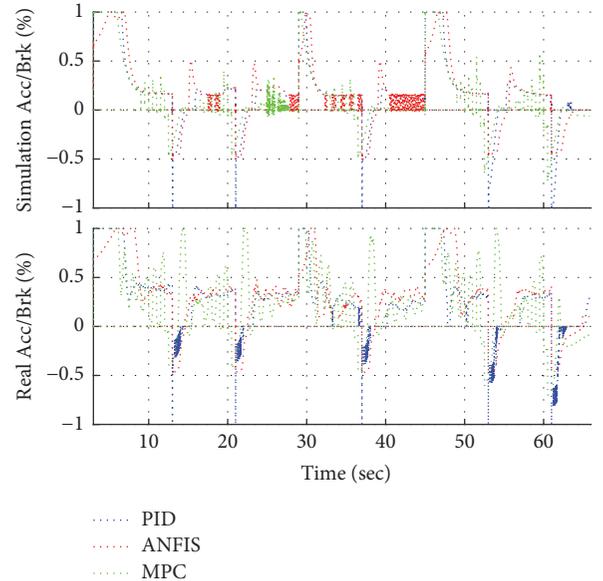


FIGURE 12: Controller output: simulation and real one.

$2 [m/s^2]$ and considering that most of the absolute acceleration errors (close to 75%) are below $0.2 [m/s^2]$ (see Figure 11). This is consistent with the speed analysis presented before.

5. Conclusion and Future Work

This work has presented the development and implementation of three longitudinal controllers for CC at low speed. The algorithms were designed in an in-house simulation framework based on a multibody vehicle model. Then, controllers were applied in an instrumented Renault Twizy. Software implementation was based on a general automated driving architecture developed in Matlab/Simulink.

The major contributions of this work include the following:

- (i) A complete framework for automated driving is presented and implemented in virtual and real vehicles.

TABLE 9: Peaks of acceleration errors [m/s²].

	PID			ANFIS			MPC		
	Sim	Real	Error	Sim	Real	Error	Sim	Real	Error
Step 1	1.20	1.14	0.06	1.05	0.92	0.13	1.17	1.16	0.01
Step 2	0.80	0.92	0.12	1.14	1.22	0.08	0.91	1.22	0.31
Step 3	0.99	1.03	0.04	1.22	1.28	0.06	1.00	1.26	0.26
Step 4	1.13	1.16	0.03	1.06	0.96	0.10	1.14	1.04	0.1
Step 5	0.94	1.17	0.23	1.15	1.21	0.06	0.92	1.38	0.54
Step 6	1.14	1.23	0.09	1.05	1.05	0.00	1.13	1.12	0.01
Step 7	1.40	1.34	0.06	1.25	1.20	0.05	1.25	1.79	0.54
Step 8	1.90	2.00	0.1	1.32	1.27	0.05	1.48	1.98	0.50

- (ii) The framework is validated as a tool for simulation-based design of ADAS functionality. Experiments show that the results obtained in the simulation tool are representative of its behavior in the real vehicle within an approximated error of 10% between platforms. The benefits include reducing the time and cost of validating automated driving algorithms while increasing performance.
- (iii) A use case of longitudinal CC based on three different control techniques is presented. The use case covers the full development process: characterization of the real vehicle, adjusting vehicle parameters in the simulator, design and implementation of desired functionality, testing and tuning in simulation platform, and validation in real vehicle.

These results open new possibilities for developing automated driving algorithms using this methodology, including new scenarios, other use cases, and more complex algorithms. Future works include the validation of the simulation framework for lateral maneuvers, such as overtaking and lane keeping. Design of controllers for cooperative maneuvers will be considered as well.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge the ESCAL Project ENABLE-S3 (with Grant no. 692455-2) for the support in the development of this work.

References

- [1] Graham Cookson, "Europe's Traffic Hotspots Measuring the impact of congestion," INRIX Research, 2016.
- [2] Connectivity and Automated Driving, "Automated Driving Roadmap," ERTRAC Working Group, 2017.
- [3] On-Road Automated Vehicle Standards Committee SAE, "Taxonomy and definitions for terms related to onroad motor vehicle automated driving systems," 2014.
- [4] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 296–305, 2014.
- [5] X. Jin, G. Yin, and J. Wang, "Robust fuzzy control for vehicle lateral dynamic stability via Takagi-Sugeno fuzzy approach," in *Proceedings of the 2017 American Control Conference, ACC 2017*, pp. 5574–5579, Seattle, WA, USA, May 2017.
- [6] Benjamin Englander et al., "Collision Avoidance System for Buses, Managing Pedestrian Detection and Alerts Near Bus Stops," in *Proceedings of the 25th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration*, Detroit, Michigan, USA, 2017.
- [7] Mariano Sans, "Semi-automatic method for assisting with parking a motor vehicle and associated device," US Patent 9,731,715. 2017.
- [8] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.
- [9] M. James Anderson, *Autonomous Vehicle Technology: A Guide for Policymakers*, Rand Corporation, Santa Monica, California, USA, 2014.
- [10] Waymo, "Waymo Safety Report On The Road to Fully Self-Drivin," Safety Report, 2017.
- [11] F. Bounini, D. Gingras, V. Lapointe, and H. Pollart, "Autonomous Vehicle and Real Time Road Lanes Detection and Tracking," in *Proceedings of the 12th IEEE Vehicle Power and Propulsion Conference, VPPC 2015*, Canada, October 2015.
- [12] S. Prialé Olivares, N. Rebernik, A. Eichberger, and E. Stadlober, "Virtual Stochastic Testing of Advanced Driver Assistance Systems," in *Advanced Microsystems for Automotive Applications 2015*, Lecture Notes in Mobility, pp. 25–35, Springer, Cham, Switzerland, 2016.
- [13] S. Khastgir, S. Birrell, G. Dhadyalla, and P. Jennings, "Development of a drive-in driver-in-the-loop fully immersive driving simulator for virtual validation of automotive systems," in *Proceedings of the 81st IEEE Vehicular Technology Conference, VTC Spring 2015*, Glasgow, UK, May 2015.
- [14] V. Milanés, J. Villagrà, J. Pérez, and C. González, "Low-speed longitudinal controllers for mass-produced cars: A comparative study," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 620–628, 2012.
- [15] P. Ioannou and Z. Xu, "Throttle and brake control systems for automatic vehicle following," *Journal of Intelligent Transportation Systems*, vol. 1, no. 4, pp. 345–377, 1994.

- [16] David González and Joshué Pérez, "Control architecture for Cybernetic Transportation Systems in urban environments," in *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium, IEEE IV 2013*, pp. 1119–1124, Australia, June 2013.
- [17] R. Lattarulo, J. Pérez, and M. Dendaluze, "A complete framework for developing and testing automated driving controllers," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 258–263, 2017.
- [18] Joshué Pérez et al., "Trajectory generator for autonomous vehicles in urban environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '13)*, pp. 409–414, Karlsruhe, Germany, May 2013.
- [19] J. Cuadrado, D. Dopico, M. A. Naya, and M. Gonzalez, "Penalty, semi-recursive and hybrid methods for MBS real-time dynamics in the context of structural integrators," *Multibody System Dynamics*, vol. 12, no. 2, pp. 117–132, 2004.
- [20] J. Cuadrado, D. Vilela, I. Iglesias, A. Martín, and A. Peña, "A multibody model to assess the effect of automotive motor in-wheel configuration on vehicle stability and comfort," in *Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics*, pp. 1083–1092, 2013.
- [21] I. Iglesias-Aguinaga, A. Martín-Sandi, and A. Pena-Rodríguez, "Vehicle modelling for real time systems application. The virtual rolling chassis," *Dyna (Spain)*, vol. 88, no. 2, pp. 206–215, 2013.
- [22] J. Pérez, V. Milanés, and E. Onieva, "Cascade architecture for lateral control in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 73–82, 2011.
- [23] Karl Johan Åström and Tore Hägglund, *Advanced PID Control*, ISA-The Instrumentation, Systems and Automation Society, Research Triangle Park, North Carolina, USA, 2006.
- [24] X. Wang, M. Fu, H. Ma, and Y. Yang, "Lateral control of autonomous vehicles based on fuzzy logic," *Control Engineering Practice*, vol. 34, pp. 1–17, 2015.
- [25] J. E. Naranjo, C. González, R. García, T. De Pedro, and M. A. Sotelo, "Using fuzzy logic in automated vehicle control," *IEEE Intelligent Systems*, vol. 22, no. 1, pp. 36–45, 2007.
- [26] T. Takagi and M. Sugeno, "Derivation of Fuzzy Control Rules from Human Operator's Control Actions," *IFAC Proceedings Volumes*, vol. 16, no. 13, pp. 55–60, 1983.
- [27] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [28] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "MPC-based approach to active steering for autonomous vehicle systems," *International Journal of Vehicle Autonomous Systems*, vol. 3, no. 2–4, pp. 265–291, 2005.
- [29] H. JIANG, Z. Wang, Q. Chen, and J. Zhu, "Obstacle avoidance of autonomous vehicles with CQP-based model predictive control," in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016*, pp. 1668–1673, Hungary, October 2016.
- [30] Ray Lattarulo et al., "A Speed Planner Approach Based On Bzier Curves Using Vehicle Dynamic Constrains and Passengers Comfort," in *in the Proceedings of the The International Symposium on Circuits and Systems*, Florence, Italy, 2018.
- [31] Q. Huang and H. Wang, "Fundamental study of jerk: Evaluation of shift quality and ride comfort," *SAE Technical Papers*, 2004.
- [32] B. Houska et al., "ACADO Toolkit Users Manual," <http://www.acadotoolkit.org>. 2009–2013.
- [33] D. Ariens, B. Houska, and H. J. Ferreau, "ACADO for Matlab Users Manual," <http://www.acadotoolkit.org>. 2010–2011.

Research Article

Multi-GPU Development of a Neural Networks Based Reconstructor for Adaptive Optics

Carlos González-Gutiérrez ¹, María Luisa Sánchez-Rodríguez,²
José Luis Calvo-Rolle ³, and Francisco Javier de Cos Juez ¹

¹Department of Exploitation and Exploration of Mines, University of Oviedo, Oviedo, Spain

²Department of Physics, University of Oviedo, Oviedo, Spain

³Department of Industrial Engineering, University of A Coruña, Ferrol, A Coruña, Spain

Correspondence should be addressed to José Luis Calvo-Rolle; jcalvo@udc.es

Received 25 November 2017; Accepted 21 February 2018; Published 28 March 2018

Academic Editor: José Manuel Andújar

Copyright © 2018 Carlos González-Gutiérrez et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aberrations introduced by the atmospheric turbulence in large telescopes are compensated using adaptive optics systems, where the use of deformable mirrors and multiple sensors relies on complex control systems. Recently, the development of larger scales of telescopes as the E-ELT or TMT has created a computational challenge due to the increasing complexity of the new adaptive optics systems. The Complex Atmospheric Reconstructor based on Machine Learning (CARMEN) is an algorithm based on artificial neural networks, designed to compensate the atmospheric turbulence. During recent years, the use of GPUs has been proved to be a great solution to speed up the learning process of neural networks, and different frameworks have been created to ease their development. The implementation of CARMEN in different Multi-GPU frameworks is presented in this paper, along with its development in a language originally developed for GPU, like CUDA. This implementation offers the best response for all the presented cases, although its advantage of using more than one GPU occurs only in large networks.

1. Introduction

Development of large telescopes is one of the biggest challenges in nowadays astronomy and astrophysics. Future construction of the Thirty-Meter Telescope (TMT) [1] and the European Extremely Large Telescope (E-ELT) [2], the two largest telescopes in the world, have originated enormous challenges for engineers and researchers [3]. One of the key elements of these telescopes is the development of the adaptive optics (AO) [4] system that helps to improve the quality of the received image.

There are several tomographic techniques employed in the image reconstruction for AO systems, like Single Conjugate Adaptive Optics (SCAO), Multiconjugate Adaptive Optics (MCAO), or Multiobject Adaptive Optics (MOAO) [5] to be used in the future E-ELT [3]. MOAO uses several reference guide stars to obtain information to reconstruct the atmosphere turbulence profile [6]. To combine this

information, it is necessary to use tomographic reconstruction algorithms. Some of the most popular are based on a matrix vector multiplication, with the control matrix being defined by either least squares (LS) [7, 8] or minimum variance techniques [9]. However, during recent years most complex solutions have been developed, like Learn and Apply (L&A) [10], or the intelligent system known as Complex Atmospheric Reconstructor based on Machine Learning (CARMEN) [11, 12]. Due to the increasing complexity and amount of data used by these algorithms [13], some of previous algorithms have been implemented in Graphics Processing Units (GPUs) [14, 15], speeding up substantially their execution and development [16, 17].

CARMEN is a tomographic reconstructor for MOAO systems created at the University of Oviedo. It was initially developed using nonparametric estimation techniques [18] and Multivariate Adaptive Regression Splines (MARS) [19], but its development using Artificial Neural Networks (ANN)

achieves great results at on-sky testing [20, 21]. ANN and deep learning have become very popular in recent years [22], and several frameworks have been developed to help researchers in their projects [23]. Most of these frameworks provide GPU acceleration, and some of them have shown good results speeding up CARMEN training and execution [24–26]. However, only one GPU has been used in previous tests, while there are already some Multi-GPU implementations both for convolutional neural networks [27, 28] and for L&A [29].

The purpose of this paper is to detail the implementation of CARMEN in different Multi-GPU environments and compare their training and execution times. The implementations include some of the most popular neural network frameworks, with their different Multi-GPU proposals and the development of a code in a native GPU language such as CUDA.

In Section 2, a most detailed explanation of CARMEN an AO will be provided. In Section 3, different approaches to Multi-GPU systems will be given, while Section 4 talks about how the selected frameworks implement this Multi-GPU approach. Section 5 describes the proposed experiment, and Section 6 shows obtained results with its analysis. Finally, conclusions are provided along with future lines in the search for improvements.

2. CARMEN

CARMEN is a tomographic reconstructor based on artificial neural networks. In Section 2.1 details about the adaptive optics system are explained, and Section 2.2 is a small summary about the neural network architecture.

2.1. Adaptive Optics. MOAO systems use several guide stars inside the field of view of the astronomical object as a reference. These stars provide information about the wave-front aberrations produced by the atmosphere. There are several techniques to combine this information in order to reconstruct the incoming wave-front.

To characterize the incoming wave-front received in large telescopes, it is common to use the Shack-Hartman Wave-front Sensor (SH-WFS) [30]. This sensor is composed by several lenses (called lenslets) with the same focal length and are focused on different photon sensors. As it can be observed in Figure 1, the incident wave-front can be split in a matrix of tilts, and the deviation from the focal spot can be calculated. Both the x and y deviations are computed, using a centroiding algorithm. This matrix of centroids is used as input to the system and allows characterizing the aberrations introduced by the atmosphere, combining the information of several reference stars.

CANARY [31, 32] is an (AO) on-sky demonstrator, principally intended for developing and testing AO concepts for the future 39 m E-ELT. It is operated on a Nasmyth platform of the 4.2 m William Herschel Telescope, one of the Isaac Newton Group of Telescopes (ING) of the Observatorio del Roque de los Muchachos (ORM), La Palma, Canary Islands, Spain. CANARY has operated in several configurations to demonstrate different AO techniques on-sky. Different sizes

and improved SH-WFS have been used through the years, which provides a wide range of configurations. Two configurations that have already been tested on-sky between 2013 and 2015 [13] will be discussed in this paper.

- (i) *CANARY Phase B1* is designed to perform observations with one Rayleigh Laser Guide Star (LGS) and up to four Natural Guide Stars (NGSs). It has a SH-WFS with 7×7 subapertures, although only 36 of them are activated due to the circular telescope pupil and secondary obscuration.
- (ii) *CANARY Phase C2* is designed for the study of Laser Tomography AO (LTAO) MOAO. There are four Rayleigh Laser Guide Stars, and the corresponding WFS have 14×14 subapertures (144 active).

Results in DRAGON [33], the successor of CANARY, will also be presented, although it is still under development at Durham University. Since it has not been fully finished and has never been tested on-sky, DRAGON have been simulated in Durham AO Simulation Platform (DASP) [34].

- (i) *DRAGON* provides a single channel MOAO system with a woofer-tweeter DM configuration, four NGSs, and four LGSs each with 30×30 subapertures, where only 704 of them are operative.

2.2. Architecture. CARMEN is a multilayer perceptron, which contains a single hidden layer. This means that it has two fully connected layers, where every neuron is connected to all the neurons in the previous layer. The output of each neuron is computed following (1), where w is the weight of each connection, x is the value of the neurons in the previous layer, b is a constant value called *bias*, and f is an activation function.

$$Y = f \left(\sum_{i=0}^n (w_i \cdot x_i) + b \right). \quad (1)$$

The number and size of the hidden layers have been the purpose of previous studies [35, 36], so the architecture of the neural network will not be analyzed in this paper, and it will follow the patterns previously defined. In the input layer, the number of neurons depends on the numbers of guide stars and the size of SH-WFS. The number of input neurons will be the numbers of subapertures of the SH-WFS, multiplied by 2 (the x and y coordinates of the centroid), also multiplied by the amount of reference stars. Following the architecture of the previous studies, the number of neurons in the hidden layer is equal to the number of input neurons. Finally, the output of the ANN is the expected wave-front slope measurements as seen by the on-axis wave-front sensor, so the number of output neurons will be the number of subapertures of the SH-WFS multiplied by two. The neural network architecture is shown in Figure 2.

3. Multi-GPU Implementation

There are different approaches about how to parallelize an artificial neural network. Although using multiple CPUs has

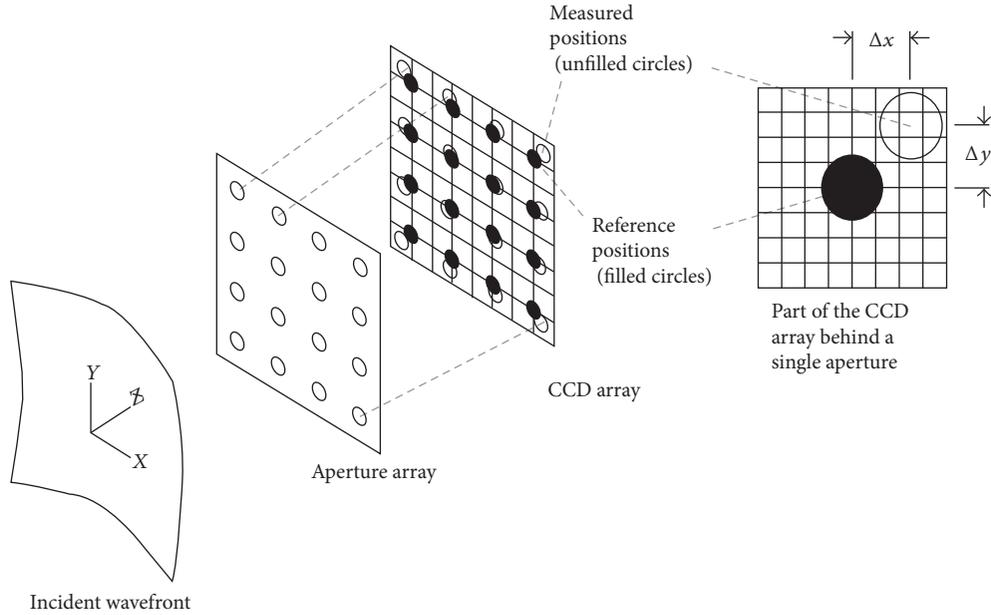


FIGURE 1: Measurement of Wave Front Tilts using a SH-WFS.

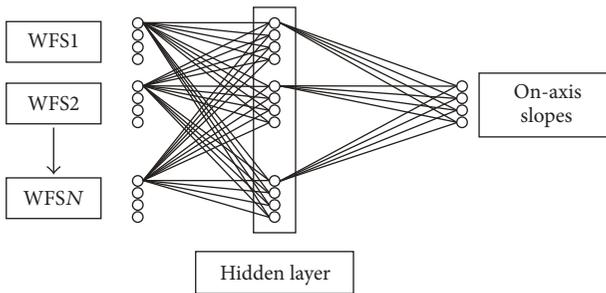


FIGURE 2: CARMEN architecture.

been proven as a good solution [37], in this paper we are going to focus on parallelizing the network by using GPUs, since it has been one of the common trends during the last years [25, 38]. However, splitting the work across different GPUs has been one of the biggest challenges that researchers have had to deal with. Two different solutions have been used to solve this problem, data parallelism and model parallelism, which will be detailed beneath. There are also new proposals, which can combine both data and model parallelism [28] or take advantage from asynchronous synchronization to train the neural networks [39, 40].

3.1. Data Parallelism. The idea behind data parallelism is to split the input data of each iteration in different GPUs. By using the stochastic gradient descent based on minibatches, it is easy to divide each minibatch, into smaller pieces, and use them as a smaller input to the network. Each GPU holds a copy of the complete model of the neural network and take the forward pass with the respective chunk of the minibatch. Once the output is obtained, gradients are calculated in the backward pass for each chunk. The last step is to collect all the

gradients, update the model values, and send the new weights back to each GPU.

In Figure 3 it is possible to observe how data flows through the system and the theoretical speeds of exchanging information between graphics cards. Although there is no need of sending data between GPUs in the forward pass, one of the biggest problems of this kind of parallelism is exchanging the gradients in the backward pass. If the number of parameters of the different layers is too high, sending the information through the PCI-Express could consume too much time. Also, it is necessary to wait until the end of the gradient exchange and the weight matrix update, to start the next iteration of the training process, which could be a huge bottleneck.

3.2. Model Parallelism. In this scenario, the neural network model is split across the GPUs. This can be done in two different ways. One easy approach is to put each layer in a separate graphics card and send the outputs of each layer to the next GPU. However, this approach is strongly limited by the number of layers and their size. The other solution is to split the model “horizontally,” so each GPU has a portion of the weights matrices from every layer.

As it can be observed in Figure 4, the outputs of each layer have to be shared between GPUs during the forward pass. This requires that each GPU needs to wait for the rest of them to be finished, so it is necessary to start all the computations almost at the same time, so all GPUs end their calculations simultaneously. During the backward pass, the error needs to be exchanged between cards also, so each matrix could be updated.

Although this method seems to exchange much more information between GPUs than the Data Parallelism idea, this is strongly dependent on the network architecture. For networks that contain large layers, it could be faster to share

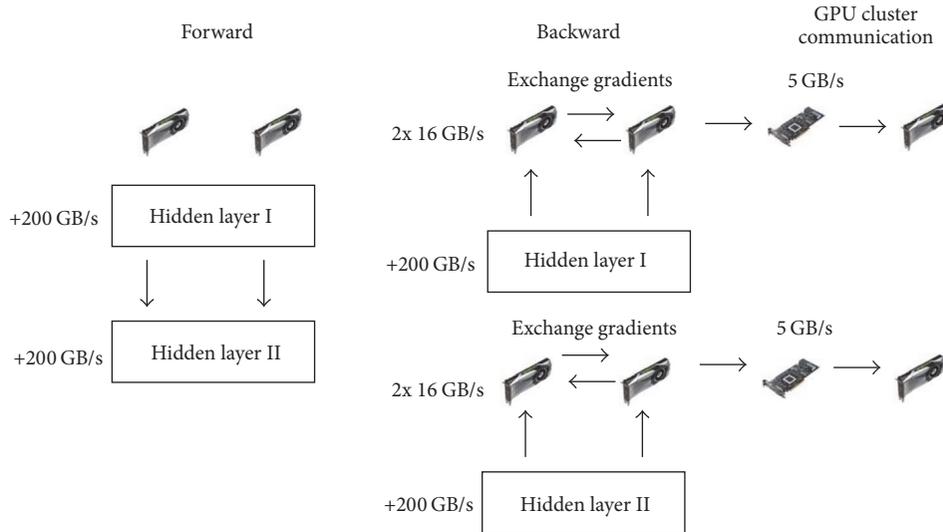


FIGURE 3: Data parallelism reproduced from [41].

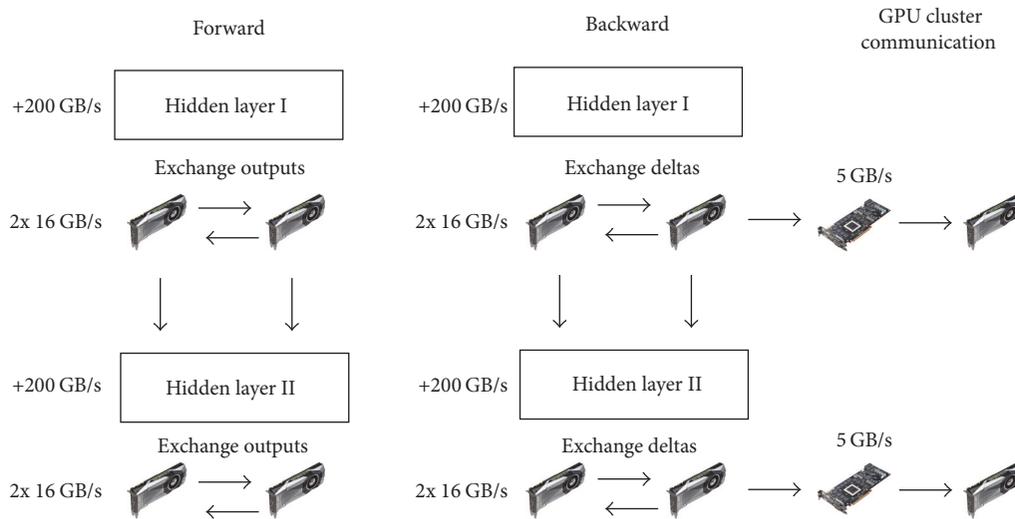


FIGURE 4: Model parallelism reproduced from [41].

only the output and the errors than to send the whole matrix to all the GPUs. Also splitting the matrices across the GPUs allows the use of bigger layers that could not fit in the memory of a single GPU.

4. Neural Network Frameworks Overview

With the recent success of Deep Learning in university and industry, dozens of neural networks frameworks have been developed [23]. For these analysis, only three of the most popular [42–44] have been selected. Although it was used in our previous papers [24, 26], Theano has been left out of the comparative due to the recent announcement of their creators not to continue its development [45].

4.1. Caffe. Caffe is a deep learning framework developed by the Berkeley Vision and Learning Center and is released

under the BSD 2-Clause license [46]. It is mainly written in C/C++ and uses CUDA to speed up the execution on the GPU, with support for Nvidia cuDNN. It provides an interface that supports data parallelism out of the box, just selecting which GPUs of the system are going to be used. However, it does not have an implementation for executing one single input across multiple-GPUs.

4.2. Torch. Torch is a scientific computing framework with wide support for machine learning algorithms. It is written in Lua and C/CUDA to provide fast execution and includes the possibility of importing modules to complete and accelerate the system.

For the multi-GPU implementation, it provides the *DataParallelTable* module, which allows parallelizing the training with the implementation of data parallelism model. By using this module, it is possible to replicate the model across

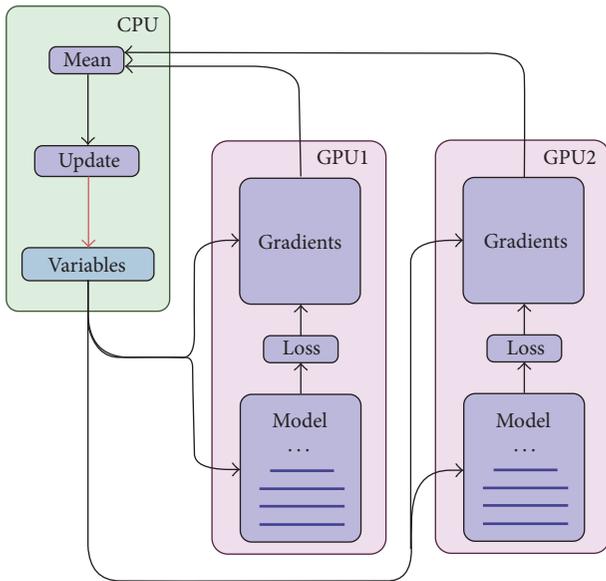


FIGURE 5: Data parallelism in TensorFlow.

different GPUs and split the minibatch in smaller chunks, sending each piece to a separate graphics card. Once the inputs are split, the framework automatically shares gradients between GPUs and updates the model after each iteration. As it happens with Caffe, a Multi-GPU implementation of the execution process is not possible without big changes in the source code.

4.3. TensorFlow. TensorFlow is an open source software library for numerical computation. It was originally developed by the Google Brain Team. It is written in C++ and CUDA and provides a Python API for an easy implementation.

Unlike the rest of the cases, two different implementations will be used. In the case of using only one GPU, all computations will take place in it. However, the multi-GPU implementation is a little more complex. A replica of the model is created on every GPU, as it is suggested in [47]. Then, each one of them computes the forward and backward pass and sends the gradients back to the CPU. The main processor calculates the means of the gradients, updates the model, and sends the information back to each GPU. Figure 5 is a schematic representation of how data parallelism is implemented in TensorFlow.

4.4. CUDA. CUDA is a programming language developed by Nvidia, which allows parallelizing different computations by using GPUs. It provides libraries such as CUBLAS, cuFFT, or cuDNN to facilitate the work of developers. Their use in all the previous presented frameworks and some of the tools and libraries provided, making CUDA a most suitable choice than other GPGPU languages like OpenCL.

In this case, the code has been developed to use model parallelism instead of data parallelism. Each weight matrix is divided among the GPUs, and all graphics cards received a full copy of the minibatch. After computing the outputs of

each layer, by using CUBLAS and cuDNN libraries, the results are shared between GPUs. In order to avoid overloading the PCI, the sharing process is asynchronous, and it starts once one of the GPUs has finished the computation of the information. However, it is not possible to compute the next layer until all the GPUs have all the outputs.

The backward process is very similar. Each GPU computes the gradients of their portion of the matrix. After that, errors are shared and every GPU updates the corresponding weights. Since it is possible to control when the data are copied from RAM, the next minibatch is copied to every GPU while updating the matrices, so it is possible to speed up even more the training process.

5. Experiment Description

In the present paper, two different measurements will be assessed. First of all, training times will be compared, evaluating changes in the time needed by the different frameworks, when the number of GPUs is changed. The second measurement is the performance of the CUDA code for execution. The other three frameworks are left out of this comparison, because data parallelism is not suitable for executing only one input, since it is not possible to split that input among several GPUs.

5.1. Training. The training benchmark is the same method used in [24–26] and will be summarized. In this case, not only learning rate, momentum, and backpropagation algorithm, Stochastic Gradient Descent (SGD) [48], are fixed, but also the minibatch size will always be 256. A comparison will be made by changing the number of GPUs used for training, from 1 to 4. Size of the dataset will increase according to the size of the neural network, although it is not relevant since increasing the size of the dataset will increase training times linearly.

To measure training times, a timer is started once the whole dataset is copied to RAM and all weights matrixes are initialized and copied to VRAM (VRAM: GPU Video RAM, RAM: CPU RAM). Timer is stopped when the whole dataset is executed and backpropagated, which is denominated as an epoch. The operation is repeated 20 times and averaged, to ensure more reliable results, which allows checking that there is no substantial difference between each epoch, less than 1%.

5.2. Execution. In execution, there will be two different measurements for CUDA code. One of them will analyze the ideal scenario, where the input is already copied into RAM and a second where the system must load the input from SSD. This differentiation is needed because it is not possible to ensure how the future integration of the reconstructor with a real telescope will be.

For this benchmark, data is in separated h5 files [49], and the output was written to a separate file. System is fed with 10,000 inputs one at a time and averaged, which allows measuring execution time. All weight matrixes are copied to VRAM before execution and they remain constant during all iterations. In one case, the copy from SSD to RAM and vice versa is considered for the time measurement. However,

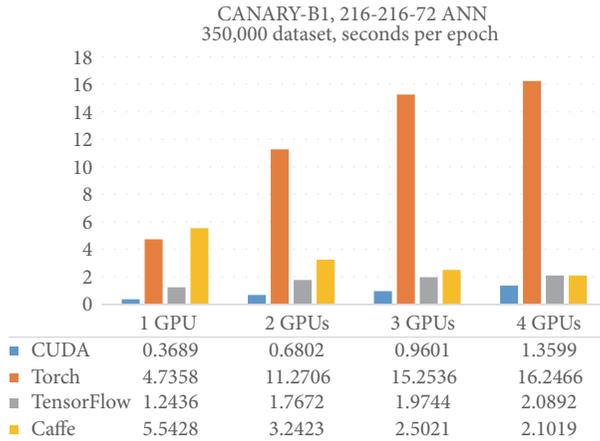


FIGURE 6: Training times per epoch for CANARY-B1.

in the other scenario, the timer is only started when data is copied to RAM.

5.3. Benchmark Equipment. The experiments were performed on a computer running on Ubuntu LTS 14.04.3, with an Intel Xeon CPU E5-1650 v3 @ 3.50 GHz, 128 Gb DDR4 memory, 4 Nvidia GeForce GTX TitanX, with 12 Gb DDR5 VRAM, and 1 Tb SSD hard drive.

6. Results

In this section, results obtained with the different frameworks and varying the number of GPUs will be presented. Also, an analysis of the results will be provided, explaining the behavior of the different frameworks.

6.1. CANARY-B1. CANARY-B1 is the smallest system analyzed. In Figure 6, it is possible to observe some interesting behavior. For instance, Torch is the slowest framework by far, and increasing the number of GPUs only provokes worst training times. Something similar could be monitored for CUDA and TensorFlow. Although both are the faster options with only one GPU, increasing the number of cards harms their performance. In this scenario Caffe, despite having good performance results with several GPUs, has the worst results with a single GPU. This behavior contrasts with the performance of the other frameworks considered. In particular when the number of GPUs increases, it is able to reduce time until matching times with TensorFlow with 4 GPUs.

In the execution with CUDA, similar results can be observed in Figure 7. Increasing the number of GPUs not only does not provide any benefit but also increases execution times substantially. For RAM execution has almost a linear relation, where doubling or tripling the number of GPUs has the same impact in times. Also, as it was analyzed in previous works [24, 25], loading data from SSD instead of directly from RAM has an important impact on performance, although in this case it could be observed how this difference is almost fixed (about 0.3 milliseconds) for every number of GPUs.

6.2. CANARY-C2. In CANARY-C2 results from Figure 8 are quite similar to those of the previous system. Torch is the

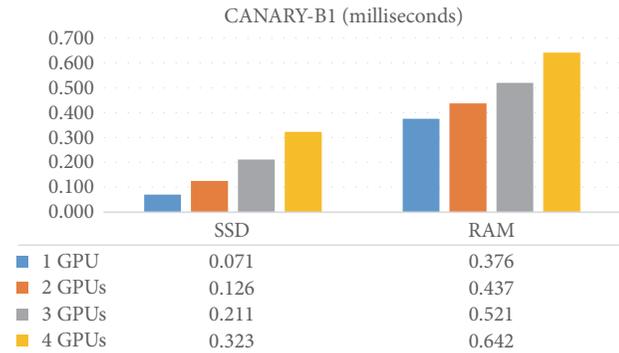


FIGURE 7: Execution times for CANARY-B1.

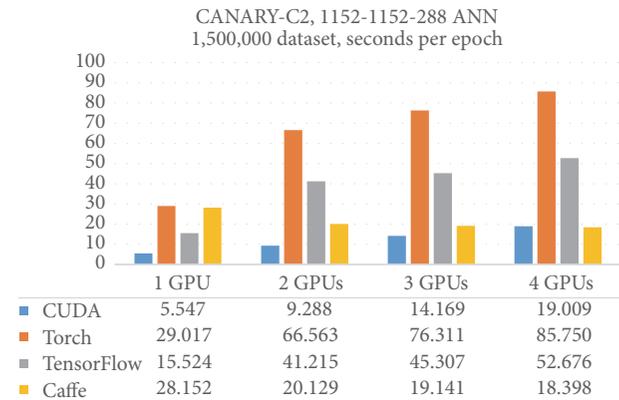


FIGURE 8: Training times per epoch for CANARY-C2.

slowest framework again and increasing the number of GPUs is very harmful for training times. In this case, TensorFlow has a similar behavior, having a huge increase in their times when escalating the number of GPUs. Again, CUDA is the fastest solution but is not able to take advantage of the increasing number of GPUs to speed up training. At last, Caffe has similar training times than torch for single GPU. However, with 4 GPUs it is able not only to decrease their training times, but also to improve the results obtained by CUDA.

As it happens in training times, CANARY-C2 shows almost the same results (Figure 9) as CANARY-B1. It is interesting to notice that even having a neural network with about 28x more weights, the increment in execution times is low, especially in the case of 4 GPUs.

6.3. DRAGON. DRAGON is the biggest network employed in this paper and has about 100x more connections than CANARY-C2. In this scenario there is a shift in some trends observed in previous networks. In Figure 10, TensorFlow is the slowest framework for every number of GPUs. Torch is the second worst option, although training times do not increase as much as in the other cases. For a network of this size, Caffe has better results than the other two frameworks, but, unlike the previous cases, it is not able to reduce training times when increasing the number of GPUs. At last, CUDA is again the fastest solution. However, in this case when the number of GPUs is increased, training times are reduced.

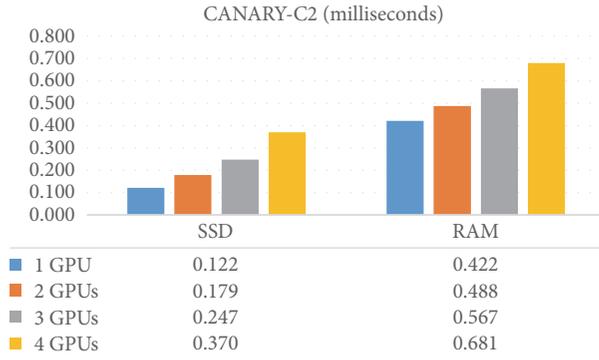


FIGURE 9: Execution times for CANARY-C2.

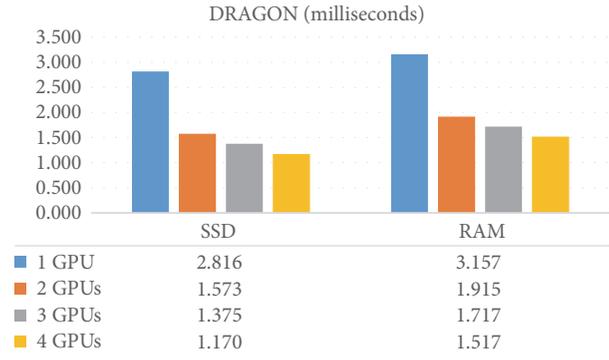


FIGURE 11: Execution times for DRAGON.

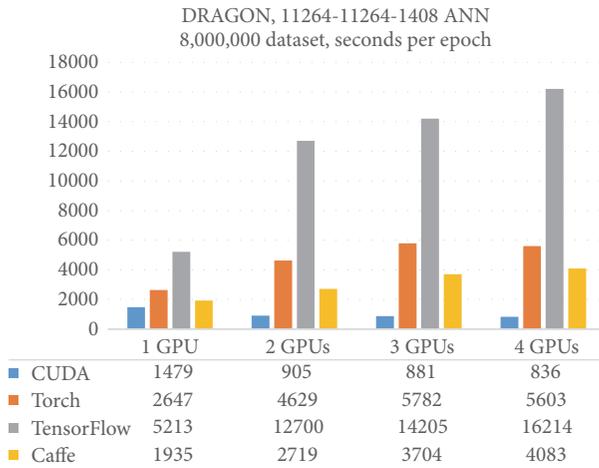


FIGURE 10: Training times per epoch for DRAGON.

A similar situation can be observed for the execution times. As it can be observed in Figure 11, increasing the number of GPUs can decrease execution times. The difference between using one and two GPUs is especially significant, although even in the case of four, execution times keep improving.

6.4. Discussion. There are lots of ideas that could be extracted from the previous results. One of them is that data parallelism implementations of Torch and TensorFlow are not suitable solutions for the present problem. Keeping this in mind, it is interesting to notice how the “full-GPU” backpropagation in Torch is quite slow for smaller systems but improves when the neural network size increases. However, for the hybrid CPU-GPU solution (Figure 5) used in TensorFlow, better results are obtained when the size is smaller. This behavior could be explained by how backpropagation in a small network cannot take advantage of the high level of parallelism provided by a GPU and consequently is better performed in a CPU. On the opposite side, when the neural network grows, not only this parallelism is better used, but also the time consumed by sending all the weight matrixes from RAM to VRAM is a high time-consuming operation.

Regarding Caffe, some differences are observed with respect to the other solutions. For small networks it is able to

reduce training times when increasing the number of GPUs. However, this is not enough to improve results provided by the CUDA code. Since the multi-GPU mode is a built-in solution, it is not possible to explain why these results take place.

At last, the model parallelism solution implemented in CUDA provides the best results for all cases. However, the inability to improve training and execution times for smaller networks, when more GPUs are available, should be analyzed. For CANARY systems, the workload is too low even for one GPU, and increasing the number only provokes time overload because of the movement of data between cards. However, when the size of the network increases, and the GPUs are fully loaded, both training and execution times start to reduce. Although DRAGON is currently the largest system available, it will keep growing in the next years [13], and bigger systems will take much more advantages of this multi-GPU implementation.

In the case of execution, an interesting behavior can be observed. Including data transfer between SSD and RAM in the experiment triggers a huge overload for all cases. There is a 0.3 milliseconds difference for every case, independently of the size of the data transfer. However, it is expected that, in a real telescope, it will not be necessary to read and copy data to the SSD, because of a better integration with the system. Also, as it happens with training, increasing the size of the network takes advantages of the number of GPUs. As it was mentioned before, the smaller networks do not provide enough workload to fill a GPU, but due to the increasing size of the sensors, it will not be a problem in the near future.

7. Conclusions and Future Lines

In this paper, some of the most popular neural network frameworks [42] and a code directly developed in CUDA have been analyzed. It is possible to extract from the exposed results that both TensorFlow and Torch are not suitable for Multi-GPU execution when the neural network used is a multilayer perceptron. In the case of Caffe, it has shown that increasing the number of GPUs in smaller networks offers some benefits, although it is not able to keep those benefits when the size increases. For the three SH configurations presented, the best solution is the code directly developed in CUDA, which has also the potential of taking better

advantages of the number of GPUs, when the size of the network increases.

However, developing these solutions in different frameworks can provide some advantages in the future. Convolutional neural networks (CNN) are very popular for image processing [50, 51], and their use could be an interesting upgrade for CARMEN architecture. Previous studies show that model parallelism could be a better solution for CNN than data parallelism [27, 28], so any of the frameworks could be faster for CNN than the CUDA solution. Also, Recurrent Neural Networks (RNN) [52, 53] should be easier to implement by using a neural network framework, since they are much more prepared for that use [42].

One last idea is the use of online learning, which means that CARMEN could be constantly training when the reconstructor is executed on-sky. Recent works show that this technique could provide an interesting boost in the performance of the reconstructor [54, 55]. By using more than one GPU, it will be possible to adapt the neural network to work faster and be much more precise correcting what is happening on sky.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors appreciate support from the Spanish Economics and Competitiveness Ministry, through Grant AYA2014-57648-P, and the Government of the Principality of Asturias (Consejería de Economía y Empleo), through Grant FC-15-GRUPIN14-017.

References

- [1] J. Nelson and G. H. Sanders, "The status of the thirty meter telescope project," in *Proceedings of the Ground-based and Airborne Telescopes II*, France, June 2008.
- [2] R. Gilmozzi and J. Spyromilio, "The European extremely large telescope (E-ELT)," *The Messenger*, vol. 127, no. 3, 2007.
- [3] S. K. Ramsay, M. M. Casali, J. C. González, and N. Hubin, "The E-ELT instrument roadmap: a status report," in *Ground-based and Airborne Instrumentation for Astronomy V, 91471Z*, Proceedings of SPIE, 2014.
- [4] J. M. Beckers, "Adaptive Optics for Astronomy: Principles, Performance, and Applications," *Annual Review of Astronomy and Astrophysics*, vol. 31, no. 1, pp. 13–62, 1993.
- [5] D. R. Andersen, K. J. Jackson, C. Blain et al., "Performance modeling for the RAVEN Multi-Object Adaptive Optics demonstrator," *Publications of the Astronomical Society of the Pacific*, vol. 124, no. 915, pp. 469–484, 2012.
- [6] O. Lardière, D. Andersen, C. Blain et al., *Multi-object adaptive optics on-sky results with Raven, 91481G*, E. Marchetti, L. M. Close, and J.-P. Véran, Eds., International Society for Optics and Photonics, 2014.
- [7] B. L. Ellerbroek, "First-order performance evaluation of adaptive-optics systems for atmospheric-turbulence compensation in extended-field-of-view astronomical telescopes," *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, vol. 11, no. 2, pp. 783–805, 1994.
- [8] T. Fusco, J.-M. Conan, G. Rousset, L. M. Mugnier, and V. Michau, "Optimal wave-front reconstruction strategies for multiconjugate adaptive optics," *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, vol. 18, no. 10, pp. 2527–2538, 2001.
- [9] M. C. Roggemann, "Optical performance of fully and partially compensated adaptive optics systems using least-squares and minimum variance phase reconstructors," *Computers and Electrical Engineering*, vol. 18, no. 6, pp. 451–466, 1992.
- [10] F. Vidal, E. Gendron, and G. Rousset, "Tomography approach for multi-object adaptive optics," *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, vol. 27, no. 11, pp. A253–A264, 2010.
- [11] F. J. de Cos Juez, F. S. Lasheras, N. Roqueñí, and J. Osborn, "An ANN-based smart tomographic reconstructor in a dynamic environment," *Sensors*, vol. 12, no. 7, pp. 8895–8911, 2012.
- [12] J. Osborn, F. J. De Cos Juez, D. Guzman et al., "Using artificial neural networks for open-loop tomography," *Optics Express*, vol. 20, no. 3, pp. 2420–2434, 2012.
- [13] A. G. Basden, D. Atkinson, N. A. Bharmal et al., "Experience with wavefront sensor and deformable mirror interfaces for wide-field adaptive optics systems," *Monthly Notices of the Royal Astronomical Society*, vol. 459, no. 2, pp. 1350–1359, 2016.
- [14] D. Romero-Laorden, J. Villazon-Terrazas, O. Martínez-Graullera, A. Ibanez, M. Parrilla, and M. S. Penas, "Analysis of Parallel Computing Strategies to Accelerate Ultrasound Imaging Processes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 12, pp. 3429–3440, 2016.
- [15] D. Romero-Laorden, J. Villazón-Terrazas, M. Santos Peñas, M. A. García-Izquierdo, and O. Martínez-Graullera, "Analysis of a software implementation of an ultrasonic signal beamformer in real-time," *RIAI - Revista Iberoamericana de Automática e Informática Industrial*, vol. 13, no. 4, pp. 462–471, 2016.
- [16] H. Ltaief and D. Gratadour, "Shooting for the Stars with GPUs," 2016, <http://on-demand.gputechconf.com/gtc/2015/video/S5122.html>.
- [17] J. Marichal-Hernández, L. F. Rodríguez-Ramos, F. Rosa, and J. M. Rodríguez-Ramos, "Atmospheric wavefront phase recovery by use of specialized hardware: Graphical processing units and field-programmable gate arrays," *Applied Optics*, vol. 44, no. 35, pp. 7587–7594, 2005.
- [18] D. Guzmán, F. J. De Cos Juez, R. Myers, A. Guesalaga, and F. S. Lasheras, "Modeling a MEMS deformable mirror using non-parametric estimation techniques," *Optics Express*, vol. 18, no. 20, pp. 21356–21369, 2010.
- [19] D. Guzmán, F. J. C. De Juez, F. S. Lasheras, R. Myers, and L. Young, "Deformable mirror model for open-loop adaptive optics using multivariate adaptive regression splines," *Optics Express*, vol. 18, no. 7, pp. 6492–6505, 2010.
- [20] J. Osborn, D. Guzman, F. J. de Cos Juez et al., "First on-sky results of a neural network based tomographic reconstructor: Carmen on Canary," in *SPIE Astronomical Telescopes + Instrumentation, 91484M*, E. Marchetti, L. M. Close, and J.-P. Véran, Eds., International Society for Optics and Photonics, 2014.
- [21] J. Osborn, D. Guzman, F. J. D. C. Juez et al., "Open-loop tomography with artificial neural networks on CANARY: on-sky results," *Monthly Notices of the Royal Astronomical Society*, vol. 441, no. 3, pp. 2508–2514, 2014.
- [22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [23] C. Gulcehre, "Deep Learning - Software Links," 2017, <http://deeplearning.net/software.links/>.

- [24] C. González-Gutiérrez, J. D. Santos, M. Martínez-Zarzuela et al., “Comparative study of neural network frameworks for the next generation of adaptive optics systems,” *Sensors*, vol. 17, no. 6, article no. 1263, 2017.
- [25] C. González-Gutiérrez, J. D. Santos-Rodríguez, R. Á. F. Díaz, J. L. C. Rolle, N. R. Gutiérrez, and F. J. de Cos Juez, “Using GPUS to speed up a tomographic reconstructor based on machine learning,” *Advances in Intelligent Systems and Computing*, vol. 527, pp. 279–289, 2017.
- [26] S. L. Suárez-Gómez, C. González-Gutiérrez, J. D. Santos-Rodríguez et al., “Analysing the performance of a tomographic reconstructor with different neural networks frameworks,” in *Proceedings of the 16th International Conference on Intelligent Systems Design and Applications*, A. M. Madureira, A. Abraham, D. Gamboa, and P. Novais, Eds., pp. 1051–1060, Springer International Publishing, Cham, Switzerland, 2017.
- [27] O. Yadan, K. Adams, Y. Taigman, and M. Ranzato, “Multi-GPU Training of ConvNets,” <https://arxiv.org/abs/1312.5853>.
- [28] A. Krizhevsky, “One weird trick for parallelizing convolutional neural networks,” <https://arxiv.org/abs/1404.5997>.
- [29] H. Ltaief, D. Gratadour, A. Charara, and E. Gendron, “Adaptive optics simulation for the world’s largest telescope on multicore architectures with multiple GPUs,” in *Proceedings of the 3rd Conference on Platform for Advanced Scientific Computing, PASC 2016*, Switzerland, June 2016.
- [30] B. C. Platt and R. Shack, “History and principles of Shack-Hartmann wavefront sensing,” *Journal of Refractive Surgery*, vol. 17, no. 5, pp. S573–S577, 2001.
- [31] R. M. Myers, Z. Hubert, T. J. Morris, E. Gendron, and N. A. Dipper, “CANARY: the On-Sky NGS/LGS MOAO Demonstrator for EAGLE,” in *Adaptive Optics Systems, 70150E*, Proceedings of SPIE, pp. 1–9, 2010.
- [32] E. Gendron, F. Vidal, M. Brangier et al., “MOAO first on-sky demonstration with CANARY,” *Astronomy & Astrophysics*, vol. 529, article no. L2, 2011.
- [33] A. P. Reeves, R. M. Myers, T. J. Morris et al., “The Durham real-time, tomographic adaptive optics test bench: progress and results,” in *SPIE Astronomical Telescopes + Instrumentation, 91485U*, E. Marchetti, L. M. Close, and J.-P. Véran, Eds., International Society for Optics and Photonics, 2014.
- [34] A. G. Basden and A. O. Durham, “Simulation Platform,” 2017, <https://gitlab.com/agb32/dasp>.
- [35] J. Osborn, F. J. De Cos Juez, D. Guzman et al., “Open-loop tomography using artificial neural networks,” in *Proceedings of the 2nd International Conference on Adaptive Optics for Extremely Large Telescopes, AO for ELT 2011*, September 2011.
- [36] M. G. Victoria, *Research of The Tomographic Reconstruction Problem by Means of Data Mining And Artificial Intelligence Technologies*, Universidad de Oviedo, 2014.
- [37] J. Dean, G. S. Corrado, R. Monga et al., Large Scale Distributed Deep Networks.
- [38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 1097–1105, Curran Associates, Inc., 2012.
- [39] S. Zhang, C. Zhang, Z. You, R. Zheng, and B. Xu, “Asynchronous stochastic gradient descent for DNN training,” in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '13)*, pp. 6660–6663, May 2013.
- [40] T. Paine, H. Jin, J. Yang, Z. Lin, and T. Huang, “GPU Asynchronous Stochastic Gradient Descent to Speed Up Neural Network Training,” <https://arxiv.org/abs/1312.6186>.
- [41] T. Dettmers, “Making deep learning accessible,” 2017, <http://timdettmers.com/>.
- [42] A. Parvat, J. Chavan, S. Kadam, S. Dev, and V. Pathak, “A survey of deep-learning frameworks,” in *Proceedings of the 2017 International Conference on Inventive Systems and Control (ICISC)*, pp. 1–7, January 2017.
- [43] S. Shi, Q. Wang, P. Xu, and X. Chu, “Benchmarking state-of-the-art deep learning software tools,” in *Proceedings of the 7th International Conference on Cloud Computing and Big Data, CCBD 2016*, pp. 99–104, China, November 2016.
- [44] S. Bahrampour, N. Ramakrishnan, L. Schott, and M. Shah, “Comparative study of deep learning software frameworks,” <https://arxiv.org/abs/1511.06435>.
- [45] P. Lamblin, “Theano 0.9.0 documentation,” 2017, <http://deeplearning.net/software/theano/>.
- [46] Y. Jia, E. Shelhamer, J. Donahue et al., “Caffe: convolutional architecture for fast feature embedding,” in *Proceedings of the ACM International Conference on Multimedia*, pp. 675–678, ACM, Orlando, Fla, USA, November 2014.
- [47] “Google Brain Team Convolutional Neural Networks - Training a Model using Multiple GPU Cards — TensorFlow,” 2017, https://www.tensorflow.org/tutorials/deep.cnn#training_a_model_using_multiple_gpu_cards.
- [48] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [49] “The HDF Group Introduction to HDF5,” 2016, <https://www.hdfgroup.org/HDF5/doc/H5.intro.html#Intro-WhatIs>.
- [50] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, “Integrated recognition, localization and detection using convolutional networks,” in *Proceedings of the International Conference on Learning Representations (ICLR2014)*, 2014.
- [51] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 1–9, Boston, MA, USA, June 2015.
- [52] A. Graves, “Generating sequences with recurrent neural networks,” Tech. Rep., 2013.
- [53] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '13)*, pp. 6645–6649, May 2013.
- [54] S. L. Suárez-Gómez, M. L. Sánchez, F. Blanco, J. Ayala, and F. J. de Cos Juez, “Successful sulfur recovery in low sulfate compounds obtained from the zinc industry: Evaporation–condensation method,” *Journal of Hazardous Materials*, vol. 336, pp. 168–173, 2017.
- [55] S. L. S. Gómez, J. D. S. Rodríguez, F. J. I. Rodríguez, and F. J. D. C. Juez, “Analysis of the temporal structure evolution of physical systems with the self-organising tree algorithm (SOTA): Application for validating neural network systems on adaptive optics data before on-sky implementation,” *Entropy*, vol. 19, no. 3, article no. 103, 2017.

Research Article

Optimal Modeling of Wireless LANs: A Decision-Making Multiobjective Approach

Tomás de Jesús Mateo Sanguino  and Jhon Carlos Mendoza Betancourt

Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática, University of Huelva, Ctra. Huelva-La Rábida S/N, Palos de la Frontera, 21819 Huelva, Spain

Correspondence should be addressed to Tomás de Jesús Mateo Sanguino; tomas.mateo@diesia.uhu.es

Received 24 November 2017; Accepted 19 February 2018; Published 26 March 2018

Academic Editor: Eloy Irigoyen

Copyright © 2018 Tomás de Jesús Mateo Sanguino and Jhon Carlos Mendoza Betancourt. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Communication infrastructure planning is a critical design task that typically requires handling complex concepts on networking aimed at optimizing performance and resources, thus demanding high analytical and problem-solving skills to engineers. To reduce this gap, this paper describes an optimization algorithm—based on evolutionary strategy—created as an aid for decision-making prior to the real deployment of wireless LANs. The developed algorithm allows automating the design process, traditionally handmade by network technicians, in order to save time and cost by improving the WLAN arrangement. To this end, we implemented a multiobjective genetic algorithm (MOGA) with the purpose of meeting two simultaneous design objectives, namely, to minimize the number of APs while maximizing the coverage signal over a whole planning area. Such approach provides efficient and scalable solutions closer to the best network design, so that we integrated the developed algorithm into an engineering tool with the goal of modelling the behavior of WLANs in ICT infrastructures. Called WiFiSim, it allows the investigation of various complex issues concerning the design of IEEE 802.11-based WLANs, thereby facilitating design of the study and design and optimal deployment of wireless LANs through complete modelling software. As a result, we comparatively evaluated three target applications considering small, medium, and large scenarios with a previous approach developed, a monoobjective genetic algorithm.

1. Introduction

Implementing Wi-Fi networks with the best use of resources while offering the best service to users requires a careful planning. WLANs can range from relatively simple installations to very complex and intricate designs so a well-documented plan must be outlined before a wireless infrastructure can be deployed [1]. In practice, network technicians must survey the space towards deciding on the WLAN arrangement, whose process involves gauging the location of APs, clients, and obstacles. This is traditionally reduced to achieve the maximum Wi-Fi coverage as decision criterion, which is translated into varying the AP placements based on signal strength measurements. Nevertheless, it is not merely drawing coverage circles on a plane and the difficulty increases when requiring a realistic view of the whole problem [2]. This indeed includes several factors such as the area morphology

(i.e., verticality and horizontality of planes), number of clients populating the WLAN (i.e., distribution, density), type of IEEE 802.11 technology (i.e., modulation scheme and frame management), effective isotropic radiated power (EIRP), and physical distance and obstacle materials (i.e., water, glass, plastic, metal, wood, and concrete). The traditional deployment is consequently no effective for practitioners due to requirements in time and cost, which may drastically reduce the WLAN performance and usability due to operability problems concerning design (e.g., interferences and frame collisions due to home devices, hidden node problem) [3].

To help reduce this gap, this paper presents WiFiSim, an engineering modelling software being developed as part of an ongoing research project for the study and design of WLANs [4–6]. The purpose of this software is to model the behavior and performance of communication networks based on the IEEE 802.11 standard. The interest lies in the realism of the

WiFiSim simulations, which provide a high level of interactivity and visual information with easy-to-interpret results through a configurable and intuitive GUI. In this paper we propose a further development beyond the study and design of WLANs: the problem of optimally deploying APs to cover Wi-Fi clients within a complex space. So this manuscript intended three significant contributions: (1) to present a new intelligent modelling approach for WLAN deployment based on a multiobjective genetic algorithm; (2) to examine its applicability into small, medium, and large scenarios; and (3) to integrate the algorithm with the previous developments and provide a complete decision-making software for ICT technicians. To achieve this goal, we propose an evolutionary genetic strategy based on nondominated sorting genetic algorithm II (NSGA-II) where the optimization approach is applied into a demanding WLAN infrastructure.

To this end, the paper is organized as follows. Section 2 introduces the tools and approaches for WLAN deployment existing in literature. Section 3 is devoted to describing the evolutionary genetic strategy followed to develop the optimization algorithm. Section 4 presents the experimentation conducted in three representative case studies. Finally, the paper provides the results and future works.

2. State of the ART

Typically, the location of APs in Wi-Fi networks is manually estimated from their power transmission. From this parameter, APs are placed at regular intervals throughout the space with a predefined distance between adjacent APs. However, this approach idealizes the signal coverage that, in reality, often finds complex environments with a wide variety of obstacles and materials. This implies designs that may result in a deficient configuration of APs due to poor or excessive coverages. When applied to large scenarios, this approach may bring a significant cost overrun. To address these shortcomings, site surveys must be combined with software modelling tools that help improve and simplify the human process [7]. However, these tools need to have a high degree of realism and simulation capabilities to be really helpful (e.g., modeling of physical spaces and network behavior in Layer 1 and Layer 2 of the OSI model). For this reason, choosing a modelling software to study and design optimal WLANs can become hard due to the large number of existing tools [8].

2.1. Modelling Software for WLAN Deployment. At present, there is a large number of applications for designing and/or planning Wi-Fi networks. Examples of research in scientific literature using these applications include NetStumbler®, a survey-site tool that facilitates the detection of WLANs using the IEEE 802.11 protocol [9]; Wi-Fi Analytics Tool™, a software that provides advanced signal strength graphs and analyzes Wi-Fi channels to optimize the Wi-Fi network setup [10]; Wi-Spy DBx, a RF spectrum analyzer designed for troubleshooting Wi-Fi issues with nearby interfering devices in the 2.4 and 5 GHz bands [11]; Ekahau HeatMapper™, an auditory tool and Wi-Fi site-survey software for home use [12]; NetSpot, a site-survey and analyzing tool that helps improve the Wi-Fi signal strength and to boost the network

speed [13]; Acrylic® WiFi Heatmaps, a site-survey and audit software aimed at generating awesome Wi-Fi heatmaps and editable reports on the RF spectrum coverage [14]; Wolf WiFi Pro®, a Wi-Fi device management software and pre-deployment toolkit for wireless professionals able to detect failure scenarios [15]; Bat-Planner, a basic planning suite for IEEE 802.11-based wireless networks [16]; TamoGraph®, a site-survey tool for collecting, visualizing, analyzing, and reporting Wi-Fi data [17]; VisualRF Plan, a wireless management suite that helps to model the RF environment and the underlying wired topology in a visual way [18]; RF3D Wifiplanner2, a modelling tool for planning and upgrading WLANs based on the study of RF signals [19]; AirMagnet Survey®, an accurate and flexible solution for planning indoor and outdoor WLANs [20]; WiTuners™, a Wi-Fi tool for site survey, automated deployment, and auditing [21]; and Ekahau Site Survey™, a professional software toolkit for site survey, spectrum analysis, and Wi-Fi planning [22], to name a few tools.

In general, all the above tools exemplify design and inspection software for wardriving, site survey, data collecting, and planning. Among the drawbacks, these applications (i) are mostly commercial software with a pay-per-use license agreement, (ii) are often natively available only for Windows platforms, (iii) do not always include structures or materials for obstacle modelling, (iv) only allow multifloor design through 2D maps, (v) do not integrate habitually algorithms for optimal WLAN deployment, and (vi) support their designs in a two-step process (i.e., definition of site requirements and planning based on RF propagation). Among them, AirMagnet Survey, WiTuners, and Ekahau Site Survey are remarkable exceptions. On the one hand, they automatically plan the AP positions and quantity needed to ensure a minimum coverage. On the other hand, they validate their designs with data collecting by auditing the real environment.

As a main disadvantage, these tools use proprietary algorithms for WLAN deployment based on RF planning (i.e., patent protected). This implies charging a licensing fee that can be fairly expensive (especially in comparison to open-source software), to make the owner too dependent on the developer (possibly less adaptable to the constantly changing needs of users) and to make the mechanics of the algorithms opaque to viewing (i.e., there is no testable information) [23]. The importance of utilizing evolutionary approaches, instead of providing a single solution as in the case of the RF-based algorithms, is to procure a set of optimal solutions to facilitate broader decision-making for users. Moreover, the above tools lack a modelling block to assist engineers in the study of wireless traffic in Layer 2. Both capabilities combined are critical when troubleshooting wireless communication issues to deploy efficient and useful WLANs. This way, the design process is improved by a cycle of three steps, much more creative, based on the generation of alternative solutions and analysis and selection of the one considered more fitting.

To introduce the developed modelling software, the main features and properties of WiFiSim are compared with the above-mentioned tools in Table 1. In sum, the pros of WiFiSim are (i) to facilitate for engineers the WLAN deployment by providing a set of optimal solutions closer to the best

TABLE 1: Comparison of features and capabilities of various applications for the study and deployment of wireless networks.

Software name	IEEE 802.11 standard	Study of Layer 1	Study of Layer 2	Obstacle modelling	Type of coverage	Main application	Automatic planning	Native platform	Size (MB)	Cost (€)
NetStumbler	a/b/g	Signal strength	No	No	2D view	Wardriving	No	Windows	1.30	Free
Wi-Fi Analytics Tool	a/b/g/n/ac	Signal strength	No	No	2D view	Wardriving	No	Windows & Android	1.8 & 0.5	Free
Wi-Spy DBx	a/b/g/n	RF spectrum	No	No	2D view	Wardriving* & RF audit	No	Windows	7.60	450–1000
Ekahau HeatMapper	a/b/g/n	Signal strength	No	No	2D map	Site survey	No	Windows	100.0	Shareware [†]
NetSpot	a/b/g/n/ac	Signal strength	No	No	2D map	Site survey	No	Windows & Mac OS X	17.0	50–500
Acrylic WiFi Heatmaps	a/b/g/n/ac	Signal strength	No	No	2D map	Site survey	No	Windows	15.9	879–1119 [●]
Wolf WiFi Pro	a/b/g/n	Signal strength	No	No	2D map	Site survey & AP management	No	Android	0.72	0–50
Bat-Planner	a/b/g/n	Signal strength	No	✓	3D view	Site survey & planning	No	Windows	N/A	Free*
TamoGraph	a/b/g/n/ac	Signal strength	No	✓	2D map	Site survey, data collecting* & planning	No	Windows	81.0	750–1000
VisualRF Plan	N/A	Signal strength	No	✓	2D map	Predeployment	No	Windows	206.9	Free
RF3D Wifiplanner2	a/b/g/n	Signal strength	No	✓	2D map	Planning	No	Windows	15.9	Shareware [♦]
AirMagnet Survey	a/b/g/n/ac	RF data & signals	No	✓	2D map & 3D view	Site survey, data collecting* & planning	RF-based algorithm	Windows	477.0	1800
WiTuners	a/b/g/n/ac	Signal strength	No	✓	2D map	Site survey, data collecting* & planning	RF-based algorithm	Windows & Android	120.1 & 7.20	760–2400 & 0–500
Ekahau Site Survey	a/b/g/n/ac	Signal strength & RF spectrum	No	✓	2D map	Site survey, data collecting* & planning	RF-based algorithm	Windows & Mac OS X	338.0	3395
WiFiSim	a/b/g/n	Signal strength	CSMA/CA RT S/CTS	✓	2D & 3D maps	Modelling & planning	SSGA & NSGA-II	Multiplatform	3.18	Free

[†]Free version with limited features; [♦]15-day trial version; * special hardware required; [●]1-year license. N/A = not available.

network design and (ii) to study WLANs in both Layer 1 and Layer 2 to avoid several communication issues concerning the PHY and MAC layers (e.g., throughput, channel utilization, frame collision, frame delay, queue length and delay, medium access delay, and jitter and hidden node problem). As far as we know, these are the main differentiating capabilities regarding other tools, whose modelling process allows practitioners to improve complex network designs. As a result, WiFiSim has been utilized by professionals on ICT and students and teachers at the University of Huelva to improve the learning and teaching of computer networking degrees over the past seven years (i.e., 10 professionals, 110 students, and 4 teachers since the 2011/12 academic year).

2.2. Approaches for WLAN Optimization. Early approaches based on computer models consisted in exhaustive searches of AP positions under strong restrictions, thus preventing being useful in more general situations [24]. Traditional forms of infrastructure planning, as used for mobile cellular networks, produce acceptable results but are generally considered too costly for wireless networks [25]. In this field, optimization techniques range from Hill climbing, random walk (RW), simulated annealing (SA), and Tabu search (TS) to genetic algorithms (GA). Hill climbing, a mathematical optimization technique within local search methods, was also applied along to metaheuristic neighborhood search algorithms (e.g., SA or TS) to prevent being trapped in local optimums. However, mathematical programming techniques are usually less preferred than metaheuristic approaches due to the difficulty for adapting math expressions to generic case studies. Moreover, TS and GA proved to be techniques with better performance than other local search approaches such as RW or SA. However, TS suffer from scalability and evolutionary genetic strategies depend largely on tuning parameters for the application [26]. Both issues—scalability and parameter tuning—were effectively addressed through an agent-based optimization approach [27] and a combination of pruning and neighborhood search algorithms [28], respectively.

Initial investigations on GAs led to deploying a single AP through a simple implementation [29]. Subsequent works were focused on modeling WLAN scenarios by increasing the number of APs [30]. GAs then evolved to include new factors such as the effect of different obstacles on the coverage performance [31]. In general, monoobjective strategies centered on optimizing a target function are more popular because of flexibility, robustness, and ease of implementation but may not be realistic enough when several design goals are sought to meet (e.g., number of APs, overall SNR, or throughput). For this reason, most of the real-world optimization problems are multiobjective in nature, which often have two or more target functions that must be met simultaneously and possibly conflict with each other. This causes having a set of optimal solutions (i.e., Pareto solutions) instead of having a single outcome. On the other hand, monoobjective algorithms may require being run every time to obtain multiple solutions. These drawbacks were addressed in this field thanks to a multiobjective genetic algorithm (MOGA) approach. Such is the case of an improved adaptive

genetic algorithm (IAGA) for WLAN deployment based on attaining the AP quantity and signal route wastage as minimum as possible [32]. Analogously, a multiobjective strategy was proposed to obtain various optimal placement configurations for different numbers of APs based on the SNR region [33]. Although closer to the work being proposed, as main disadvantage, these algorithms were not provided as a modelling software for the complete study of the WLAN operation in Layer 1 and Layer 2 (i.e., not interactive and exhaustive).

In this paper, we specifically used NSGA-II, one of the most representative approaches of MOGAs due to its ability to find multiple Pareto solutions in a simple execution [34]. As a main advantage, this approach allows easily dealing with concave and discontinuous Pareto boundaries, combined with a crowding operator, allowing obtaining a wider set of optimal solutions than other GAs of first and second generation as the niched Pareto genetic algorithm (NPGA) [35], nondominated sorting genetic algorithm (NSGA) [36], strength Pareto evolutionary algorithm (SPEA) [37], Pareto archived evolution strategy (PAES) [38], Pareto envelope-based selection algorithm (PESA) [39], microgenetic algorithm (Micro-GA) [40], improved niched Pareto genetic algorithm (NPGA2) [41], improved envelope-based selection algorithm (PESA-II) [42], or improved strength Pareto evolutionary algorithm (SPEA2) [43], among others.

3. Modelling System

The first implementation of WiFiSim—acronym for Wireless Fidelity Simulator—consisted of an intelligent modelling software developed in Java™ with the Eclipse framework. WiFiSim allows modeling various parameters in the PHY and MAC layers of the OSI model with support for the IEEE 802.11a/b/g/n standards, including stand-alone configuration of APs and wireless clients (e.g., interbeacon frame, rate and sensitivity, Cartesian coordinates, transmission power, packet size, RTS threshold, and packet load distribution), support for a customized library of materials, and control of the medium access (i.e., CSMA/CA mechanism, RTS/CTS mechanism, and back-off algorithm). The validation of models in WiFiSim is done in a similar way as in AirMagnet Survey, WiTuners, and Ekahau Site Survey but using complementary tools, for instance, WiFi Analyzer. This application allows us to manually conduct real WLAN tests, such as measuring the signal sensibility, antenna aiming, or APs detection [44]. Thus, once nodes and obstacles are added to a wireless scenario, we can simulate network models with a high degree of realism and study signal loss-related problems. Further details are described in [4, 5].

In order to facilitate the graphical deployment of WLANs, WiFiSim was firstly extended with a steady-state genetic algorithm (SSGA). This allowed users to dynamically work on 3D maps and specify whether the APs must be automatically arranged anywhere inside the buildings or fixed on the walls according to common ICT installation requirements. The main advantage of using SSGA is that multiple optimal placement solutions can be obtained for the same configuration of APs (i.e., from 1 AP to 8 APs). This provides more



FIGURE 1: WiFiSim's control panel: (1) graphical deployment for Layer 1, (2) steady-state GA module, (3) multiobjective GA module, and (4) study of Layer 2.

alternatives to the network designer than automatic RF-based algorithms. Moreover, SSGA is able to automatically plan APs in both empty spaces (i.e., when the nodes are not known a priori) and populated environments (i.e., when the nodes are fixed beforehand). Hence, SSGA seeks to optimize the coverage when designing empty planes or maximize it on customer locations according to the user requirements. Briefly explained, SSGA consisted in the following steps: (1) begin with a selection of candidate solutions (i.e., positions for a specific number of APs), (2) evaluate the solutions according to the maximum coverage as objective, (3) evolve these solutions by using parent selection, crossover, mutation, and replacement with subsequent generation improvements as long as the best solution does not change, and (4) provide those best solutions to the user. Additional details can be found in [6].

3.1. Multiobjective Genetic Algorithm. As a complementary strategy to SSGA, a MOGA module has been integrated into WiFiSim. The major improvement of using MOGA compared to SSGA is that multiple optimal placement configurations consisting of different numbers of APs can be simultaneously obtained from a single run. That way, the functionalities of WiFiSim were extended by not only selecting up to 8 APs with SSGA but also collecting 32 simultaneous solutions per WLAN design with NSGA-II. It gives users a greater range of optimization and experimentation as shown in Figure 1.

The modelling technique was developed using jMetal, a Java framework for developing metaheuristic algorithms [45]. NSGA-II basically consists of a parent population P_t and a descendant population Q_t , both of size N , combined to form a new population $R_t = P_t \cup Q_t$ of size $2N$. While P_t is randomly created based on an initialization method, Q_t is generated from P_t using a tournament, crossing, and mutation process.

Then R_t is classified into Pareto fronts (F_i) according to a nondominated ordering process and a fitness function such that $R_t = \{F_1, F_2, F_3, \dots, F_i\}$. Instead of randomly choosing the Pareto fronts to form the next generation of size N , NSGA-II takes the best solution fronts and discards the worst fronts. This is efficiently done by computing the crowding distance (d_i) of their solutions, meaning the value of the search space around a solution that is not occupied by another solution in the population:

$$d_j^m = d_j^m + \frac{f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}}{f_m^{\max} - f_m^{\min}}, \quad (1)$$

where m stands for each of the objective functions and j means the solution taken from the Pareto front for which $j + 1$ and $j - 1$ stand for the neighboring solutions. As a result, NSGA-II ensures the best and more diversity results since the solutions are distanced from each other when the population converges towards the optimal Pareto front after several generational cycles. The problem formulation and the genetic operators used are discussed in the following sections.

3.2. Map Structure and Objectives. The floor maps were codified in NSGA-II by means of a cell division method. In the case of the SSGA, maps were divided—depending on the floor shape and obstacles available—into square cells, rectangular cells, and mini-cells with resolution up to $1 \times 1 \text{ m}^2$ (i.e., 32×32 pixels by default). The NSGA-II approach was improved, treating the cells closer to reality by using not only standard cells but also triangular and polymorphic shapes. This resulted in a more efficient cell division method since it completely covers the whole useful space (Figure 2).

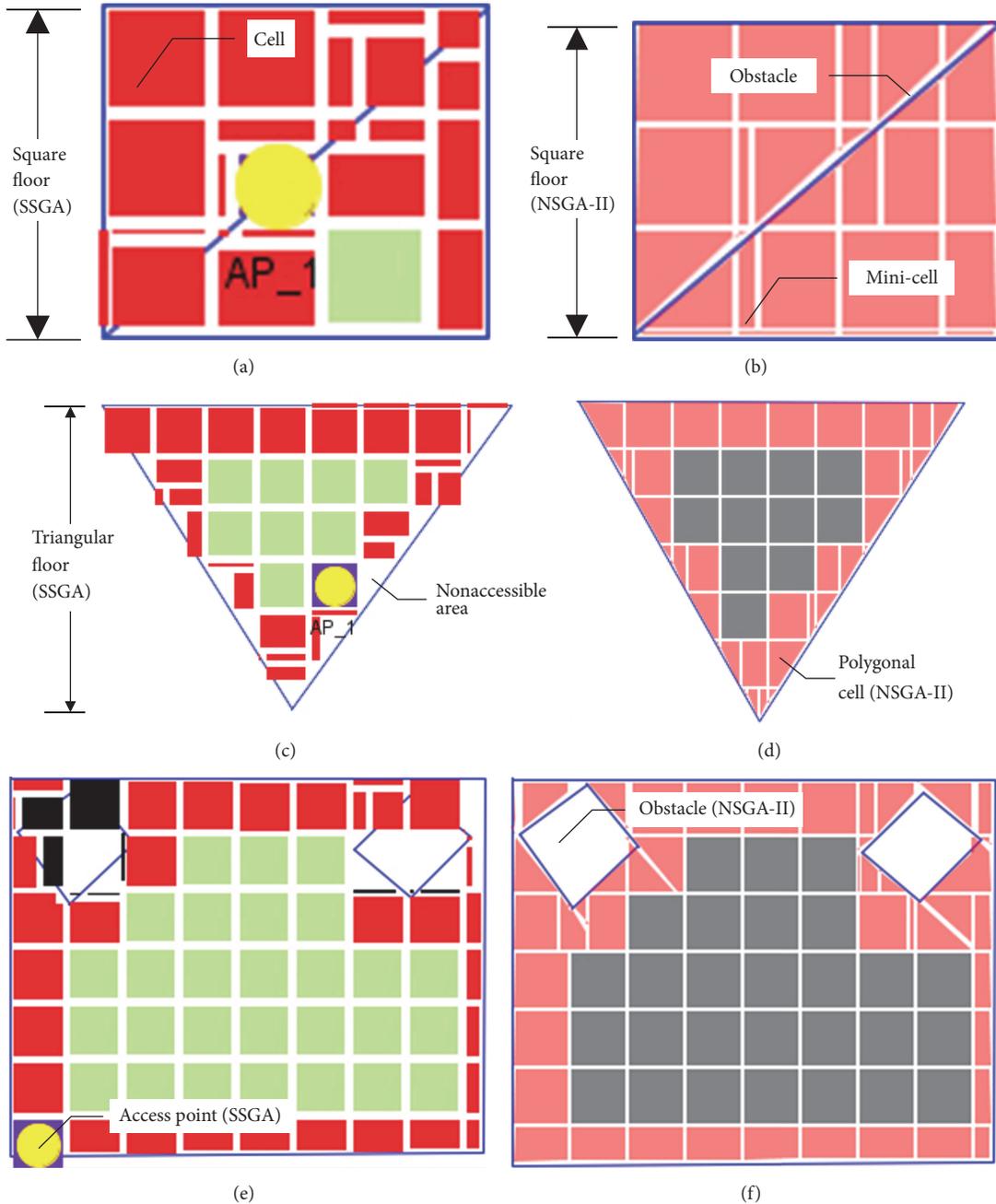


FIGURE 2: Comparison of the grid structure in SSGA (left) and NSGA-II (right) where red, green, and grey cells stand for covered areas: (a)-(b) square building with diagonal obstacle, (c)-(d) triangular building, and (e)-(f) rectangular building with inner obstacles (e.g., pillars).

There were two optimization objectives in this problem: minimize the number of APs and maximize the number of cells covering the whole planning area. These objectives were in conflict since reducing the number of APs inevitably implies decreasing the average signal as well, and vice versa. The first decision criterion was to minimize the number of APs to reduce the infrastructure cost as much as possible. As second decision criterion, NSGA-II computes the signal attenuation by distance and obstacle at each cell for every AP configuration (i.e., from 1 to 32 APs). The free space path

loss (FSPL) is calculated as a function of the distance and frequency, whose amount of signal loss is given in decibels by

$$\text{FSPL} = 20 \log_{10}(d) + 20 \log_{10}(f) + 32.44, \quad (2)$$

where d is the distance from the transmitter (in Km) and f is the frequency (in MHz). The loss signal by obstacle is then subtracted to (2) and the total power received at each cell is modeled as follows:

$$P_{rx} = P_{tx} - (\text{FSPL} - L_o), \quad (3)$$

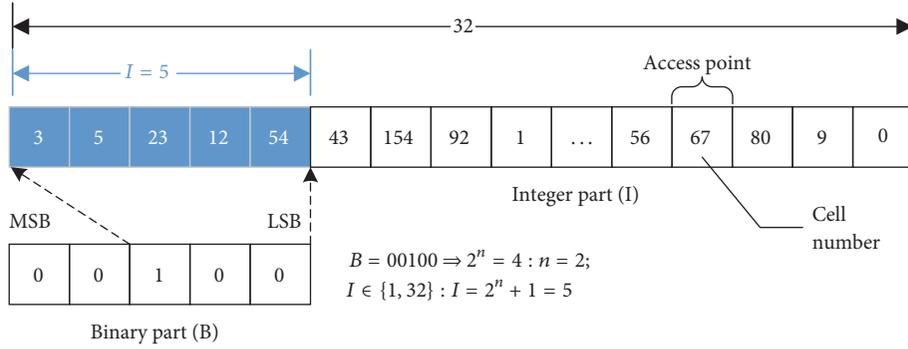


FIGURE 3: Coding scheme of the chromosome utilized in NSGA-II.

where P_{rx} is the power received at each cell by every AP, P_{tx} is the power emitted for each AP, and L_o is the loss signal due to the obstacle presence. Once the power signal is calculated at each map cell, two specific user constraints defined in the algorithm options are evaluated. These are the maximum bandwidth and Wi-Fi protocol. Since the rate and modulation scheme used in the WLAN set the media sensibility (i.e., rate in Mbps versus power in dBm), this serves to filter out the solution calculated by the algorithm in (3) and satisfy the design constraints. Therefore, the planning of the AP location plays a key role in maximizing the WLAN transmission coverage and communication throughput.

3.3. Coding Scheme and Chromosome Selection. The number of APs and their locations were concatenated to form a chromosome (Figure 3). We modified jMetal to use a double chromosome structure in NSGA-II consisting of a binary part and an integer part that codifies the two decision variables. The first one stands for 5 bits whose equivalent decimal value stands for the number of APs to be evaluated in the integer part of that chromosome. The second part contains 32 values that identify the cell number where the APs will be located. The number of cells will depend on the map size, number of floors, and obstacle shapes.

The chromosome selection is based on a fitness function and the stochastic universal sampling selection, where each individual of the population is valued in a nondominated Pareto rank after a selection process, becoming part of the next generation of those chromosomes with lower rank value [46]. In case of individuals with the same rank value, a crowding operator based on (1) is applied to discriminate individuals. The lower the value of the crowding operator, the greater the diversity of solutions that a chromosome has.

3.4. Mutation and Crossover Operators. The mutation process is applied independently to the binary and integer part of the chromosomes according to values of probability that initially are chosen very low (e.g., 3%). Specifically, we have used the integer flip mutation method for the integer part, which takes random positions of a chromosome to change their values. In case of generating a chromosome with more than one AP in the same cell position, the algorithm will consider them as the same AP and filter out this solution

in favor of others presenting the same number of APs but in different map cells. As a result, the algorithm discards those anomalous chromosomes before passing to the next generation. Regarding the binary part, we used a bit flip method with similar mutation to the integer flip technique but considering the range [0, 1].

The crossover process is also applied to the integer and binary parts severally. In particular, a uniform crossover scheme (UX) was applied to the integer part in which the values of two chromosomes are compared individually and swapped with a fixed probability, typically 0.5. Regarding the binary part, we used a one-point crossover operator in which all the values from a point of a chromosome are interchanged with the corresponding values in the other chromosome compared. Both crossover operators were entirely programmed in jMetal for this project.

3.5. Modelling Improvements. Beyond being an evolutionary genetic algorithm, NSGA-II was conceived to exploit its full capability as an intelligent modelling software. To this end, NSGA-II was integrated into WiFiSim to afford the parameter configuration of the multiobjective algorithm. First, because the generation, mutation, crossing, and selection of the population are processes cyclically done by NSGA-II, we end the algorithm execution by setting a maximum number of evaluations. This value, fixed to 5,000 by default as depicted in Figure 1, can be customized by users to experiment with different results. Once the algorithm finishes, it enters the process of ranking solutions, filters out those that are anomalous, and sorts them according to the number of APs that compose it. Hence, users can move in an orderly way from one solution to another—by means of a cursor—to study the arrangement of up to 32 access points. Conversely, in SSGA, when a replacement exists after an iteration, the algorithm reevaluates the population to determine whether it improves or not. In case of remaining stuck during 500 iterations, SSGA ends.

Secondly, NSGA-II allows search for optimal solutions based on requiring a minimum percentage of allowable coverage in the WLAN infrastructure according to a Wi-Fi bandwidth and technology (e.g., 1, 6, 11, 54, 108, 130, 150, or 300 Mbps for IEEE 802.11n). In this way, NSGA-II models the relationship between transmission rate (Mbps) and

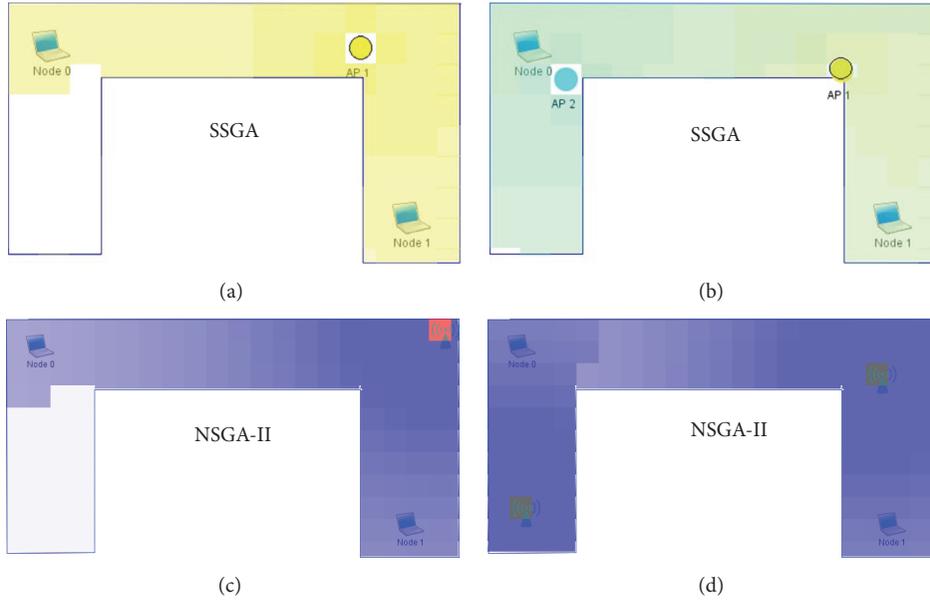


FIGURE 4: Comparison of solutions for SSGA and NSGA-II on a small scenario: (a)–(c) one AP and (b)–(d) two APs.

power (–dBm) for each AP technology (i.e., sensitivity to environment). This stands for an additional improvement giving users a greater optimization and experimentation grade compared to other WLAN planning tools or MOGA approaches.

Moreover, we have modified NSGA-II to develop a multithread version to parallelize the effort in processing the algorithm’s objectives. In this sense, we experienced that the higher processing requirements were spent in the evaluation of the strength signal at each map location for each AP configuration. The complexity proportionally increases with the number of resulting cells because of the number of building floors, map scale, and obstacles included. To this end, WiFiSim was programmed to use up to 8 simultaneous threads (set to 4 by default).

4. Experimentation

This section shows a comparative study about the performance and scalability of the SSGA and NSGA-II approaches. To this end, three target applications were considered: small, medium, and large scenarios. Experimenting with large environments has serious concerns compared with small and medium scenarios. Larger areas have tendency to contain more living rooms and obstacles to form a more complex morphology, thus resulting in a higher division of the map cells. This implies higher search spaces to compute with significant impact on uncertainty and execution time. In order to evaluate the tests, the parameter settings for both algorithms are summarized in Tables 2 and 3.

4.1. WLAN Design on Small Scenarios. This case study consisted in fixing two wireless clients on a small symmetric scenario where one and two APs were deployed anywhere

TABLE 2: Parameter settings for SSGA.

Seed	Seed 1
Mutation type	By destruction
Selection tournament size	Minimum
Crowding	By parents
AP power	200 mW, Cisco Aironet 1100

TABLE 3: Parameter settings for NSGA-II.

Threads	4
Population size	500
Max. evaluations	5000
Integer crossover probability	90%
Binary crossover probability	90%
Integer mutation probability	3%
Binary mutation probability	3%
AP power	200 mW, standard
Protocol	IEEE 802.11g
Maximum bandwidth	54 Mbps
Minimum coverage	25%

on the map. The map was set with a scale of 1:65 meters per pixel, which is equivalent to 29.28 m². This case study served to demonstrate how NSGA-II and SSGA managed their solutions to optimally deploy the WLANs. We observed that a solution with one AP did not cover the whole map (74% in Figure 4(a) and 82% in Figure 4(c)) whereas a solution considering two APs covered the complete scenario (Figures 4(b) and 4(d)). However, the algorithms successfully found similar solutions to optimally reach the two wireless clients in their locations.

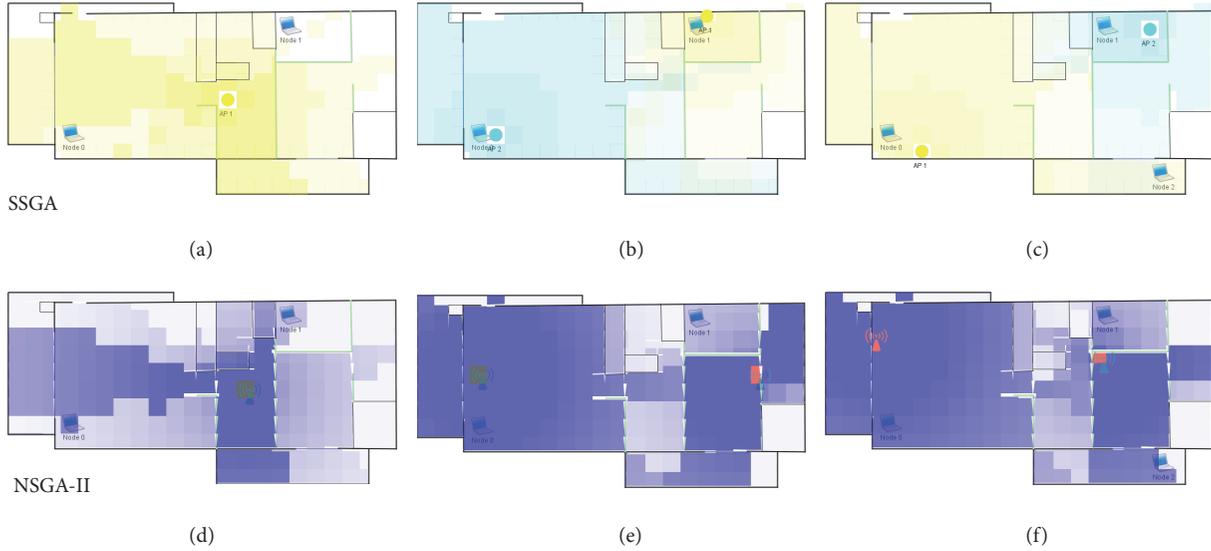


FIGURE 5: Comparison of solutions for SSGA and NSGA-II: (a)–(d) one AP covering one client, (b)–(e) two APs covering two clients, and (c)–(f) two APs covering three clients.

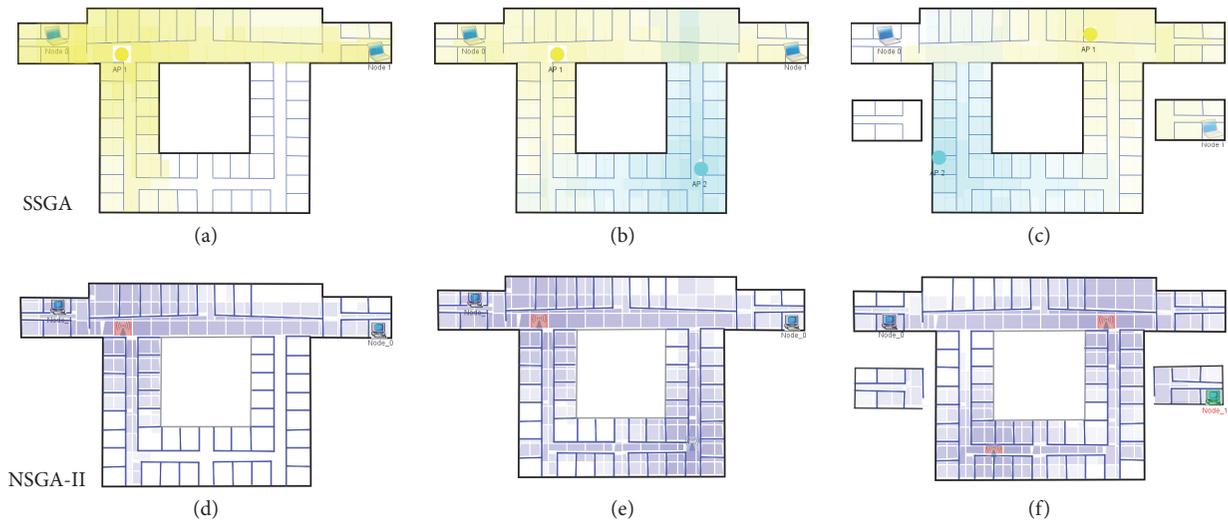


FIGURE 6: Comparison of solutions for SSGA and NSGA-II: (a)–(d) one AP for two nodes, (b)–(e) two APs for two nodes, and (c)–(f) two APs in two floors.

4.2. WLAN Design on Medium Scenarios. For this test we used a real map house with two bedrooms, two bathrooms, a dining room, a living room, and two terraces on a floor of 237.91 m^2 (Figure 5). The map was set with a scale of 1 : 32 meters per pixel. The scenario was modelled using a special contour-type material to form the outlines of the building (black thick lines with -95 dB), concrete walls for thin inner walls (green lines with -18 dB), construction beams for internal walls (black thin lines with -10 dB), a special “not evaluable” material for areas like closets (black lines with -80 dB), and a glass material for the doors accessing the terraces (blue lines with -2 dB).

The case study consisted in deploying one AP covering one client, two APs covering two clients, and two APs covering three clients. To this end, we placed the wireless

clients at the living room (node 0), a bedroom (node 1), and a terrace (node 2). Although SSGA and NSGA-II achieved similar coverage percentages on the map (between 64% and 91% for the three examples), we observed that SSGA provided a distribution more centered on the nodes than NSGA-II when the number of nodes was higher than the APs. On the contrary, NSGA-II tried to provide a more global and diverse coverage on the map (i.e., more balanced).

4.3. WLAN Design on Large Scenarios. The following experiment was conducted in a typical office environment consisting of a $73 \times 40 \text{ m}^2$ building with rooms, corridors, and open spaces with a scale of 1 : 10 meters per pixel corresponding to 2920 m^2 (Figure 6). Because of this structure, we created an exclusion zone around a landscaped courtyard at the center

TABLE 4: Performance of SSGA in a small scenario.

Target APs	Solutions found	Min. coverage	Max. coverage	AP location	Time (s)
1	1	64%	74%	Walls	0.10 ± 0.00
2	3	80%	84%	Walls	0.30 ± 0.94
3	4	80%	88%	Walls	0.36 ± 0.05
4	5	88%	92%	Walls	0.42 ± 0.04
5	6	96%	96%	Walls	0.66 ± 0.07
6	6	96%	100%	Walls	0.99 ± 0.10
7	4	100%	100%	Walls	0.82 ± 0.10
8	5	100%	100%	Walls	0.74 ± 0.05
<i>Average</i>					<i>0.54 ± 0.06</i>
1	1	64%	74%	Anywhere	0.10 ± 0.00
2	5	80%	84%	Anywhere	0.37 ± 0.08
3	3	84%	88%	Anywhere	0.32 ± 0.04
4	4	88%	92%	Anywhere	0.44 ± 0.05
5	6	92%	96%	Anywhere	0.69 ± 0.11
6	6	92%	96%	Anywhere	0.60 ± 0.08
7	6	96%	100%	Anywhere	0.62 ± 0.08
8	9	100%	100%	Anywhere	1.04 ± 0.11
<i>Average</i>					<i>0.52 ± 0.07</i>

of the building using the “not evaluable” special obstacle. The goal was to prevent the algorithms from including APs in uninteresting areas and considering only working areas to look for the optimal solutions. In addition, the environment was modelled using a contour line for the building perimeter (black lines with -95 dB), brick walls to separate the office rooms (blue lines with -3 dB), and a “roof” obstacle with 7 dB of attenuation to separate the two building floors.

The case study consisted in observing how the algorithms deployed one AP for two nodes, two APs for two nodes, and two APs in two floors located at the building walls. From the results, we obtained similar wireless coverage in the three examples (67%, 89%, and 99%, resp.). Moreover, we observed that SSGA and NSGA-II found similar AP distribution with slight differences. This suggests a great accuracy and repeatability on the algorithm solutions, thus providing a high stability regardless of the strategy chosen.

4.4. Time Analysis. In order to evaluate the complexity of the SSGA and NSGA-II strategies, we extended the above case studies with additional configurations and conducted a comprehensive analysis on the time cost. The methodology consisted in computing the average times and standard deviations of the algorithm solutions for a series of 10 records per setting. To this end, we captured the calls to the Java methods of SSGA and NSGA-II with VisualVM 1.3.9. The maps were modeled with 8, 36, and 120 building structures for the small, medium, and large scenarios, respectively, in a floor area of 800×600 pixels of resolution. For it, the experiments were performed with an Intel® Core™ i7 (2.6 GHz, 16-GB RAM).

From the results, we found in general that the SSGA executions lasted more; the higher the map resolution, the

higher the number of floor obstacles and the higher the target nodes or APs to search (Tables 4, 6, and 8). In addition, we found no evidence that the position of the APs within the map influenced the execution times (i.e., at the walls or anywhere) other than the number of solutions computed. These arguments are also valid for NSGA-II except for the execution times in medium and large environments which improved more with higher number of nodes deployed in the map (Tables 5, 7, and 9). This suggests that the uncertainty decreases as more clients are covered on the map. In other words, covering a single node requires more possible solutions of APs to be computed (i.e., more processing time) than covering several clients at the same time (i.e., more fitted solutions).

In relation to the size of the scenarios, we encountered significant differences in the execution times. In this sense, we found a better performance of NSGA-II compared with SSGA being more remarkable for the medium scenario. Moreover, it should be also mentioned that while SSGA requires one run per optimal solution searching (for one AP, two APs, etc.), NSGA-II computes a set of 32 optimal solutions in a single execution. This makes the NSGA-II approach significantly more efficient than the SSGA strategy in global terms.

With respect to the map resolutions, we attained times ranging from 0.54 ± 0.06 s to 2.25 ± 0.14 s for the small scenarios, from 27.76 ± 2.87 s to 40.06 ± 5.95 s for the medium scenarios, and from 11.98 ± 1.59 minutes to 17.15 ± 1.78 minutes for the large scenarios. This means that the algorithms are very sensitive to the map resolution, with the processing times being more suitable for small and medium scenarios but not so scalable for large environments due to the higher search spaces to compute. Nonetheless, these

TABLE 5: Performance of NSGA-II in a small scenario.

Nodes deployed	Min. APs	Min. coverage	Max. coverage	AP location	Time (s)
1	1	65%	87%	Walls	1.47 ± 0.07
2	2	100%	100%	Walls	1.54 ± 0.10
3	2	100%	100%	Walls	1.77 ± 0.05
4	2	100%	100%	Walls	2.10 ± 0.01
5	2	100%	100%	Walls	2.31 ± 0.11
6	2	100%	100%	Walls	2.54 ± 0.10
7	2	100%	100%	Walls	2.90 ± 0.10
8	2	100%	100%	Walls	3.15 ± 0.09
<i>Average</i>					<i>2.22 ± 0.07</i>
1	1	65%	87%	Anywhere	1.40 ± 0.11
2	2	100%	100%	Anywhere	1.57 ± 0.08
3	2	100%	100%	Anywhere	1.88 ± 0.09
4	2	100%	100%	Anywhere	2.13 ± 0.14
5	2	100%	100%	Anywhere	2.23 ± 0.20
6	2	100%	100%	Anywhere	2.49 ± 0.16
7	2	100%	100%	Anywhere	3.05 ± 0.13
8	2	100%	100%	Anywhere	3.31 ± 0.24
<i>Average</i>					<i>2.25 ± 0.14</i>

TABLE 6: Performance of SSGA in a medium scenario.

Target APs	Solutions found	Min. coverage	Max. coverage	AP location	Time (s)
1	1	76%	76%	Walls	0.54 ± 0.21
2	6	91%	93%	Walls	15.45 ± 1.88
3	5	91%	95%	Walls	20.07 ± 4.31
4	11	93%	98%	Walls	39.79 ± 8.93
5	8	96%	100%	Walls	36.66 ± 6.39
6	13	97%	100%	Walls	72.59 ± 8.74
7	12	98%	100%	Walls	72.39 ± 9.43
8	19	98%	100%	Walls	63.03 ± 7.72
<i>Average</i>					<i>40.06 ± 5.95</i>
1	1	76%	76%	Anywhere	0.83 ± 0.10
2	5	88%	92%	Anywhere	10.17 ± 1.20
3	5	92%	94%	Anywhere	19.88 ± 3.25
4	9	94%	97%	Anywhere	31.96 ± 4.30
5	10	95%	99%	Anywhere	36.87 ± 3.51
6	6	96%	100%	Anywhere	23.29 ± 2.06
7	12	97%	100%	Anywhere	67.79 ± 10.50
8	14	98%	100%	Anywhere	124.41 ± 22.47
<i>Average</i>					<i>39.40 ± 5.93</i>

times are in accordance with the times typically achieved by implementations based on genetic algorithms as described through [47–49]. As a solution, we enlarged the workspace of WiFiSim to reduce the resolution when simulating large scenarios. With this goal, the map resolution was changed from 1 : 10 meters by pixel in a workspace of 800×600 pixels to 1 : 20 meters by pixel in a workspace of 1600×1200 pixels.

Considering the large scenario and the worst execution times from Tables 8 and 9, we obtained a reduction from 39.08 min to 15.65 min for SSGA and from 16.37 min to 6.93 min for NSGA-II. This allows reducing the processing cost to more manageable times, thus resulting in an improvement of 40.05% and 42.34%, respectively. As a result, this test suggests that larger workspaces must be utilized to simulate larger

TABLE 7: Performance of NSGA-II in a medium scenario.

Nodes deployed	Min. APs	Min. coverage	Max. coverage	AP location	Time (s)
1	1	43%	100%	Walls	32.79 ± 2.94
2	2	56%	100%	Walls	29.87 ± 2.46
3	2	59%	100%	Walls	30.11 ± 2.31
4	3	54%	100%	Walls	25.29 ± 2.26
5	3	57%	100%	Walls	24.69 ± 3.01
6	3	53%	100%	Walls	26.39 ± 3.18
7	3	57%	100%	Walls	27.64 ± 3.07
8	3	65%	100%	Walls	27.68 ± 1.70
<i>Average</i>					28.05 ± 2.62
1	1	42%	100%	Anywhere	30.31 ± 3.17
2	2	63%	100%	Anywhere	33.44 ± 1.47
3	2	61%	100%	Anywhere	30.16 ± 2.40
4	3	63%	100%	Anywhere	24.74 ± 2.91
5	4	72%	100%	Anywhere	24.61 ± 2.72
6	4	70%	100%	Anywhere	24.11 ± 2.48
7	4	72%	100%	Anywhere	26.95 ± 2.86
8	4	75%	100%	Anywhere	27.83 ± 4.89
<i>Average</i>					27.76 ± 2.87

TABLE 8: Performance of SSGA in a large scenario.

Target APs	Solutions found	Min. coverage	Max. coverage	AP location	Time (min)
1	1	67%	67%	Walls	0.45 ± 0.06
2	7	87%	95%	Walls	5.05 ± 0.79
3	6	94%	98%	Walls	5.32 ± 0.55
4	6	97%	99%	Walls	7.34 ± 0.64
5	9	99%	100%	Walls	15.59 ± 0.90
6	6	100%	100%	Walls	17.80 ± 3.03
7	12	100%	100%	Walls	29.06 ± 3.49
8	9	100%	100%	Walls	18.96 ± 2.00
<i>Average</i>					12.45 ± 1.43
1	1	64%	64%	Anywhere	0.74 ± 0.13
2	1	94%	94%	Anywhere	3.14 ± 0.50
3	3	98%	98%	Anywhere	4.21 ± 0.80
4	7	99%	100%	Anywhere	11.79 ± 1.20
5	8	99%	100%	Anywhere	18.59 ± 2.38
6	11	100%	100%	Anywhere	22.23 ± 0.29
7	17	100%	100%	Anywhere	37.40 ± 5.82
8	18	100%	100%	Anywhere	39.08 ± 3.13
<i>Average</i>					17.15 ± 1.78

environments, being besides more appropriate when drawing small details in large scenarios due to the larger view.

5. Conclusions and Future Works

WLAN design and planning is a complex task that demands of engineers high analytical and troubleshooting skills to

offer the best network performance and usage. While hand-operated traditional methodologies are not effective for practitioners due to cost and time, computer-aided systems must provide a high degree of realism and modelling capabilities to be useful. To contribute to this field, this paper presented an engineering tool—called WiFiSim—developed to facilitate the automatic WLAN planning of complex environments

TABLE 9: Performance of NSGA-II in a large scenario.

Nodes deployed	Min. APs	Min. coverage	Max. coverage	AP location	Time (min)
1	3	33%	99%	Walls	15.79 ± 1.12
2	3	32%	99%	Walls	14.55 ± 0.09
3	3	29%	99%	Walls	14.48 ± 1.52
4	4	39%	98%	Walls	12.63 ± 0.57
5	6	50%	99%	Walls	13.62 ± 1.85
6	6	60%	99%	Walls	10.80 ± 0.74
7	8	62%	100%	Walls	10.14 ± 0.62
8	8	65%	100%	Walls	10.40 ± 0.96
<i>Average</i>					<i>12.80 ± 0.94</i>
1	3	35%	99%	Anywhere	16.37 ± 1.36
2	3	32%	99%	Anywhere	11.93 ± 2.31
3	3	27%	99%	Anywhere	13.07 ± 1.89
4	5	47%	99%	Anywhere	10.83 ± 1.22
5	5	44%	99%	Anywhere	10.39 ± 1.46
6	6	46%	99%	Anywhere	9.85 ± 1.86
7	7	60%	99%	Anywhere	12.65 ± 0.52
8	6	55%	100%	Anywhere	10.66 ± 2.07
<i>Average</i>					<i>11.98 ± 1.59</i>

and to assist in decision-making prior to real deployment. This tool that affords the complete behavior modeling of Wi-Fi networks in Layer 1 and 2 of the OSI model has been improved with an optimization algorithm based on evolutionary genetic. The goal was to procure a set of suboptimal solutions very close to the best for the AP location problem considering the largest covered area and maximum strength signal as decision criteria.

With this purpose, a previous approach based on a monoobjective genetic algorithm (i.e., SSGA) was extended with a multiobjective genetic algorithm (i.e., NSGA-II). Thus, in addition to obtaining several AP locations for up to 8 different selectable configurations, the main advantage of using NSGA-II was the capability to attain 32 solutions at once with several simultaneous configurations of APs. This approach was also designed to satisfy the user design constraints concerning the transmission rate and Wi-Fi technology used in the WLAN.

To assess both approaches and highlight the advantages of the second algorithm, several scenarios with small, medium, and large areas were used to simulate typical Wi-Fi environments embodying an office, a house, and a campus. The aim of the various tests was to evaluate the consistency of the algorithms—and how they computed their optimal solutions—along different scenarios and map resolutions in relation to the structure complexity and the search spaces. From the results, we encountered that the algorithm executions lasted more; the higher the map resolution, the higher the number of map obstacles and the higher the number of targets (i.e., nodes or APs). Moreover, we found no significant difference in the execution times due to the position of the APs within the map (i.e., at the walls or anywhere). Concerning the map size, the algorithms achieved very affordable

processing times for small and medium scenarios, not so scalable for large environments due to the higher search spaces. Although the times were in accordance with the typical requirements for GA-based implementations, the execution cost can be reduced to more manageable times (up to 40.05% and 42.34%) using larger workspaces to model larger environments (e.g., from a map resolution of 1:10 meters by pixel in a workspace of 800 × 600 pixels to 1:20 meters by pixel in a workspace of 1600 × 1200 pixels). As a result, we conclude that the NSGA-II approach achieved higher performance in general terms compared with the SSGA strategy.

Regarding the present, we are currently working on connecting WiFiSim with an eLearning tool for the programming, study, and distribution of wireless protocols based on an institutional web repository. In addition, future developments are focused on improving several functional and technical capabilities of WiFiSim. In this sense, we are working to extend the features on the PHY and MAC layers to enhance the software realism. Regarding the PHY layer, this comprises new signal measurements given by the BER, EIRP, and RSSI, an antennae modeling, support for IEEE 802.11ac/ad, larger Wi-Fi vendor library, and the cost of the resulting infrastructures. As for the MAC layer, this consists in providing the complete IEEE 802.11 frame format, including new frames as the RIFS, EIFS, and PS-Poll frames, new connection procedures as the association, reassociation, authentication, and deauthentication, and the protection mode for mixed scenarios with DSSS and OFDM, as well as power management by the TIM and DTIM mechanisms.

Finally, a set of selected videos and the modelling simulation software are offered for the free use and evaluation at <http://www.uhu.es/tomas.mateo/wifisim>.

Abbreviations

AP:	Access point
BER:	Bit error rate
CA:	Collision avoidance
CSMA:	Carrier sense multiple access
CTS:	Clear to send
DSSS:	Direct sequence spread spectrum
DTIM:	Delivery traffic indication map
EIFS:	Extended interframe space
EIRP:	Effective isotropic radiated power
FSPL:	Free space path loss
GA:	Genetic algorithm
GUI:	Graphical user interface
IEEE:	Institute of Electrical & Electronics Engineers
IAGA:	Improved adaptive genetic algorithm
ICT:	Information and communication technologies
LAN:	Local area network
MAC:	Media access control
MICRO-GA:	Microgenetic algorithm
MOGA:	Multiobjective genetic algorithm
NPGA:	Niched Pareto genetic algorithm
NPGA2:	Improved niched Pareto genetic algorithm
NSGA-II:	Nondominated sorting genetic algorithm II
OFDM:	Orthogonal frequency division multiplexing
OSI:	Open system interconnection
PAES:	Pareto archived evolution strategy
PESA:	Pareto envelope-based selection algorithm
PHY:	Physical
PS-POLL:	Power save poll
RF:	Radio frequency
RIFS:	Reduced interframe space
RSSI:	Received signal strength indicator
RTS:	Request to send
RW:	Random walk
SA:	Simulated annealing
SNR:	Signal to noise ratio
SPEA:	Strength Pareto evolutionary algorithm
SSGA:	Steady-state genetic algorithm
TIM:	Traffic indication map
TS:	Tabu search
WI-FI:	Wireless Fidelity
WIFISIM:	Wireless Fidelity Simulator
WLAN:	Wireless area network.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to express their very great appreciation to F. A. Márquez Hernández, D. Ortiz Fuentes, B. Moriña Arias, and M. E. Pedraza Escudero for their valuable and constructive work that helped to improve this research.

References

- [1] J. Kennington, E. Olinick, and D. Rajan, "Introduction to optimization in wireless networks," *International Series in Operations Research and Management Science*, vol. 158, pp. 1–6, 2011.
- [2] D. Hucaby, *CCNA Wireless 640-722 Official Cert Guide*, Cisco Press, 2014.
- [3] D. Vassis and G. Kormentzas, "Performance analysis of IEEE 802.11 ad hoc networks in the presence of hidden terminals," *Computer Networks*, vol. 51, no. 9, pp. 2345–2352, 2007.
- [4] T. D. J. Mateo Sanguino and F. A. Márquez, "Optimal design and deployment of wireless LANs based on evolutionary genetic strategy," *Advances in Intelligent Systems and Computing*, vol. 474, pp. 283–291, 2016.
- [5] T. J. Mateo Sanguino, F. A. Márquez Hernández, and C. Serrano López, "Evaluating a computer-based simulation program to support wireless network fundamentals," *Computers & Education*, vol. 70, pp. 233–244, 2014.
- [6] T. De J. Mateo Sanguino, C. S. Lopez, and F. A. M. Hernandez, "WiFiSiM: An educational tool for the study and design of wireless networks," *IEEE Transactions on Education*, vol. 56, no. 2, pp. 149–155, 2013.
- [7] S. Vasudevan, *A Simulator for analyzing the throughput of IEEE 802.11b Wireless LAN Systems [Master, thesis]*, Virginia Polytechnic Institute and State University, 2005.
- [8] C. Karthik and J. Paul, "Optimal design of wireless local area networks (WLANs) using simulation," in *Proceedings of the 2009 IEEE Military Communications Conference, MILCOM 2009, USA, October 2009*.
- [9] S. U. Nnebe, G. N. Onoh, and C. O. Ohaneme, "Empirical Analysis of Signal-to-Interference Ratio Variations in IEEE 802.11b WLAN," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 1, no. 2, pp. 263–270, 2012.
- [10] W. Zhu, F. Gao, Z. Song, and K. He, "Theoretical Analysis and Test of EMF in TDFI Bus," in *Progress in Electromagnetics Research Symposium Abstracts*, Guangzhou, China, 2014.
- [11] J. Gu, J. Zheng, and S. Zhang, "A graphical user interface in WLAN monitoring and management system," *Journal of Networks*, vol. 7, no. 4, pp. 644–651, 2012.
- [12] E. Neha Sharma and G. C. Lall, "Enhancement and Characterization of Indoor Propagation Models," *International Journal of Scientific & Engineering Research*, vol. 2, no. 11, pp. 1–7, 2011.
- [13] I. Aldasouqi and W. A. Salameh, "Using GIS in Designing and Deploying Wireless Network in City Plans," *International Journal of Computer Networks*, vol. 6, no. 4, pp. 66–75, 2014.
- [14] J. Scheuner, G. Mazlami, and D. Schöni, "Probr - a generic and passive WiFi tracking system," in *Proceedings of the 2016 IEEE 41st Conference on Local Computer Networks (LCN)*, pp. 495–502, IEEE, Dubai, United Arab Emirates, November 2016.
- [15] "Wolf WiFi VIRI Help Documentation," Tech. Rep., 2011, <https://twitter.com/wolfwifi>.
- [16] J. Hunt, "Working successful motion control via standard Ethernet," *The Journal of Industrial Network Connectivity*, vol. 71, pp. 33–38, 2012.
- [17] S. Vigneshwaran and C. Selvaraj, "Implementation of WLAN N and Estimation of Co Channel and Adjacent Channel Interference," *Journal of Computer Engineering Information Technology*, vol. 5, no. 2, 2016.
- [18] S. Blomqvist, *Improvement Proposal for Wireless Office Networks [BSc Degree]*, Mälardalen University, 2017.

- [19] R. J. Bartz, *Mobile Computing Deployment and Management: Real World Skills for CompTIA Mobility+ Certification and Beyond*, Wiley & Sons, 2015.
- [20] L. Phifer, "Managing WLAN Risks with Vulnerability Assessment," in *White Paper*, Core Competence, Inc., 2008.
- [21] J. Kruys and L. Qian, "RF Spectrum, Usage and Sharing," in *Sharing RF Spectrum with Commodity Wireless Technologies*, Signals and Communication Technology, pp. 3–14, Springer Netherlands, Dordrecht, 2011.
- [22] T. Vanhatupa, "Wi-Fi Capacity Analysis for 802.11ac and 802.11n - Theory & Practice," in *White Paper*, Ekahau, Inc., 2013.
- [23] "Optimus Information, Inc. Open-Source vs. Proprietary Software Pros and Cons," in *White paper*, 2015.
- [24] K. M. Reineck, *Evaluation and Comparison of Network Simulation Tools [Master, thesis]*, University of Applied Sciences, Bonn-Rhein-Sieg and Fraunhofer-Gesellschaft, 2008.
- [25] R. M. Whitaker and S. Hurley, "Evolution of planning for wireless communication systems," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences, HICSS 2003*, USA, January 2003.
- [26] A. Mc Gibney, M. Klepal, and D. Pesch, "Agent-based optimization for large scale WLAN design," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 470–486, 2011.
- [27] A. Mc Gibney, M. Klepal, and D. Pesch, "A Wireless Local Area Network Modeling Tool for Scalable Indoor Access Point Placement Optimization," in *Proceedings of the 2010 Spring Simulation Multiconference*, vol. 163, pp. 1–8, 2010.
- [28] M. Unbehauen and M. Kamenetsky, "On the Deployment of Picocellular Wireless Infrastructure," *IEEE Wireless Communications Magazine*, vol. 10, no. 6, pp. 70–80, 2003.
- [29] S. A. Chen, Y. H. Lee, R. Y. Yen et al., "Optimal prediction tool for wireless LAN using genetic algorithm and neural network concept," in *Proceedings of APCC/OECC'99 - 5th Asia Pacific Conference on Communications/4th Optoelectronics and Communications Conference*, pp. 786–789 vol.1, Beijing, China, October 1999.
- [30] R.-H. Wu, Y.-H. Lee, and S.-A. Chen, "Planning system for indoor wireless network," *IEEE Transactions on Consumer Electronics*, vol. 47, no. 1, pp. 73–79, 2001.
- [31] X. Huang, U. Behr, and W. Wiesbeck, "Automatic cell planning for a low-cost and spectrum efficient wireless network," in *Proceedings of the IEEE Global Telecommunication Conference (GLOBECOM'00)*, pp. 276–282.
- [32] S. Liu, C. Zhong, L. Zhang, and L. Zhang, "Optimization design of wireless local area network based on improved adaptive genetic algorithm," in *Proceedings of the 6th World Congress on Intelligent Control and Automation, WCICA 2006*, pp. 3357–3360, China, June 2006.
- [33] K. Maksuriwong, V. Varavithya, and N. Chaiyaratana, "Wireless LAN access point placement using a multi-objective genetic algorithm," in *Proceedings of the System Security and Assurance*, pp. 1944–1949, usa, October 2003.
- [34] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [35] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, pp. 82–87, June 1994.
- [36] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *Evolutionary Computation*, vol. 2, pp. 221–248, 1994.
- [37] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [38] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.
- [39] D. W. Corne, J. D. Knowles, and M. J. Oates, "The Pareto envelope based selection algorithm for multiobjective optimization," in *Parallel Problem Solving from Nature PPSN VI*, vol. 1917 of *Lecture Notes in Computer Science*, pp. 869–878, Springer, New York, NY, USA, 2000.
- [40] C. A. C. Coello and G. T. Pulido, "A micro-genetic algorithm for multiobjective optimization," in *Evolutionary Multi-Criterion Optimization: First International Conference, EMO 2001 Zurich, Switzerland, March 7–9, 2001 Proceedings*, vol. 1993 of *Lecture Notes in Computer Science*, pp. 126–140, Springer, Berlin, Germany, 2001.
- [41] M. Erickson, A. Mayer, and J. Horn, "The Niched Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems," in *Evolutionary Multi-Criterion Optimization*, vol. 1993 of *Lecture Notes in Computer Science*, pp. 681–695, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.
- [42] D. Corne, N. Jerram, J. Knowles, and M. Oates, "PESA-II: Region Based Selection in Evolutionary Multiobjective Optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 283–290, 2001.
- [43] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization," in *Proceedings of the Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, 2001.
- [44] C. Pei, Z. Wang, Y. Zhao et al., "Why it takes so long to connect to a WiFi access point," in *Proceedings of the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, May 2017.
- [45] J. J. Durillo, A. J. Nebro, and E. Alba, "The jMetal framework for multi-objective optimization: design and architecture," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '10)*, pp. 4138–4325, July 2010.
- [46] A. A. Márquez Hernández, *Cooperación entre el Mecanismo de Inferencia y la Base de Reglas usando Algoritmos Genéticos Multiobjetivo [Ph.D. thesis]*, University of Huelva, 2008.
- [47] D. Noever and S. Baskaran, *Steady-State vs. Generational Genetic Algorithms: A Comparison of Time Complexity and Convergence Properties*, Santa Fe Institute Working Papers, 1992.
- [48] B. M. Kim, Y. B. Kim, and C. H. Oh, "A Study on the Convergence of Genetic Algorithms," *Computers & Industrial Engineering*, vol. 33, no. 3–4, pp. 581–588, 1997.
- [49] J. Seo, E. Ko, and Y. Kim, "Performance Comparison of GPUs with a Genetic Algorithm based on CUDA," in *Proceedings of the Database 2014*, pp. 36–40.

Research Article

Design, Modelling, and Implementation of a Fuzzy Controller for an Intelligent Road Signaling System

José Manuel Lozano Domínguez  and Tomás de J. Mateo Sanguino 

Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática, University of Huelva, Ctra. Huelva-La Rábida S/N, Palos de la Frontera, 21819 Huelva, Spain

Correspondence should be addressed to José Manuel Lozano Domínguez; josemanuel.lldominguez@alu.uhu.es

Received 23 November 2017; Accepted 24 January 2018; Published 13 March 2018

Academic Editor: Eloy Irigoyen

Copyright © 2018 José Manuel Lozano Domínguez and Tomás de J. Mateo Sanguino. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Crossing points are not always 100% visible for drivers due to different factors (e.g., poor road maintenance, occlusion of vertical signs, and adverse weather conditions). USA estimated in 2015 the number of traffic accidents involving pedestrians and vehicles in 70,000 of whom 5,376 resulted in deceased people. To contribute in this field, this paper presents the design, implementation, and testing of a smart prototype system applied to pedestrian crossings—not regulated by semaphores—which try to reduce the accident rate on roads. The hardware and software system consists of a set of autonomous, intelligent, and wireless low-cost devices that generate a visual warning barrier perceived by drivers from a suitable distance when pedestrians traverse a crosswalk. In this way, drivers can reduce the speed of their vehicles and stop safely. The system's intelligence is carried out by a fuzzy controller that performs sensory fusion at both low level and high level with various types of sensors from local and neighboring devices. The tests conducted have determined an average success of 94.64% and a precision of 100%, thus corresponding with a very good test according to a ROC analysis. As a result, the system proposed has been patented and extended to international PCT.

1. Introduction

According to a recent report published by Goodyear and RACE (acronym for Real Automobile Club of Spain), around 10,000 accidents of pedestrians occur every year in Spain of whom 9,289 happened in urban areas, 222 resulted in death, and 2/3 were attributed to drivers [1]. Other studies also estimate 40% of the accidents when pedestrians are crossing through the right place [2]. According to a barometer from CIS—a Spanish sociological investigation center—the main causes of the accidents are mistakes and distractions of drivers (8.25%), lack of civic education (7.87%), poor road conditions (6.80%), and deficient signaling in some tracks (6.58%), among others [3]. A similar study published in USA estimated in 70,000 the number of traffic accidents in 2015 with pedestrians and vehicles involved. Of them, 5,376 resulted in deceased people, which has significantly increased in the last decade up to 15% according to the National Highway Traffic Safety Administration [4]. Its study determined that 78% of the accidents occur in low visibility

conditions, resulting in 74% at nighttime and 4% during the sunrise or sunset. Thus, pedestrian crossings are not always 100% visible due to different factors: (i) poor road maintenance (e.g., blurring lines due to the vehicle passing), (ii) occlusion of vertical signs (vegetation, large vehicles, etc.), and (iii) adverse weather conditions (e.g., rain, fog, or obscurity), among others. The distance at which drivers begin to brake the vehicle is decisive in the severity of the accident and the prevention of it. Hence, to improve the perception of drivers upon pedestrians is a key goal whether accidents or the severity of injuries in zebra crossings needs to be reduced [5, 6].

There are different solutions in the state of the art with the common aim of decreasing the number of accidents in crosswalks. They are classified into devices on board vehicles to detect pedestrians and approaches located on the road to detect both pedestrians and vehicles.

Regarding the first group, techniques and devices being included in vehicles aim to actively detect pedestrians to alert drivers. Such is the case of a prototype based on RADAR,

TABLE I: Comparison of features and capabilities for various approaches in the state of the art.

System reference	Safety	Autonomous	Wireless communication	Environment interaction	Public work	Cost
[7–9]	On car	No	No	✓	No	High
[10]	On road	No	No	No	✓	High
[11]	On road	✓	No	No	✓	High
[12]	On road	No	No	✓	✓	High
[13]	On road	No	No	✓	✓	High
[14]	On road	No	No	✓	✓	High
[15]	On road	No	No	✓	✓	High
[17]	On road	No	No	✓	✓	High
[19]	On road	No	No	✓	✓	High
[20]	On road	✓	✓	No	✓	High
Proposed	On road	✓	✓	✓	No	Low

camera, and sensory fusion devised to warn of possible collisions [7]. This system is similar to that of cameras, sensors on windshields, and bumpers that other proposals such as Ford Mondeo, Mercedes S Class, or Nissan from the UC3M (Universidad Carlos III de Madrid) implement on cars [8, 9]. However, these systems require the collaboration of the automotive industry to standardize and implement electronics on board. Moreover, these approaches represent personal devices not available to all users. That is, the system belongs to the car’s owner and it is not permanently available on public roads to all users.

In relation to the second group, very different concepts exist as for the type of installation, size, and price. For instance, a road sign formed by a luminous marquee over a pedestrian crossing that incorporates spotlights oriented towards the pavement to improve the driver’s visualization on pedestrians [10]; a trapezoidal speed bump placed on the road composed of passive lighting such as small light bulbs, LEDs (light-emitting diode), or optical fiber [11]; a section elevated above the track level like a trapezoidal highlight, where the road sign is made of electroluminous diodes activated by the pedestrian presence in zones of pressure placed at the access [12]; the control of traffic lights by means of the activation by weight of a tile placed on the sidewalk [13]; a horizontal road signaling system for crosswalks consisting of long-range and short-range optical sensors on a vertical support to detect pedestrians and vehicles [14]; a system with photoelectric emitters/receivers placed on the sidewalk to detect pedestrians that activates luminous devices located both in the periphery of the crosswalk and vertically on the sidewalk [15]; a proposal of super-safe smart crosswalk that detects persons at the entrance and exit of the pedestrian crossing and projects a virtual light barrier to warn drivers [16]; an embedded system on the sidewalk that carries a camera in charge of taking images of the pedestrian crossing to visually warn drivers when people is traversing [17]; a Japanese system implemented by Philips that warns pedestrians about the proximity of electrical silent vehicles near crosswalks [18]; a device that protects zebra crossings through warning lights placed on the sidewalk, which contains presence sensors for pedestrians and vehicles [19]; and a crosswalk alert

system based on a mast over the sidewalk which includes intermittent lights, solar cells, and wireless communication as a means of synchronizing the signaling with the opposite mast and where the trigger is a mechanical switch operated by the pedestrians [20].

The solution proposed in this paper is classified into the second group presented in the state of the art, that is, experimental systems placed on the road that detect both pedestrians and vehicles. Despite the variety of proposals found in the literature about this, they have not been widely deployed in our cities yet. The main reason is the high cost of some approaches requiring a fixed installation on the road pavement or the need for large supporting structures over the sidewalk. Our proposed system comprises a set of smart sensor devices with capability to alert drivers. The action performed consists in differentiating if there are people traversing the zebra crossing, then communicating that situation to the rest of the system nodes, and turning on synchronously the signaling as a visual barrier to alert drivers in order to safely stop their vehicles.

In summary, Table 1 compares the main features of some representative proposals in the state of the art in relation to our solution. The proposed system has several innovative features: (1) low-cost installation since it does not require public works; (2) minimum impact on traffic and users due to its small size; (3) autonomy since it does not require an electrical wiring infrastructure; and (4) intelligent control since it interacts with the environment.

Since the system is based on modular devices acting autonomously, this allows us to adapt the number of nodes depending on the crosswalk topology (one-way route, two-way route, multilane route, etc.). This in turn constitutes another advantage because if a node stays out of service, the rest of the system can keep on operating. Another interesting aspect is its low cost compared with other existing proposals, being an advantage for the final real deployment in urban environments. The system neither needs expensive installation in roads nor mobilizes large machines to do electrical wiring ditches. The size of the system modules is small, being placed on the road directly. Neither big luminous panels nor infrastructures placed in the road or sidewalks are necessary,

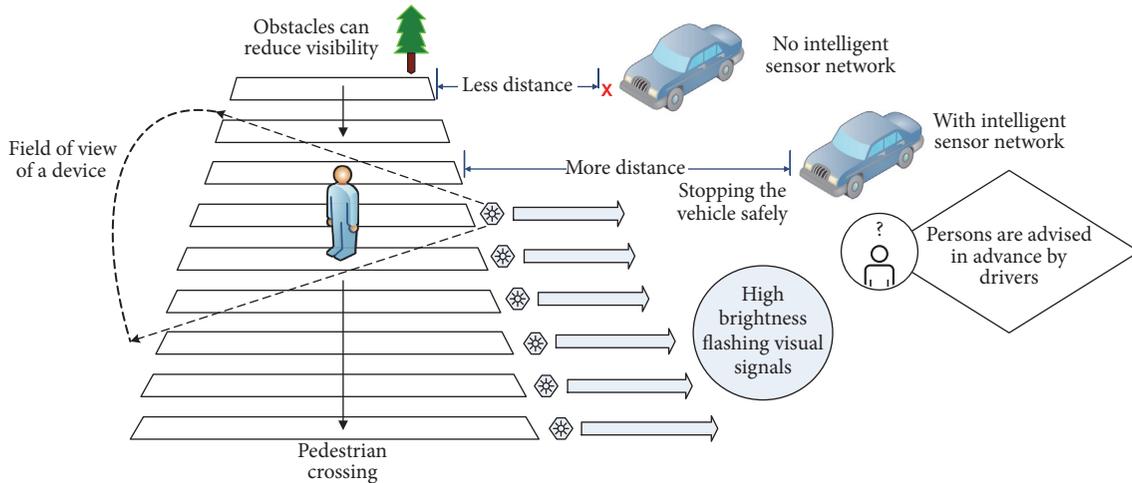


FIGURE 1: General depiction of the smart road signaling system.

thus presenting a minimal impact on users. Moreover, it is not necessary to connect the system to the power grid, thus obtaining energetic and economical savings. Nevertheless, the proposed system presents some disadvantages regarding the other solutions mentioned as it can be the autonomy beyond ~30 hours at full capacity. Finally, devices can get dirty by the vehicle transits and then loose energy-gathering capacity by being formed by solar cells.

This paper aims at the following significant contributions: (1) design, implement, and prove the feasibility of an autonomous smart system able to actively interact with the environment to detect pedestrians on zebra crossings; (2) generate an innovative way of light signaling to prevent drivers from pedestrians traversing roads; (3) quantitatively measure and analyze the system impact on the improvement of the road safety. Thus, this manuscript is structured as follows: Section 2 describes the general system, hardware elements, and modelling software designed for doing sensory fusion. Section 3 shows the experiments carried out and the results. Finally, Section 4 presents the findings as well as the future works of the proposed system.

2. Smart Road Signaling System

Based on the goals, the intelligent road signaling system consists of a variable number of devices which has been patented and extended through an international cooperation treaty to ARIPO (African Regional Intellectual Property Organization), OAPI (Organisation Africaine de la Propriété Intellectuelle), Eurasia, and Europe [21]. The devices aforesaid are arranged on longitudinal form around the crosswalk limits—depending on the lane topology—until covering the total road width and separated several centimeters between them so that each one covers a sector (Figure 1). Hence, when a pedestrian is detected by a device, the system is wirelessly synchronized to generate a visual intermittent signal which creates the sensation of a visual barrier over the road. This is caught by drivers and perceived as a warning light signal.

2.1. Hardware Description. The system is based on small autonomous devices that comprises a control unit, detection unit, signaling unit, power unit, real-time unit, and data storage unit (Figure 2). A transducer for the pedestrian detection is oriented towards the interior of the crosswalk while another transducer for the vehicle detection and the signaling unit is oriented towards the traffic flow. The cost of the prototype including the electronics and housing is about 1225€ per device, which maintains the advantage over other higher cost solutions.

2.1.1. Control Unit. The prototype is based on a 32-bit RISC (reduced instruction set computing) microcontroller (80 MHz, 1-MB RAM) with integrated wireless module including the IEEE 802.11 b/g/n/d/e/i/k/r standards [22]. Its function is to manage the electronics within the same node, process both the internal and external data, and provide intelligence to the overall system. To this end, the developed prototype utilizes proximity sensors that provide measurements proportional to distance unlike the presence sensors used in other road signaling approaches that only deliver binary signs of type “all/nothing.” This feature allowed us to perform analyses on object proximity over time, which offers a major operation capability in contrast to conventional presence sensors that only determine if an obstacle exists or not. As main benefit, this lets us to immerse the detection sensors within the road along the pedestrian crossing (i.e., the sensors are not located in the sidewalks as for the most approaches in the state of the art).

The intelligence is supported by operating rules based on fuzzy logic—which helps to decrease the number of false positives [23]—and other coordinated techniques [24] that process the information of the internal sensors of the same device and from other neighboring devices. This strategy allows improving the system accuracy and discerning between vehicles and people to generate visible signs only when objects are pedestrians and not vehicles. Furthermore, the nodes are connected through a WLAN (wireless local

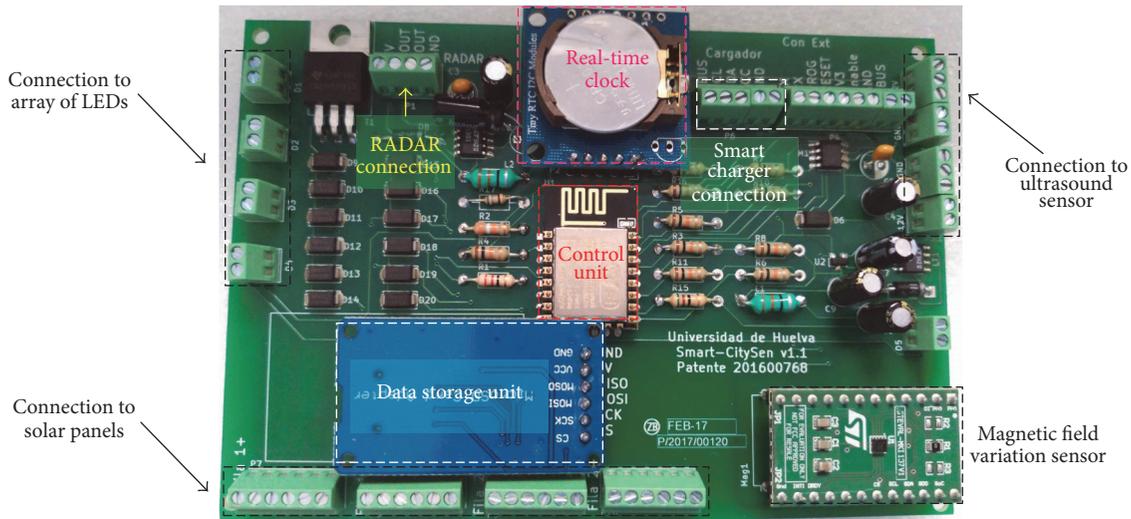


FIGURE 2: Outline of the inner electronics of the prototype system.

area network) with RF (radio frequency) technology whose function is to synchronize the devices' visual signposting through the system.

2.1.2. Detection Unit. Three sensors based on ultrasound, magnetic field variation, and RADAR were used in order to sense different objects around the crosswalks. Each of the sensors is oriented towards the type of object to be detected (i.e., ultrasound to pedestrians, magnetic and RADAR to vehicles). It is thereby possible to detect a pedestrian when entering to the zebra crossing from the entry points at the sidewalks or across any intermediate position from the road (e.g., when walking diagonally). In addition, it is possible to determine if there are vehicles circulating upon the pedestrian crossing. As an example, when a detection unit perceives a pedestrian approximation, the control unit activates the signaling. On the contrary, when a detection unit identifies a vehicle, the system disables the luminous barrier to avoid false positives.

The ultrasound sensor works at 42 KHz and can detect pedestrians at a distance between 0.6 and 5 meters. The magnetic field sensor provides detection in the x -, y -, and z -axes with a sensibility up to ± 16 Gauss, being capable of detecting vehicles circulating at 50 Km/h with a resolution in the order of centimeters. Additionally, the RADAR improves the sense of other vehicles that come frontally to the crosswalk from a distance between 10 and 20 meters (e.g., bicycles compounded of materials as aluminum or carbon not detected by the magnetic sensor). To this end, we utilized a Doppler-based RADAR working in the X-band in CW (continuous wave) acquisition mode with EIRP (effective isotropic radiated power) of 15 dBm.

2.1.3. Signaling Unit. The prototype system includes a set of high brightness LEDs which are visible under both low contrast and high contrast conditions (i.e., during day and night). An array of four LEDs is oriented to drivers to alert the pedestrians' detection while another one is directed

oppositely towards the pedestrians to indicate the system activity and facilitate the decision-making before crossing. The selected LEDs produce white cold light (7000°K) up to 140,000 mCd (4.82 lm) and can be seen from an angle of vision of $12^{\circ} \pm 6^{\circ}$. The road signaling presents an intermittent pattern (i.e., activation of 75 ms during 5 s) which is managed by the control unit through a low consumption strategy performed by means of a PWM (pulse-width modulation) control. The frequency has been selected experimentally so that the refreshment of the LEDs is not perceived by the human eye at the same time that it reduces the system consumption.

2.1.4. Power Unit. The supply of the system consists of a matrix of 12 solar panels of monocrystalline Silicon of $92 \times 54 \times 3 \text{ mm}^3$ with maximum power of 15 W, a rechargeable Li-Po (Lithium polymer battery) of 9000 mAh (3.7 V, 1C) with autonomy up to 29.89 hours, and a smart charger circuit with PPM (power path management). This allows supporting 25.52 Wh/day or energy consumed in a day by the system without getting any additional energy. To size the power supply unit, we used the latitude of Huelva (37.2583N , -6.9508W) during December as the period of minor daily solar insolation ($2.12 \text{ kWh/m}^2/\text{d}$)—or energy produced by the Sun in one m^2 per day—and a solar cell performance of 75% as reference. See Table 2 for further details.

In order to increase the autonomy of the system, a strategy to reduce the power consumption depending on the hour of the day was implemented. For it, a study carried out by means of Google™ Maps allowed us to know the time of the major pedestrians' attendance taking several ROIs (region of interest) of the city as reference (e.g., shopping centers and the historic city center). It was determined that the hours with major agglomeration of people were mainly distributed in the time slots between 12–14 hours and 18–20 hours. In addition, it gave us an idea of the hourly intervals of highest danger for pedestrians. Accordingly, in Figure 3, "Activated"

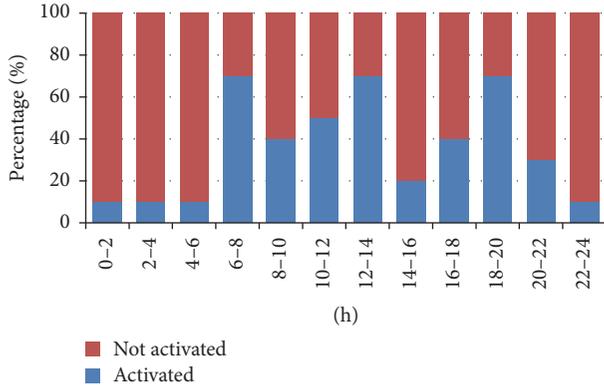


FIGURE 3: Timetable for the low-power consumption strategy. In blue, the estimated percentage of time that pedestrian crossings are typically used by people. In red, the estimated percentage of time that pedestrian crossings are not used by people.

TABLE 2: Average daily solar insolation (kWh/m²/day) in Huelva order by month.

Month	Insolation
January	2.15
February	3.20
March	4.53
April	5.52
May	6.82
June	7.25
July	8.13
August	7.27
September	6.00
October	4.10
November	2.60
December	2.12
Average of year	4.98

Estimated time of battery charging: 2 hours and 12 minutes.

(blue label) indicates the estimated percentage of the hourly section in which pedestrian crossings are utilized by people, while “No Activated” (red label) means the percentage in which the pedestrian crossings are not utilized by people. The whole strategy consisted in fitting the LED lighting and the reading frequency of the system’s transducers based on the pedestrian attendance and battery charge level. This allows increasing the battery life up to 5.8% in average. Although not significant, the electrical consumption has been one of the major challenges in this project.

2.1.5. Additional Units. To take a count of the operations carried out by the system such as the pedestrian detection, errors in units, and operating parameters, each device incorporates a data storage unit and a real-time clock unit. The storage electronics comprises an external solid state memory (i.e., 8-GB microSD card) that allows extending the storage capacity to implement a local database in each device. This database



FIGURE 4: Example of the prototype developed on the road.

stores all the events that happen both in the surrounding environment and in the inner device for further analysis. To this end, the real-time unit provides a date and hourly stamp for all the actions carried out by a device.

2.1.6. Wireless Communication. The wireless communication unit is entrusted to transfer data between the system’s nodes within the WLAN. One of the devices has the role of AP (access point), whose function is to manage and control the network operation. The rest of the devices work as clients such that when one of them detects an object, a broadcast message is sent towards the rest of the nodes in the WLAN.

The PHY (physical) layer has been implemented within the microwave band at 2.4 GHz, the MAC (media access control) layer utilizes the IEEE 802.11n standard at 300 Mbps with optional WPA/WPA2 (Wi-Fi protected access) encryption [25], the network layer uses unicast and broadcast IPv4 (Internet Protocol version 4) addresses, the transport layer has been implemented with UDP (user datagram protocol) since it accelerates the message delivery with regard to TCP (transmission control protocol) by dispensing with ACK (acknowledgment) messages as discussed in [26], and the application layer is based on the M3 (Machine-to-Machine Measurement) framework to exchange structured information between nodes [27, 28]. This standard declares semantic rules that help to identify clearly the source of a message, what fields it contains, the values of the fields, and its units.

2.1.7. Placement of the Device. The prototype’s housing has been constructed with a CNC (computer numerical control) machine that protects the inner electronics (Figure 4). The case is arranged in a matrix of 4×3 high quality tempered glasses (10H hardness, transmittance of 88–92%) with enough thickness to support the weight of the vehicles up to $8.16 \cdot 10^6$ N/m² in compression and preventing the solar panels’ surface from being damaged. The complete structure fulfills the IP67 standard of protection against dust and liquids. It has been fixed on the road by means of bituminous adherent (e.g., warm tar), epoxy glue (i.e., thermostable polymer hardened by a catalyst agent), or mechanical anchorage by means of screws (e.g., wall plug and lag screw). The housing does not need public works for its installation on the road. Moreover, it is reusable when a reasphalting of the road surface is required. The shape of the case presents a nonslipping knurling texture to avoid falls of pedestrians or bicycles, as well as a trapezoidal profile with

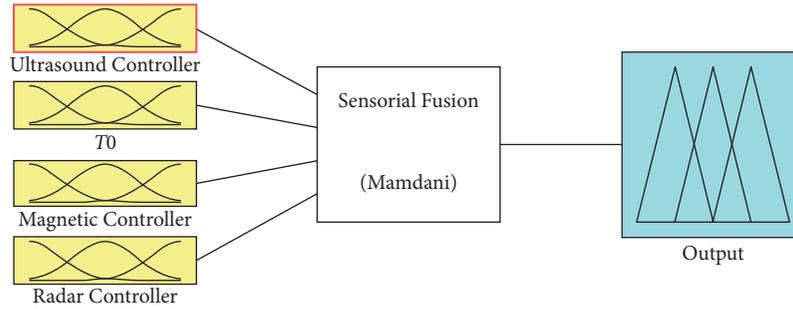


FIGURE 5: Structure of the fuzzy inference system.

a maximum size of $540 \times 500 \times 45 \text{ mm}^3$ and 45° of angle of attack to facilitate the vehicle movement. These features comply with the recommendations of the Spanish Ministry of Promotion [29].

2.2. Fuzzy Model. The device's intelligence consists of a sensory fusion process of association, correlation, and combination of information based on fuzzy logic. Specifically, we have used the three-level abstraction model of Dasarathy [30] to combine sensor measurements and high-level decisions (e.g., if a pedestrian exists or not upon the crosswalk). The measurements are fused in a cooperative way when coming from different sensors within the same device (i.e., ultrasound, RADAR, and magnetic field) and redundantly when belonging to the same sensor in several neighboring devices.

The model implemented is based on a Mamdani fuzzy controller with linguistic rules such as “If X_1 is A_1 and . . . and X_n is A_n , Then Y is B ” in which both antecedents and consequents are labels and rules obtained with an expert system [31]. Systems described in [32, 33] present an inference system similar to that described in this paper. Considering the number of tags (t) and input variables (n) to use, the complexity is raised to (t^n). Since the system has a high number of inputs and tags, a hierarchical structure with several controllers was designed to keep the rules in a number easily manageable by the expert. The membership of a fuzzy set is defined by trapezoidal functions—as they adequately model the system behavior and are not computationally complex—where the conjunction and implication operators use the minimum T-norm [34]. Moreover, the defuzzification process uses the FITA method (i.e., First Inter, Then Aggregate) since it is more consistent than FATI [35] and the MVP (Maximum Value Point) weighting method, also called weighted average method [36].

The system's sensory fusion has been modelled through the Fuzzy Logic Toolbox™ from Matlab® (Figure 5). This implementation is similar to that described in [37]. The “Ultrasound Controller” block manages the fusion of the ultrasound sensors from the local device and the remote ones at the neighboring nodes; its output indicates whether an obstacle exists on the pedestrian crossing. The “Magnetic Controller” is in charge of fusing the sensors of magnetic field variation and determining vehicles arriving at the zebra crossing; its output updates a temporary variable “ T_0 ” useful

to control slow traffic or stationary traffic that could generate false positives due to vehicles staying a long period of time over the crosswalk. The “Radar Controller” is entrusted of fusing the RADAR sensors and determining if an object approaching the zebra crossing is vehicle or pedestrian. Finally, the “Sensorial Fusion” block determines whether a pedestrian on the crosswalk exists by fusing the outputs from the previous fuzzy controllers.

To sum up, the signaling unit within a device is activated in case of positive detection (i.e., a pedestrian) and the whole system is synchronized by a broadcast message sent over the network to activate the nodes' signaling units. When the sensory fusion detects a vehicle, through either the magnetic sensor or the RADAR, an inhibition message is sent to avoid false positives. As an example, a number of cases with pedestrians and vehicles—either circulating, stopped, or parked—interacting around a crosswalk is shown (Figure 6). The system activity, formed by several nodes, is depicted at the bottom of the zebra crossings. The devices are colored in green when an object is detected on the pedestrian crossing by the ultrasound sensor, in blue when detected by the magnetic sensor, in red when detected by both the magnetic sensor and the RADAR, and in black when there is no presence of pedestrian or vehicle.

The various scenarios consist of a crosswalk without any agent (Case 0), a pedestrian entering and walking through the crosswalk (Cases 1 and 2), a vehicle navigating along the road (Case 3), a vehicle approaching the zebra crossing while a pedestrian begins to cross (Case 4), a vehicle approaching the crosswalk when a pedestrian is crossing (Case 5), a vehicle stopping or parking (Case 6), a vehicle stopping when a pedestrian begins to cross (Case 7), a pedestrian crossing and a vehicle stopping near the crosswalk (Case 8), and a vehicle detected by the magnetic and ultrasound sensors but not detected by the RADAR (Case 9), as well as a vehicle stopping while a pedestrian was crossing (conflictive case). The last example represents the most problematic scenario, which is addressed for the following fuzzy logic.

2.2.1. Ultrasound Fuzzy Controller. This block utilizes three inputs comprising normalized measurements from the ultrasound sensors at the local, right, and/or left nodes. The normalization allows using more than one transducer with different radiation patterns to adapt the azimuthal and horizontal range based on the device's position regarding

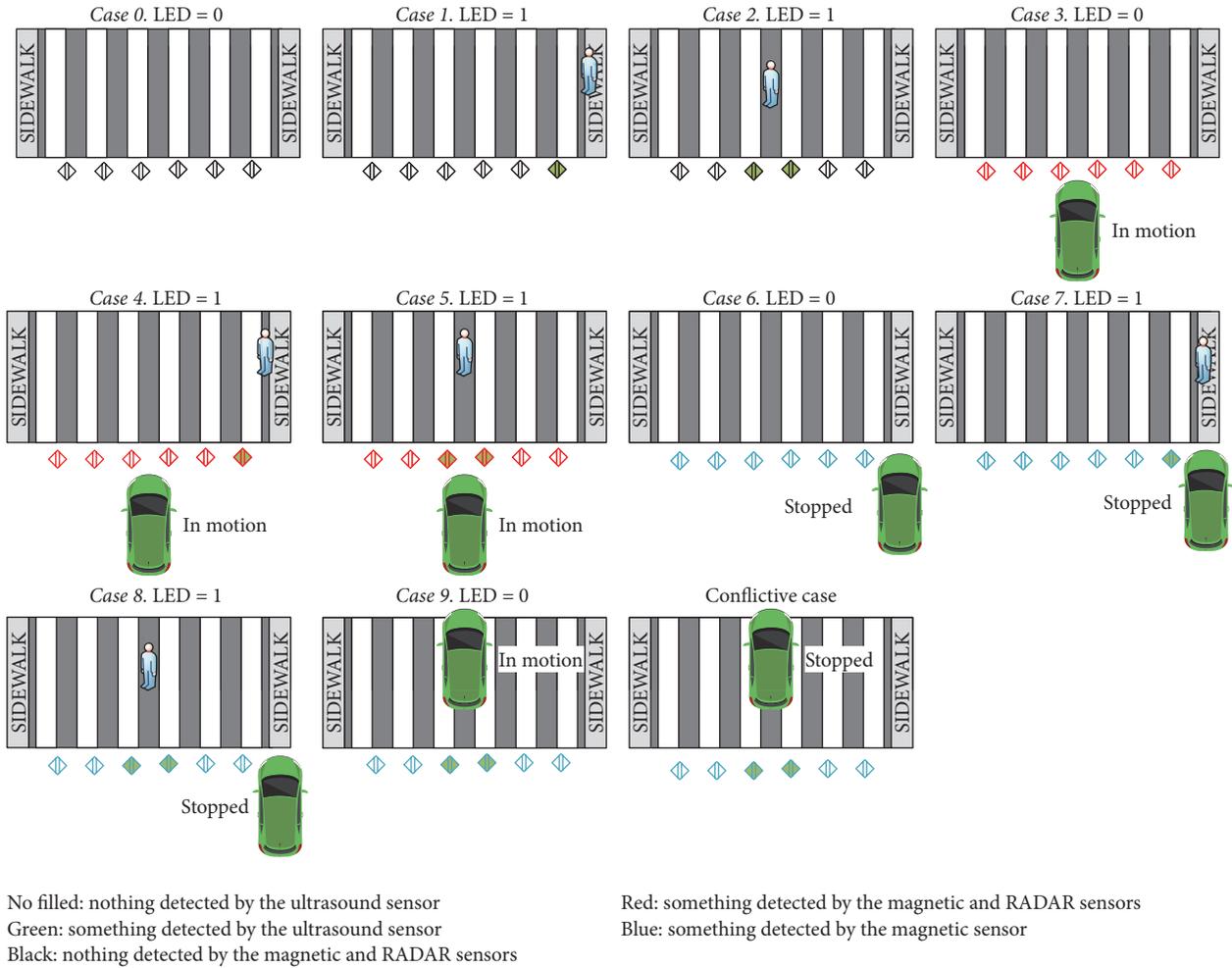


FIGURE 6: Case studies of the system operation.

the way (e.g., in the middle of the crosswalk or close to the sidewalk). The behavior of the ultrasound sensor versus temporary samples can be observed in Figure 7.

Each input has three tags meaning the distance of the obstacle in front of the sensor, being *Far*, *Medium*, and *Near* (Figure 8). The *Far* tag indicates that the object is not detected or is detected in a ROI external to the crosswalk. The *Medium* tag means that the object is detected in an uncertainty area or it is not detected with enough precision by the sensor. The *Near* tag points out an obstacle existing on the zebra crossing that is clearly perceived.

The output of the controller provides values within $[0, 1]$, whose range stands for the grade in which an object exists over the crosswalk expressed by means of two labels (*No* and *Yes*). Thus, a value near zero (0) means that an obstacle does not exist while a value close to one (1) indicates that it exists. In that case, it must be inferred by the rest of the fuzzy system whether the object is vehicle or pedestrian to activate the signaling unit.

The rules base and tags of the fuzzy controller were tuned experimentally by an expert system (Table 3). Since an ultrasound sensor typically has a nonlinear response whose

error increases due to distance, it was established that three inputs declaring a *Far* tag after the defuzzification process means no obstacle on the crosswalk. On the contrary, two inputs with the *Medium* tag or one input with the *Near* tag signifies the presence of an obstacle.

2.2.2. Magnetic Fuzzy Controller. This block uses an input per axis (x , y , and z) with three labels for each one called $Far\alpha$, $NearN\alpha$, and $Near\alpha$, being (α) the corresponding axis (Figure 9). The $Far\alpha$ tag stands for the value during the idle state of the magnetic field sensor, while $NearN\alpha$ and $Near\alpha$ indicate a magnetic field variation below and above the idle state, respectively. This variation is represented versus temporary samples (Figure 10).

The output of the magnetic fuzzy controller has been modelled with two tags ranging from 0 to 1 (*No* and *Yes*), where values near zero mean no vehicle close to the zebra crossing and values near one indicate the presence of a vehicle. Similarly, both the rules base and tags of the fuzzy controller were experimentally fitted by the expert system (Table 4). It was determined that a variation of the idle state at least in two of three axes of the magnetic sensor indicates

TABLE 3: Rule base for the ultrasound fuzzy controller.

Rule number	Local device	Left neighbor	Right neighbor	Obstacle
1	Far	Far	Far	No
2	Far	Far	Medium	No
3	Far	Far	Near	✓
4	Far	Medium	Far	No
5	Far	Medium	Medium	✓
6	Far	Medium	Near	✓
7	Far	Near	Far	✓
8	Far	Near	Medium	✓
9	Far	Near	Near	✓
...
27	Near	Near	Near	✓

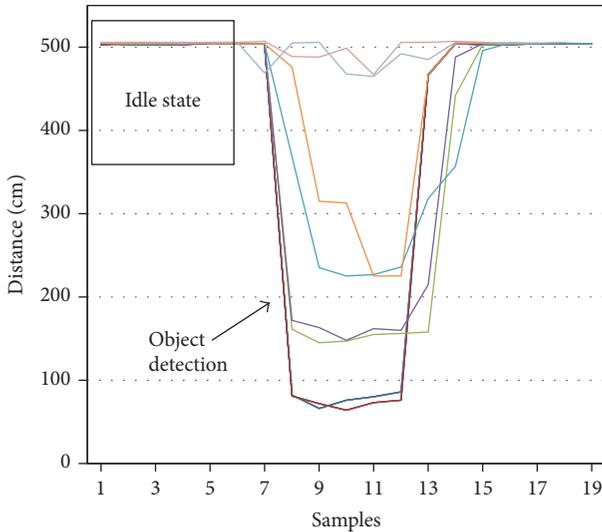


FIGURE 7: Detection of a pedestrian when crossing a crosswalk at different distances expressed in several colors. Blue and red show the detection of a pedestrian crossing at 100 cm away from the sensor, green and purple crossing at 200 cm, orange and light blue crossing at 300 cm, and pink and light purple crossing at 400 cm. For each distance there are two colors meaning the test crossing from right to left and vice versa. The range measurements taken by the sensor are shown on the y -axis while the temporary samples are observed on the x -axis.

presence of a vehicle driving around the zebra crossing; otherwise a vehicle does not exist.

2.2.3. Temporary Variable. The output of the magnetic fuzzy controller has the additional function of updating a counter called “ T_0 ” to measure the time elapsed since the sensor did not detect a vehicle, that is, to count the time elapsed since the output of this controller was near zero. This variable is useful to resolve conflictive cases—as that depicted in Figure 6—where it is necessary to know whether a vehicle is sited over the zebra crossing (i.e., constant values close to one).

TABLE 4: Rule base for the magnetic fuzzy controller.

Rule number	x -axis	y -axis	z -axis	Vehicle
1	Far x	Far y	Far z	No
2	Far x	Far y	Near Nz	No
3	Far x	Far y	Near z	No
4	Far x	Near Ny	Far z	No
5	Far x	Near Ny	Near Nz	✓
6	Far x	Near Ny	Near z	✓
7	Far x	Near y	Far z	No
8	Far x	Near y	Near Nz	✓
9	Far x	Near y	Near z	✓
...
27	Near x	Near y	Near z	✓

According to this logic, the temporary fuzzy controller was modelled with three tags called *Little*, *Medium*, and *Much* (Figure 11). The *Little* tag stands for a small period of time since the magnetic sensor detected the presence of a vehicle, *Medium* indicates that a certain time has elapsed—but not sufficiently large—since there was a vehicle on the zebra crossing (e.g., stopped), and *Much* indicates an interval of time large enough since the magnetic sensor detected the presence of a vehicle (i.e., parked).

2.2.4. RADAR Fuzzy Controller. The inputs defined in this controller block are the *State* and *Speed* variables (Figure 12). The first one stands for a digital output from the RADAR that points out the presence of an object in the FOV (field of view) of the sensor. So, the *State* tag indicates if there is an object present, where values close to zero (0) specify no object existing (*low*) and values close to one (1) indicate an object being detected (*high*). The *Speed* variable consists of two tags (*Minor* and *Major*) to designate values higher and lower than 5 Km/h. This value—experimentally determined (Figure 13)—stands for a speed threshold above which any detection belongs with certainty to a vehicle (i.e., everything that is not a pedestrian moving on foot).

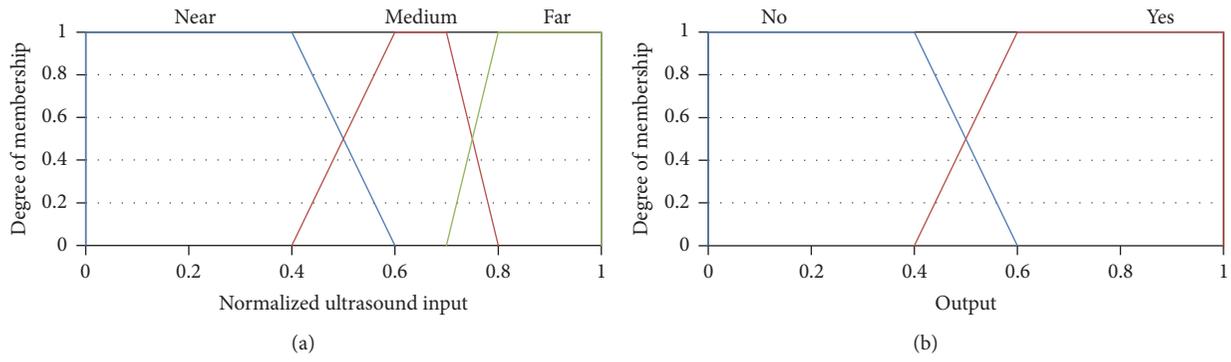


FIGURE 8: Labels for the ultrasound fuzzy controller: (a) input and (b) output.

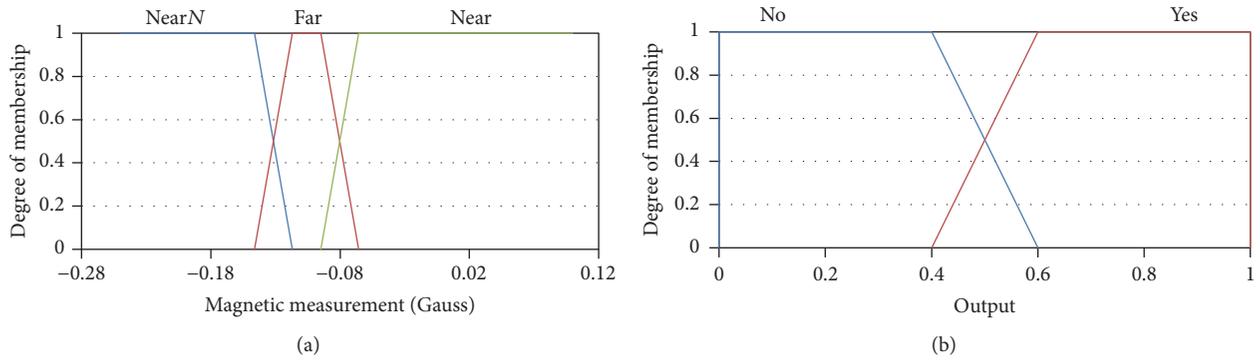


FIGURE 9: Labels for the magnetic fuzzy controller: (a) input for an axis and (b) output.

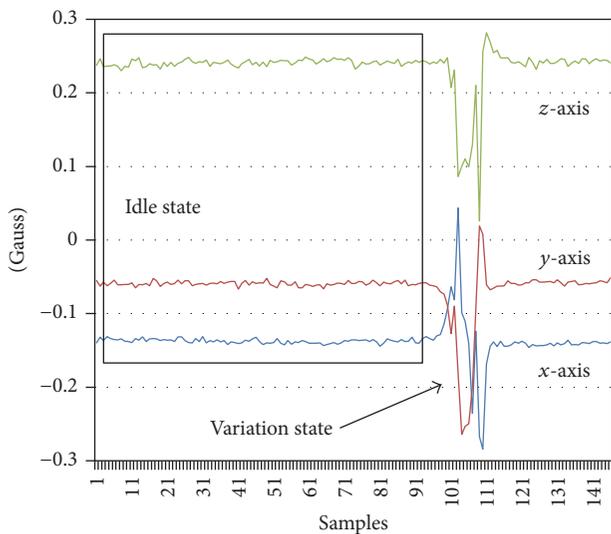


FIGURE 10: Detection of a vehicle by the magnetic sensor. The idle state and variation state are caused by the penetration of a car in the magnetic field. The Gauss measurements taken by the sensor are shown on the y -axis while the temporary samples are observed on the x -axis.

The output of the fuzzy controller is represented by the *Pedestrian* and *Vehicle* tags in the range within $[0, 1]$, meaning the values near zero (0) a *Pedestrian* and the values near one

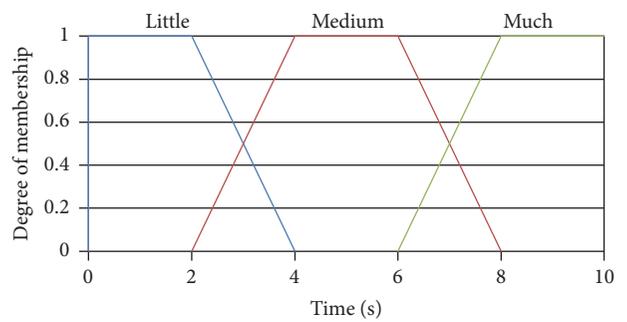


FIGURE 11: Labels for the temporary variable (T_0).

(1) a *Vehicle*. The rules base of the fuzzy controller has been made experimentally by the expert system, which determined that an object speed declared as *Major* with a state designed as *High* stands for a vehicle (Table 5). In any other case, the object will be potentially a *Pedestrian* or a *Vehicle* at low speed, whose nature must be inferred by the rest of the fuzzy system.

2.2.5. *Sensory Fusion Controller*. This block is entrusted to fuse all the fuzzy controllers previously described (i.e., ultrasound, magnetic, RADAR, and “ T_0 ”) whose function is generating an output to control the LED signaling unit. The output values have a range within $[0, 1]$, where values near zero (0) indicate that the signaling unit must not be activated

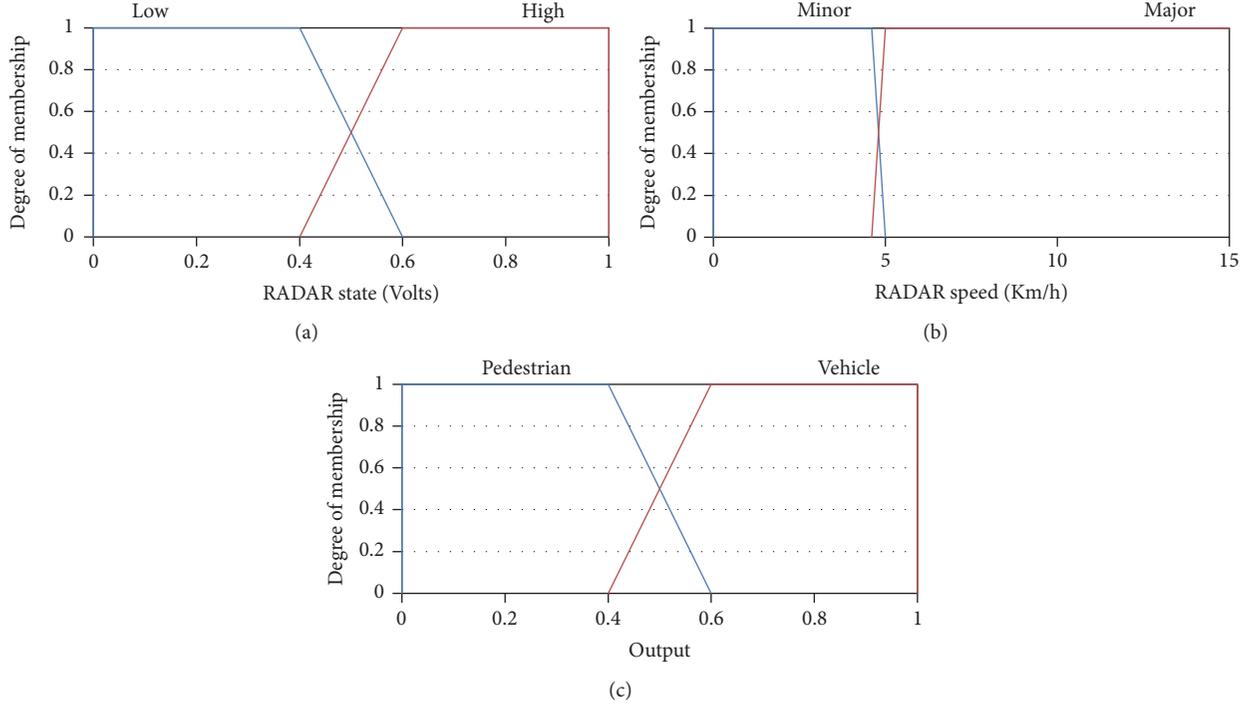


FIGURE 12: Labels for the RADAR fuzzy controller: (a) input for State, (b) input for Speed, and (c) output.

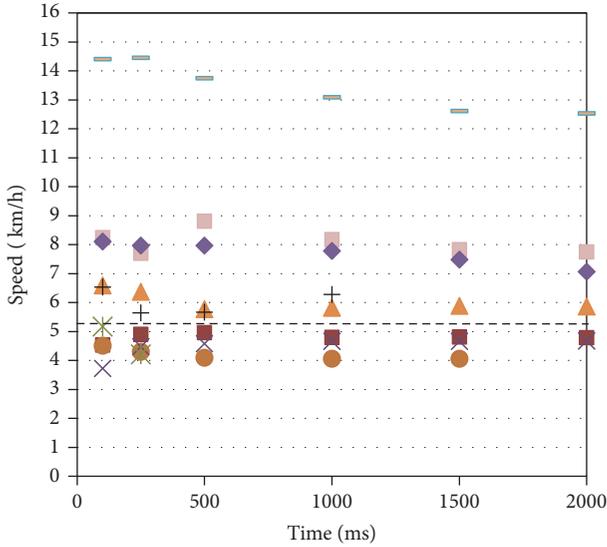


FIGURE 13: Estimation of the speed threshold between pedestrians and vehicles. Red squares, purple cross, and orange circles below the dashed line stand for a pedestrian. The rest are vehicles.

and values near one (1) indicate that the signaling unit must be activated to alert drivers about pedestrians detected on the crosswalk.

The expert system settled that a pedestrian is detected when the output of the ultrasound fuzzy controller indicates *Yes*, the magnetic fuzzy controller points out *No*, the “ T_0 ” variable indicates the *Little* or *Medium* tags, and the output of the RADAR fuzzy controller designates *Pedestrian*. In

TABLE 5: Rule base for the RADAR fuzzy controller.

Rule number	State	Speed	Output
1	Low	Minor	Pedestrian
2	Low	Major	Pedestrian
3	High	Minor	Pedestrian
4	High	Major	Vehicle

this case, it is necessary to activate the LED signaling unit. Otherwise, the pedestrian warning lighting will be disabled (Table 6).

3. Results and Discussion

The experimentation carried out with the prototype system consisted in a total of 240 hours of hardware and software integration, 160 hours of tests in laboratory under controlled conditions, and 65 hours of tests in a real environment. ROC (receiver operating characteristic) analysis, as described in [38], was conducted on the data test achieved from the real scenario to obtain the sensibility against specificity of the system (Table 7). The performance was obtained from a confusion matrix of 2×2 elements that relates positive (p) and negative (n) results. In this way, the sensibility or true positive rate (TPR) can be defined as the success rate as follows:

$$TPR = \frac{TP}{(TP + FN)}, \quad (1)$$

where TP stands for the true positives and FN stands for the false negatives. Anyway, the false positive rate (FPR) or

TABLE 6: Rule base for the whole sensory fusion controller.

Rule number	Ultrasound	T0	Magnetic	RADAR	Output
1	No	Much	No	Pedestrian	Deactivate
2	No	Much	No	Vehicle	Deactivate
3	No	Much	Yes	Pedestrian	Deactivate
4	No	Much	Yes	Vehicle	Deactivate
5	No	Medium	No	Pedestrian	Deactivate
6	No	Medium	No	Vehicle	Deactivate
7	No	Medium	Yes	Pedestrian	Deactivate
8	No	Medium	Yes	Vehicle	Deactivate
9	No	Little	No	Pedestrian	Deactivate
10	No	Little	No	Vehicle	Deactivate
11	No	Little	Yes	Pedestrian	Deactivate
12	No	Little	Yes	Vehicle	Deactivate
13	Yes	Much	No	Pedestrian	Deactivate
14	Yes	Much	No	Vehicle	Deactivate
15	Yes	Much	Yes	Pedestrian	Deactivate
16	Yes	Much	Yes	Vehicle	Deactivate
17	Yes	Medium	No	Pedestrian	Activate
18	Yes	Medium	No	Vehicle	Deactivate
19	Yes	Medium	Yes	Pedestrian	Deactivate
20	Yes	Medium	Yes	Vehicle	Deactivate
21	Yes	Little	No	Pedestrian	Activate
22	Yes	Little	No	Vehicle	Deactivate
23	Yes	Little	Yes	Pedestrian	Deactivate
24	Yes	Little	Yes	Vehicle	Deactivate

TABLE 7: Contingency table for the ROC analysis.

	Actual value		Total
	p	n	
Prediction			
p'	True positives (TP)	False positives (FP)	P'
n'	False negatives (FN)	True negatives (TN)	N'
Total	P	N	

1 – specificity can be defined from Table 7 as the false alarm rate according to the following expression:

$$FPR = \frac{FP}{(FP + TN)}, \quad (2)$$

where FP means false positives and TN means true negatives. Finally, the accuracy (ACC) can be defined as follows:

$$ACC = \frac{(TP + TN)}{(P + N)}, \quad (3)$$

where P and N stand for the total positives and negatives, respectively. The tests involved two types of trials: (i) activation of the LED signaling by pedestrians on the zebra crossing and (ii) inhibition of the LED signaling by vehicle traffic on the road. The first trial included the following categories: one pedestrian, two or more pedestrians walking in the same direction, two or more pedestrians walking in



FIGURE 14: Location of the scenario under study.

opposite directions, a buggy baby, and a bicycle. The second trial included a car, a bicycle, and a motorcycle passing the crosswalk.

To this end, the scenario utilized consisted of a crosswalk sited in “Sector Pp1 Cruz de Montaña” of Bollullos Par del Condado, Huelva, Spain (37.34N, -6.55W). This zebra crossing was selected because it is located on a 180-meter straight track being the speed limited to 30 Km/h (Figure 14). Three prototypes of smart device were installed to cover a crosswalk of 9 meters wide by 5 meters depth. According to (1)–(3), a positive discrimination threshold was established when the object detection was triggered on the first 2/3 of the crosswalk. In other words, a detection on the last 1/3 of the zebra crossing was considered negative in the tests for not achieving the minimum safety requirements for pedestrians.

TABLE 8: Results obtained in the ROC analysis.

Study case	Test	Speed (m/s)	TPR	FPR	ACC	Success (%)
Pedestrian	148	0.88 ± 0.04	0.8133	0	1	81.33
Buggy baby	33	0.84 ± 0.05	0.972	0	1	97.2
Pedestrian group in same direction	17	0.94 ± 0.05	0.95	0	1	95
Pedestrian group in opposite directions	16	1.05 ± 0.04	0.9433	0	1	94.33
Bicycle as pedestrian	13	1.88 ± 0.14	1	0	1	100
Vehicle (i.e., car, bicycle, and motorcycle)	17	≥ 2.22 < 5.55	1	0	1	100
Average	40.66 ± 53.1	1.18 ± 0.43	0.9464	0	1	94.64
Total number of tests	244					

The detection of persons in the direction of the crosswalk consisted in traversing the zebra crossing in different directions: (i) longitudinal movement from right to left and from left to right, meaning as the movement crossing the road from side to side; (ii) transversal movement from back to front and from front to back, meaning as the movement by which a user can enter the crosswalk through any interior point of it; and (iii) diagonal movement from front to back, from right to left and vice versa, meaning as the oblique movement that a user can make to shorten path.

The tests for detecting the vehicle flow over the crosswalk in the direction of the road consisted in inhibiting the lightning barrier and detecting true negatives with a speed higher than 5 Km/h. Recall that this type of vehicles should not activate the LED lighting by themselves when crossing over the system.

The series of tests attained an average speed of 1.18 ± 0.43 m/s for pedestrians and a speed between 2.22 and 5.55 m/s for vehicles. After the experimentation, an average success of 94.64% and a precision of 100% were obtained. According to the ROC analysis, this corresponds to a very good test with $TPR = [0.9, 0.97]$ and lack of false positives. Specifically analyzing the different case studies (Table 8), we found that the best result was obtained for the detection of vehicles in the road direction, as well as bicycles, buggy babies, and group of pedestrians in the crosswalk direction. On the contrary, we found worse results in both the detection of a person and several people crossing simultaneously in opposite directions. On the one hand, this suggests that the greater the volume of the objects to be detected, the more efficient the intelligent road signaling system (e.g., bicycles or buggy babies versus a person). This is explained due to the collocation of the sensors at the asphalt level, which reduces the effectiveness when detecting low limbs from persons compared to the chest or other parts of the body with major size. On the other hand, the study suggests that bicycles and motorcycles passing the crosswalk are not a trouble for the type of sensors used, both the RADAR and the ultrasound. To sum up, Figure 15 shows a comparative graphic with the case studies addressed and their TPR values.

4. Conclusions

According to studies, 40% of the accidents in which pedestrians are involved are produced when crossing for the right

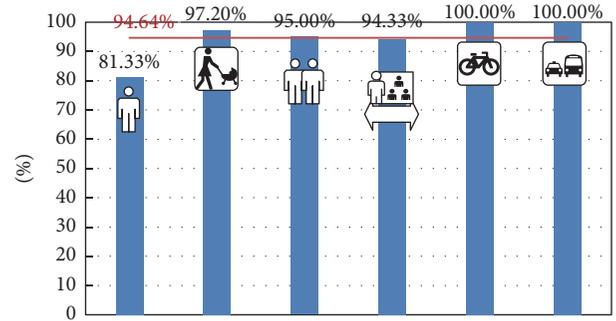


FIGURE 15: TPR of the classifier modelled by the fuzzy controller.

place. This happens—in part—because crosswalks are not 100% visible for drivers due to different reasons: (i) poor road maintenance, (ii) occlusion of vertical signs, and/or (iii) adverse weather conditions, among others.

To help reduce accidents, this paper proposed a prototype signaling system set on the road that alerts drivers when they are approaching zebra crossings when a pedestrian is traversing. This system, formed by a set of autonomous, intelligent, and low-cost devices, implements an object detection and alerts fuzzy controller that activates luminous signals so that drivers can stop safely.

A study on the state of the art about several patents and commercial solutions—including luminous road markers, speed bumps, and vertical signs—revealed that the main innovative features of the prototype system are the artificial intelligence and low-cost installation. On the one hand, the intelligence allows contradistinguishing whether an obstacle traversing a crosswalk is pedestrian or vehicle, thus interacting actively with the environment. On the other hand, the installation is based on a low-cost fitting that removes the need for public works on the road since the autonomous device includes solar-based rechargeable technology instead of a power grid infrastructure.

The tests conducted to validate the system over a total of 65 hours consisted in (i) detecting different types of users walking over a crosswalk (i.e., individual pedestrians or grouped, buggy babies, and bicycles) and (ii) trying the intelligent discrimination of vehicles to avoid false positives (i.e., cars, bicycles, and motorcycles). An exhaustive ROC analysis with a total of 244 tests provided an average success

of 94.64% and a precision of 100%. This suggests a very good test in global terms, while the greater the volume of the object to be detected, the better the detection (e.g., bicycles or buggy babies versus a single person). By way of conclusion, a video with the construction and validation of the prototype is available in the web page of the project at <http://www.uhu.es/tomas.mateo/smartycitysen>.

Future works are focused to improve several technical and functional aspects of the prototype, including size, power consumption, electronics, and code. Regarding the hardware, the energetic deficit supported by the system can be still considered somewhat high. Incorporating elements of lower consumption and/or inducing the control unit to sleep periods without affecting the detection capacity (e.g., modem-sleep, light-sleep, or deep-sleep) can reduce the number of PV (photovoltaic) panels at the same time that reduces the case size. Regarding the software, the upgrade to more advanced control techniques would improve the system accuracy and reliability. On the one hand, the object detection would be enhanced by means of pattern analysis and recognition based on FFT (fast Fourier transform) techniques. On the other hand, the implementation of machine learning techniques (e.g., based on genetic algorithms) would automate the calibration of the system variables—and their translation into diffuse linguistic labels to form the knowledge base—regardless of the system installation (i.e., road conditions). Finally, the developed prototype could also be improved by extending its functionalities to enhance the safety of dependent people (e.g., acoustic signaling for blind people).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors are very grateful to the MAPFRE Foundation for financing this research (Ref. BIL/15/P3/047) thanks to the grant “Ignacio H. de Larramendi.” They also express their gratitude to M. A. Rodríguez Román, M. Redondo González, J. M. Enrique Gómez, and E. Durán Aranda for the valuable help given to this project.

Supplementary Materials

The supplementary material includes a video that shows the prototype construction process, the tests to which the devices have been subjected, and the results obtained. (*Supplementary Materials*)

References

- [1] RACE-GOODYEAR, “Atropello de Peatones en Zona Urbana,” Tech. Rep., 2013.
- [2] Centro de Estudios Ponle Freno-AXA, “Atropello a Peatones 2014,” Tech. Rep., 2014, <http://www.antena3.com/a3document/2015/03/06/DOCUMENTS/00803/00803.pdf>.
- [3] Centro de Investigaciones Sociológicas, “Barómetro de Mayo 2016 - Avance de Resultados,” Tech. Rep., 2016.
- [4] National Highway Traffic Safety Administration, “Traffic Safety Facts 2015 Data – Pedestrians,” Tech. Rep., Department of Transportation, National Highway Traffic Safety Administration, Washington, DC, USA, 2017, <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812375>.
- [5] F. Bella and M. Silvestri, “Effects of safety measures on driver’s speed behavior at pedestrian crossings,” *Accident Analysis & Prevention*, vol. 83, pp. 111–124, 2015.
- [6] J.-L. Martin and D. Wu, “Pedestrian fatality and impact speed squared: Cloglog modeling from French national data,” *Traffic Injury Prevention*, pp. 1–8, 2017.
- [7] S. Suzuki, P. Raksincharoensak, I. Shimizu, M. Nagai, and R. Adomat, “Sensor fusion-based pedestrian collision warning system with crosswalk detection,” in *Proceedings of the 2010 IEEE Intelligent Vehicles Symposium, IV 2010*, pp. 355–360, USA, June 2010.
- [8] L. Moya, “Ford Añadirá Detección de Peatones a sus Coches,” Tech. Rep., The MIT Technology Review, 2014, <https://www.technologyreview.es/s/4539/ford-anadira-deteccion-de-peatones-sus-coches>.
- [9] D. Olmeda, C. Premebida, U. Nunes, J. M. Armingol, and A. De La Escalera, “Pedestrian detection in far infrared images,” *Integrated Computer-Aided Engineering*, vol. 20, no. 4, pp. 347–360, 2013.
- [10] J. Ortiz Carranza, “Señalizador para Paso de Cebra,” *OEPM*, Article ID ES1034252U, 1996.
- [11] J. Bonallach Aguado and I. Casanovas Viladomiu, “Casanovas Viladomiu, I. Disposición de Señalización para Tramos de Vía Pública,” *OEPM*, Article ID ES1060684U, 2005.
- [12] A. Carballa Figueras, “Paso de Peatones con Señalización Luminosa,” *OEPM*, Article ID ES2321794U, 2009.
- [13] J. Portoles Ibañes, E. Algora Sebastia, A. Benavente Piote, and E. Ballester Berna, “Vado Peatonal Cerámico Inteligente,” *OEPM*, Article ID ES1072508Y, 2010.
- [14] J. M. Pelayo Sanchez, “Illuminated Signaling Device for Pedestrian Crossings,” *OEPM*, Article ID ES1067629-U, 2008.
- [15] A. Pozo Macho, “Dispositivo luminoso que alerta a los conductores de peatones cruzando por pasos de cebra,” *OEPM*, Article ID S2311398, p. A1, 2009.
- [16] B. Siuru, “A Super-safe Smart Crosswalk,” *Electronics Now*, vol. 70, no. 2, pp. 48–86, 1999.
- [17] G. Matrella and D. Marani, “An embedded video sensor for a smart traffic light,” in *Proceedings of the 2011 14th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, DSD 2011*, pp. 769–776, Finland, September 2011.
- [18] A. A. Ballestín, “Philips Crea un Paso de Peatones LED que Avisa de la Presencia de Vehículos Silenciosos,” Tech. Rep., 2010, <http://es.engadget.com>.
- [19] Co. Kg. RTB GmbH, “Device for protecting pedestrian crossing,” *Google Patents*, Article ID E202004006444, 2004.
- [20] D. Robin Green, J. Ward, and N. Wyper, *Solar-powered wireless crosswalk warning system*, Google Patents, 2008.
- [21] J. M. Lozano Domínguez, T. J. Mateo Sanguino, and M. A. Rodríguez Román, “Autonomous Detection and Signaling Crosswalk System,” *OEPM*, Article ID P201600768, 2016.
- [22] IEEE Computer Society, “IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access

- Control (MAC) and Physical Layer (PHY) Specifications,” Tech. Rep., 2016.
- [23] K. Kapitanova, S. H. Son, and K. Kang, “Using fuzzy logic for robust event detection in wireless sensor networks,” *Ad Hoc Networks*, vol. 10, no. 4, pp. 709–722, 2012.
- [24] J. M. Andújar and A. J. Barragán, “Hybridization of fuzzy systems for modeling and control,” *RIAI - Revista Iberoamericana de Automática e Informática Industrial*, vol. 11, no. 2, pp. 127–141, 2014.
- [25] A. H. Adnan, M. Abdirazak, A. B. M. S. Sadi et al., “A comparative study of WLAN security protocols: WPA, WPA2,” in *Proceedings of the 3rd International Conference on Advances in Electrical Engineering, ICAEE 2015*, pp. 165–169, Bangladesh, December 2015.
- [26] M. Masirap, M. H. Amaran, Y. M. Yussoff, R. A. Rahman, and H. Hashim, “Evaluation of reliable UDP-based transport protocols for Internet of Things (IoT),” in *Proceedings of the 2016 IEEE Symposium on Computer Applications and Industrial Electronics, ISCAIE 2016*, pp. 200–205, mys, May 2016.
- [27] A. Gyrard, S. K. Datta, C. Bonnet, and K. Boudaoud, “Standardizing generic cross-domain applications in Internet of Things,” in *Proceedings of the 2014 IEEE Globecom Workshops, GC Wkshps 2014*, pp. 589–594, USA, December 2014.
- [28] A. Gyrard, S. K. Datta, C. Bonnet, and K. Boudaoud, “Cross-domain internet of things application development: m3 framework and evaluation,” in *Proceedings of the 3rd International Conference on Future Internet of Things and Cloud, FiCloud 2015*, pp. 9–16, Italy, August 2015.
- [29] Ministerio de Fomento, “Guía para el proyecto y ejecución de obras de señalización horizontal,” Tech. Rep., Secretaría General Técnica, 2012.
- [30] B. V. Dasarathy, “Sensor fusion potential exploitation-innovative architectures and illustrative applications,” *Proceedings of the IEEE*, vol. 85, no. 1, pp. 24–38, 1997.
- [31] E. H. Mamdani, *Applications of fuzzy set theory to control systems: a survey*, *Fuzzy Automata and Decision Processes*, 1997.
- [32] B. Sugiarto, “The use of fuzzy logic for data classification in sensor node on wireless sensor and actuator network (WSAN) system,” in *Proceedings of the 2014 Makassar International Conference on Electrical Engineering and Informatics, MICEEI 2014*, pp. 109–113, Makassar, Indonesia, November 2014.
- [33] S. A. U. R. Omer and E. Muhammad, “Design of intelligent air conditioner controller using fuzzy logic,” in *Proceedings of the 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies, ICIEECT 2017*, Karachi, Pakistan, April 2017.
- [34] R. A. Aliev, “Fuzzy Sets and Fuzzy Logic,” in *Fundamentals of the Fuzzy Logic-Based Generalized Theory of Decisions*, pp. 1–58, Springer-Verlag, Berlin, Germany, 2013.
- [35] A. M. Ibrahim, “Embedded fuzzy applications,” in *Fuzzy Logic for Embedded System Applications*, pp. 69–98, Elsevier Inc, 2004.
- [36] J. J. Buckley and Y. Hayashi, “Can approximate reasoning be consistent?” *Fuzzy Sets and Systems*, vol. 65, no. 1, pp. 13–18, 1994.
- [37] H. R. Jayatileke, W. R. De Mel, and H. U. W. Ratnayake, “Real-time fuzzy logic speed tracking controller for a DC motor using Arduino Due,” in *Proceedings of the 2014 7th International Conference on Information and Automation for Sustainability, ICIAfS 2014*, Sri Lanka, December 2014.
- [38] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

Research Article

Neural Models for Imputation of Missing Ozone Data in Air-Quality Datasets

Ángel Arroyo ¹, Álvaro Herrero,¹ Verónica Tricio,²
Emilio Corchado,³ and Michał Woźniak⁴

¹Department of Civil Engineering, University of Burgos, Burgos, Spain

²Department of Physics, University of Burgos, Burgos, Spain

³Departamento de Informática y Automática, University of Salamanca, Salamanca, Spain

⁴Department of Systems and Computer Networks, Wrocław University of Science and Technology, Wrocław, Poland

Correspondence should be addressed to Ángel Arroyo; aarroyop@ubu.es

Received 5 December 2017; Accepted 31 January 2018; Published 8 March 2018

Academic Editor: Eloy Irigoyen

Copyright © 2018 Ángel Arroyo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ozone is one of the pollutants with most negative effects on human health and in general on the biosphere. Many data-acquisition networks collect data about ozone values in both urban and background areas. Usually, these data are incomplete or corrupt and the imputation of the missing values is a priority in order to obtain complete datasets, solving the uncertainty and vagueness of existing problems to manage complexity. In the present paper, multiple-regression techniques and Artificial Neural Network models are applied to approximate the absent ozone values from five explanatory variables containing air-quality information. To compare the different imputation methods, real-life data from six data-acquisition stations from the region of Castilla y León (Spain) are gathered in different ways and then analyzed. The results obtained in the estimation of the missing values by applying these techniques and models are compared, analyzing the possible causes of the given response.

1. Introduction and Related Work

The ozone (O_3) is an odorless, colorless, and highly reactive gas composed of three oxygen atoms. It is formed both in the Earth's upper atmosphere (stratospheric ozone) and at ground level (tropospheric ozone). It can be “good” or “bad” for people's health and for the environment, depending on its concentration levels and location in the atmosphere [1].

Stratospheric O_3 is formed naturally through the interaction of solar UltraViolet (UV) radiation with molecular oxygen (O_2). Ground-level or “bad” ozone is not emitted directly into the air. In the 1950s, hydrocarbons and nitrogen oxides (NO_x) were identified as the two key chemical precursors of photochemical smog and its concomitant high concentrations of O_3 and other photochemical oxidant [2]. The majority of ground-level O_3 is formed from the photochemical oxidation of Volatile Organic Compounds

(VOCs) in the presence of NO and other NO_x . Significant sources of VOCs are chemical plants, gasoline pumps, oil-based paints, autobody shops, and print shops. NO_x result primarily from high temperature combustion, and its most significant sources are power plants, industrial furnaces and boilers, and motor vehicles [3].

1.1. Importance of Ozone. The O_3 exposition can cause damage in different ways. In the stratosphere, reduced O_3 levels as a result of O_3 layer depletion mean less protection from the sun's rays and more exposure to UltraViolet B (shortwave) rays (UVB) radiation at the Earth's surface [4]. The effects on human health of the O_3 layer depletion have been much analyzed, increasing the amount of UVB that reaches the Earth's surface. UVB causes nonmelanoma skin cancer and plays a major role in malignant melanoma development. In addition, UVB has been linked to the development of certain cataracts, negative effects in patients with asthma, and

other chronic respiratory disease. With respect to ground-level O_3 , and its effects on human health, breathing O_3 can trigger a variety of health problems. People with asthma and other chronic respiratory disease are a large and growing segment of the population and are also known to be especially susceptible to the effects of O_3 exposure. On days with high levels of O_3 , people with asthma tend to experience increased respiratory symptoms [3]. The layer O_3 depletion has also negative effects on the process of the development of plants, effects on the marine ecosystems like a direct reduction in phytoplankton production, negative effects on materials like biopolymers, and so forth. Tropospheric O_3 does not provide the protective function that it fulfills in the stratosphere, being high reactivity. Its strong oxidizing capacity, when its levels rise above the natural background, can cause adverse effects in materials (derived from its corrosive effects), on vegetation and ecosystems.

The present work focuses on tropospheric O_3 , which is a risk for the air quality [3]. Given the increase in O_3 levels in the troposphere, it is currently considered one of the most important atmospheric pollutants.

1.2. Ozone Level Monitoring. Around the world there are numerous data-acquisition networks for the measurement of O_3 levels and other pollutants, which consist of many stations in different locations where different sensors measure corresponding magnitudes. These network stations acquire data at periodic intervals of time (periods between ten and fifteen minutes are the most frequent ones) but frequently appear missing or corrupted data. In Europe, data are considered as corrupted when not meeting the Council Decision 97/101/EC of January 27, 1997 [5], which establish a reciprocal exchange of information and data from networks and individual stations measuring ambient air pollution within the Member States. Some of these networks provide information about the validity of the data, indicating through codes if the data is correct, it has not been possible to acquire, or it is corrupt, but in other occasions this type of information is not provided while the data are still missing. Some reasons for such failures have been pinpointed [6], namely, a damaged cable, the loss of proper electrical grounding, half-melted frost or snow on the dome, communications failure, and so forth. Some of these causes are temporary and may disappear spontaneously, but other ones require the intervention of a maintenance task force, and therefore errors persist for different periods of time. The absence of valid data may also be due to reasons such as the following: mishandling of samples, low signal-to-noise ratio, measurement error, nonresponse, or deleted aberrant value [7]. This is a problem for the analysis of the information coming from the measurement networks, and the imputation of these missing data [8] is necessary. Any of the variables acquired in network stations may suffer from the problem of the absence of data. If many data variables are omitted or corrupted in the same record, the whole sample must be withdrawn, when some models are applied [9], for subsequent tasks such as control, classification, forecast. Alternatively, if data for the same pollutant are missing in several adjacent rows, removing that variable may also be an alternative solution. In conclusion, having a complete set of

data is necessary to perform a reliable study and to apply some models that cannot deal with missing data.

1.3. Missing Values and Related Work. The standard classification of missing data phenomenon [10] includes different situations:

- (i) Missing Completely At Random (MCAR), when the probability of an instance (case) having a missing value for a variable does not depend on either the known values or the missing data.
- (ii) Missing At Random (MAR), when the probability of an instance having a missing value for a variable may depend on the known values but not on the value of the missing data itself.
- (iii) Not Missing At Random (NMAR), when the probability of an instance having a missing value for a variable could depend on the value of that variable.

As previous authors have pointed out, the complexity varies between these patterns of missing data [11]. Usually, in the case of air-quality data, missing values are associated with MAR or MCAR. The circumstances that may interfere with the acquisition of the data are many and not easily predictable [12].

To solve the missing data problem, a wide variety of different methods have been applied up to now [8, 10, 13]. These imputation methods (IMs) are usually classified as follows:

- (i) Single imputation (SI): the method fills in one value for each missing one [12].
- (ii) Multiple imputation (MI): multiple simulated values are generated at the same time [14].

The univariate and multivariate imputation methods differ in which the approximation of the missing values of the variable under study are calculated from the rest of the values of the very same variable (univariate) or using values of the rest of the variables (multivariate) [12].

With the aim of reducing the complexity of other MI applied methods [11], the present paper focuses on single and multivariate imputation for the O_3 magnitude in air pollution datasets. To do so, multiple-regression (linear and nonlinear) techniques together with Artificial Neural Networks (ANN) are applied to real-life datasets obtained from public air-quality networks.

Up to now, different Artificial-Intelligence (AI) techniques have been applied for imputation of missing data. In [7] imputation methods based on six different techniques are compared: *K*-Nearest Neighbors (KNN), Fuzzy *K*-Means (FKM), Singular Value Decomposition, Bayesian Principal Component Analysis (bPCA) and Multiple Imputations by Chained Equations. These methods are applied to four datasets split into two groups of various sizes: small datasets (Iris and *E. coli*) and large datasets (breast cancers 1 and 2). bPCA and FKM appeared to be the most robust imputation methods in the tested conditions.

In [15] the accuracy of different imputation methods is evaluated: MissForest (MF) and Multiple Imputation based

on Expectation-Maximization (MIEM), along with two other imputation methods: Sequential Hot-Deck and Multiple Imputation based on Logistic Regression (MILR). The models are applied over fourteen binary datasets, with a range of missing data rates between 5% and 50%. The results from 10-fold Cross-Validation (CV) show that the performance of the imputation methods varies substantially between different classifiers and at different rates of missing values.

Although many imputation methods have been proposed up to now, scant attention has been paid to validate ANN for such a task, taking advantage of their regression capability [16]. Among these previous studies, ANN have been applied for the estimation of lost values in [17], where the main goal is identifying Learning Disabilities (LD) in children at early stages. In [18], authors proposed a SI approach relying on a Multilayer Perceptron (MLP) whose training is conducted with different learning rules, and a MI approach based on the combination of MLP and KNN. 24 real and simulated datasets from the UCI repository, the Promise repository, and mldata.org were exposed to a perturbation experiment with random generation of monotone missing data pattern.

In [19] six different types of ANN are proposed as IM: MLP and its variations (the Time-Lagged Feedforward Network (TLFN)), the Generalized Radial-Basis-Function (GRBF) network, the Recurrent Neural Network (RNN), and its variations (the Time Delay Recurrent Neural Network (TDRNN)). Additionally, the Counterpropagation Fuzzy-Neural Network (CFNN) along with different optimization methods is applied for infilling missing daily total precipitation and extreme temperature series from 15 weather stations. The standard MLP and TLFN appear to provide the most accurate reconstruction of missing precipitation and daily extreme temperatures records with results for the R correlation coefficient between the observed and the reconstructed daily series close to 1.

In [20] a novel nonparametric algorithm named Generalized regression neural network Ensemble for Multiple Imputation (GEMI) is proposed. Additionally, a SI version of this approach (GESI) is proposed. The algorithms were tested on 98 synthetic and real-world datasets. All simulation results show the advantages of GEMI as compared with conventional algorithms. GEMI has heavy memory storage requirements but outperformed other SI algorithms.

In [21] fifteen real and simulated datasets are exposed to a perturbation experiment, based on the random generation of missing values. Several architectures and learning algorithms for the MLP are tested and compared with three classic imputation procedures: mean/mode imputation, regression, and hot-deck [22].

In [23] a methodology based on Gaussian Mixture Model (GMM) and Extreme Learning Machine (ELM) is developed and tested on some datasets from the UCI Machine Learning Repository and the LIACC regression repository. GMM is used to model the data distribution which is adapted to handle missing values, while ELM enables devising a Multiple Imputation strategy for final estimation. The combination of GMM and ELM is shown to be superior in almost all tested cases over the method based on conditional mean imputation.

In [24] a SI approach relying on a MLP and a MI approach based on the combination of MLP and K -NN is proposed. The models are applied to 18 real and simulated datasets like domains such as biology, medicine, chemistry, electronics, social surveys, census, and business. For datasets with only quantitative variables MIMLP model provided the best results, with IMLP being the best method for datasets with categorical variables.

In [25] a two-stage hybrid model for filling the missing values using fuzzy c -means clustering and MLP is proposed. It is applied to a Wine dataset with a 1% to 5% of generated missing values and the accuracy of the model is checked using the Mean Absolute Percentage Error (MAPE). The MAPE obtained for stage 2 (MLP regression to the obtained dataset as a result of applying fuzzy c -means in stage 1) is 4.95% for 1% missing-value records and 8.36% for 5% missing-value records.

In the case of air-quality data, few imputation methods have been proposed up to now. In [13], an important set of SI: Listwise, Unconditional mean, Modified Median, Principal Component-based, Expectation-Maximization (EM) (Regularized-EM), and MI methods are applied to three datasets with the most important pollutant variables (NO , NO_2 , NO_x , CO , O_3 , PM_{10} , and $\text{PM}_{2.5}$) and a percentage of missing data among the 3.85% and the 23.52% depending on the year. Missing data of the eight variables are imputed in order to assess the effectiveness of the methods applied. In general, MI tends to yield more scattered values than its counterparts, mainly when the variables have many voids and they correlate poorly to the other variables like CO with 43.5% of missing data in 2006 and they correlate poorly to the other variables.

In [11] some methods for the imputation of missing air-quality data are compared: in the context of SI (linear, spline, and nearest neighbor interpolations), MI (regression-based imputation, multivariate nearest neighbor, Self-Organizing Maps (SOM), and Multilayer Backpropagation (MLBP) nets) and hybrid methods of the aforementioned. The dataset uses the most common pollutants: NO_x , NO_2 , O_3 , PM_{10} , SO_2 , and CO concentrations, all on a time-scale of one per hour (hourly averaged), together with four meteorological parameters. The performance of the proposed univariate missing data interpolation was limited, and in general they were able to fill only very short gaps of contiguous missing data. The general performance of the applied imputation methods was fair good when considering the pollutants (NO_x , NO_2 , O_3 , PM_{10} , SO_2 , and CO) which are the most important ones in terms of air-quality modelling, but not so good regarding meteorological variables. The results suggested that SOM and MLBP are the methods of choice for air-quality data imputation and even better results can be achieved by using the MI.

1.4. Main Contributions. The main contributions of this work are as follows:

- (i) Deep study of the real-life human health protection task in Spanish region of Castilla y León.
- (ii) Multisensor of O_3 data analysis.

- (iii) Experimental evaluation of the proposed approach based on multiple-regression techniques together with ANN models.

To the best of authors knowledge, this is the first approach of imputation methods of O_3 based on both MLP and Radial-Basis-Function Networks.

The rest of this paper is organized as follows. Section 2 presents the techniques and models applied. Section 3 details the real-life case study that is addressed in present work, while Section 4 describes the experiments and results. Finally, Section 5 sets out the main conclusions and future work.

2. Regression Techniques and ANN Models

In order to fill missing or corrupted values of O_3 in high dimensional datasets with air-quality information, two regression techniques and two ANN models have been applied in present study. This set of techniques applied as imputation methods is described in this section.

2.1. Regression Techniques. Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable [26].

The general purpose of multiple regressions [27] is to learn more about the relationship between several independent or predictor variables and a dependent or criterion variable.

2.1.1. Multiple Linear Regression. Multiple linear regression (MLR) attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data [28]. Every value of the independent variable (x) is associated with a value of the dependent variable (y). The population regression line for p explanatory variables

$$x_1, x_2, \dots, x_p \quad (1)$$

is defined to be

$$u_y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p. \quad (2)$$

This line describes how the mean response u_y changes with the explanatory variables. The observed values for y vary about their means u_y , and are assumed to have the same standard deviation σ . The fitted values b_0, b_1, \dots, b_p estimate the parameters $\beta_0, \beta_1, \dots, \beta_p$ of the population regression line.

Since the observed values for y vary about their means u_y , the multiple-regression models include a term for this variation. The model is expressed as DATA = FIT + RESIDUAL, where the ‘‘FIT’’ term represents the expression $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$. The ‘‘RESIDUAL’’ term represents the deviations of the observed values y from their means u_y , which are normally distributed with mean 0 and variance σ . The notation for the model deviations is ε .

Formally, the model for multiple linear regression, given n observations, is [28]

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \varepsilon_i \quad (3)$$

for $i = 1, 2, \dots, n$.

2.1.2. Multiple Nonlinear Regression. A Multiple Nonlinear Regression (MN-LR) is a form of regression analysis in which observational data are modelled by a function which is a nonlinear combination of the model parameters and depends on one or more independent variables [29]. The data are fitted by a method of successive approximations.

The parameters can take the form of an exponential, trigonometric, power, or any other nonlinear function. To determine the nonlinear parameter estimates, an iterative algorithm is typically used.

$$y = f(X, B) + \varepsilon, \quad (4)$$

where B represents nonlinear parameter estimates to be computed, X is the dependent or criterion variables, and ε represents the error terms.

2.2. Artificial Neural Networks. Artificial Neural Networks (ANN), also known as Artificial Neural Systems (ANS), connectionist systems, adaptive networks, and distributed and parallel processing are simplified models of natural neural systems. The following definition, given by Hecht-Nielsen in 1989 [30], formalizes the concept of ANN:

An ANN is a parallel processing computer system distributed, consisting of a set of elementary processing units equipped with a small local memory and interconnected in a network through connections with associated weights. Each processing unit has one or more input connections and a single output connection that links to many collateral connections as desired. All processing associated with an elementary unit is a local, i.e. depends only on the values that take input signals from the unit and the internal state of the same.

2.2.1. Multilayer Perceptron (MLP). The MLP consists of a system of simple interconnected neurons or nodes. The nodes are connected by weights and output signals which are a function of the sum of the inputs to the node modified by a simple nonlinear transfer, or activation, function. The architecture consists of several layers of neurons; the input layer serves to pass the input vector to the network. The terms ‘‘input vectors’’ and ‘‘output vectors’’ refer to the inputs and outputs of the MLP and can be represented as single vectors [31]. A MLP may have one or more hidden layers and finally an output layer. MLP are fully connected, with each node connected to every node in the next and previous layer.

To perform a comprehensive comparison, the MLP is trained with the following algorithms:

- (1) Levenberg-Marquardt backpropagation (LM)

- (2) Gradient Descent with momentum and adaptive learning rate backpropagation (GDX) [32]
- (3) Batch Training with weight and bias learning rules (TB)
- (4) Scaled Conjugate Gradient backpropagation (SCG)
- (5) Bayesian Regularization backpropagation (BR).

2.2.2. Radial-Basis-Function Networks (RBFN). In a RBFN [33] each unit in the hidden layer of this network has its own centroid, and, for each input vector $x = (x_1, x_2, \dots, x_n)$, it computes the distance between x and its centroid. Its output of the unit is calculated as a nonlinear function of this distance.

Assuming that there are r input nodes and m output nodes, the overall response function without considering nonlinearity in an output node has the following form [34]:

$$\sum_{i=1}^M W_i * K\left(\frac{x - z_i}{\sigma_i}\right) = \sum_{i=1}^M W_i * g\left(\frac{\|x - z_i\|}{\sigma_i}\right), \quad (5)$$

where $M \in \mathbb{N}$ is the number of units in the hidden layer, $W_i \in \mathbb{R}^m$ is the vector of weights linking the i th hidden-layer unit to the output nodes, x is an input vector, K is a radially symmetric kernel function of a unit in the hidden layer, z_i and σ_i are the centroid and smoothing factor of the i th kernel node, respectively, and $g: [0, \infty) \rightarrow \mathbb{R}$ is a function called the activation function, which characterizes the kernel shape.

3. Case Study

In present study, data from air-quality stations in Castilla y León (CyL) are analyzed. CyL is a Spanish region located at the north-center of the Iberian Peninsula. It is composed of nine provinces and it is the most extensive region of Spain with a total surface of 94,226 square kilometers and the sixth with more population: 2,435,797 habitants. Gross Domestic Product (GDP) in CyL represents the 5.3% of country's GDP [35]. Climate in CyL approaches what is known as the continental ocean, characterized by cold winters and hot summers with short spring and autumn periods.

CyL region provides a wide network of stations [36] for the acquisition of air-quality data. These data are public available according to the Open Data Initiative from the Spanish Government [37].

Stations from this network have some interesting characteristics:

- (1) Stations are classified in types: urban, background, and oriented to the vegetation protection [36].
- (2) These stations collect the fundamental air-quality pollutants, and among them is the O_3 , which is the objective pollutant of this study. Daily averages data [38] of each pollutant are provided in each location.
- (3) This data presents empty or corrupted data in all of its variables in some rows and in a reasonable percentage to be estimated.

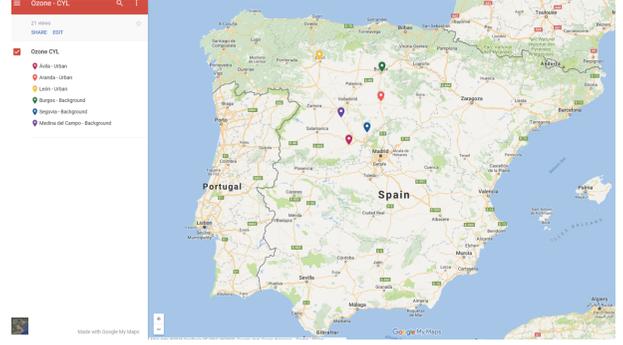


FIGURE 1: Location of the six selected stations in CyL, by Google Maps.

In the present study, pollutant data recorded in six different stations from the CyL network are analyzed. Daily data averages from years 2000 to 2008 have been selected. For some periods of time within the selected time window, data are not available for all the variables and, thus, the whole example is rejected for the study. Three of the stations are located in the center of the cities and labeled as urban stations; these stations are oriented to the protection of the human health. The other three stations are background stations and are also oriented to the protection of the human health. These stations measure a greater number of pollutants than the other type of stations and are the most important ones in terms of air quality, and many of them are not collected at the stations for the vegetation protection. This fact is important for the determination of the O_3 missing values, as this gas is especially harmful for human health.

The three urban stations considered in present study are as follows:

- (1) Ávila. “Bus Station” station. Geographical coordinates: 40.65914, -4.68237 ; 1150 meters above sea level (masl).
- (2) Aranda de Duero. “Jardines de Don Diego” station. Geographical coordinates: 41.67111, -3.68388 ; 801 masl.
- (3) León. “Avda. San Ignacio de Loyola” station. Geographical coordinates: 42.60388, -5.58722 ; 838 masl.

The three background stations are as follows:

- (1) Burgos. “Fuentes Blancas” station. Geographical coordinates: 42.33611, -3.63611 ; 929 masl.
- (2) Segovia. “Acueducto” station. Geographical coordinates: 40.95555, -4.11055 ; 951 masl.
- (3) Medina del Campo (Valladolid). “Bus Station” station. Geographical coordinates: 41.31638, -4.90916 ; 721 masl.

Figure 1 shows the location of the six selected stations that have been studied in the present paper.

The pollutants gathered in the above-mentioned stations and analyzed in the present study are as follows:

TABLE 1: Correlation matrix of the six variables in the dataset.

	O ₃	CO	NO	NO ₂	PM10	SO ₂
O ₃	1.000	-0.123	-0.161	-0.202	0.072	-0.013
CO	-0.123	1.000	0.360	0.412	0.358	0.299
NO	-0.161	0.360	1.000	0.540	0.233	0.330
NO ₂	-0.202	0.412	0.540	1.000	0.330	0.257
PM10	0.072	0.358	0.233	0.330	1.000	0.251
SO ₂	-0.013	0.299	0.330	0.257	0.251	1.000

TABLE 2: Percentage of missing and corrupted data for each one of the analyzed variables.

	O ₃	NO	NO ₂	CO	PM10	SO ₂
Missing	8.104%	8.020%	8.034%	8.554%	9.131%	8.196%
Corrupted	1.857%	2.047%	1.815%	2.926%	2.413%	1.801%
Total	9.961%	10.067%	9.849%	11.480%	11.544%	9.997%

- (1) Ozone (O₃), $\mu\text{g}/\text{m}^3$, secondary pollutant. See Section 1.
- (2) Carbon monoxide (CO), mg/m^3 , primary pollutant. It is an odorless, colorless gas formed by the incomplete combustion of fuels. When people are exposed to CO gas, the CO molecules will displace the oxygen in their bodies and lead to poisoning [39].
- (3) Nitric oxide (NO), $\mu\text{g}/\text{m}^3$, primary pollutant. NO is a colorless gas which reacts with ozone undergoing rapid oxidation to NO₂, predominant in the atmosphere [39].
- (4) Nitrogen dioxide (NO₂), $\mu\text{g}/\text{m}^3$, primary pollutant. From the standpoint of health protection, nitrogen dioxide has set exposure limits for long and short duration [39].
- (5) Particulate matter (PM10), $\mu\text{g}/\text{m}^3$, primary pollutant. These particles remain stable in the air for long periods of time without falling to the ground and can be moved significant distances by the wind. It is defined by the ISO as follows: “particles which pass through a size-selective inlet with a 50% efficiency cut-off at 10 μm aerodynamic diameter. PM10 corresponds to the ‘thoracic convention’ as defined in ISO 7708:1995, Clause 6” [40].
- (6) Sulphur dioxide (SO₂), $\mu\text{g}/\text{m}^3$, primary pollutant. It is a gas. It smells like burnt matches. Its smell is also suffocating. SO₂ is produced by volcanoes and in various industrial processes. In the food industry, it is also used to protect wine from oxygen and bacteria [39].

Primary pollutants are injected into the atmosphere directly. Secondary pollutants are formed in the atmosphere through chemical and photochemical reactions from the primary pollutants [36].

All data from these six variables were normalized for the study. On the other hand, all of them are highly decorrelated. Table 1 shows the correlation matrix of the six pollutants of the case study.

It is worth mentioning that O₃ is the most independent pollutant, as its correlation coefficients with the rest of the variables are close to zero.

There are a total of 13,526 samples, as one sample per day (daily average) was collected for the twelve months of every year, between years 2000 and 2008, in the six stations analyzed in this study. Missing or corrupted data appear in all the variables in some rows, which are omitted for the study.

Table 2 shows the percentage of missing or corrupted data presented in each variable in the whole dataset.

All the samples with at least one missing or corrupted value were removed from the dataset.

4. Experiments, Results, and Discussion

The main target of this paper is to fill missing O₃ values in air pollution datasets. To do so, several imputation methods are comprehensively compared as described below.

4.1. Experimental Settings. The imputation methods described in Section 2 are applied to different datasets, all of them with the six variables described in Section 3:

- (1) The Whole Dataset (WD), comprising the 13,526 samples: results for this datasets are shown in Section 4.2.
- (2) The Season Dataset (SD): samples in WD are split in four subsets according to the four seasons of the year: spring (3,453 samples), summer (3,349 samples), autumn (3,295 samples), and winter (3,429 samples). Results for this dataset are shown in Section 4.3.
- (3) The Type station Dataset (TD): samples in WD are split into two subsets according to the type of the station where the data come from; “urban” (6,763 samples) or “background” (6,763 samples). Results for this datasets are shown in Section 4.4.

For the three datasets, both statistical and neural imputation methods were applied and the performance is calculated through n -fold Cross-Validation (CV). The main idea behind CV is to split data, normally many times, for estimating the

TABLE 3: Linear regression and nonlinear regression results for the WD.

Method	MSE		Time (s)	
	Mean	STD	Mean	STD
MLR	$5.490E - 06$	$2.311E - 08$	0.089	0.216
MN-LR	$5.415E - 06$	$2.437E - 08$	2.143	0.254

TABLE 4: Radial-basis function network results for the WD.

# of neurons	MSE		Time (s)	
	Mean	STD	Mean	STD
10	$5.104E - 06$	$2.723E - 08$	0.050	1.091
30	$5.108E - 06$	$1.273E - 08$	0.050	1.091
50	$5.105E - 06$	$2.513E - 08$	0.047	0.098

risk, error, or performance of each algorithm. Part of data (the training samples) is used for training each algorithm, and the remaining part (the validation samples) is used for validating the algorithm(s). Then, CV selects the algorithm with the smallest estimated risk [41]. CV prevents from overfitting because the training sample is independent of the validation sample. The number of the k parameters (data partitions) was 10 for all the experiments in the present study. It means that 90% of the data are used for training and 10% for validation. In the case of neural models, the training process is repeated ten times (one for each *fold*). In the case of MLP, training is also repeated for each training algorithm (see Section 2.2). For all the experiments the Mean and the Standard Deviation (STD) of the Mean Square Error (MSE) for the ten *folds* are presented in Tables 3–11. The Mean and the STD of the execution time (in seconds) are also presented in Tables 3–11 for the 10 *folds*.

For MLP and RBFN different network topologies have been applied: combinations of 10, 20, and 30 neurons in the hidden layer. Additionally, in the case of MLP, the model is trained 10 times with the same combination of parameters to reduce the effect of randomness and get more statistically significant results.

4.2. Results from the Whole Dataset. In this section, results in terms of MSE and execution time when applying MLR, MN-LR, RBFN, and MLP to the WD are presented.

In Tables 3 and 4, it can be observed that the MSE Mean values for the determination of the O_3 are very similar for the three applied methods (MLR, MN-LR, and RBFN). In the case of RBFN, slightly lower values of MSE are obtained, with the lowest one being obtained with 10 neurons in the hidden layer. Regarding execution times, the MN-LR method turns out to be the slowest and RBFN the quicker. The high values of STD for the runtime in the case of RBFN are due to the fact that it greatly varies from one fold to the others.

As it can be seen in Table 5, the LM, SCG, and BR training algorithms present the lowest values of MSE Mean in all cases (10, 30, and 50 neurons) and very close to those shown in Tables 3 and 4. The lowest value of MSE was obtained with the LM learning algorithm and 50 neurons. The learning algorithm that attained the worst results (in terms of MSE)

is GDX. With respect to execution time, the SCG algorithm attained the best results, while LM and BR are the second best ones, while TB was the slowest of the five algorithms. Obviously, the training algorithms take more time when 50 neurons are defined in the hidden layer, the TB algorithm being the one with greatest effect.

4.3. Results from the Season Dataset. In Tables 6–8 results of applying MLR, MN-LR, RBFN, and MLP to subsets with data from the four seasons of the year (spring, summer, autumn, and winter) are presented.

In Tables 6 and 7 the 3 methods present similar values in MSE Mean, and the lowest MSE Mean is achieved by the RBFN with 50 neurons in the hidden layer for the summer season. The MSE Mean values are higher than that observed for the WD. The season of the year with the lowest values of MSE Mean is the summer. One reason may be that there are few variations in pollution conditions during summer time. This is due to the small variation in weather conditions during summer as well as low industrial activity and traffic in urban areas due to vacation time. Furthermore, correlation coefficients in more than 20 pollutants analyzed in [42] are higher for measurements in the summer compared with correlations for measurements over all days combined. The season of the year with the worst results in the calculation of the MSE has been the autumn in the case of the two regression techniques and RBFN, although the differences between the three seasons (spring, summer, and autumn) is not significant. In terms of execution time, it is probed once again that MN-LR is the slowest method, while RBFN is the quickest one, returning very similar results for the four seasons of the year.

In Table 8, similarly to Table 5, the training algorithms that achieve the best results in terms of MSE Mean are LM, SCG, and BR. LM achieved the best value of MSE Mean in 10 of the 12 cases shown in Table 8, being exceeded by BR by a minimum value for the winter and spring seasons with a configuration of 10 neurons. GDX records the worst MSE values in the 12 cases shown in Table 8. Again, the best MSE Mean is obtained for the summer season, reducing the MSE Mean in comparison with those registered by RBFN. The season of the year with the worst results in the calculation of

TABLE 5: Multilayer perceptron results for the WD.

# of neurons	Training algorithm	MSE		Time (s)	
		Mean	STD	Mean	STD
10	LM	$4.731E-06$	$5.143E-08$	0.070	0.287
	GDX	$1.129E-04$	$6.825E-05$	0.317	0.287
	TB	$5.889E-05$	$4.973E-05$	0.642	0.022
	SCG	$5.216E-06$	$1.514E-07$	0.060	0.001
	BR	$4.775E-06$	$1.092E-07$	0.074	0.003
30	LM	$4.599E-06$	$1.015E-07$	0.102	0.442
	GDX	$4.045E-04$	$3.523E-04$	0.481	0.442
	TB	$4.223E-05$	$2.087E-05$	1.420	0.025
	SCG	$5.162E-06$	$1.19E-07$	0.063	0.001
	BR	$4.727E-06$	$5.667E-08$	0.102	0.005
50	LM	$4.512E-06$	$8.952E-08$	0.160	1.080
	GDX	$1.541E-04$	$3.722E-04$	0.648	1.080
	TB	$4.812E-05$	$3.032E-05$	2.156	0.051
	SCG	$5.014E-06$	$8.322E-08$	0.068	0.001
	BR	$4.731E-06$	$1.099E-07$	0.161	0.010

TABLE 6: Linear regression and nonlinear regression results for the Season Dataset.

Subset	Method	MSE		Time (s)	
		Mean	STD	Mean	STD
Spring	MLR	$1.895E-05$	$1.406E-07$	0.085	0.208
	MN-LR	$1.895E-05$	$1.242E-07$	0.169	0.298
Summer	MLR	$2.101E-05$	$1.447E-07$	0.085	0.215
	MN-LR	$1.343E-05$	$1.365E-07$	0.665	0.321
Autumn	MLR	$2.106E-05$	$2.079E-07$	0.085	0.208
	MN-LR	$2.101E-05$	$1.447E-07$	0.677	0.259
Winter	MLR	$1.895E-05$	$1.406E-07$	0.088	0.214
	MN-LR	$1.895E-05$	$1.242E-07$	0.168	0.274

TABLE 7: Radial-basis function network results for the Season Dataset.

Subset	# of neurons	MSE		Time (s)	
		Mean	STD	Mean	STD
Spring	10	$1.845E-05$	$1.687E-07$	0.046	0.096
	30	$1.847E-05$	$1.06E-07$	0.050	0.098
	50	$1.846E-05$	$1.297E-07$	0.047	0.096
Summer	10	$1.308E-05$	$1.549E-07$	0.045	0.098
	30	$1.310E-05$	$1.389E-07$	0.045	0.096
	50	$1.306E-05$	$1.344E-07$	0.047	0.096
Autumn	10	$1.986E-05$	$1.176E-07$	0.046	0.096
	30	$1.987E-05$	$2.323E-07$	0.046	0.097
	50	$1.987E-05$	$2.199E-07$	0.045	0.095
Winter	10	$1.845E-05$	$1.687E-07$	0.046	0.100
	30	$1.847E-05$	$1.060E-07$	0.045	0.093
	50	$1.846E-05$	$1.297E-07$	0.046	0.096

TABLE 8: Multilayer perceptron results for the Season Dataset.

Subset	# of neurons	Training algorithm	MSE		Time (s)	
			Mean	STD	Mean	STD
Spring	10	LM	$1.696E-05$	$2.550E-07$	0.063	0.086
		GDX	$3.298E-04$	$8.39E-04$	0.183	0.086
		TB	$5.870E-05$	$2.590E-05$	0.383	0.049
		SCG	$1.959E-05$	$3.439E-07$	0.056	0.001
		BR	$1.695E-05$	$1.988E-07$	0.076	0.015
	30	LM	$1.531E-05$	$4.107E-07$	0.068	0.217
		GDX	$6.652E-04$	$1.051E-03$	0.210	0.217
		TB	$1.069E-04$	$3.909E-05$	0.473	0.019
		SCG	$1.870E-05$	$2.857E-07$	0.055	0.005
		BR	$1.562E-05$	$3.915E-07$	0.071	0.225
	50	LM	$1.473E-05$	$4.138E-07$	0.080	0.181
		GDX	$6.286E-04$	$1.200E-03$	0.253	0.181
		TB	$1.564E-04$	$1.402E-04$	0.770	0.075
		SCG	$1.809E-05$	$3.768E-07$	0.056	0.001
		BR	$1.580E-05$	$3.697E-07$	0.089	0.490
Summer	10	LM	$1.009E-05$	$1.203E-07$	0.059	0.0663
		GDX	$4.718E-04$	$4.791E-04$	0.166	0.0663
		TB	$7.65E-05$	$6.231E-05$	0.355	0.0396
		SCG	$1.217E-05$	$2.749E-07$	0.053	0.0007
		BR	$1.010E-05$	$1.436E-07$	0.064	0.0059
	30	LM	$9.171E-06$	$2.713E-07$	0.065	0.440
		GDX	$8.070E-04$	$1.446E-03$	0.202	0.440
		TB	$9.117E-05$	$5.051E-05$	0.500	0.047
		SCG	$1.118E-05$	$4.118E-07$	0.056	0.001
		BR	$9.822E-06$	$3.107E-07$	0.073	0.004
	50	LM	$8.673E-06$	$3.284E-07$	0.083	0.225
		GDX	$4.572E-04$	$9.864E-04$	0.253	0.225
		TB	$1.269E-04$	$6.38E-05$	0.743	0.019
		SCG	$1.089E-05$	$1.561E-07$	0.057	0.001
		BR	$9.851E-06$	$2.165E-07$	0.085	0.003
Autumn	10	LM	$1.622E-05$	$2.146E-07$	0.061	0.101
		GDX	$1.598E-04$	$3.096E-04$	0.168	0.101
		TB	$7.248E-05$	$3.589E-05$	0.351	0.058
		SCG	$1.904E-05$	$2.617E-07$	0.055	0.001
		BR	$1.628E-05$	$7.680E-07$	0.071	0.009
	30	LM	$1.495E-05$	$3.564E-07$	0.067	0.196
		GDX	$1.045E-03$	$1.520E-03$	0.204	0.196
		TB	$1.09E-04$	$6.737E-05$	0.506	0.048
		SCG	$1.808E-05$	$2.756E-07$	0.054	0.001
		BR	$1.522E-05$	$3.456E-07$	0.069	0.001
	50	LM	$1.401E-05$	$1.926E-06$	0.079	0.307
		GDX	$5.676E-04$	$1.700E-03$	0.240	0.307
		TB	$1.005E-04$	$6.447E-05$	0.734	0.029
		SCG	$1.758E-05$	$5.509E-07$	0.054	0.001
		BR	$1.559E-05$	$6.591E-07$	0.083	0.103

TABLE 8: Continued.

Subset	# of neurons	Training algorithm	MSE		Time (s)	
			Mean	STD	Mean	STD
Winter	10	LM	$1.573E-05$	$2.204E-07$	0.060	0.105
		GDX	$3.380E-04$	$8.630E-04$	0.170	0.105
		TB	$9.792E-05$	$1.398E-04$	0.356	0.008
		SCG	$1.811E-05$	$3.452E-07$	0.055	0.001
		BR	$1.558E-05$	$1.229E-06$	0.065	0.016
		LM	$1.423E-05$	$2.490E-07$	0.070	0.122
	30	GDX	$5.721E-04$	$7.005E-04$	0.215	0.122
		TB	$1.253E-04$	$4.611E-05$	0.490	0.035
		SCG	$1.710E-05$	$3.509E-07$	0.056	0.001
		BR	$1.475E-05$	$2.85E-07$	0.103	0.096
		LM	$1.355E-05$	$6.746E-07$	0.080	0.155
		GDX	$6.788E-04$	$5.702E-04$	0.247	0.155
50	TB	$1.324E-04$	$1.253E-04$	0.757	0.011	
	SCG	$1.670E-05$	$4.094E-07$	0.055	0.001	
	BR	$1.478E-05$	$2.93E-07$	0.080	0.115	

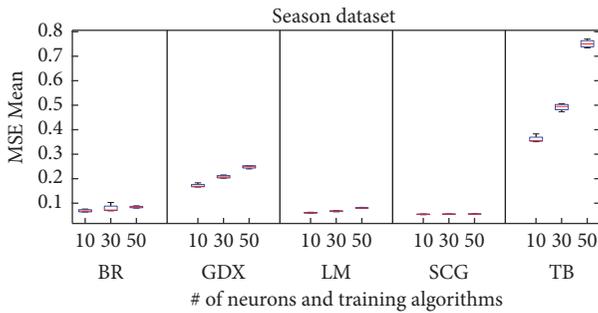


FIGURE 2: Boxplot for the MLP applied to the Season Dataset (SD).

the MSE has been the spring, although the difference in this term between spring, autumn, and winter is minimal.

In terms of execution time, it can be said that the SCG algorithm is the fastest one (in terms of mean execution time), with slight variations (low STD). LM and BR perform very well according to runtime with very similar result. Finally, the TB algorithm is the slowest one in the 12 cases shown in Table 8. This algorithm is very sensitive in its execution time to the increase in the number of neurons in the hidden layer. It is worth mentioning that the best value in terms of execution time has been obtained by SCG and for the summer season, the same for which the best value of MSE is achieved.

Figure 2 shows the boxplot for the results shown in Table 8. Each box represents the MSE Mean values for the whole dataset (four seasons), for a certain number of neurons and a training algorithm.

In Figure 2 it can be observed that the LM and SCG training algorithms outperform the other algorithms and that the TB algorithm achieved the worst results. It is also worth mentioning that, in general terms, increasing the number of neurons in the hidden layer causes an increase in the MSE due to the loss of generalization capability of the models

(especially in the TB and GDX algorithms). The difference between the 25th and 75th percentiles is also higher in the case of algorithms achieving poor results in the Season Dataset, especially for the TB training algorithm.

4.4. Results from the Station Type Dataset. Tables 9–11 show the results of applying the four techniques to two different subsets, according to the station type: urban or background (see Section 4.1 for further details).

MLR, according to Sections 4.2 and 4.3, again MN-LR, and RBF achieve similar results in estimating the MSE, but in this occasion it is higher than in Sections 4.2 and 4.3. In turn, the urban stations get a better MSE than the background stations; this indicates that the pollution levels are more constant in the urban stations than in the ones furthest from the center. In terms of the execution time, the MN-LR is again the slowest method. The RBFN shows a more efficient response than the regression methods.

For MLP, and in similar way compared to the other datasets (Sections 4.2 and 4.3), the training algorithms which achieved the lowest MSE Mean values are LM, SCG, and BR. These values are similar to those obtained by RBFN (Table 10) and lower than the values associated with the regression techniques in Table 9. According to the station type, generally speaking, lower MSE values were obtained for the “urban” stations, in comparison with “background” stations. The lowest MSE value was obtained for “urban” stations with 50 neurons and LM algorithm. The only training algorithm which returns higher values of MSE Mean for the “urban” stations than for the “background” stations is GDX. This happened for the three different numbers of neurons, while the other four algorithms get lower values of MSE for the “urban” stations for the different numbers of neurons in the hidden layer. The lower value in the MSE makes the “urban” stations easier to estimate the missing O_3 values; this is due to fewer variations in the pollution values in

TABLE 9: Linear regression and nonlinear regression results for the Station Type Dataset.

Subset	Method	MSE		Time (s)	
		Mean	STD	Mean	STD
Urban	MLR	$9.922E - 06$	$3.297E - 08$	0.053	0.190
	MN-LR	$9.352E - 06$	$3.840E - 08$	1.172	0.249
Background	MLR	$1.148E - 05$	$2.495E - 08$	0.092	0.214
	MN-LR	$1.141E - 05$	$6.084E - 08$	0.204	0.297

TABLE 10: Radial-basis function network results for the Station Type Dataset.

Subset	# Neurons	MSE		Time (s)	
		Mean	STD	Mean	STD
Urban	10	$8.480E - 06$	$4.001E - 08$	0.047	0.086
	30	$8.486E - 06$	$3.531E - 08$	0.047	0.094
	50	$8.477E - 06$	$5.212E - 08$	0.047	0.083
Background	10	$8.559E - 06$	$4.078E - 08$	0.047	0.095
	30	$9.654E - 06$	$3.401E - 08$	0.047	0.096
	50	$1.141E - 05$	$6.084E - 08$	0.045	0.093

the predictive variables throughout the year in this type of stations.

In terms of execution time, the SCG algorithm is the quickest one in the six cases shown in Table 11 followed by LM, with no big difference depending on the number of neurons in the hidden layer, only a little faster with 10 neurons. The slowest train algorithm is again TB in the six cases, as it was identified from these results with the exposed results in Tables 5 and 8.

4.5. Discussion. The two applied regression techniques (MLR and MN-LR) obtained similar values of MSE in most cases, in terms of both Mean and STD. However, MN-LR obtained poor results according to execution time (Tables 3, 6, and 9), even worse than the slowest training algorithm for MLP (GDX and TB in Tables 5, 8, and 11).

For the ANN models (RBFN and MLP), different combinations of neurons in the hidden layer were compared. For the sake of brevity, only the results for 10, 30, and 50 neurons (Tables 4, 5, 7, 8, 10, and 11) have been included in the present paper. In the case of RBFN, the best execution times are achieved, outperforming the fastest algorithm for MLP (SCG in Tables 5, 8, and 11). In the case of MLP, varying results have been obtained, depending on the training algorithm applied, obtaining the best results (in terms of MSE) when learning through the LM and SCG algorithms. SCG algorithm additionally is the fastest one. GDX has been identified as the algorithm with worst error, as can be seen in Tables 5, 8, and 11. No significant improvement is observed in the estimation of missing values according to MSE when increasing the number of neurons of the hidden layer. On the contrary, the selection of the training algorithm has been identified as a key factor when applying MLP. An increase in the number of neurons in RBFN does not affect considerably the accuracy of the results in terms of MSE and execution time (see Tables 4, 7, and 10). MLP achieved

a better value of MSE if the training algorithm is properly selected.

Taking into account the different datasets, the lowest MSE for the Season Dataset is obtained for the summer season when applying the LM training algorithm with 50 neurons, without big differences between the other three seasons of the year. For the spring, autumn, and winter seasons the best MSE corresponds to the LM algorithm combined with 50 neurons. In terms of execution time, the fastest experiment was that applying RBF with 10 and 30 neurons for the summer season, SCG for 50 neurons in the case of spring season, RBF with 50 neurons for autumn season, and RBF with 30 neurons for the winter season. In the case of the Station Type Dataset, “urban” stations, the best results in terms of MSE for the “urban” stations and for the summer season are accompanied by the lowest execution times. It must be mentioned that good results have been obtained, in terms of MSE, when applying the four imputation methods to the WD. This fact indicates no great variations neither between the weather seasons of the year nor between the analyzed types of station (“urban” and “background”).

5. Conclusions

In the present work, several different imputation methods are proposed for dealing with missing O_3 values in multi-dimensional real-life datasets with air-quality information. To do this, two multiple-regression techniques (linear and nonlinear) and two ANN models (RBFN and MLP) with different training algorithms and different number of neurons in the hidden layer have been compared. As a validation scheme, 10-fold cross-validation has been applied to the different datasets. The imputation task has been carried out firstly on the complete dataset, and on different datasets, where the original data are split according to two criteria: according to the season and according to the station type.

TABLE II: Multilayer perceptron results for the Station Type Dataset.

Subset	# of neurons	Training algorithm	MSE		Time (s)	
			Mean	STD	Mean	STD
Urban	10	LM	$6.785E-06$	$2.563E-07$	0.062	0.092
		GDX	$8.761E-05$	$1.457E-04$	0.223	0.092
		TB	$5.551E-05$	$2.967E-05$	0.411	0.029
		SCG	$8.305E-06$	$3.095E-07$	0.058	0.001
		BR	$6.572E-06$	$2.574E-07$	0.069	0.002
		LM	$6.237E-06$	$1.037E-07$	0.081	0.176
	30	GDX	$2.256E-04$	$4.360E-04$	0.292	0.176
		TB	$6.583E-05$	$2.762E-05$	0.834	0.028
		SCG	$7.752E-06$	$1.808E-07$	0.058	0.001
		BR	$6.527E-06$	$4.274E-07$	0.086	0.004
		LM	$6.059E-06$	$7.950E-08$	0.110	0.230
		GDX	$6.550E-04$	$1.165E-04$	0.379	0.230
	50	TB	$6.975E-05$	$2.578E-05$	1.168	0.026
		SCG	$7.527E-06$	$1.559E-07$	0.062	0.002
BR		$6.437E-06$	$1.117E-07$	0.108	0.014	
LM		$6.964E-06$	$7.935E-08$	0.064	0.125	
Background	10	GDX	$1.125E-04$	$2.734E-04$	0.226	0.125
		TB	$1.109E-04$	$7.399E-05$	0.444	0.069
		SCG	$8.106E-06$	$2.258E-07$	0.056	0.002
		BR	$7.036E-06$	$1.47E-07$	0.069	0.008
		LM	$6.469E-06$	$6.612E-08$	0.077	0.204
		GDX	$1.9E-04$	$2.6E-04$	0.294	0.204
	30	TB	$8.202E-05$	$2.319E-05$	0.848	0.012
		SCG	$7.707E-06$	$1.100E-07$	0.059	0.001
		BR	$6.813E-06$	$3.061E-07$	0.079	0.010
		LM	$6.184E-06$	$6.764E-08$	0.107	0.317
		GDX	$3.1E-04$	$8.9E-04$	0.383	0.317
		TB	$7.926E-05$	$5.183E-05$	1.205	0.070
	50	SCG	$7.625E-06$	$9.986E-08$	0.062	0.001
		BR	$6.743E-06$	$1.253E-07$	0.111	0.006

The following conclusions are worth mentioning:

- (1) MLR and MN-LR attained very similar results in terms of MSE and execution time. These results are slightly worse than those obtained by the two ANN models (RBFN and MLP). The lowest value of MSE has been obtained for the WD (applying MN-LR technique) and the highest one for the SD (also applying MN-LR technique).
- (2) In the case of RBFN, slight differences have been obtained when varying the number of neurons in the hidden layer, in terms of both the MSE and the execution time. The best results have been obtained for the WD (with 10 neurons in the hidden layer) and the worst for the SD (with 50 neurons in the hidden layer), as it happened for MLR and MN-LR.
- (3) In the case of MLP, the best results are achieved when using the LM training algorithm and a number of 50 neurons in the hidden layer. As in the previous case

(RBFN), the best results are obtained for the WD and the worst results for the SD, with small differences between the results in the three datasets. These are the best result from the whole experimentation in the present paper. The results obtained by MLP improve those obtained by RBFN, only when applying the LM training algorithm.

- (4) The CV technique guarantees reliability of the results when dealing with large datasets.

As future work, the application of additional artificial-intelligence models for the imputation of O_3 and other pollutants is proposed, comparing the results with those obtained in the present study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] The Pubchem Project, "Ozone," <https://www.ncbi.nlm.nih.gov/pubmed/>.
- [2] W. L. Chameides, F. Fehsenfeld, and M. O. Rodgers, "Ozone precursor relationships in the ambient atmosphere," *Journal of Geophysical Research: Atmospheres*, vol. 97, no. 5, pp. 6037–6055, 1992.
- [3] United States Environmental Protection Agency, "What is Ozone?" <https://www.epa.gov/ozone-pollution-and-your-patients-health/what-ozone>.
- [4] United States Environmental Protection Agency, "Health and Environmental Effects of Ozone Layer Depletion," <https://www.epa.gov/ozone-layer-protection/health-and-environmental-effects-ozone-layer-depletion>.
- [5] European Union Law, "Directive 2008/50/EC of the European Parliament and of the Council of 21 May 2008 on ambient air quality and cleaner air for Europe," 2008, <http://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX:32008L0050>.
- [6] C. C. Turrado, M. D. C. M. López, F. S. Lasheras, B. A. R. Gómez, J. L. C. Rollé, and F. J. D. C. Juez, "Missing data imputation of solar radiation data under different atmospheric conditions," *Sensors*, vol. 14, no. 11, pp. 20382–20399, 2014.
- [7] P. Schmitt, J. S. Mandel, and M. Guedj, "A Comparison of Six Methods for Missing Data Imputation," *Journal of Biometrics Biostatistics*, p. 6, 2015.
- [8] T. D. Pigott, "A Review of Methods for Missing Data," *Educational Research and Evaluation*, vol. 7, no. 4, pp. 353–383, 2001.
- [9] Á. Arroyo et al., "A hybrid intelligent system for the analysis of atmospheric pollution: a case study in two European regions," *Logic Journal of the IGPL*, vol. 25, no. 6, pp. 915–937, 2017.
- [10] J. L. Schafer, "Multiple imputation: a primer," *Statistical Methods in Medical Research*, vol. 8, no. 1, pp. 3–15, 1999.
- [11] H. Junninen, H. Niska, K. Tuppurainen, J. Ruuskanen, and M. Kolehmainen, "Methods for imputation of missing values in air quality data sets," *Atmospheric Environment*, vol. 38, no. 18, pp. 2895–2907, 2004.
- [12] A. Plaia and A. L. Bondi, "Single imputation method of missing values in environmental pollution data sets," *Atmospheric Environment*, vol. 40, no. 38, pp. 7316–7330, 2006.
- [13] M. P. Gómez-Carracedo, J. M. Andrade, P. López-Mahía, S. Muniategui, and D. Prada, "A practical comparison of single and multiple imputation methods to handle complex missing data in air quality datasets," *Chemometrics and Intelligent Laboratory Systems*, vol. 134, pp. 23–33, 2014.
- [14] J. L. Schafer, *Analysis of Incomplete Multivariate Data*, Hall/CRC Monographs on Statistics & Applied Probability, Chapman & Hall, New York, NY, USA, 1997.
- [15] S. Ghorbani and M. C. Desmarais, "Performance Comparison of Recent Imputation Methods for Classification Tasks over Binary Data," *Applied Artificial Intelligence*, vol. 31, no. 1, pp. 1–22, 2017.
- [16] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: a review," *Neural Computing and Applications*, vol. 19, no. 2, pp. 263–282, 2010.
- [17] M. D. Julie and B. Kannan, "Attribute reduction and missing value imputing with ANN: Prediction of learning disabilities," *Neural Computing and Applications*, vol. 21, no. 7, pp. 1757–1763, 2012.
- [18] M. S. Gashler, M. R. Smith, R. Morris, and T. Martinez, "Missing value imputation with unsupervised backpropagation," *Computational Intelligence*, vol. 32, no. 2, pp. 196–215, 2016.
- [19] P. Coulibaly and N. D. Evora, "Comparison of neural network methods for infilling missing daily weather records," *Journal of Hydrology*, vol. 341, no. 1-2, pp. 27–41, 2007.
- [20] I. A. Gheyas and L. S. Smith, "A neural network-based framework for the reconstruction of incomplete data sets," *Neurocomputing*, vol. 73, no. 16-18, pp. 3039–3065, 2010.
- [21] E.-L. Silva-Ramírez, R. Pino-Mejías, M. López-Coello, and M.-D. Cubiles-de-la-Vega, "Missing value imputation on missing completely at random data using multilayer perceptrons," *Neural Networks*, vol. 24, no. 1, pp. 121–129, 2011.
- [22] R. R. Andridge and R. J. A. Little, "A review of hot deck imputation for survey non-response," *International Statistical Review*, vol. 78, no. 1, pp. 40–64, 2010.
- [23] D. Sovilj, E. Eirola, Y. Miche et al., "Extreme learning machine for missing data using multiple imputations," *Neurocomputing*, vol. 174, pp. 220–231, 2016.
- [24] E.-L. Silva-Ramírez, R. Pino-Mejías, and M. López-Coello, "Single imputation with multilayer perceptron and multiple imputation combining multilayer perceptron and k-nearest neighbours for monotone patterns," *Applied Soft Computing*, vol. 29, pp. 65–74, 2015.
- [25] S. Azim and S. Aggarwal, "Hybrid model for data imputation: using fuzzy c means and multi layer perceptron," in *Proceedings of the 4th IEEE International Advance Computing Conference (IACC '14)*, pp. 1281–1285, Gurgaon, India, February 2014.
- [26] University of Yale, "Linear Regression," <http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm>.
- [27] K. Pearson and A. Lee, "On the generalised probable error in multiple normal correlation," *Biometrika*, vol. 6, no. 1, pp. 59–68, 1908.
- [28] University of Yale, "Multiple Linear Regression," 2017, <http://www.stat.yale.edu/Courses/1997-98/101/linmult.htm>.
- [29] M. H. Kutner, C. J. Nachtsheim, and J. Neter, *Applied Linear Regression Models*, McGraw Hill.
- [30] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '89)*, vol. 1, pp. 593–605, Washington, DC, USA, June 1989.
- [31] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric Environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.
- [32] C. C. Yu and B. D. Liu, "A backpropagation algorithm with adaptive learning rate and momentum coefficient," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2002.
- [33] R. P. Lippmann, "Pattern Classification Using Neural Networks," *IEEE Communications Magazine*, vol. 27, no. 11, pp. 47–50, 1989.
- [34] J. Park and I. W. Sandberg, "Universal approximation using radial basis function networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991.
- [35] European Commission, "Castilla y León," 2016, <https://ec.europa.eu/growth/tools-databases/regional-innovation-monitor/base-profile/castilla-y-le%C3%B3n-0>.

- [36] Castilla y León Government, “Castilla y León Air Quality network,” https://medioambiente.jcyl.es/web/jcyl/Medio-Ambiente/es/Plantilla66y33//_/./...
- [37] Government of Spain, “Open Data,” <http://datos.gob.es/>.
- [38] Castilla y León Government, “Castilla y León Open Data,” https://datosabiertos.jcyl.es/web/jcyl/set/es/medio-ambiente/calidad_aire_estaciones/1284212701893.
- [39] National Center for Biotechnology Information, “PubChem Compound,” 2017, <https://www.ncbi.nlm.nih.gov/pubmed/>.
- [40] Dekati, *ISO 23210 Measurements of PM10 and PM2.5*, 2017, <https://www.dekati.com/applications/stationary-source-emissions/emission-monitoring/pm10-and-pm25-according-to-iso23210>.
- [41] S. Arlot and A. Celisse, “A survey of cross-validation procedures for model selection,” *Statistics Surveys*, vol. 4, pp. 40–79, 2010.
- [42] I. Levy, C. Mihele, G. Lu, J. Narayan, and J. R. Brook, “Evaluating multipollutant exposure and urban air quality: Pollutant interrelationships, neighborhood variability, and nitrogen dioxide as a proxy pollutant,” *Environmental Health Perspectives*, vol. 122, no. 1, pp. 65–72, 2014.

Research Article

Intelligent Torque Vectoring Approach for Electric Vehicles with Per-Wheel Motors

Alberto Parra ¹, Asier Zubizarreta ², Joshué Pérez,¹ and Martín Dendaluze²

¹*TecNALIA Research & Innovation, Industry and Transport Division, Donostia, Spain*

²*Department of Automatics and System Engineering, Faculty of Engineering of Bilbao, University of the Basque Country (UPV/EHU), Bilbao, Spain*

Correspondence should be addressed to Alberto Parra; alberto.parra@tecnalia.com

Received 24 November 2017; Revised 15 January 2018; Accepted 23 January 2018; Published 25 February 2018

Academic Editor: José Manuel Andújar

Copyright © 2018 Alberto Parra et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Transport electrification is currently a priority for authorities, manufacturers, and research centers around the world. The development of electric vehicles and the improvement of their functionalities are key elements in this strategy. As a result, there is a need for further research in emission reduction, efficiency improvement, or dynamic handling approaches. In order to achieve these objectives, the development of suitable Advanced Driver-Assistance Systems (ADAS) is required. Although traditional control techniques have been widely used for ADAS implementation, the complexity of electric multimotor powertrains makes intelligent control approaches appropriate for these cases. In this work, a novel intelligent Torque Vectoring (TV) system, composed of a neuro-fuzzy vertical tire forces estimator and a fuzzy yaw moment controller, is proposed, which allows enhancing the dynamic behaviour of electric multimotor vehicles. The proposed approach is compared with traditional strategies using the high fidelity vehicle dynamics simulator Dynacar. Results show that the proposed intelligent Torque Vectoring system is able to increase the efficiency of the vehicle by 10%, thanks to the optimal torque distribution and the use of a neuro-fuzzy vertical tire forces estimator which provides 3 times more accurate estimations than analytical approaches.

1. Introduction

The need for reducing global warming, air pollution, and oil dependency has motivated not only the use of renewable energies, but also some paradigm changes in other areas, such as transportation systems, where the development of electric vehicles (EV) has become a key strategy [1]. The interest in vehicles with electrified powertrains (fully electric as well as hybrid) has increased in the last years, becoming one of the main research areas in the automotive industry [2].

The integration of electric motors in propulsion systems provides not only better energy efficiency and lower pollution, but also increased controllability, as electric motors offer better response time [3]. These features are fueling a notable interest in the development of Advanced Driver-Assistance Systems (ADAS) that enhance not only the dynamic behaviour of the vehicle, but also its efficiency and energy consumption [4].

Traditional control approaches have been widely used to implement ADAS during the last decades. However, electrified propulsion systems offer wider complexity (and multiple topologies) than internal combustion propulsion systems. Due to this, intelligent control approaches have become one of the main research interests lately, as they can manage complex systems more easily than traditional approaches.

One of the most complete ADAS for enhancing the dynamic behaviour and stability of an electric vehicle with per-wheel motors is Torque Vectoring (TV) [5], which focuses on the optimal driving torque distribution. Several strategies can be used to control the torque distribution in a TV approach, with most of them being based on controlling the moment along the vertical axis of the vehicle (yaw moment) [6]. For this purpose, three main strategies are used [7]: the first one is based on the distribution of the torque among the driving wheels [6, 8–12]; the second one is based

on using active vehicle roll control systems to vary the lateral load distribution [13, 14]; finally, the third strategy consists in implementing a steering system in both axles [15, 16]. This work focuses on torque distribution strategies, as an all-wheel drive vehicle has been selected as case study.

Torque distribution approaches have been implemented conventionally using a wide variety of control algorithms. Among the traditional approaches, simpler ones, such as proportional-integral-derivative control (PID) based ones [6], or more advanced algorithms, such as Model Predictive Control (MPC) [17] or Sliding Mode Control (SMC) [9], have been proposed. The latter provide enhanced results, although their computational cost is higher than PID based approaches. On the other hand, intelligent approaches, such as neural networks [10] or fuzzy logic systems [11], have been demonstrated to be able to provide good results with lower computational cost, making them a good alternative to the implementation of torque distribution approaches.

In order to achieve an effective driving torque distribution, the knowledge of the tire forces is crucial [18]. Direct measurement of these forces, however, is a complex and difficult task, and, therefore, the design of proper estimators is required. Nevertheless, this issue is not always considered in the works proposed in the literature, which are based on perfect estimations of these forces, which is not a real-case scenario.

Among the approaches proposed to estimate the vehicle tire forces, the most common one is the use of estimators based on tire models, such as the linear tire model [19], Dugoff's model [20], or semiempirical models such as the Pacejka's tire model [18]. Other works do not consider the tire model and estimate vertical tire forces based on longitudinal and lateral load transfers as well as the static loads on each wheel [21–25]. Intelligent modelling approaches have also been used, which reduce the need for knowing the complex tire dynamics and even allow model adaptation. For instance, neural networks, extended Kalman filters, and recursive least squares approaches are combined in [26] to estimate tire lateral force and grip potential identification even in aggressive manoeuvres. In [27] a longitudinal, lateral, and vertical tire force estimator based on fuzzy logic is used, which requires the slip angle estimation to operate and provides high correlation. Both approaches provide more accurate estimations compared with traditional approaches, although their applicability presents some inconveniences, as they require variables difficult to measure.

In summary, intelligent approaches have been demonstrated to be a suitable alternative to ADAS development, providing balanced performance versus computational cost. However, proper tire force estimations are required to guarantee this theoretical performance in a real-case scenario. In the literature, most works consider perfect estimations or use estimators based on physical variables difficult to measure, which require expensive sensors, or use complex models whose parameters are difficult to identify. This issue reduces the implementability and performance of the approaches proposed in most works in real-case scenarios.

In order to solve these issues, this work presents a novel intelligent Torque Vectoring approach, composed of

two intelligent algorithms: first, an adaptive neuro-fuzzy inference system (ANFIS) estimator for the tire vertical forces based on exclusively measurable variables; second, a fuzzy yaw moment controller, which controls both vehicle yaw rate and sideslip angle, as they are some of the most representative vehicle dynamics variables. The proposed approach is able to enhance electric vehicle dynamics and their efficiency. To demonstrate its effectiveness, the ANFIS estimator and the resulting intelligent TV system have been validated considering several scenarios in the Dynacar high fidelity dynamic simulator, using an E-Class vehicle and comparing the obtained results with other previous works from the literature.

The rest of the paper is divided as follows: In Section 2 the proposed intelligent Torque Vectoring approach is detailed. In Section 3, the simulation framework and experimental setup are presented. In Section 4, the results of the validation carried out are explained. Finally, in Section 5, the main conclusions are presented.

2. Intelligent Torque Vectoring System

In this section the proposed intelligent Torque Vectoring system is detailed. Its main purpose is to distribute the driving torque among the different actuated wheels, so that vehicle handling and stability is improved. It can be divided into 5 subsystems (Figure 1): the lateral torque distribution is carried out using a fuzzy yaw moment controller, whose reference is calculated using a yaw rate reference generator; the longitudinal torque distribution is given by a self-defined torque distribution algorithm, which uses an ANFIS vertical tire forces estimator; the data provided by the longitudinal and lateral distributions is used to calculate the final torque distribution.

2.1. Yaw Rate Reference Generator. The developed intelligent TV system approach is composed of a lateral torque distribution approach and a longitudinal one. The first is based on the control of the yaw moment of the vehicle; this is, it requires an appropriate yaw rate reference for its proper performance.

For the calculation of the desired yaw rate reference, the well-known *bicycle model* is used, as it provides a good balance between accuracy and computational cost [28]. In order to further increase computational performance, the following assumptions and simplifications are carried out: the center of gravity is assumed to be at a height of zero; the variation of the vertical force of each tire will not be taken into account; small slip angles assumption will be considered, so that $\sin(\beta) = \beta$ and $\cos(\beta) = 1$ (linear region); and the coefficient of lateral stiffness of the tire will be constant, defined by the ratio of the lateral force to the slip angle.

It must be noted that this model of reduced complexity is exclusively used for real-time execution in the controller. In addition, some of these simplifications are reasonable for passenger cars, as they are not driven until the limits of the tires.

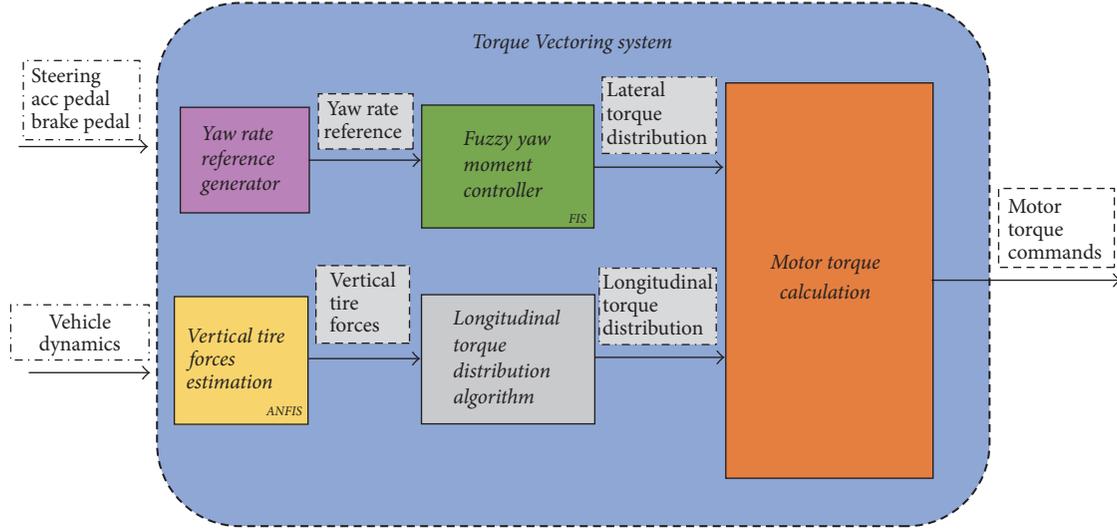


FIGURE 1: Proposed intelligent torque vectoring system.

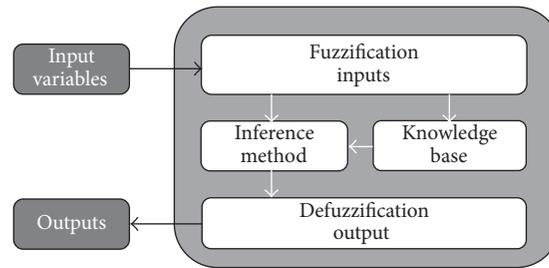


FIGURE 2: General diagram representing fuzzy logic approaches.

This way, the yaw rate reference equation is [28]

$$\dot{\psi}_{\text{ref}} = \frac{V}{L + (m/L)(b/C_{\alpha F} - a/C_{\alpha R})V^2} \delta, \quad (1)$$

where m is the total mass of the vehicle located in the center of gravity, a and b are the distance to the center of gravity of the front and rear axle, respectively, L is the distance between axles, δ is the angle of rotation of the front wheels, $C_{\alpha F}$ and $C_{\alpha R}$ are the lateral stiffness coefficients of the front and rear wheels, respectively, and V is the vehicle speed.

However, for safety reasons it is necessary to limit the value of the yaw rate reference generated. In this case, the limit has been set as follows [6]:

$$|\dot{\psi}_{\text{ref,max}}| = \left| \frac{a_y}{V} \right|, \quad (2)$$

where a_y is the vehicle lateral acceleration.

2.2. Fuzzy Yaw Moment Controller. The fuzzy yaw moment controller handles the lateral torque distribution (τ_{lat}) for the vehicle. Hence, considering both the yaw rate and slip angle of the vehicle, this system calculates the torque percentage to be applied to each side of the vehicle. This way, if $\tau_{\text{lat}} = 0$, all the torque will be applied to the wheels of the right side; and

if $\tau_{\text{lat}} = 1$, all the torque is applied to the wheels of the left side.

This subsystem is based on fuzzy logic, which is an extension of Boolean logic by Zadeh in 1965 [29] based on the mathematical theory of fuzzy sets. It enables an abstraction from the model and mathematical formulations by translating expert knowledge into rules without renouncing a notable level of fine-tuning capabilities.

The most common fuzzy logic system structure is shown in Figure 2. First a fuzzification process must be carried out to transform the input data into fuzzy sets to continue with an inference system, based on the developed rules. Finally, a defuzzification method is necessary in order to convert the result given by the inference system into the exact value to, in this particular case, ensure appropriated control.

The proposed fuzzy logic controller is based on the Mamdani fuzzy model, as it provides a more intuitive tuning [30]. In order to calculate the torque percentage to be applied to each side of the vehicle, τ_{lat} , the controller requires three inputs: the yaw rate error, its derivative, and the side slip angle error. The yaw rate error and its derivative are calculated considering the reference detailed in the previous subsection. The slip angle error is calculated considering that the reference sideslip angle of the vehicle will be zero, in

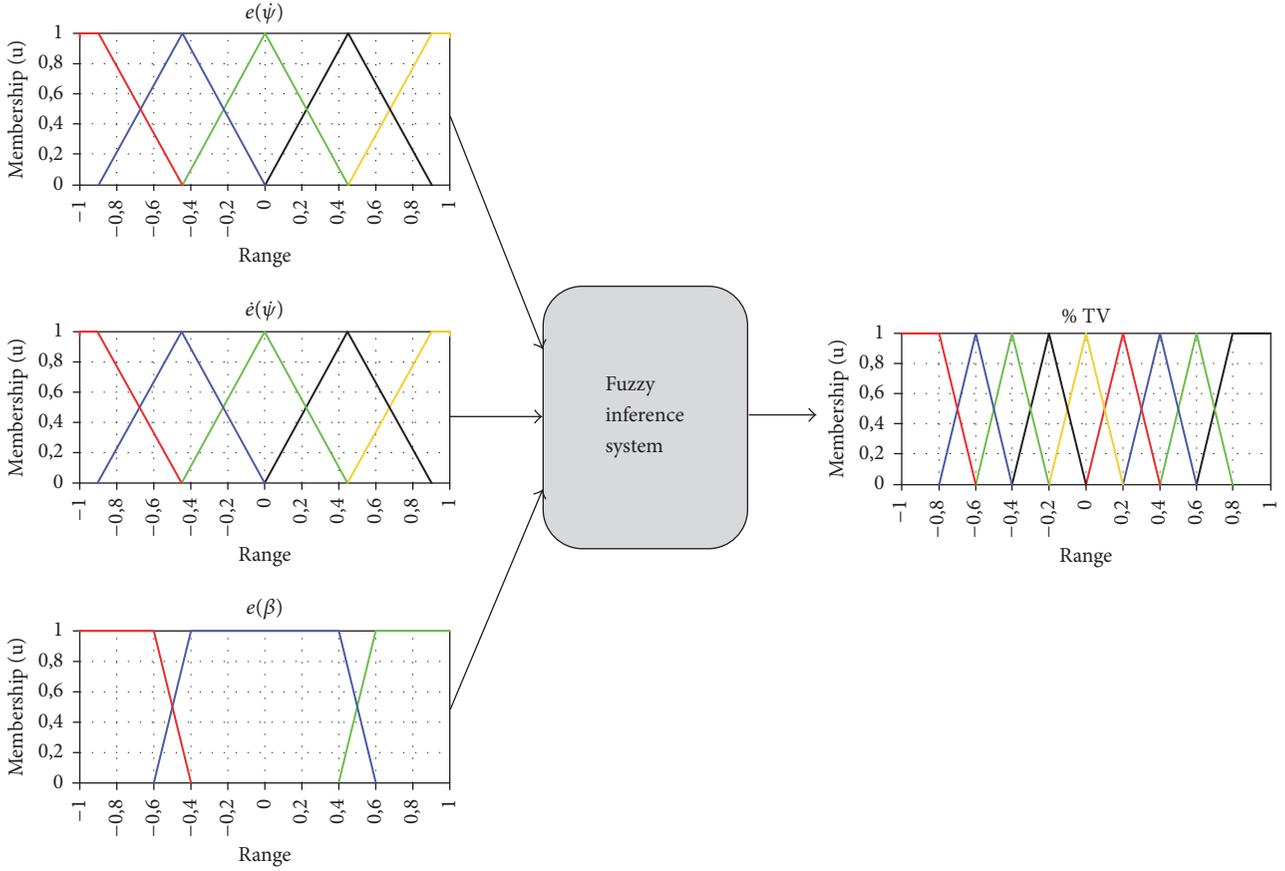


FIGURE 3: Fuzzy logic controller proposed.

order to reduce the real value of the slip and achieve a neutral handling.

The actual vehicle sideslip angle value is calculated using the following equation [31]:

$$\beta = \arctan\left(\frac{V_y}{V_x}\right), \quad (3)$$

where V_y and V_x are the vehicle speed in longitudinal (x) and lateral (y) local axes.

For the design of the fuzzy system the following structure has been implemented. First, a distribution of 5 membership functions has been chosen for the yaw rate error $e(\psi)$ and for its derivative $\dot{e}(\psi)$, and three membership functions for the input of lateral slip angle, $e(\beta)$. Considering the typical sideslip angle and yaw rate values achieved by a passenger car, it has been considered that five membership functions cover the whole range for the first two variables with a good level of accuracy. Moreover, the sign of sideslip angle determines if the vehicle has a neutral, understeering or oversteering behaviour. Therefore, it has been considered that only 3 membership functions are needed for this last variable. All of them have been selected as trapezoidal and triangular. The trapezoidal ones have been selected for the boundaries of each variable and for the membership functions of sideslip angle, since the proposed controller tries to minimize this variable and, therefore, accuracy is not the highest priority.

The triangular ones have been used for the rest of variables, as they provide computationally efficient calculations [32] maintaining acceptable smoothness on the response, suitable to be implemented in conventional automotive Electronic Control Units (ECUs).

And finally, for the output, the torque percentage to be applied to each side of the vehicle, τ_{lat} , a more complex distribution of membership functions has been chosen, nine in this case, aiming to achieve a response as accurate and smooth as possible.

The structure of the developed fuzzy controller is shown in Figure 3 including the membership functions.

Subsequently the corresponding rules have been implemented based on the knowledge about the system and human driving datasets. Table 1 shows the names and description of the membership functions, while Tables 2–4 show the implemented rules.

2.3. ANFIS Vertical Tire Forces Estimator. The dynamic behaviour of the vehicle depends heavily on the tire forces, as these model the contact force between the wheels and the road. However, their estimation is one of the most complex issues in vehicle dynamics, as the tire/road contact dynamics depends on a number of different variables. Direct measurement of these forces is not always a solution either, as these forces are very difficult to measure.

TABLE 1: Membership functions names.

Names	Description
NVL	Negative very large
NL	Negative large
NM	Negative medium
NS	Negative small
ZE	Zero
PS	Positive small
PM	Positive medium
PL	Positive large
PVL	Positive very large

TABLE 2: Rules for negative yaw rate error derivative.

$\dot{e}(\psi) < 0$	$e(\psi)$					
	NL	NS	ZE	PS	PL	
	NL	ZE	NS	NM	NL	NVL
$e(\beta)$	NS	ZE	ZE	NS	NM	NL
	ZE	ZE	ZE	ZE	NS	NL
	PS	PM	PS	ZE	ZE	NS
	PL	PL	PM	PS	ZE	ZE

TABLE 3: Rules for zero yaw rate error derivative.

$\dot{e}(\psi) = 0$	$e(\psi)$					
	NL	NS	ZE	PS	PL	
	NL	ZE	NS	NM	NL	NVL
$e(\beta)$	NS	PS	ZE	NS	NM	NL
	ZE	PM	PS	ZE	NS	NM
	PS	PL	PM	PS	ZE	NS
	PL	PVL	PL	PM	ZE	ZE

TABLE 4: Rules for positive yaw rate error derivative.

$\dot{e}(\psi) > 0$	$e(\psi)$					
	NL	NS	ZE	PS	PL	
	NL	ZE	ZE	NS	NS	NM
$e(\beta)$	NS	PS	ZE	ZE	NS	NS
	ZE	PM	PS	ZE	ZE	ZE
	PS	PL	PM	PS	ZE	ZE
	PL	PVL	PL	PM	PS	ZE

In this section, a novel ANFIS vertical tire forces estimator is proposed. The proposed estimator provides real-time and accurate estimations of the tire forces, which can be exploited by ADAS to increase the safety, stability, and efficiency of vehicles. Hence, this estimator will be used to perform the longitudinal dynamics torque distribution.

The proposed estimator is based on an ANFIS that is based on a fuzzy system that uses a learning algorithm derived from neural network theory to determine its parameters (fuzzy sets and fuzzy rules) by processing data samples [33]. For that purpose, the Takagi-Sugeno model is used, as it

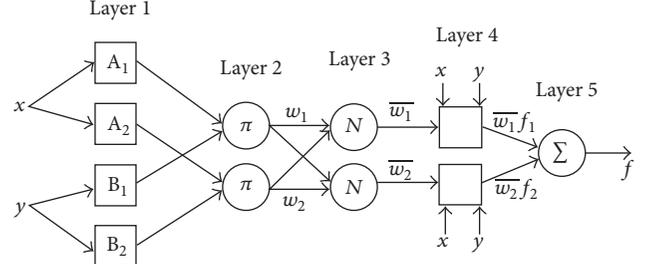
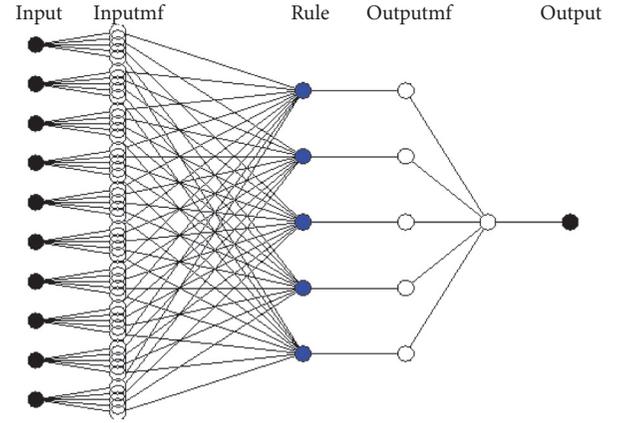


FIGURE 4: ANFIS structure [2].



Logical operations

● And

FIGURE 5: ANFIS estimators structure.

is computationally efficient [34] and has been demonstrated to be appropriate for modelling applications. This approach maintains the fuzzy logic capability of converting human knowledge into a quantitative process and rules while it solves the membership functions iterative tuning process problem thanks to the neural networks learning ability. A simplified diagram of an ANFIS structure is shown in Figure 4.

The proposed estimator uses measurable variables to operate, which is one of the main contributions of this work compared with those analyzed in the bibliography. The input data is composed of 10 variables: the steering angle; the x , y , and z linear acceleration and speed components of the center of gravity (CoG) of the vehicle; and the 3 angular speeds associated with the local axes. All of them can be easily measured using commercially available sensors such as Inertial Measurement Units (IMU), Global Positioning Systems (GPS), and steering angle sensors. The output data is composed of the vertical force of the selected tire. The structure of the ANFIS designed for each wheel is detailed in Figure 5.

The proposed ANFIS estimator is composed of 4 layers. In the first of them 7 membership functions for each input have been developed. These membership functions are of Gaussian type, as they provide better precision than triangular ones [32, 35]. In the second one, the rules established by the learning process appear. In the third layer the ratio calculation and

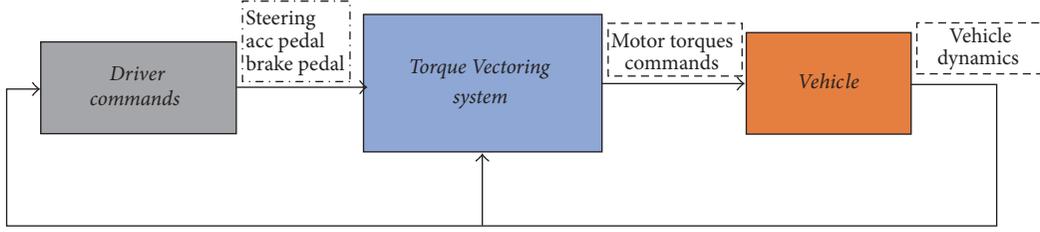


FIGURE 6: General test framework, based on Dynacar [36, 37].

TABLE 5: Parameters of the hybrid method training.

Range of influence	0.5
Squash factor	1.25
Accept ratio	0.5
Reject ratio	0.15

normalisation are carried out to finally, in the fourth layer, add all the signals.

The method selected for the generation of the fuzzy inference system is Subclustering, due to the high number of inputs. The training method chosen is the hybrid method, which is a combination of least squares and backpropagation gradient descent method. The parameters of this process are detailed in Table 5.

The data used for the training and testing of the proposed ANFIS structure have been obtained from a simulation of a vehicle running on the Nurburgring circuit during one lap (simulation time of 800 s and sample time of 50 ms). The simulation has been obtained from the high fidelity vehicle dynamics simulator Dynacar [36], where, for each vertical tire force estimation, the aforementioned 10 variables have been captured.

2.4. Longitudinal Torque Distribution Algorithm. The proposed longitudinal torque distribution approach calculates the longitudinal torque distribution percentage τ_{long} to be applied to the wheels of each axle (front and rear) considering the vertical normal forces generated by the wheels at each axle f_{wheels} . This way, if $\tau_{\text{long}} = 0$, all the torque will be provided by the wheels of the rear axle, and if $\tau_{\text{long}} = 1$, all the torque will be provided by the wheels of the front axle.

This way, this subsystem allows sending greater torque commands to the motors whose wheels have more grip. For that purpose, a simple but effective torque distribution algorithm is proposed, based on the maximum normal force that can be applied in an axle (the front one has been taken as reference). This way,

$$\tau_{\text{long}} = \frac{f_{\text{wheels}}}{f_{\text{max}}}, \quad (4)$$

where f_{wheels} is the combination of the vertical forces of each wheel of an axle (left and right), given by the vertical tire forces estimator, and f_{max} is the maximum normal force that can be applied in the front axle (this is, considering that the

whole mass is actuating only in one axle). For the study case selected, this final value is 20208 N.

2.5. Motor Torque Calculation. This subsystem calculates the exact motor torque command to be applied to each wheel τ_{ij} , based on the torque requested by the driver using the throttle τ , and the longitudinal τ_{long} and lateral τ_{lat} torque distribution percentages:

$$\begin{aligned} \tau_{\text{fl}} &= \tau \tau_{\text{long}} \tau_{\text{lat}} \\ \tau_{\text{fr}} &= \tau \tau_{\text{long}} (1 - \tau_{\text{lat}}) \\ \tau_{\text{rl}} &= \tau (1 - \tau_{\text{long}}) \tau_{\text{lat}} \\ \tau_{\text{rr}} &= \tau (1 - \tau_{\text{long}}) (1 - \tau_{\text{lat}}). \end{aligned} \quad (5)$$

3. Validation Framework

In this section the validation methodology used is explained, including the selected vehicle, the simulation environment, and the proposed manoeuvres and testing scenarios.

Figure 6 shows a general overview of the implemented control concept. Three main blocks can be considered: first, the driver command, which is generated by an automated driving algorithm that simulates the behaviour of a standard driver, guaranteeing that the performance of each test is not dependent on the driver; second, the control block, which includes the intelligent Torque Vectoring approach detailed in the previous section, with all subsystems; finally, the vehicle model, which simulates the highly nonlinear dynamics of the vehicle using Dynacar software [36] and provides the time evolution of the measurable variables.

3.1. Dynacar High Fidelity Dynamics Simulator. The vehicle model is implemented in Dynacar, which is a high fidelity vehicle dynamics simulation platform developed by Tecnia Research & Innovation [36]. This vehicle simulation software has been validated through several racetrack tests [37, 38] and it can be used in a model-in-the-loop framework to test the performance of different automotive aimed control systems.

One of the features of Dynacar is the possibility of activating an automated driver mode, which simulates a standard driver. This allows reducing the effect of the driver ability when analyzing the results of the developed ADAS and, hence, allowing better comparison.

Dynacar's vehicle physical model simulation engine is based on a multibody model and integrated in C code [39].

TABLE 6: Vehicle principal characteristics.

Mass [kg]	1830
I_x [kgm^2]	928.1
I_y [kgm^2]	2788.5
I_z [kgm^2]	3234.0
Wheelbase [m]	3.05
Front axis track [m]	1.6
Rear axis track [m]	1.6

This C code has also been implemented in Simulink. The solver is run with a sample time of 1 ms. Tires are modelled using implementation of the Pacejka “Magic Formula” model, extensively used by most car manufacturers as an industry standard for vehicle model simulations [18].

Table 6 shows the main characteristics of the vehicle, which is an E-Class vehicle.

3.2. Controller Implementation. The control block is implemented in a Xilinx Zynq XC7Z020 SoC, whose inputs and outputs are connected to Dynacar. This allows testing the real-time performance of the proposed intelligent TV control approach and does not require the use of a whole vehicle thanks to the Dynacar’s model-in-the-loop approach.

The selected hardware is composed of two parts. The first is the programmable logic part, which is a full FPGA. And the other part is the processing system, which is composed of an ARM CPU of two cores and 800 MHz clock rate. In addition, this board has several I/O peripherals, such as digital and analog inputs/outputs ports and communication buses.

The ARM core has been used to implement the different subsystems of the proposed intelligent TV approach (Figure 2) proposed in the previous section. For that purpose, the developed approach has been implemented first in Matlab Simulink and then compiled to C code that runs in the ARM processor.

3.3. Manoeuvres and Scenarios. Dynacar’s framework allows simulating and testing the developed intelligent TV controller in different scenarios and with a set of different standardized manoeuvres: a skid-pad [40] (Figure 7) and a double lane change manoeuvre [41] (Figure 8).

On the one hand, the objective of the skid-pad test is to measure the car’s cornering ability on a flat surface while making a constant-radius turn. This test is one of the FSAE Dynamics Events [40], but, as it is designed for formula type vehicles, it is necessary to adapt it to a passenger vehicle. In that sense, the diameters of the circles have been modified considering the rules for road designs [42]. These rules state that for a speed limit of 40 km/h the minimum radius of a curve has to be 60 meters. However, as this radius is defined to ensure the stability of the car at that speed, in this work the radius has been reduced in order to take the vehicle to its limits and then be able to evaluate the correct performance of the developed controller. Therefore, it has been decreased until 20 meters, converting the test into a challenging scenario for an E-Class vehicle.

On the other hand, the double lane change manoeuvre is detailed in the ISO 3888 specification [41] (Figure 8). In this test, the vehicle enters the course at a particular speed and the throttle is released. The driver then attempts to negotiate the course without striking the cones. The test speed is progressively increased until either instability occurs or the course can no longer be negotiated successfully. Such a severe manoeuvre effectively demonstrates the cornering capability of a vehicle when driving at the friction limit in both directions and, therefore, many car manufacturers and research institutions consider this test to be a suitable manoeuvre for assessing advanced vehicle dynamics control systems. This manoeuvre is typically performed as a closed-loop driving test and is used to adjust the dynamics of a vehicle based on the subjective evaluations of professional drivers.

4. Results

In this section the results obtained during the validation of the developed intelligent TV control approach are analyzed. For that purpose, first the proposed ANFIS vertical force estimator is validated with the results obtained with Dynacar and the analytical estimator proposed in [43], in order to demonstrate its accuracy. Then, the overall intelligent TV approach including all subsystems (Figure 2) is validated, comparing its results with a more traditional solutions. In the case of the intelligent TV algorithm its performance is compared with a PID TV controller based on [6] with a constant longitudinal torque distribution, while the ANFIS estimator’s performance is compared with the analytical estimator proposed in [43].

4.1. ANFIS-Based Vertical Tire Force Estimation. In order to test the effectiveness of the approach, the data obtained from the proposed estimator is compared with (a) Dynacar’s internal high fidelity tire model and (b) the model-based analytical estimator proposed in [43], whose results have been validated in the cited work, but it requires data from variables that are difficult to measure. The aforementioned two manoeuvres have been used to validate the force estimator.

Figures 9 and 10 and Tables 7 and 8 show the results obtained for each manoeuvre and each wheel. They show that the ANFIS estimator is able to reduce the error between 38% and 79% compared with the analytical approach (depending on the wheel and the manoeuvre), being able to eliminate the peaks obtained by the analytical estimator due to transient conditions. It has to be noted that the errors obtained for the analytical model correspond to the errors obtained in [43].

In addition, the real-time performance of the proposed estimator has been analyzed, requiring 0.9 ms to run, which is appropriate for automotive applications.

4.2. Intelligent Torque Vectoring System. In order to validate the ability of the proposed intelligent TV approach to enhance the dynamic handling, first, the results associated with the skid-pad test will be analyzed.

In order to determine the effectiveness of the approach, the critical speed of the vehicle has to be defined first. This

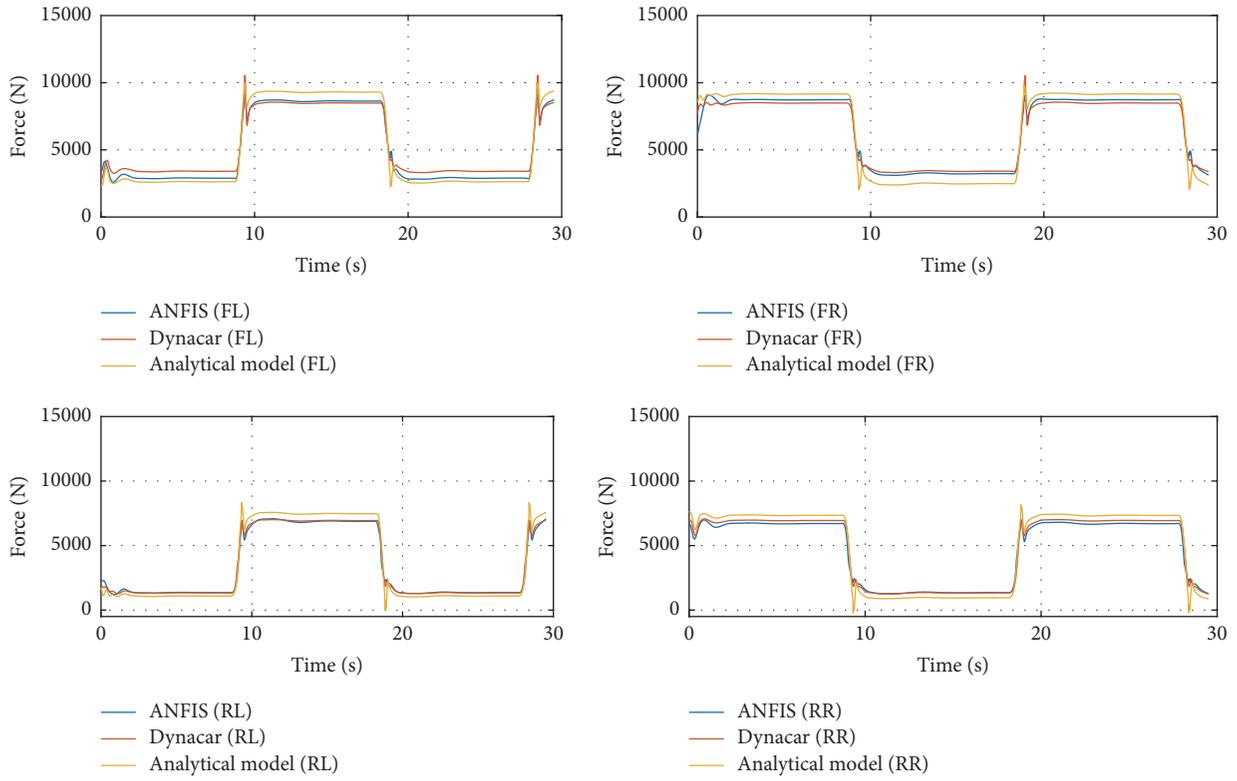


FIGURE 9: F_z estimation, skid-pad.

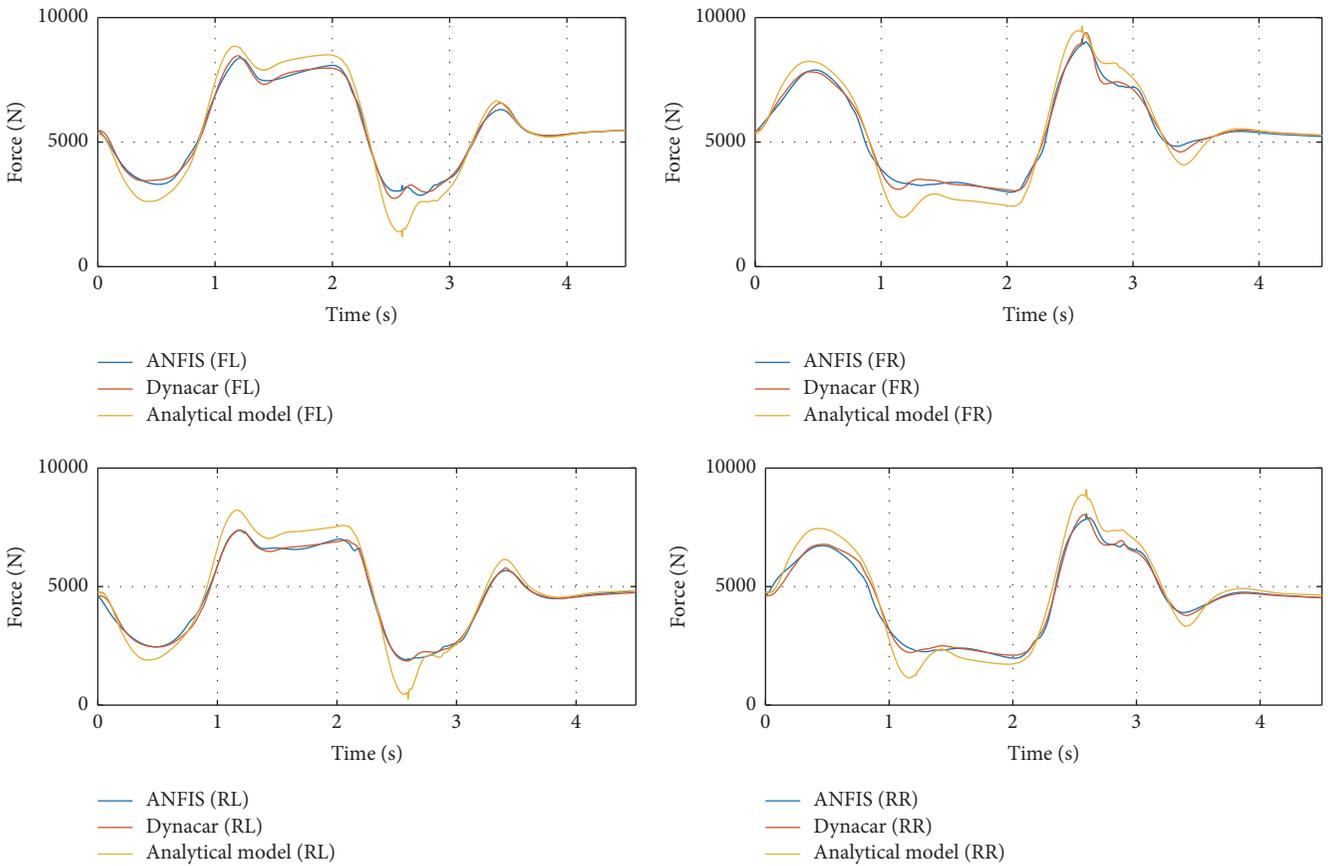


FIGURE 10: F_z estimation, double lane change.

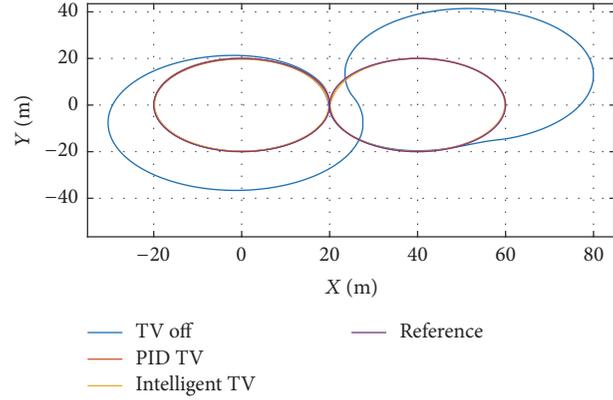


FIGURE 11: Skid-pad trajectory.

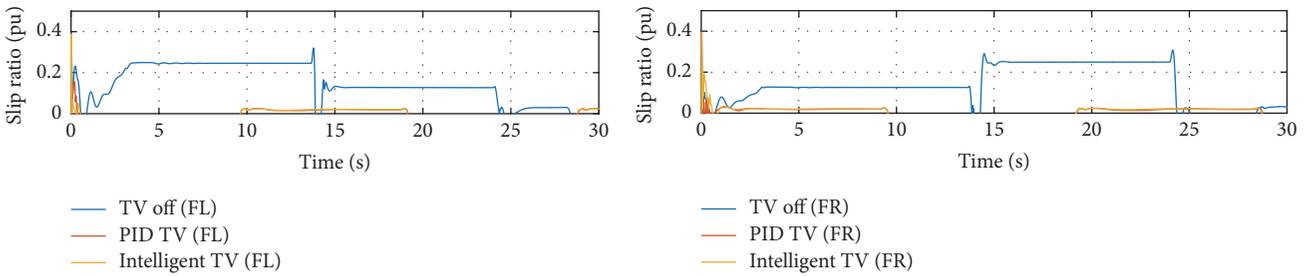


FIGURE 12: Wheels slip ratio.

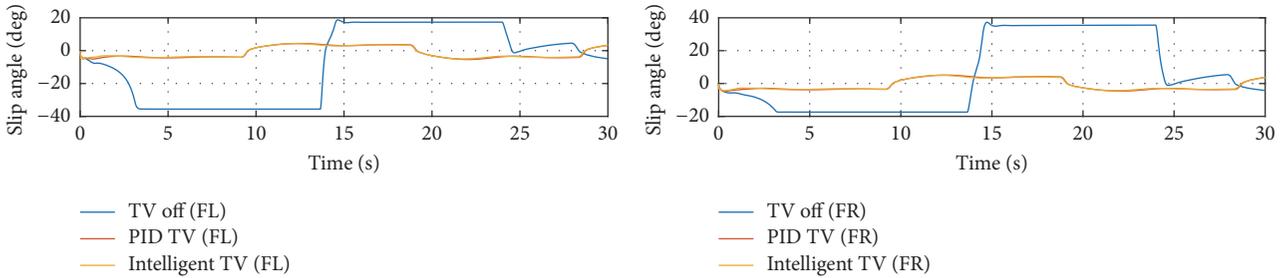


FIGURE 13: Wheels slip angles.

critical speed is the maximum speed that allows the vehicle to perform the skid-pad test correctly with no TV control. For that purpose, no TV system has been activated, and the skid-pad test has been carried out increasing the speed in each test until the vehicle has not been able to follow the reference trajectory. This critical speed has been experimentally defined as 47 km/h, providing a theoretical lateral acceleration of 0.86 g.

Once this critical speed limit is detected, the skid-pad test has been executed activating the proposed intelligent TV approach and a PID based TV approach. Results are shown in Figure 11. It can be appreciated that when no TV controller is active, the vehicle is not able to track the desired trajectory, due to understeering, but that TV approaches allow executing this test even at the critical speed.

The undesirable behaviour at the critical speed when no TV is activated can be further appreciated in Figures 12 and

13, which show the slip ratio and slip angle of the front wheels. As it can be seen, these wheels cannot transmit the requested force to the road and they slip, which can be detected by high values of the slip ratio and angles in the TV off case. Furthermore, when either TV control is active (fuzzy TV and PID TV), slip ratio and slip angle are reduced to 19% and 23°, respectively, which implies higher traction forces on the tires. This implies higher speeds in the skid-pad tests, as shown in Figure 14. Moreover, higher lateral acceleration and yaw rate values are achieved without losing stability as illustrated in Figures 15 and 16.

Figure 16 illustrates the yaw rate evolution of the vehicle. As it can be seen, the proposed intelligent Torque Vectoring approach provides a correct tracking of the reference, while reducing the overshoots. This allows increasing the cornering ability of the vehicle, reducing the risk of under- and oversteering. In fact, the difference between the yaw rate

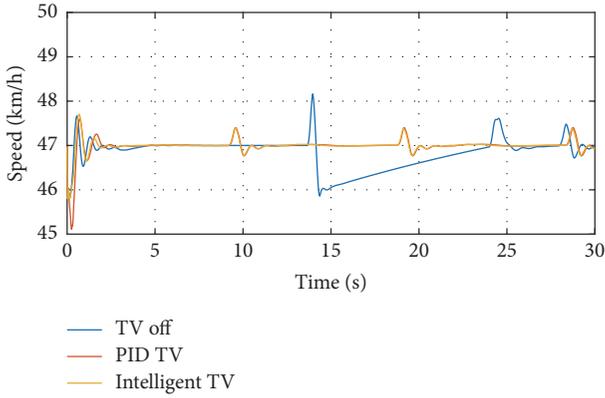


FIGURE 14: Vehicle speed.

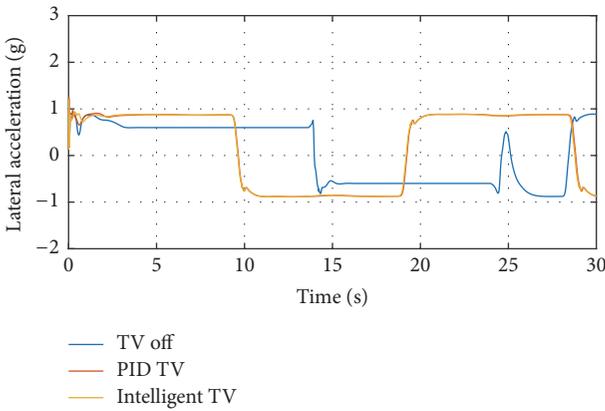


FIGURE 15: Vehicle lateral acceleration.

TABLE 9: Mechanical energy comparison.

	Energy mech [kWh]
No TV	0.4219
PID TV	0.2091
Intelligent TV	0.1895

reference and the obtained values for the case where the TV is deactivated shows understeering behaviour as has been explained.

Moreover, if a mechanical energy consumption analysis is carried out, the proposed intelligent Torque Vectoring approach allows not only a correct yaw rate tracking, increased cornering ability, and reduced slip ratio, but also an increase in the efficiency of the vehicle. Efficiency results are shown in Table 9, where the proposed approach is compared with the PID based TV and no TV cases. As can be seen, an increase of 10% can be achieved.

After analyzing the skid-pad performance, the double lane change scenario will be studied. In order to perform this test, an initial speed of 50 km/h has been selected, and a constant torque reference has been applied to the motors (2300 Nm total torque). This provides a longitudinal acceleration of 0.35 g approximately, allowing to obtain a final

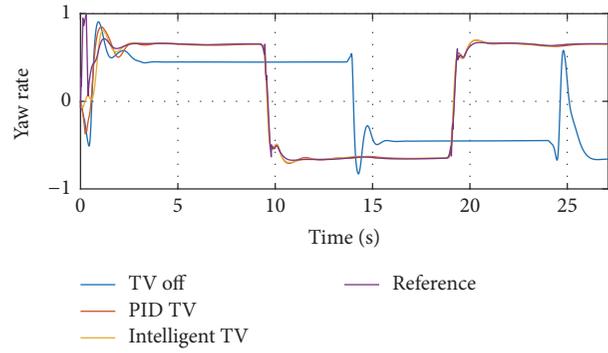


FIGURE 16: Yaw rate tracking.

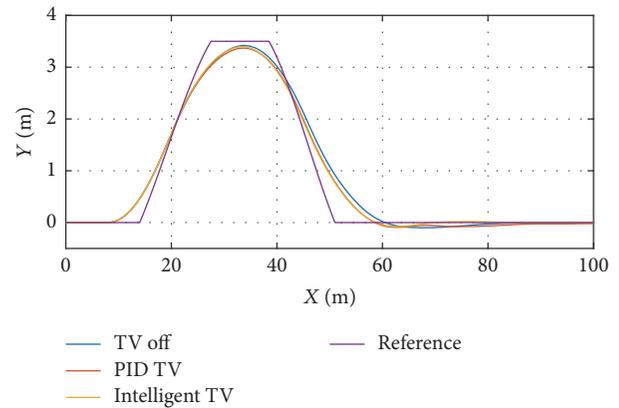


FIGURE 17: Double lane change trajectory.

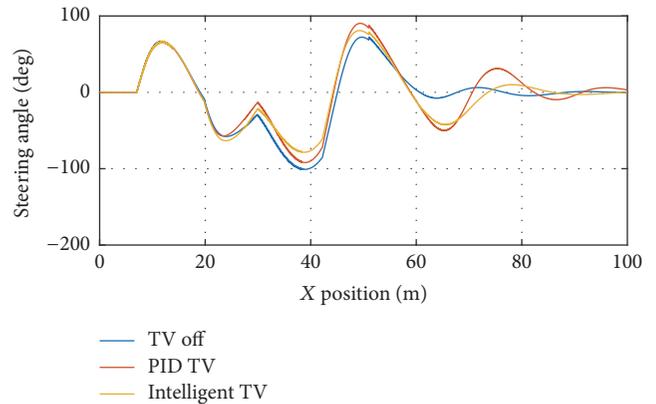


FIGURE 18: Steering angle.

speed of almost 90 km/h, covering the most common speed range of passenger vehicles in medium speed roads.

Simulation results for this scenario are shown from Figures 17–20. Figure 17 shows the trajectory followed by the vehicle for each case. As can be seen in this figure, the trajectory when the proposed TV approach is activated is closer to the lateral double lane change manoeuvre reference, demonstrating a vehicle handling improvement in such a challenging manoeuvre. In addition, Figure 18 shows that this

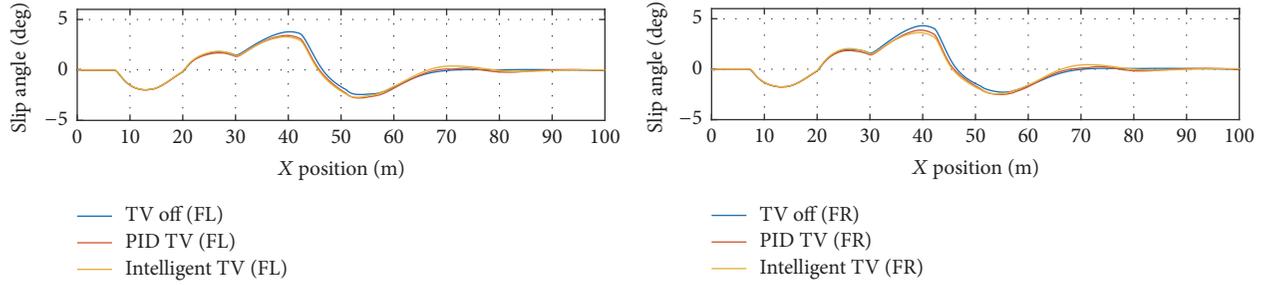


FIGURE 19: Wheels slip angles.

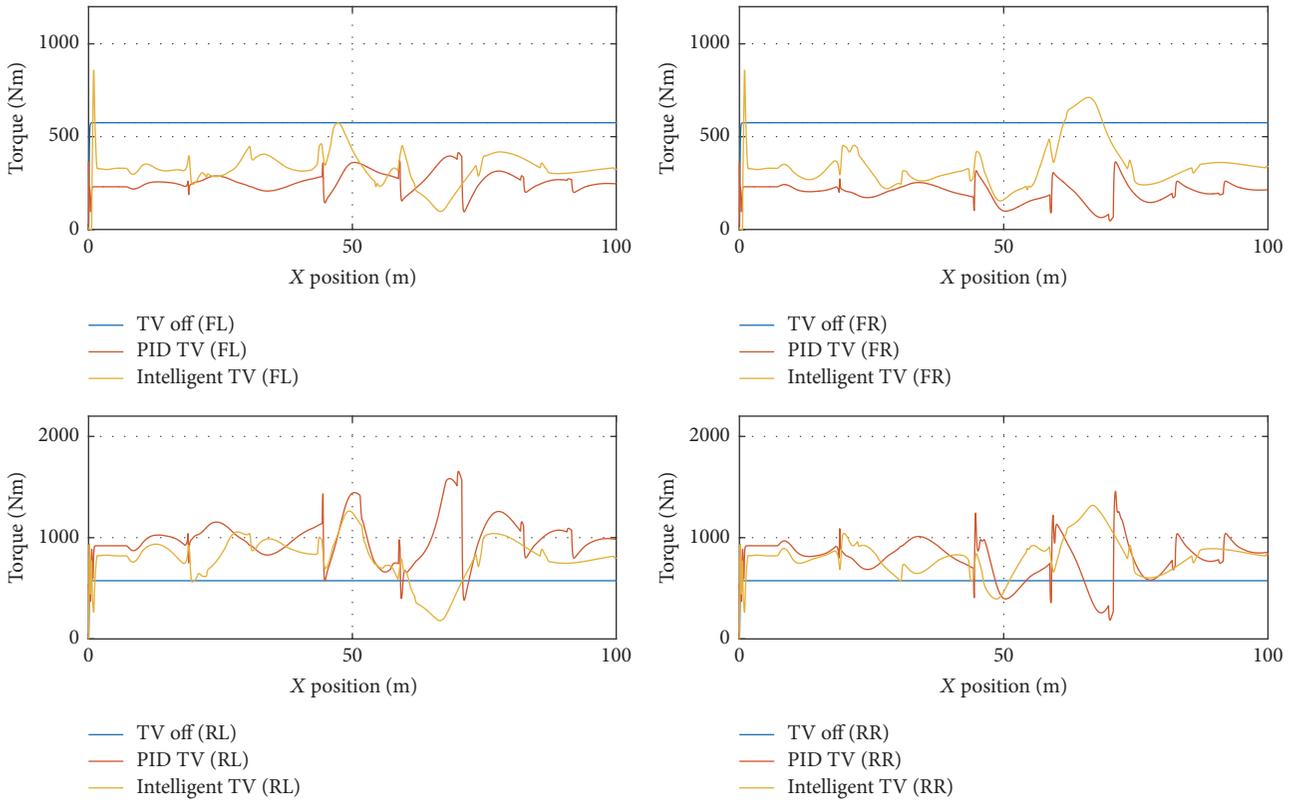


FIGURE 20: Wheel torque.

better tracking is achieved with lower values of steering angle, which means a reduction of the slip angle of about 1.5° as can be seen in Figure 19. The maximum reduction of this value is achieved by the proposed intelligent TV controller, which allows maximizing the forces transmitted to the road. Finally, wheel torque is illustrated in Figure 20, showing the correct torque distribution made by the controller designed.

5. Conclusions

The development of real-time capable, accurate, and efficient ADAS is a key issue for the development of vehicles with independent in-wheel motors. In this work a novel intelligent Torque Vectoring (TV) system, composed of a neuro-fuzzy vertical tire forces estimator and a fuzzy yaw moment controller, has been proposed.

The proposed approach considers both lateral and longitudinal torque distributions. The longitudinal distribution is based on a neuro-fuzzy vertical tire forces estimator that is based exclusively on measurable variables, which is an important contribution compared with the existing estimators. The estimated forces are used to determine the percentage of torque to be applied to the wheels of the rear and front axles, so that the maximum grip can be achieved.

On the other hand, the lateral torque distribution is achieved using a fuzzy yaw moment controller. This controller allows distributing the torque laterally (right and left wheels), to minimize wheel slip and enhance cornering capabilities. The overall torque distribution is calculated by taking into account both distributions.

Results demonstrate the ability to enhance vehicle dynamics of the intelligent Torque Vectoring System in

various scenarios. On the one hand, it was able to increase the stability in an evasive manoeuvre, such as double lane change, allowing the vehicle to follow better the desired trajectory, which is a critical safety issue in such manoeuvre. On the other hand, in the skid-pad test, a significant wheel slip ratio and slip angle reduction (19% and 23°, resp.) have been shown, resulting in an understeering behaviour reduction. This has allowed the vehicle to better match the yaw rate reference (33% error reduction) and then be able to follow the desired trajectory, demonstrating the cornering improvement provided by the correct torque distribution. Additionally, the proposed intelligent TV algorithm presents an improvement regarding a more traditional approach of the state of the art, providing more efficient driving (10% mechanical energy consumption reduction).

Future work will include a more sophisticated design for the use of the estimated tire vertical forces in the intelligent Torque Vectoring controller, resulting in a more elaborate controller, to improve its performance. Moreover, the implementation of the TV System in the logical part of a SoC will be considered in order to decrease its cycle time.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The research leading to these results has been supported by the ECSEL Joint Undertaking under Grant agreement no. 662192 (3Ccar). This Joint Undertaking receives support from the European Union Horizon 2020 research and innovation program and the ECSEL member states.

References

- [1] ACEA, European Automobile Manufacturers Association - New passenger car registrations by alternative fuel type in the European Union Q4 2015, 2016.
- [2] A. Al-Hmouz, J. Shen, R. Al-Hmouz, and J. Yan, "Modeling and simulation of an Adaptive Neuro-Fuzzy Inference System (ANFIS) for mobile learning," *IEEE Transactions on Learning Technologies*, vol. 5, no. 3, pp. 226–237, 2012.
- [3] M. Nikowitz, *Advanced Hybrid and Electric Vehicles*, Springer, 2016.
- [4] A. Broggi, A. Zelinsky, Ü. Özgüner, and C. Laugier, "Intelligent Vehicles," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., Springer, Cham, Switzerland, 2016.
- [5] S. Inoue, T. Ozawa, H. Inoue, P. Raksincharoensak, and M. Nagai, "Cooperative lateral control between driver and ADAS by haptic shared control using steering torque assistance combined with direct yaw moment control," in *Proceedings of the 19th IEEE International Conference on Intelligent Transportation Systems (ITSC '16)*, pp. 316–321, Brazil, November 2016.
- [6] L. De Novellis, A. Sorniotti, P. Gruber, and A. Pennycott, "Comparison of feedback control techniques for torque-vectoring control of fully electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 8, pp. 3612–3623, 2014.
- [7] K. Shimada and Y. Shibahata, "Comparison of three active chassis control methods for stabilizing yaw moments," *SAE Technical Papers*, 1994.
- [8] L. De Novellis, A. Sorniotti, P. Gruber, L. Shead, V. Ivanov, and K. Hoeppeing, "Torque vectoring for electric vehicles with individually controlled motors: State-of-the-art and future developments," *World Electric Vehicle Journal*, vol. 5, no. 2, pp. 617–628, 2012.
- [9] A. Haddoun et al., "Sliding mode control of EV electric differential system," in *Proceedings of the ICEM*, Chania, Greece.
- [10] A. Haddoun, M. E. H. Benbouzid, D. Diallo, R. Abdessemed, J. Ghouili, and K. Srairi, "Modeling, analysis, and neural network control of an EV electrical differential," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2286–2294, 2008.
- [11] K. Jalali, T. Uchida, J. Mcphee, and S. Lambert, "Integrated stability control system for electric vehicles with in-wheel motors using soft computing techniques," *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, vol. 2, no. 1, pp. 109–119, 2009.
- [12] H. Kahveci, H. I. Okumus, and M. Ekici, "An electronic differential system using fuzzy logic speed controlled in-wheel brushless DC motors," in *Proceedings of the 4th International Conference on Power Engineering, Energy and Electrical Drives*, pp. 881–885, Turkey, May 2013.
- [13] D. E. Williams and W. M. Haddad, "Nonlinear control of roll moment distribution to influence vehicle yaw characteristics," *IEEE Transactions on Control Systems Technology*, vol. 3, no. 1, pp. 110–116, Mar 1995.
- [14] Y. Hu, L. Yang, B. Yan, T. Yan, and P. Ma, "An online rolling optimal control strategy for commuter hybrid electric vehicles based on driving condition learning and prediction," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 4312–4327, 2016.
- [15] I. Yang, S. Byun, B. Seo, D. Lee, and D. S. Han, "Integrated control systems of active front steering and direct yaw moment control using dynamic inversion," in *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IEEE IV '13)*, pp. 1303–1306, Australia, June 2013.
- [16] C. Shen, G. Yin, and X. Jin, "Study on the active steering system with direct yaw-moment control for electric vehicle with electric motored wheels," in *Proceedings of the 35th Chinese Control Conference (CCC '16)*, pp. 8945–8950, China, July 2016.
- [17] E. Siampis, E. Velenis, and S. Longo, "Model Predictive torque vectoring control for electric vehicles near the limits of handling," in *Proceedings of the European Control Conference (ECC '15)*, pp. 2553–2558, Austria, 2015.
- [18] H. B. Pacejka, "Tyre and Vehicle Dynamics, BH, 2005".
- [19] J. Kim, H. Lee, and S. Choi, "A robust road bank angle estimation based on a proportional-integral Hfilter," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 226, no. 6, pp. 779–794, 2012.
- [20] R. Ghandour, A. Victorino, M. Doumiati, and A. Charara, "Tire/road friction coefficient estimation applied to road safety," in *Proceedings of the 18th Mediterranean Conference on Control and Automation (MED '10)*, pp. 1485–1490, June 2010.
- [21] W. Cho, J. Yoon, S. Yim, B. Koo, and K. Yi, "Estimation of tire forces for application to vehicle stability control," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 638–649, 2010.
- [22] M. Doumiati, A. Victorino, D. Lechner, G. Baffet, and A. Charara, "Observers for vehicle tyre/road forces estimation:

- Experimental validation,” *Vehicle System Dynamics*, vol. 48, no. 11, pp. 1345–1378, 2010.
- [23] S. Antonov, A. Fehn, and A. Kugi, “Unscented Kalman filter for vehicle state estimation,” *Vehicle System Dynamics*, vol. 49, no. 9, pp. 1497–1520, 2011.
- [24] M. Doumiati, A. Victorino, A. Charara, and D. Lechner, “Lateral load transfer and normal forces estimation for vehicle safety: Experimental test,” *Vehicle System Dynamics*, vol. 47, no. 12, pp. 1511–1533, 2009.
- [25] A. Rabhi, N. K. M’Sirdi, and A. Elhajjaji, “Estimation of contact forces and tire road friction,” in *Proceedings of the Mediterranean Conference on Control and Automation*, pp. 1–6, 2007.
- [26] M. Acosta and S. Kanarachos, “Tire lateral force estimation and grip potential identification using Neural Networks, Extended Kalman Filter, and Recursive Least Squares,” *Neural Computing and Applications*, pp. 1–12, 2017.
- [27] R. Jayachandran, S. D. Ashok, and S. Narayanan, “Fuzzy logic based modelling and simulation approach for the estimation of tire forces,” in *Proceedings of the 2013 International Conference on Design and Manufacturing*, vol. 64, pp. 1109–1118, India, July 2013.
- [28] H. Wallentowitz, *Vertical and Lateral Dynamics of Passenger Vehicles*, Automotive Technology II, Institute of Automotive Engineering, Aachen University of Technology, Germany, 2005.
- [29] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [30] M. Blej and M. Azizi, “Comparison of Mamdani-type and Sugeno-type fuzzy inference systems for fuzzy real time scheduling,” *International Journal of Applied Engineering Research*, vol. 11, no. 22, pp. 11071–11075, 2016.
- [31] D. L. Milliken and W. F. Milliken, “Race Car Vehicle Dynamics,” *Society of Automotive Engineers*, 1995.
- [32] Y. Bai and D. Wang, “Fundamentals of Fuzzy Logic Control - Fuzzy Sets, Fuzzy Rules and Defuzzifications,” in *Advanced Fuzzy Logic Technologies in Industrial Applications, Advances in Industrial Control*, Y. Bai, H. Zhuang, and D. Wang, Eds., Springer, London, UK, 2006.
- [33] J. S. R. Jang, “ANFIS: adaptive-network-based fuzzy inference system,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [34] M. Asadi, “Optimized Mamdani fuzzy models for predicting the strength of intact rocks and anisotropic rock masses,” *Journal of Rock Mechanics and Geotechnical Engineering*, vol. 8, no. 2, pp. 218–224, 2016.
- [35] K. M. Passino and S. Yurkovich, *Fuzzy Control*, 1997.
- [36] Dynacar by Tecnalía, <http://www.dynacar.es/en/home.php>.
- [37] A. Peña, I. Iglesias, J. J. Valera, and A. Martín, “Development and validation of Dynacar RT software, a new integrated solution for design of electric and hybrid vehicles,” in *Proceedings of the 26th Electric Vehicle Symposium (EVS 2012)*, pp. 2026–2032, USA, May 2012.
- [38] M. Dendaluce, I. Iglesias, A. Martín, P. Prieto, and A. Peña, “Race-Track testing of a torque vectoring algorithm on a motor-in-wheel car using a model-based methodology with a hil and multibody simulator setup,” in *Proceedings of the 19th IEEE International Conference on Intelligent Transportation Systems (ITSC 2016)*, pp. 2500–2505, Brazil, November 2016.
- [39] J. Cuadrado, D. Vilela, I. Iglesias, A. Martín, and A. Peña, “A multibody model to assess the effect of automotive motor in-wheel configuration on vehicle stability and comfort,” in *Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics*, pp. 1083–1092, 2013.
- [40] <https://www.fsaeonline.com/>.
- [41] <https://www.iso.org/standard/57253.html>.
- [42] Ministerio de Transportes y Comunicaciones, *Manual De Carreteras Diseno Geometrico*, 2014.
- [43] A. Rezaeian, R. Zarringhalam, S. Fallah et al., “Novel tire force estimation strategy for real-time implementation on vehicle applications,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2231–2241, 2015.

Research Article

WH-EA: An Evolutionary Algorithm for Wiener-Hammerstein System Identification

J. Zambrano ¹, J. Sanchis ², J. M. Herrero ² and M. Martínez ²

¹Universidad Politécnica Salesiana, Cuenca, Ecuador

²Instituto Universitario de Automática e Informática Industrial, Universitat Politecnica de Valencia, València, Spain

Correspondence should be addressed to J. Zambrano; jzambranoa@ups.edu.ec

Received 22 November 2017; Accepted 27 December 2017; Published 20 February 2018

Academic Editor: José Manuel Andújar

Copyright © 2018 J. Zambrano et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Current methods to identify Wiener-Hammerstein systems using Best Linear Approximation (BLA) involve at least two steps. First, BLA is divided into obtaining front and back linear dynamics of the Wiener-Hammerstein model. Second, a refitting procedure of all parameters is carried out to reduce modelling errors. In this paper, a novel approach to identify Wiener-Hammerstein systems in a single step is proposed. This approach is based on a customized evolutionary algorithm (WH-EA) able to look for the best BLA split, capturing at the same time the process static nonlinearity with high precision. Furthermore, to correct possible errors in BLA estimation, the locations of poles and zeros are subtly modified within an adequate search space to allow a fine-tuning of the model. The performance of the proposed approach is analysed by using a demonstration example and a nonlinear system identification benchmark.

1. Introduction

Nonlinearities are present to a greater or lesser extent in all real processes. When nonlinearities are weak, linear models can be successfully used to forecast the evolution of variables or to design control schemes. Currently, a lot of methods to build linear models can be found in the literature [1–5]. However, when nonlinearities are hard, linear models just can be used in a specific operation range. If process operating range is large, cause-effect relationship should be represented by a nonlinear model. An alternative to nonlinear system modelling is to describe the process phenomena using rigorous first-principles formulation [6–8]; nevertheless, in most cases, it can be a very challenging task. Another alternative is the use of soft computing methods for process identification. In this framework, nonlinear system identification has attracted considerable interest of researchers over the past few years. Nowadays, nonlinear identification is an open research topic where some benchmark problems have been proposed [9–12] and real measurement data are available for testing and validate different nonlinear identification methods (e.g., DaISy database [13]).

One of the most challenging problems regarding nonlinear system identification is the selection of a good model structure. Currently there are several structures based on neural networks [14], block-oriented models [15, 16], Volterra series [17], NARMAX models [18], and fuzzy models [19], among others. A review of black box methods to nonlinear identification can be found in Suykens and Vandewalle [20].

In this paper, block-oriented models are considered, which are a class of nonlinear representations consisting of linear time-invariant (LTI) systems coupled with nonlinear static functions (NL) [21]. Within this class of models, the most popular ones are Wiener (LTI-NL), Hammerstein (NL-LTI), Wiener-Hammerstein (LTI-NL-LTI), and Hammerstein-Wiener (NL-LTI-NL) models [15]. Nowadays, several methods to identify these models can be found in the literature. An interesting classification of contributions that have been developed until the last decade can be found in Lopes dos Santos et al. [22].

Block-oriented models are attractive for their simplicity and great capabilities to model nonlinear dynamic systems [23–28]. Specifically, Wiener-Hammerstein models have proved to be able describe several systems like a paralyzed

skeletal muscle [29, 30], a limb reflex control system [31], a DC-DC converter [32], a heat exchanger system and a superheater-desuperheater in a boiler system [33], and a thermal process [34], among others [35]. Not only does the study of block-oriented models address parameters estimation, but also these structures are used to implement modern control strategies [36–40].

In the context of block-oriented models, knowledge of process dynamics can be a good starting point for identification [52]. In this regard, the Best Linear Approximation or BLA of a nonlinear system [52–55] can be used. For the specific case of Wiener-Hammerstein models, the BLA does not provide information about the dynamics of each LTI block. Therefore, all BLA-based Wiener-Hammerstein identification methods have concentrated their efforts on the BLA division to generate initial estimates and avoid suboptimal local minima in an optimization procedure. In Sjöberg et al. [45], both LTI subsystems are initialized with all possible BLA partitions and least squares optimization is applied to fit the nonlinearity. Although identification results are very good, the number of possible partitions (combinations) grows with the number of poles and zeros of the BLA and therefore the computational cost required for this method can be very high.

To avoid multiple BLA divisions, in Lauwers [44] and Sjöberg et al. [45] an “advanced” method is proposed where both LTI subsystems are overparameterized with all poles and zeros of the BLA. This method is formulated as a linear-in-the-parameters total-least-squares problem for which the back LTI subsystem is inverted and basis functions are used to represent both linear subsystems. To minimize the effect of overparameterization, an order reduction technique is applied. However, since the formulation is based on neglecting the effect of disturbances, the solution is in general not consistent if there is noise on the output. In addition, the BLA is required to be invertible.

Another approach to initialize Wiener-Hammerstein models is presented by Westwick and Schoukens [42], where the poles and/or zeros of the BLA are classified by using a nonlinear transformation of the input and the output residuals (quadratic/cubic BLA). On the same context of QBLA/CBLA and in line with “brute-force” method, Westwick and Schoukens [46] propose a scanning technique for a rapid evaluation of all possible BLA partitions between both LTI blocks of the Wiener-Hammerstein system. With this evaluation, the vast majority of possible partitions are discarded. Both proposals based on QBLA/CBLA show excellent results and overcome some disadvantages of “brute-force” and “advanced” methods; however, the QBLA/CBLA estimation can be difficult (high variance) since it is estimated from the BLA residuals.

In a more recent work, Schoukens et al. [41] propose a more robust method based on QBLA/CBLA. Unlike the two previous proposals, the BLA is split into a nonparametric framework. This avoids mainly the parameterization of the QBLA that can be tedious given that the number of poles and zeros tends to be high. Once the front and the back dynamics of the Wiener-Hammerstein model have been identified, a parameterization of both LTI blocks is required in an additional step. This step can be complicated because a linear phase shift can be present in the nonparametric estimate.

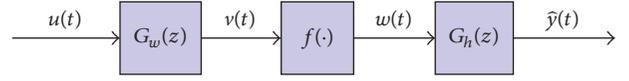


FIGURE 1: Wiener-Hammerstein model.

To avoid QBLA/CBLA estimation, in Vanbeylen [43] a fractional model parameterization based in multiplicities (powers) of poles and zeros is presented. The problem is formulated in the frequency domain and fractional exponents indicate which poles and zeros belong to each subsystem after an optimization problem is solved. Once the poles and zeros of the BLA have been classified, both LTI blocks must be parameterized in an additional step.

All methods mentioned here, each with its advantages and disadvantages, identify Wiener-Hammerstein models from the BLA. However, all require high user interaction to parameterize the LTI blocks and/or a final optimization to refit all parameters of the Wiener-Hammerstein model. In this work, we show that it is possible to obtain a good Wiener-Hammerstein model from the BLA by solving a single optimization problem, where user interaction is only required at the beginning, just configuring simple parameters of an evolutionary algorithm.

Hereinafter, this paper is organized as follows. In Section 2, Wiener-Hammerstein systems formulation is revisited together with some relevant information about the BLA. The proposed evolutionary algorithm, WH-EA, is presented and described in detail in Section 3, while its application and results on a numerical example and on the benchmark data SYSID’09 are presented in Section 4. Finally, in Section 5, some conclusions are reported.

2. Background

2.1. Wiener-Hammerstein Model. A Wiener-Hammerstein model consists of two LTI subsystems $G_w(z)$ and $G_h(z)$ surrounding a static nonlinear function $f(v(t))$ (Figure 1). Both LTI subsystems can be represented in the discrete-time domain as rational transfer functions in factorized form:

$$\begin{aligned} v(t) &= G_w(z) u(t) \\ &= K_w \frac{\prod_{i=1}^{n_b} (z - z_{w_i}) / (1 - z_{w_i})}{\prod_{i=1}^{n_a} (z - p_{w_i}) / (1 - p_{w_i})} u(t), \\ \hat{y}(t) &= G_h(z) w(t) \\ &= K_h \frac{\prod_{i=1}^{n_d} (z - z_{h_i}) / (1 - z_{h_i})}{\prod_{i=1}^{n_c} (z - p_{h_i}) / (1 - p_{h_i})} w(t), \end{aligned} \quad (1)$$

where z is the discrete-time operator, K_w , $p_{w_1} \cdots p_{w_{n_a}}$ and $z_{w_1} \cdots z_{w_{n_b}}$ represent the static gain, poles, and zeros of the front LTI block, respectively, and K_h , $p_{h_1} \cdots p_{h_{n_c}}$ and $z_{h_1} \cdots z_{h_{n_d}}$ represent the static gain, poles, and zeros of the back LTI block, respectively.

The nonlinearity can be represented as a linear combination of a finite set (M) of basis functions:

$$w(t) = f(v(t)) = \sum_{m=1}^M \beta_m f_m(v(t)), \quad (2)$$

where $v(t)$ and $w(t)$ are the input and output of the static nonlinearity, β_m are weighting parameters to be estimated, and f_m are basis functions.

From (1) and (2), the output of the Wiener-Hammerstein model is analytically related to the input through the following expression:

$$\hat{y}(t, \theta) = G_h(z, \theta_h) f(\theta_{NL}, G_w(z, \theta_w) u(t)), \quad (3)$$

where

$$\begin{aligned} \theta_w &= [K_w, z_{w_1}, z_{w_2} \cdots z_{w_b}, p_{w_1}, p_{w_2} \cdots p_{w_a}], \\ \theta_{NL} &= [\beta_1, \beta_2, \dots, \beta_m], \\ \theta_h &= [K_h, z_{h_1}, z_{h_2} \cdots z_{h_d}, p_{h_1}, p_{h_2} \cdots p_{h_c}], \\ \theta &= [\theta_w, \theta_{NL}, \theta_h]. \end{aligned} \quad (4)$$

The challenge is to find the best θ so that the predicted output $\hat{y}(t, \theta)$ is as close as possible to the measured output $y(t)$. Without prior knowledge of the system, this identification problem is not easy to solve, because there are some inconveniences that must be overcome:

- (i) Parameters n_a , n_b , n_c , and n_d are not known (i.e., the structure of the LTI blocks is unknown).
- (ii) The order and basis functions for nonlinearity are not known.
- (iii) Without adequate initial values of θ , it is quite possible that the optimization process, trying to find the best θ , gets stuck in a local minimum.
- (iv) Internal variables $v(t)$ and $w(t)$ are not measurable.

As a complement to the formulation presented, the following assumptions are made about the system.

Assumption 1. The nonlinear system to be identified can be described by (3).

Assumption 2. The Wiener-Hammerstein system will be identified from an input/output data set $\{u(t), y(t)\}_{t=1}^N$. The input signal $u(t)$ is Gaussian or equivalent (see Section 2.2 for more details), while the measured output $y(t)$ may be corrupted by stationary additive noise $n(t)$. It is further assumed that the noise is independent of the input excitation signal:

$$y(t) = y_0(t) + n(t). \quad (5)$$

Assumption 3. There is no cancellation of poles and zeros and all poles of both LTI subsystems must be within the unit circle.

Assumption 4. Nonlinearity is static and its current output $w(t)$ only depends on the current input $v(t)$ (i.e., the nonlinearity has no memory).

2.2. The Best Linear Approximation (BLA) of a Wiener-Hammerstein System. The BLA of a nonlinear system for a given class of excitation signals is a linear model that minimizes the expected mean square error between the true output of the nonlinear system and the output of the linear model [56]:

$$G_{BLA}(z) = \arg \min_{G(z)} E[|y(t) - G(z)u(t)|^2], \quad (6)$$

where $u(t)$ is the input that excites the nonlinear system, $y(t)$ is the measured output, and E is the expectation operator. An alternative way to obtain the BLA of a nonlinear system is in a nonparametric framework:

$$G_{BLA}(j\omega_k) = \frac{S_{yu}(j\omega_k)}{S_{uu}(j\omega_k)}, \quad (7)$$

where $S_{yu}(j\omega_k)$ is the cross-power spectrum between the output $y(t)$ and the input $u(t)$ and $S_{uu}(j\omega_k)$ is the auto power spectral density of $u(t)$ [54, 57].

The BLA depends on the excitation power spectrum (bandwidth and amplitude level) and excitation probability density function. Therefore, obtaining the BLA is restricted to the type of input signal that excites the process. Most estimation methods to obtaining the BLA use Gaussian noise signals or equivalent [58].

When a nonlinear system is excited with a Gaussian signal or equivalent, according to Bussgang's theorem [59], the nonlinearity can be replaced by a constant (K_{NL}). Therefore, in the specific case of a Wiener-Hammerstein system, the BLA can be defined by the following expression:

$$G_{BLA}(z) = K_{NL} G_{wh}(z), \quad (8)$$

where $G_{wh}(z)$ represents the dynamics of the nonlinear system:

$$G_{wh}(z) = \frac{\prod_{i=1}^{n_b+n_d} (z - z_i) / (1 - z_i)}{\prod_{i=1}^{n_a+n_c} (z - p_i) / (1 - p_i)} u(t). \quad (9)$$

It is evident that $p_1 \cdots p_{n_a+n_c}$ and $z_1 \cdots z_{n_b+n_d}$ are the poles and zeros that must be assigned to $G_w(z)$ and $G_h(z)$. Although the BLA does not provide information to distinguish the dynamics between both LTI subsystems, knowledge of the overall dynamics of a Wiener-Hammerstein system is a good starting point to identify such systems.

3. The Evolutionary Algorithm (WH-EA)

In this paper, the identification of a Wiener-Hammerstein system is addressed as an optimization problem, which is formulated considering the following issues:

- (i) The BLA is estimated in the first instance.
- (ii) The poles and zeros of the BLA must be classified to find the dynamics of the front and back of the Wiener-Hammerstein model.
- (iii) The pole-zero locations of the BLA can change moderately to improve modelling errors.

- (iv) Without loss of generality, it is possible to model a Wiener-Hammerstein system considering that both linear blocks have unit gain (gains from $G_w(z)$ and $G_h(z)$ are not part of the optimization problem).
- (v) The nonlinear static function is modelled as a piecewise function represented by a set of points in the v, w plane.

To explain in detail how WH-EA works, this section has been divided into three parts. First part explains how Wiener-Hammerstein model is coded for individuals in the population; in addition, the optimization problem statement is presented. Second part explains in detail the customized genetic operators developed. Finally, the third part explains the general procedure of WH-EA.

3.1. Optimization Problem Statement. Changing pole/zero locations of the BLA to improve modelling error implies new estimates around the known values. These locations for both linear subsystems are coded in a single vector as follows:

$$\mathbf{P} = [z_{c_1}, \dots, z_{c_{nc}}, z_{r_1}, \dots, z_{r_{nr}}, z_{i_1}, \dots, z_{i_{nc}}, p_{c_1}, \dots, p_{c_{mc}}, p_{r_1}, \dots, p_{r_{mr}}, p_{i_1}, \dots, p_{i_{mc}}], \quad (10)$$

where $z_{r_1}, \dots, z_{r_{nr}}$ and $p_{r_1}, \dots, p_{r_{mr}}$ contain the locations of the real zeros and poles, respectively, and $z_{c_1}, \dots, z_{c_{nc}}$ and $z_{i_1}, \dots, z_{i_{nc}}$ contain the real and imaginary parts of complex conjugate zeros, respectively, while $p_{c_1}, \dots, p_{c_{mc}}$ and $p_{i_1}, \dots, p_{i_{mc}}$ contain the real and imaginary parts of complex conjugate poles, respectively. The values of nc , nr , mc , and mr depend on the number of zeros and poles (real and/or complex conjugates) of the BLA.

Poles and zeros contained in (10) must be classified to obtain the dynamics of the front and back blocks of a Wiener-Hammerstein model. This classification is performed using a binary vector:

$$\mathbf{C} = [xz_1, \dots, xz_{nc+nr}, xp_1, \dots, xp_{mc+mr}]. \quad (11)$$

The first part of the vector, $\mathbf{C} (xz_1, \dots, xz_{nc+nr})$, is associated with $z_{c_1}, \dots, z_{c_{nc}}, z_{r_1}, \dots, z_{r_{nr}}$ and indicates the zeros classification, while its second part, $\mathbf{C} (xp_1, \dots, xp_{mc+mr})$, is associated with $p_{c_1}, \dots, p_{c_{mc}}, p_{r_1}, \dots, p_{r_{mr}}$ indicating the poles classification. Note that imaginary parts are not considered for classification since they are already associated with their corresponding real parts. It is assumed that if $xz_{ith} = 1$, the corresponding i th element of \mathbf{P} with $i = 1, \dots, nc + nr$ (i.e., a real zero or a pair of complex conjugated zeros) will belong to the subsystem $G_w(z)$; otherwise, it will belong to the subsystem $G_h(z)$. In the same way, this correspondence can be applied to classify the poles using xp_1, \dots, xp_{mc+mr} .

For example, if a nonlinear system is approximated by a BLA with four poles, $p_{1,2} = -0.32 \pm 0.77j$, $p_3 = -0.11$, and $p_4 = 0.17$, and three zeros, $z_{1,2} = 1.41 \pm 0.56j$, $z_3 = 1.1$, then $nc = 1$, $nr = 1$, $mc = 1$, and $mr = 2$, \mathbf{P} would be structured as $[1.41, 1.1, 0.56, -0.32, -0.11, 0.17, 0.77]$, and vector \mathbf{C} should contain five elements whose values switch between zero and one as the algorithm evolves. By way of illustration if $\mathbf{C} = [1, 0, 0, 1, 1]$, then $G_w(z)$ would have two zeros and two poles:

$z_{1,2} = 1.41 \pm 0.56j$, $p_3 = -0.11$, $p_4 = 0.17$, while $G_h(z)$ would have a zero and two poles: $z_3 = 1.1$, $p_{1,2} = -0.32 \pm 0.77j$.

With respect to nonlinear static function, let us consider that it is represented by a set of n points:

$$\mathbf{B} = [v_1, \dots, v_n, w_1, \dots, w_n], \quad (12)$$

where the pairs $(v_1, w_1), \dots, (v_n, w_n)$ correspond to their coordinates in a two-dimensional v - w plane. The location of these points and the interpolation method used will determine the quality of the captured static nonlinearity.

The proposed evolutionary algorithm is based on stochastic population of candidate solutions (individuals). Each individual contains genetic information related to

- (i) the pole/zero locations in the Z -plane of the linear subsystems (\mathbf{P}),
- (ii) the point coordinates representing the nonlinear static function (\mathbf{B}),
- (iii) and the pole/zero classification for blocks $G_w(z)$ and $G_h(z)$ (\mathbf{C}),

such that any Wiener-Hammerstein model ((1) and (2)) can be easily described from this coded information. Recall that gains from linear blocks are assumed to be 1 and that parameters n_a , n_b , n_c , and n_d will be implicitly optimized and they will depend on the structure of vector \mathbf{C} .

To find the best set of parameters, an optimization problem is stated based on a prediction-error method and the typical mean-squared error criterion (although any other criteria can be used in the proposed method, such as the mean absolute or maximum error criteria):

$$\varepsilon(t, \boldsymbol{\theta}) = y(t) - \hat{y}(t, \boldsymbol{\theta}), \quad (13)$$

$$J(\boldsymbol{\theta}) = \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t, \boldsymbol{\theta}), \quad (14)$$

where $\boldsymbol{\theta} = [\mathbf{P}, \mathbf{B}, \mathbf{C}]$ and the solution of the optimization problem is stated as

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}), \quad (15)$$

where $\hat{\boldsymbol{\theta}}$ contains the genetic information from the best individual at the end of generations.

3.2. Genetic Operators. Customized mutation and crossover operators will be developed taking in mind the problem at hand: to identify all parameters of the Wiener-Hammerstein model in a single optimization trial. Figure 2 shows the structure of an individual as well as the genetic operators developed on each piece of genetic information. Note that i and g have been introduced into the formulation. Subscript i represents an individual in the population, while the superscript g indicates the current population.

The specific mutation and crossover operators designed are randomly selected to maintain a balance between exploration and exploitation of the search space. Mutation operations are used to maintain genetic diversity, while crossover

<p style="text-align: center;">Pole-zero locations</p> <p style="text-align: center;">Zeros</p> $P_i^g = \left\{ \left[\begin{array}{cc} zc_1, \dots, zc_{nc} & zr_1, \dots, zr_{nr} \\ \text{Real values} & \text{Imag. values} \end{array} \right] \dots \right.$ <p style="text-align: center;">Poles</p> $\dots \left[\begin{array}{cc} pc_1, \dots, pc_{mc} & pr_1, \dots, pr_{mr} \\ \text{Real values} & \text{Imag. values} \end{array} \right] \left. \right\}$	<p>Mutation M.1</p> <p>Crossover C.1</p>
<p style="text-align: center;">Static nonlinearity</p> $B_i^g = \left\{ \left[\begin{array}{cc} v_1, \dots, v_n & w_1, \dots, w_n \\ \text{Abscissa} & \text{Ordinate} \end{array} \right] \right\}$	<p>Mutation M.2</p> <p>Mutation M.3</p> <p>Crossover C.2</p>
<p style="text-align: center;">Pole-zero classification</p> $C_i^g = \left\{ \left[\begin{array}{cc} xz_1, \dots, xz_{nc+nr} & xp_1, \dots, xp_{mc+mr} \\ \text{Zeros } G_w \text{ or } G_h & \text{Poles } G_w \text{ or } G_h \end{array} \right] \right\}$	<p>Mutation M.4</p>

FIGURE 2: Structure of individual and genetic operations performed on each piece of genetic information.

operations allow genetic information from the best individuals to be combined and disseminated throughout the generations. Further details on how the algorithm works will be given in Section 3.3.

3.2.1. Location in the Z-Plane of Poles and Zeros. Theoretically in a Wiener-Hammerstein model, the pole-zero locations of $G_w(z)$ and $G_h(z)$ subsystems correspond to the pole-zero locations of the BLA; however, it is well known that once the BLA has been divided, a refit can be used to improve the modelling error. In this regard, the proposed algorithm considers that while the BLA is divided and nonlinearity is captured, the pole-zero locations can change subtly.

Both operations used on this portion of genetic information produce offspring vector $\tilde{\mathbf{P}}^g$, which directly inherits from its parent \mathbf{P}_i^g all the genetic information except in a gene. This gene will be selected using a random integer number $r_{zp} \in [1, \dots, nr + 2nc + mr + 2mc]$ and modified according to the corresponding genetic operator *mutation M.1* or *crossover C.1*.

Mutation M.1. The selected gene is mutated to explore in an individualized way new pole-zero locations of the BLA. A new location \tilde{P}_j^g is determined by a random number N_{zp} with Gaussian distribution:

$$\tilde{P}_j^g = \begin{cases} P_{i,j}^g + N_{zp}(0, \sigma^2(g)) & \text{if } (j = r_{zp}) \\ P_{i,j}^g & \text{otherwise,} \end{cases} \quad (16)$$

where $j = 1 \dots nr + 2nc + mr + 2mc$. $P_{i,j}^g$ and \tilde{P}_j^g represent the j th elements of vectors \mathbf{P}_i^g and $\tilde{\mathbf{P}}^g$, respectively.

The new locations for poles and zeros are explored within a search space defined by \mathbf{P}^{\min} and \mathbf{P}^{\max} ; therefore, $P_j^{\min} \leq \tilde{P}_j^g \leq P_j^{\max}$, where P_j^{\min} and P_j^{\max} are the j th elements of vectors \mathbf{P}^{\min} and \mathbf{P}^{\max} , respectively (see Section 3.3 for more details on search space for poles and zeros).

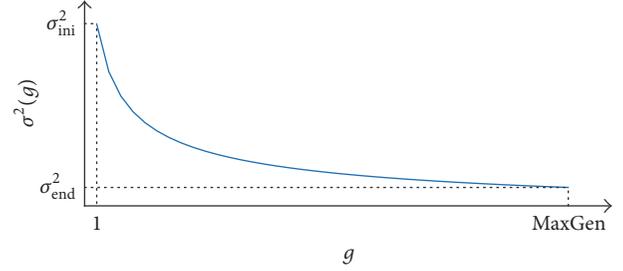


FIGURE 3: Variation of standard deviation over generations to control the aggressiveness of mutations.

Aggressiveness of mutations can be controlled through the standard deviation:

$$\sigma^2(g) = \frac{\Delta_s}{100} \left(\frac{\sigma_{ini}^2}{\sqrt{1 + g * \sigma_{ratio}^2}} \right), \quad (17)$$

$$\sigma_{ratio}^2 = \frac{(\sigma_{ini}^2 / \sigma_{end}^2)^2 - 1}{\text{MaxGen} - 1},$$

where MaxGen is the predefined number of algorithm generations; σ_{ratio}^2 is the rate at which the standard deviation will decrease from σ_{ini}^2 to σ_{end}^2 as the generations pass (see Figure 3); the parameter Δ_s bounds the limits of the interval in which the selected gene can be moved. For this mutation, $\Delta_s = P_{r_{zp}}^{\max} - P_{r_{zp}}^{\min}$. Variation of $\sigma^2(g)$ will allow mutations to be more subtle in the last generations to achieve a fine-tuning of the corresponding parameters.

Crossover C.1. The selected gene is formed using genetic information from the parent, $P_{i,j}^g$, combined with the corresponding genetic information from the best individual, $P_{best,j}^g$, in the current population:

$$\tilde{P}_j^g = \begin{cases} \frac{P_{i,j}^g + P_{best,j}^g}{2} & \text{if } (j = r_{zp}) \\ P_{i,j}^g & \text{otherwise,} \end{cases} \quad (18)$$

where $j = 1 \dots nr + 2nc + mr + 2mc$.

3.2.2. Nonlinear Static Function. As the algorithm evolves, points for nonlinear static function must be located adequately in the v - w plane. Here any type of interpolation can be used to capture the static nonlinearity. To achieve a good fit, mutations M.2 and M.3 plus a crossover operation are used. Both mutations used on this portion of genetic information produce offspring vector $\tilde{\mathbf{B}}^g$, which directly inherits from its parent \mathbf{B}_i^g all the genetic information except in two genes. This pair of genes represents the coordinates of the point that will be modified. Unlike mutation operations, the crossover operation generates offspring with a single modified gene which corresponds to the ordinate of a point. Given the correspondence between the abscissa and the ordinate of a point, for the three operations a single integer random

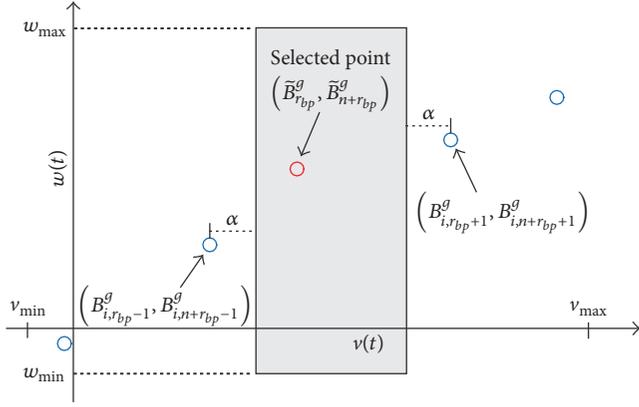


FIGURE 4: Bounds for mutation M.2. Grey area indicates $\tilde{B}_{r_{bp}}^g$ and $\tilde{B}_{n+r_{bp}}^g$ feasible space.

number $r_{bp} \in [1, n]$ will allow us to select the gene(s) to be modified.

Mutation M.2. This genetic operation allows us to explore in the v - w plane new positions for the points. The mutation in both genes is handled by random numbers (N_v, N_w) with Gaussian distribution:

$$\tilde{B}_j^g = \begin{cases} B_{i,j}^g + N_v(0, \sigma^2(g)) & \text{if } (j = r_{bp}) \\ B_{i,j}^g + N_w(0, \sigma^2(g)) & \text{if } (j = n + r_{bp}) \\ B_{i,j}^g & \text{otherwise} \end{cases} \quad (19)$$

with $j = 1 \dots 2n$. $B_{i,j}^g$ and \tilde{B}_j^g represent the j th elements of vectors \mathbf{B}_i^g and $\tilde{\mathbf{B}}^g$, respectively. To avoid overlapping points, bounds for mutations on the abscissa axis are set depending on the selected point to mutate ($\tilde{B}_{r_{bp}}^g$) and the location of its neighbors according to

- (i) $B_{i, r_{bp}}^g + \alpha < \tilde{B}_{r_{bp}}^g < B_{i, r_{bp}+1}^g - \alpha$; if $r_{bp} = 1$,
- (ii) $B_{i, r_{bp}-1}^g + \alpha < \tilde{B}_{r_{bp}}^g < B_{i, r_{bp}+1}^g - \alpha$; if $r_{bp} = 2 \dots n - 1$,
- (iii) $B_{i, r_{bp}-1}^g + \alpha < \tilde{B}_{r_{bp}}^g < B_{i, r_{bp}}^g - \alpha$; if $r_{bp} = n$,

where α is a user-defined parameter that indicates how close the points can be located. To achieve a good fit of the nonlinearity α must be small, relative to the search space on the abscissa axis defined by v_{\min} and v_{\max} . Note that the horizontal boundaries for the endpoints are delimited by their position and their left or right neighbor, respectively. Bounds for mutations on ordinate axis are fixed and equal for all points. This allows each point to move freely throughout the search space on the ordinate axis defined by w_{\min} and w_{\max} . The vertical and horizontal bounds for a selected point are illustrated in Figure 4. When mutation M.2 is required, the selected point can be changed to a new position within the grey rectangle. Details on how to determine v_{\min} , v_{\max} , w_{\min} , and w_{\max} will be given in Section 3.3.

To achieve a fine-tuning of all nonlinearity points, mutations' aggressiveness can be controlled through standard

deviation (17) as in mutation M.1. Note that $\Delta_s = w_{\max} - w_{\min}$ is constant for all mutations over ordinate axis, while for abscissa axis mutations, Δ_s can be calculated as

$$\Delta_s = \begin{cases} B_{i, r_{bp}+1}^g - \alpha - v_{\min}; & \text{if } (r_{bp} = 1) \\ B_{i, r_{bp}+1}^g - B_{i, r_{bp}-1}^g - 2\alpha & \text{if } (r_{bp} = 2 \dots n - 1) \\ v_{\max} - B_{i, r_{bp}-1}^g - \alpha & \text{if } (r_{bp} = n). \end{cases} \quad (20)$$

Mutation M.2 has great potential to explore the searching space. This genetic operation will locate points where there are slope changes. During first generations, it is useful to shape nonlinearity, while in last ones, it allows a refinement. However, when a point is located where there is a slope change, it could be kept in this location until the end of generations, especially when there are abrupt changes in the slope. Because jumps between points are not allowed with mutation M.2, when a point is kept in a place where there is a significant change of slope, one or more points would remain trapped to the left or right of it. This would lead to having redundant points in a segment that would not require so many or worse, to having a segment (curvature) that would not contain enough points. To avoid this drawback, the exploration in the search space is complemented with mutation M.3.

Mutation M.3. This genetic operation is designed to concentrate as many points as possible on the curvatures that nonlinearity can have. Therefore, it will be required that each point can be displaced on the abscissa axis by jumping one or more positions of the other points. Let us define a segment as the horizontal space between two consecutive points (so for n points there will be $n - 1$ segments); then a random integer number $r_s \in [1, n - 1]$ will indicate to which segment the selected point will move. The first half of the offspring vector is found using the following expression:

$$\tilde{B}_j^g = \begin{cases} \frac{B_{i, r_s}^g + B_{i, r_s+1}^g}{2} & \text{if } (j = r_{bp}) \\ B_{i,j}^g & \text{otherwise} \end{cases} \quad (21)$$

with $j = 1 \dots n$. Note that if $r_{bp} = r_s$ or $r_{bp} = r_s - 1$, the corresponding point will not make a jump but it will be located at the midpoint between its current position and the position of the point on the right or left, respectively. As in the mutation M.2 to prevent points from getting too close together, the α parameter is also used in this mutation; therefore, a jump is conditioned to the space available in the selected segment to accommodate a new point. Minimum space should be 2α . If this condition is not met, r_s must be regenerated to randomly search for another segment.

To provide a smooth transition between adjacent segments, gene mutation corresponding to the position on the ordinates axis is performed using a quadratic interpolation. To do that, three neighboring points are required. The second half of the offspring vector is found using the following expression:

$$\tilde{B}_j^g = \begin{cases} \left[(\tilde{B}_{j-n}^g)^2, \tilde{B}_{j-n}^g, 1 \right] * \begin{bmatrix} k_2 \\ k_1 \\ k_0 \end{bmatrix} & \text{if } (j = n + r_{bp}) \\ B_{i,j}^g & \text{otherwise} \end{cases} \quad (22)$$

with $j = n + 1 \cdots 2n$. \tilde{B}_{j-n}^g is the ν -coordinate of the selected point to mute which can be found with (21); k_0 , k_1 , and k_2 are the coefficients of the quadratic polynomial Ψ defined by three adjacent points selected once the new ν -coordinate of the point that is mutating is known. The three adjacent points can be selected directly when a point has mutated to the first or last segment,

$$\Psi = \begin{cases} f((B_{i,1}^g, B_{i,n+1}^g); (B_{i,2}^g, B_{i,n+2}^g); (B_{i,3}^g, B_{i,n+3}^g)) : & \text{if } (r_s = 1) \\ f((B_{i,n}^g, B_{i,2n}^g); (B_{i,n-1}^g, B_{i,2n-1}^g); (B_{i,n-2}^g, B_{i,2n-2}^g)) : & \text{if } (r_s = n - 1) \end{cases}, \quad (23)$$

while if the point has mutated to a nonextreme segment, the three adjacent points can be selected using the two points that

define that segment plus one on its right or left. For more effective exploration, a random number $r_{3p} \in (0, 1]$ is used for selection:

$$\Psi = \begin{cases} f((B_{i,r_s-1}^g, B_{i,n+r_s-1}^g); (B_{i,r_s}^g, B_{i,n+r_s}^g); (B_{i,r_s+1}^g, B_{i,n+r_s+1}^g)) : & \text{if } (r_{3p} \leq 0.5) \\ f((B_{i,r_s}^g, B_{i,n+r_s}^g); (B_{i,r_s+1}^g, B_{i,n+r_s+1}^g); (B_{i,r_s+2}^g, B_{i,n+r_s+2}^g)) : & \text{otherwise} \end{cases}. \quad (24)$$

Figure 5 illustrates how a jump occurs with mutation M.3. Notice that the new ordinate is calculated according to the polynomial formed by the two points of the segment plus a point to the right: that is, $r_{3p} > 0.5$. After a jump has occurred, an ascending reordering of the points with respect to the abscissa values is necessary.

Crossover C.2. This genetic operation works just like crossover C.1 and is applied only to vary the position of a point on the ordinate axis:

$$\tilde{B}_j^g = \begin{cases} \frac{B_{i,j}^g + B_{\text{best},j}^g}{2} & \text{if } (j = n + r_{bp}) \\ B_{i,j}^g & \text{otherwise} \end{cases} \quad (25)$$

with $j = n + 1 \cdots 2n$. $B_{\text{best},j}^g$ is the j th element of vector $\mathbf{B}_{\text{best}}^g$, which corresponds to the individual in the current population g with the best fitness value.

3.2.3. Pole-Zero Classification. Due to the stochastic nature of evolutionary algorithms, the binary values of (11) will change as the algorithm evolves, generating different structures of $G_w(z)$ and $G_h(z)$. The evolution of this piece of genetic information is handled by a simple mutation operator.

Mutation M.4. Unlike the previous ones, this operator generates a new vector $\tilde{\mathbf{C}}^g$ that depends entirely on the effects of mutation, meaning that for this piece of genetic information there is no information exchange between generations. This allows free testing of different structures for $G_w(z)$ and $G_h(z)$ to avoid premature convergence. When this operation is required, a random process will generate the mutation vector:

$$\tilde{C}_j^g = \begin{cases} 1 & \text{if } N_c \leq 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (26)$$

with $j = 1 \cdots nc + nr + mc + mr$. \tilde{C}_j^g is the j th element of vector $\tilde{\mathbf{C}}^g$. N_c is a random number with standard uniform distribution on the open interval $(0, 1)$. Note that the structure of $\tilde{\mathbf{C}}^g$ is built under two considerations: the LTI subsystems cannot be improper and the sum of zeros and the sum of the poles between both subsystems must be equal to the number of zeros and poles of the BLA, respectively.

3.3. WH-EA Description. WH-EA is an elitist evolutionary algorithm that evolves a population of NP individuals. Each individual contains three portions of genetic information related to the parameters of a Wiener-Hammerstein model ($\theta_i^g = [\mathbf{P}_i^g, \mathbf{B}_i^g, \mathbf{C}_i^g]$). Like any other evolutionary algorithm, WH-EA is inspired by biological evolution over generations. Starting from an initial population, new generations are created using information of the current generation g and performing crossover and/or mutation operations and selection based on the fitness of the new individuals. Algorithm 1 shows a pseudocode of main steps performed in WH-EA, whereas details of all the parameters that the algorithm uses are shown in WH-EA Parameters.

Initialize the Population. The initial population $[\mathbf{P}^0 \ \mathbf{B}^0 \ \mathbf{C}^0]$ contains NP individuals generated within a search space delimited by lower and upper bounds. For each piece of genetic information, the lower and upper bounds can be determined from the information provided by the BLA (location of poles and zeros and static gain).

Real and imaginary values of poles and zeros from the BLA without modifications are introduced as part of the first

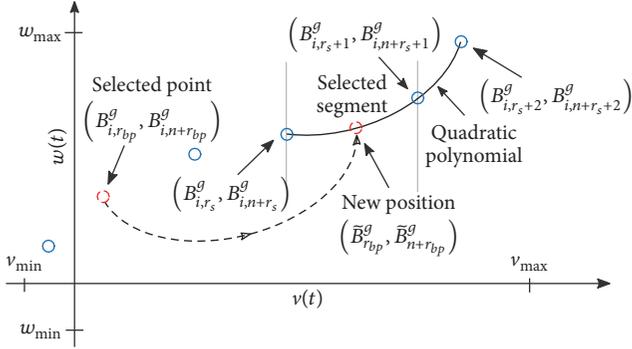


FIGURE 5: Mutation M.3 with $r_{bp} = 2$, $r_s = 4$, and $r_{3p} > 0.5$. Jump to the selected segment (dashed line). Quadratic polynomial (solid line).

- (1) Initialise the population;
- (2) Evaluate fitness of all population;
- (3) **for** $g = 1$ to MaxGen **do**
- (4) Find θ_{best}^g
- (5) Random selection of a individual (r_1);
- (6) Compute $\gamma(g)$;
- (7) **if** $r_{pznl} \leq \gamma(g)$ **then**
- (8) Compute $\tilde{\mathbf{B}}^g$ using Algorithm 2;
- (9) **else**
- (10) Compute $\tilde{\mathbf{P}}^g$ using Algorithm 3;
- (11) **end if**
- (12) **if** $r_c \leq \xi$ **then**
- (13) Compute $\tilde{\mathbf{C}}^g$ using Mutation M.4;
- (14) **end if**
- (15) Update population;
- (16) **end for**
- (17) Print θ_{best}^{MaxGen}

ALGORITHM 1: Pseudocode of WH-EA.

individual \mathbf{P}_1^0 of the initial population. For the rest, mutation M.1 (16) is used but fixing the parent vector as \mathbf{P}_1^0 , that is, the BLA.

Therefore, mutation M.1 must be executed $NP - 1$ times under the above conditions to generate mutated versions of the BLA. Initialization and mutation M.1 are performed considering a search space delimited by lower bound \mathbf{P}^{\min} and upper bound \mathbf{P}^{\max} which are set around the values of the BLA:

$$\begin{aligned} P_j^{\min} &= P_{1,j}^0 - \Upsilon_j^{\min}, \\ P_j^{\max} &= P_{1,j}^0 + \Upsilon_j^{\max}, \end{aligned} \quad (27)$$

where Υ_j^{\min} and Υ_j^{\max} represent the j th elements of the user-defined vectors $\mathbf{\Upsilon}^{\min}$ and $\mathbf{\Upsilon}^{\max}$, respectively, indicating how much the BLA's pole/zero position can change as the algorithm evolves.

On the other hand, the initial population corresponding to two-dimensional points must be generated within a search space defined horizontally by the minimum (v_{\min}) and

maximum (v_{\max}) amplitude of $v(t)$ (the output of $G_w(z)$) and vertically by the minimum (w_{\min}) and maximum (w_{\max}) amplitude of $w(t)$ (the input to $G_h(z)$). Although $v(t)$ and $w(t)$ are not known, v_{\min} and v_{\max} depend on the input $u(t)$ and $G_w(z)$. Since $G_w(z)$ is a linear system, the following expressions can be used:

$$\begin{aligned} v_{\min} &= \Omega * u_{\min}, \\ v_{\max} &= \Omega * u_{\max}, \end{aligned} \quad (28)$$

where u_{\min} and u_{\max} are the minimum and maximum values of the signal $u(t)$, respectively, and Ω is a scaling factor which depends on the pole/zero locations and the static gain of $G_w(z)$. Without loss of generality, it is possible to model a Wiener-Hammerstein system considering that both linear blocks have unit gain.

Regarding w_{\min} and w_{\max} , if the input $u(t)$ with mean u_{mean} enters $G_w(z)$ regardless of its structure, the mean of the output signal v_{mean} will be equal to u_{mean} . This is not the case when the signal $v(t)$ enters the nonlinear block since the gain K_{NL} might provide an offset to $w(t)$. However, since $G_h(z)$ is a linear block, the mean of the output y_{mean} will be equal to the mean of the incoming signal w_{mean} . With this information, the straight line of Figure 6 can be drawn, and w_{\min} and w_{\max} can be found using the following equations:

$$\begin{aligned} w_{\min} &= y_{\text{mean}} + K_{NL} (\Omega u_{\min} - u_{\text{mean}}), \\ w_{\max} &= y_{\text{mean}} + K_{NL} (\Omega u_{\max} - u_{\text{mean}}). \end{aligned} \quad (29)$$

As can be seen from (28) to (29), the search space for non-linearity depends on the input signal, output signal, static gain of the BLA, and Ω which is a user-defined parameter. Since both linear subsystems will be estimated with unit gain, neither of these will amplify their input signals, therefore $v_{\min} > u_{\min}$ and $v_{\max} < u_{\max}$. For these two conditions to be met, Ω must be less than one. In the same way, it must be observed that $w_{\min} < y_{\min}$ and $w_{\max} > y_{\max}$. If Ω is less than one, the first pair of conditions will always be met; however, there is no guarantee that the second pair of conditions will be met. Since it is possible to perform this check prior to the execution of the algorithm, if the second pair of conditions are not met, Ω must be increased, but considering that it must be less than one. It should also be taken into account that if Ω is too large, the search space will be larger than necessary, so the algorithm will cost more to estimate static nonlinearity.

To initialize this portion of genetic information, the n points are uniformly distributed between Ωu_{\min} and Ωu_{\max} and located on the straight line shown in Figure 6. These points are introduced as part of the first individual in the population \mathbf{B}_1^0 . The corresponding genetic information for the rest of individuals is generated using mutation M.2 (19), but considering that the parent vector is always \mathbf{B}_1^0 .

Finally, genetic information corresponding to pole-zero classification is initialized directly using mutation M.4 (26) NP times.

Evaluate Fitness. Performance of each individual in the population is defined by a fitness criterion which can be calculated

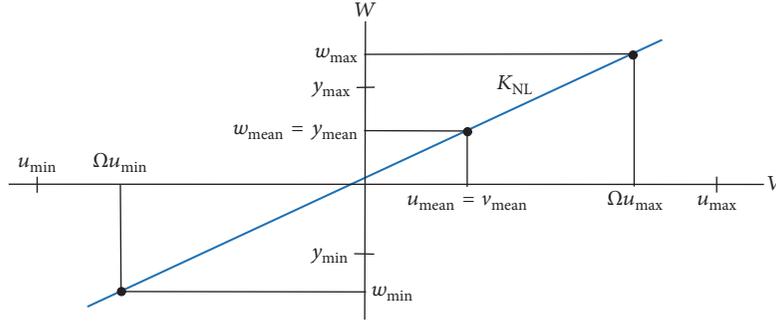


FIGURE 6: Information to define the search space for the nonlinear function.

using (14), where θ is obtained from the encoded information in $[\mathbf{P}_i^g, \mathbf{B}_i^g, \mathbf{C}_i^g]$.

The Offspring. Once population has been initialized, for each generation a random integer number $r_1 \in [1, NP]$ will be used to select the parent from which an offspring $\tilde{\mathbf{P}}^g$ will be generated. As can be seen in Algorithm 1, not all genetic operators are applied at the same time to generate offspring; this can help to expand diversity and avoid premature convergence.

One or two pieces of the offspring genetic information will be randomly selected for modification according to their respective genetic operators. A random number $r_{pznl} \in (0, 1]$ chooses between modifying the portion related to static nonlinearity using Algorithm 2 or the portion of genetic information related to pole/zero locations using Algorithm 3. The probability for this selection is handled by the control parameter $\gamma(g)$ defined as

$$\gamma(g) = \frac{\gamma_{ini}}{\sqrt{1 + g * \gamma_{rat}}}, \quad (30)$$

$$\gamma_{rat} = \frac{(\gamma_{ini}/\gamma_{end})^2 - 1}{\text{MaxGen} - 1},$$

where γ_{rat} is the rate at which the probability $\gamma(g)$ will decrease from initial probability γ_{ini} to final probability γ_{end} as generations pass; therefore, $0 < \gamma_{end} < \gamma_{ini} \leq 1$. If $\gamma_{ini} = 1$, the probability of modifying the genetic information of nonlinearity in the first generations will be high, while the probability of modifying the location of poles and zeros will be low. On the other hand, if $\gamma_{end} = 0.5$, in the final generations, the algorithm will modify with equal probability both portions of genetic information. The selection of these values is justified by the fact that pole/zero locations are known and they will only be fine-tuned within a suitable search space to amend possible errors in the BLA estimation, whereas nonlinearity is completely unknown, so the algorithm should focus more on this portion of genetic information during first generations.

Variation of genetic information corresponding to the classification of poles and zeros for both LTI subsystems is handled by a comparison between a random number $r_c \in (0, 1]$ and the probability $\xi \in (0, 1]$. The value of probability ξ is defined by the user and will be constant throughout

- (1) **if** $r_{nmc} \leq \delta_{nl}$ **then**
- (2) Compute $\eta(g)$
- (3) **if** $r_{nm} \leq \eta_{min} + \eta(g)$ **then**
- (4) Mutation M.2;
- (5) **else**
- (6) Mutation M.3;
- (7) **end if**
- (8) **else**
- (9) Crossover C.2;
- (10) **end if**

ALGORITHM 2: Modify two-dimensional points for nonlinear function.

- (1) **if** $r_{lmc} \leq \delta_{zp}$ **then**
- (2) Mutation M.1;
- (3) **else**
- (4) Crossover C.1;
- (5) **end if**

ALGORITHM 3: Modify pole/zero locations.

the evolution of the algorithm. Figure 7 shows the behaviour of the control parameters (probabilities) used to select the portions of genetic information that will be modified in each generation.

Algorithm 2 is used to modify the genetic information related to nonlinear static function. The control parameter $\delta_{nl} \in (0, 1]$ indicates the probability with which the mutation (either M.2 or M.3) or crossover C.2 will be used. Probability of selecting M.2 or M.3 is variable with respect to the generations. During first generations, mutation M.3 is not necessary, since the nonlinearity can be captured thanks to the two-dimensional points movements due to mutation M.2 and crossover C.2 operations. Since it is very likely that nonlinearity includes one or more curvatures, as the algorithm evolves mutation M.3 will be required to concentrate as many points as possible on these curvatures. The variable

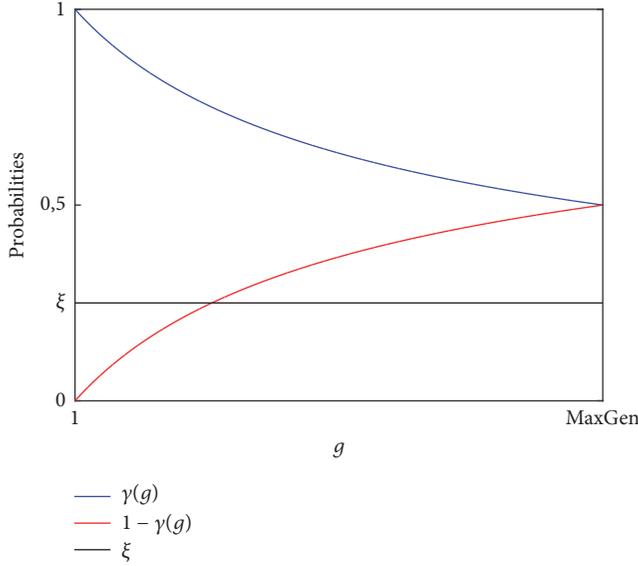


FIGURE 7: Control parameters of WH-EA for selection of the genetic information to be modified in generation g .

probability for selection between both mutations is defined by

$$\eta(g) = (1 - \eta_{\min}) - (1 - \eta_{\min}) \frac{g}{\text{MaxGen}}, \quad (31)$$

where $\eta_{\min} \in (0, 0.5]$ is a user-defined parameter indicating the minimum probability with which the mutation M.2 can be selected. Note that according to (31) and Algorithm 2, the maximum probability is 1 and occurs in the first generation. As the algorithm evolves, this probability will decrease linearly until it reaches η_{\min} in the last generation. When Algorithm 2 is required a random number $r_{nmc} \in (0, 1]$ will allow us to select either a mutation or crossover C.2. If a mutation is selected, a new random number $r_{mm} \in (0, 1]$ will allow us to select between mutation M.2 or mutation M.3.

On the other hand, Algorithm 3 is used to modify the genetic information related to pole/zero locations using mutation M.1 or crossover C.1. The control parameter $\delta_{zp} \in (0, 1]$ indicates the probability with which each genetic operation will be used. Since crossover C.1 causes offspring to inherit genetic information from the best individual, a small value of δ_{zp} may lead to premature convergence, whereas a value closer to 1 will cause the algorithm to converge very slowly. When Algorithm 3 is required, a random number $r_{lmc} \in (0, 1]$ will determine the genetic operation to be used.

Update. It is based on a competition between the generated offspring and the individuals of the population. The contestant with the best fitness will be the one who wins the competition. From a randomly selected individual, the offspring starts to compete until defeating an individual; when this happens the descendant will take his place in the population and the algorithm continues with the next generation. If the offspring comes to compete with all individuals and could not win, this will be discarded and the algorithm will pass to the next generation.

4. Application of WH-EA and Results

WH-EA was tested on a numerical example and on the benchmark for nonlinear system identification in (SYSID'09) [9], where a Wiener-Hammerstein system is selected as test object. The benchmark is not intended as a competition, but as a tool to compare the possibilities of different methods to deal with this specific nonlinear structure.

For both cases, the BLA was estimated with the MATLAB System Identification Toolbox [60] using a Box-Jenkins (BJ) structure. Besides, trends and means were only removed for the BLA identification. The following parameters of the algorithm were set in common for both estimates: $\xi = 0.25$; $\delta_{zp} = 0.75$; $\delta_{nl} = 0.75$; $\eta_{\min} = 0.35$; in addition, initial and final standard deviations for mutations were set to 20 and 1, respectively.

4.1. Numerical Example. A Wiener-Hammerstein system with the following structure was designed (where tansig is the hyperbolic tangent sigmoid transfer function):

$$\begin{aligned} G'_w(z) &= \frac{0.1190}{(z - 0.9048)}, \\ w'(t) &= 0.45 \text{tansig}(2.80v'(t)), \\ G'_h(z) &= -0.01426 \\ &\cdot \frac{(z - 1.0510)(z + 1)}{(z - 0.9746 + 0.03656j)(z - 0.9746 - 0.03656j)}. \end{aligned} \quad (32)$$

A Gaussian excitation signal of 6 dB was filtered with a cut-off frequency of 6 Hz and used as input signal. The system was simulated and 120000 input/output samples were recorded and separated in two parts: the estimation data set $t_n \in [1001, 70000]$ for identification purposes and the test data set $t_n \in [71001, 120000]$ for validation purposes (in both data sets first 1000 samples were ignored to avoid transient effects). Furthermore, additive white Gaussian noise with a Signal-to-Noise Ratio (SNR) of 45.32 dB was added to the output.

The identification of the BLA was carried out and the model obtained was expressed in factored form:

$$\begin{aligned} G_{\text{BLA}} &= -1e^{-3} \\ &\cdot \frac{(z - 1.0508)(z + 0.9631)}{(z - 0.9749 + 0.0366j)(z - 0.9749 - 0.0366j)(z - 0.9045)}. \end{aligned} \quad (33)$$

The root mean square of the error (e_{RMS}) obtained with this linear model on test data was of 0.0414.

According to the BLA structure, vector \mathbf{P}_1^0 was coded with $nc = 0$, $nr = 2$, $mc = 1$, and $mr = 1$ as follows:

$$\mathbf{P}_1^0 = [1.0508, -0.9631, 0.9749, 0.9045, 0.0366]. \quad (34)$$

The search space for nonlinear function was defined with $\Omega = 0.28$, while elements of \mathbf{Y}^{\min} and \mathbf{Y}^{\max} were set to 0.01 (except bounds for the zero $z = -0.9631$ that were set to 0.1, since it influences slightly the dynamics and should have freedom of movement during tuning).

TABLE 1: Performance of the numerical example estimation using different numbers of points n to represent static nonlinearity.

n	NRMSE (%)	eRMS
8	99.179	$1.183e^{-3}$
10	99.396	$1.109e^{-3}$
12	99.564	$1.044e^{-3}$

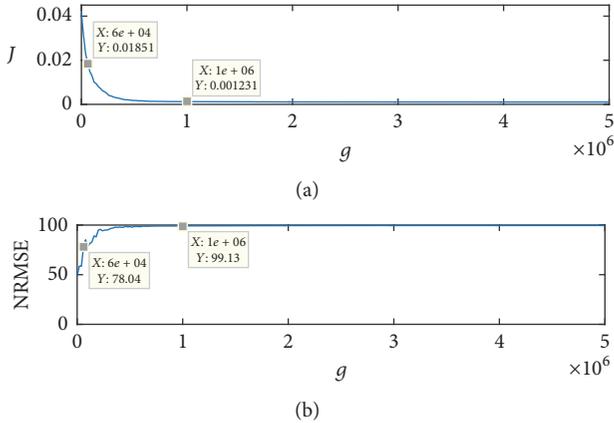


FIGURE 8: Convergence graph (a) and NRMSE of the captured nonlinearity (b) for Wiener-Hammerstein estimation with $n = 12$.

WH-EA was executed 3 times with $\text{MaxGen} = 5 \cdot 10^6$ and different number of points was chosen for the nonlinearity. For all trials, the algorithm was initialized with 60000 individuals and the minimum distance between two points was set to $\alpha = (v_{\max} - v_{\min})/6n$. For each estimated Wiener-Hammerstein model, the eRMS on the test data was computed; in addition, the normalized root mean square error (NRMSE) criterion was used to quantify the goodness of fit between real and captured nonlinearity. The results are reported in Table 1; for all cases piecewise linear interpolation was used to connect the n points.

The poles and zeros of the BLA were correctly classified in the three tests carried out. As can be seen in Table 1, the eRMS of the Wiener-Hammerstein models decreased as the quality of the captured nonlinearity increases. A reasonable model was scored with 12 points considering that the RMS noise was $9.98e^{-4}$. A convergence graph for this model is shown in Figure 8, during WH-EA execution, at 6e4th generation the poles and zeros of the BLA were correctly classified, and from there the best individual of each generation conserved the genetic information for this classification. Since the noise RMS is known, at $g = 1e6$ the performance of the model was good enough, so the algorithm could have been stopped. Anyway, 5e6 generations have been allowed in order to demonstrate the great precision that the algorithm can achieve.

In Figure 9, pole/zero locations of the BLA, the obtained Wiener-Hammerstein model, and real system are compared. Notice how WH-EA has moved initial locations trying to get to the true values improving modelling error.

On the other hand, a graphical comparison between real and captured nonlinearity is shown in Figure 10. Linear

subsystems of the estimated Wiener-Hammerstein model are represented by (35), while the ordered pairs for the nonlinear static function are shown in Table 2.

$$G_w(z) = \frac{0.0259}{z - 0.9048},$$

$$G_h(z) = -0.01960 \quad (35)$$

$$\frac{(z - 1.0512)(z + 0.9713)}{(z - 0.9746 + 0.0365j)(z - 0.9746 - 0.0365j)}$$

4.2. *Nonlinear System Identification Benchmark.* The system to be modelled is an electronic nonlinear circuit with a Wiener-Hammerstein structure (see Figure 11). This system was built by Vandersteen [61] and presented as a benchmark problem for system identification by Schoukens et al. [9].

The first linear dynamic system $G_1(s)$ is designed as a third-order Chebyshev filter (pass-band ripple of 0.5 dB and cut-off frequency of 4.4 kHz). The second linear dynamic system $G_2(s)$ is a third-order inverse Chebyshev filter (stop-band attenuation of 40 dB starting at 5 kHz). This system has a transmission zero in the frequency band of interest. This can complicate the identification significantly, because the inversion of such a characteristic is difficult. The system was excited with a filtered Gaussian signal (cut-off frequency 10 kHz). Data used for estimation corresponds to interval $t_n \in [1, 100000]$, whereas test data corresponds to the remaining part $t_n \in [101001, 188000]$. In order to analyse the performance of estimation methods, the mean value of the simulation error (μ), the standard deviation of the error (std), and the root mean square value of the error (eRMS) must be calculated on test and estimation data [9].

Since first 5000 data samples just contain quantization noise, a set of 95000 input/output data $t_n = 5001, \dots, 100000$ was used to estimate the BLA. Multiple simulations were performed considering different combinations of poles and zeros for the input/output model and for the noise model. For each BJ model, the eRMS on test data set $t_n = 101001, \dots, 188000$ was computed. The BLA was obtained with 6 poles, 5 zeros, and one sample delay for the input/output model and 3 zeros and 3 poles for the noise model. The BLA is fully described with $K_{\text{NL}} = 0.7840$ and the pole-zero pattern shown in Figure 12. The eRMS of this linear model was of 56.159 mV on test data and 43.143 mV after removing trends and means. According to the BLA structure, vector \mathbf{P}_1^0 was coded with $nc = 1, nr = 4, mc = 2, \text{ and } mr = 2$ as follows:

$$\begin{aligned} \mathbf{P}_1^0 = [& 0.7605, -0.2733, 0, -3.4122, \\ & -30.2553, 0.6501, 0.7314, \dots, \\ & 0.8912, 0.8289, 0.7004, 0.4358, 0.1692]. \end{aligned} \quad (36)$$

During BLA estimation stage, different noise models were tested and it was observed that all poles, real zeros within the unitary circle, and complex zeros where they are located correspond to the dominant dynamics of the system, while real zeros outside the unitary circle were more likely to vary

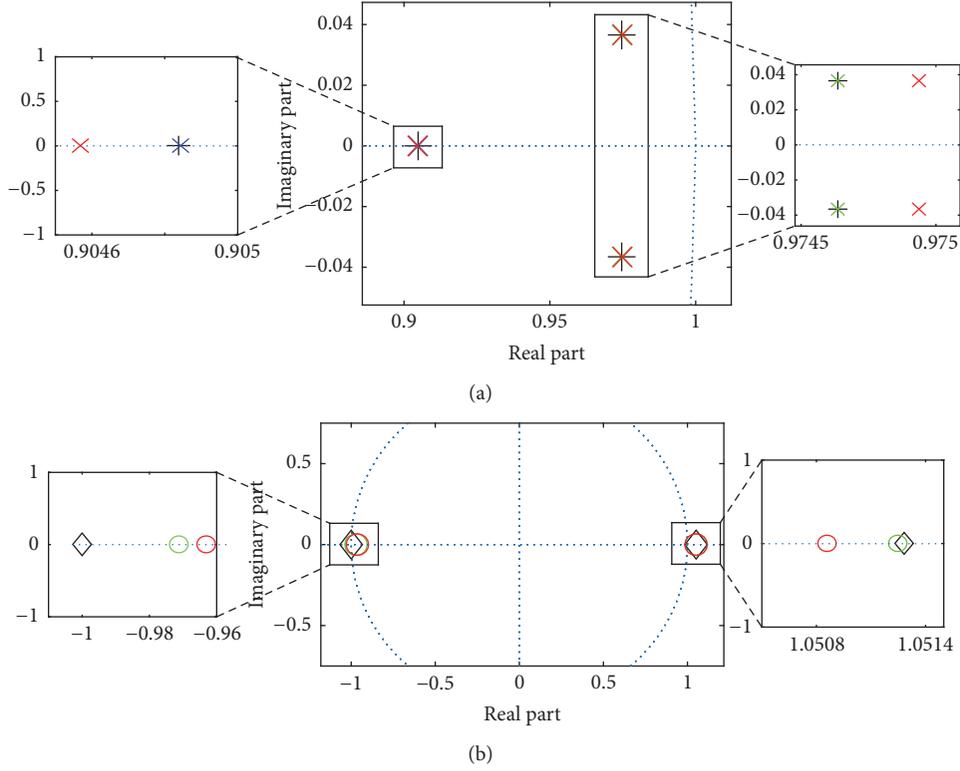


FIGURE 9: (a) Poles of the real system (black +), the BLA (red ×), and the estimated G_w (blue ×) and G_h (green ×) models. (b) Zeros of the real system (black ◇), the BLA (red o), and the estimated G_h model (green o).

TABLE 2: Coordinates of the estimated nonlinearity with $n = 12$.

i	1	2	3	4	5	6	7	8	9	10	11	12
v_i	-0.9694	-0.5794	-0.3716	-0.2479	-0.1251	0.1038	0.1996	0.2893	0.3363	0.4739	0.7648	1.0808
w_i	-0.3303	-0.3250	-0.2894	-0.2354	-0.1415	0.1178	0.2031	0.2558	0.2754	0.3126	0.3309	0.3317

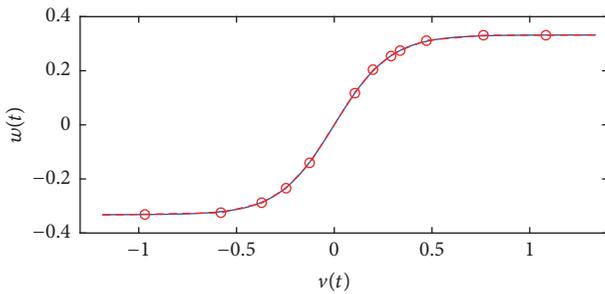


FIGURE 10: Comparison between true (solid-blue) and estimated nonlinearity defined as a piecewise linear function (dashed-red) using pairs $[v_i, w_i]$ (red circles).

their location. This information was used to define the search space for refining the location of poles and zeros:

$$\begin{aligned} \mathbf{Y}^{\min} &= [0.025, 0.025, 0.025, 1, 10, 0.025, 0.025, \dots, \\ &\quad 0.025, 0.025, 0.025, 0.025, 0.025], \\ \mathbf{Y}^{\max} &= [0.025, 0.025, 0.025, 1, 10, 0.025, 0.025, \dots, \\ &\quad 0.025, 0.025, 0.025, 0.025, 0.025]. \end{aligned} \quad (37)$$

Bounds (37) limit search space in the system dominant dynamics within ± 0.025 , while for $z = -30.255$ and $z = -3.412$, limits are between ± 10 and ± 1 , respectively.

Static nonlinearity is represented by piecewise linear functions with $n = 8$ points. Its search space was defined with $\Omega = 0.51$ and the minimum distance between two points was calculated with $\alpha = (v_{\max} - v_{\min})/10n$. The algorithm was initialized with 5000 individuals and $3e7$ generations were executed. The performance of the estimated Wiener-Hammerstein model is shown in Table 3. In Figure 13, it is depicted how the algorithm has distributed pole/zero locations for both linear subsystems. Notice how some of them were displaced to improve the modelling error. The

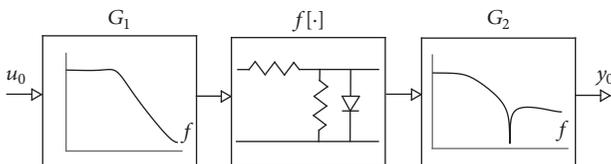


FIGURE 11: Wiener-Hammerstein benchmark.

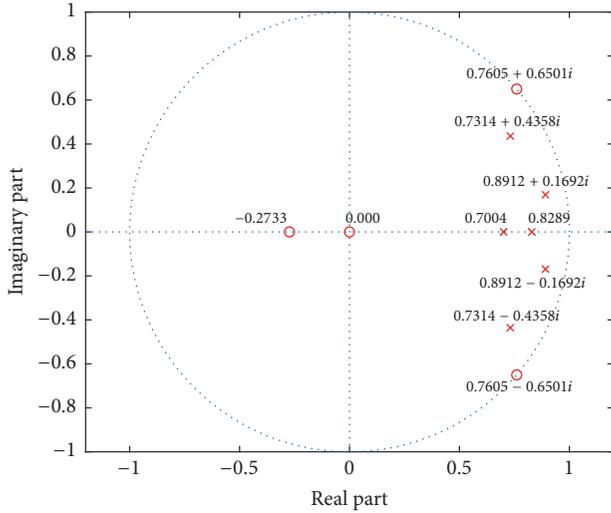


FIGURE 12: Poles (×) and zeros (o) of the BLA for the benchmark data (SYSID'09). Two real zeros fall outside the plot in -30.2553 and -3.4122 .

outputs of the estimated Wiener-Hammerstein model, the linear model error, and the nonlinear model error on test data are shown in Figure 14, while the DFT spectra of this signals are shown in Figure 15. Captured nonlinearity is plotted in Figure 16.

Final estimated linear blocks $G_w(z)$ and $G_h(z)$ are shown in (38) and (39) respectively, while coordinates for nonlinear function are shown in Table 4. The estimated Wiener-Hammerstein model contains 26 parameters of which 14 are used to represent the static nonlinearity (without end points since they can be located anywhere on their respective end segments; nevertheless, these segments slopes are taken into account). Figure 16 shows how mutation M.2 and mutation M.3 located the two-dimensional points to capture the nonlinearity. As expected, due to the effect of mutation M.3, most of them were concentrated on the curvature.

$$G_w(z) = 6.5e^{-4} \frac{(z + 0.0138)(z + 2.9034)(z + 26.76)}{(z - 0.7243)(z - 0.7324 + 0.4361j)(z - 0.7324 - 0.4361j)}, \quad (38)$$

$$G_h(z) = 0.0120 \frac{(z + 0.2635)(z - 0.7575 + 0.6513j)(z - 0.7575 - 0.6513j)}{(z - 0.8191)(z - 0.8899 + 0.1688j)(z - 0.8899 - 0.1688j)}. \quad (39)$$

4.3. Discussion. In contrast to other methods which generate good initial estimates by splitting the poles and zeros of the BLA, WH-EA allows us to identify Wiener-Hammerstein models avoiding high user interaction which is an advantage compared to methods using QBLA, where at least two intermediate procedures are required before fine-tuning all parameters of the Wiener-Hammerstein model.

The $eRMS$ of 0.306 mV achieved with WH-EA on test data is quite acceptable considering that the RMS of the quantization noise is 0.189 mV. With respect to the initial model (BLA), the error was reduced by a factor of 183.52 thanks

to the captured nonlinearity and the updated pole/zero locations. Table 5 shows other proposals that have been tested on the benchmark. It can be appreciated that the $eRMS$ of this paper is slightly higher than others; however, not all estimated models have the same complexity. Some of them use complex models with a greater number of parameters processing raw data before identification, while in this work, WH-EA is fed raw input/output data without preprocessing operations.

Comparing WH-EA with the proposals of Westwick and Schoukens [42] and Vanbeylen [43] whose models have the same complexity as the model estimated in this paper, the results are quite similar; however, to obtain a good final model with these two proposals, it is required that the BLA be estimated with high precision. In Vanbeylen [43] at the BLA division phase, a false position of a pole or zero could cause the values of the fractional powers to be close to 1/2 which would cause the user to make a bad decision and the BLA is badly divided. This problem is much more critical in Westwick and Schoukens [42] since the method is based on a graphical comparison between the poles and zeros of the BLA and the poles and zeros of the QBLA. With WH-EA, this problem does not occur, since the evolutionary algorithm contemplates possible errors that can be made in the estimation of the BLA. During the algorithm evolution, the binary code used for the classification of the poles and zeros of the BLA can be changed without user interaction as the false positions of the poles and zeros are corrected. This is an important advantage of WH-EA, since it is very likely that the BLA estimate is subject to errors due to noise and nonlinearity effects; this has been experimentally demonstrated; for this reason, many proposals carry out a final readjustment of the parameters of the Wiener-Hammerstein model.

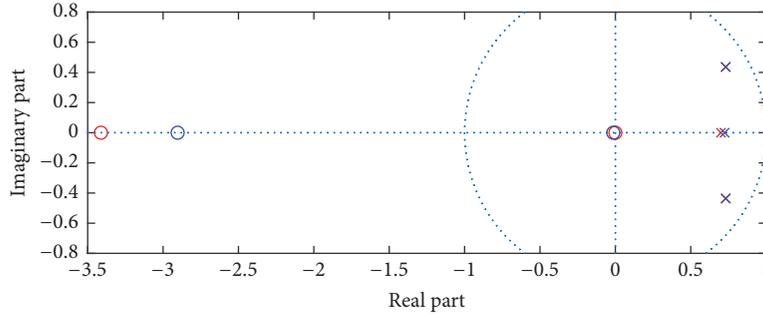
About the computational complexity, an iteration in WH-EA involves the random selection of an individual from the population, the offspring generation according to the genetic operations indicated in Section 3, the calculation of the objective function of the offspring, and the population update. This algorithm was implemented on MATLAB® software and was run on a personal computer with core i7 processor of 2,6 GHz and 16 Gb of RAM. The objective function was easily implemented using the filter command to simulate both linear subsystems, while the nonlinearity was interpolated using the interp1 command. For the benchmark system, the average time consumed per 50 iterations was 1.16 s.

5. Conclusions

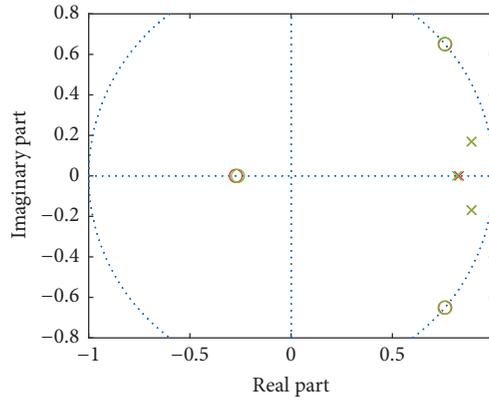
In this paper, a new method (WH-EA) to identify Wiener-Hammerstein systems in a single step is proposed. The proposal estimates all parameters of the Wiener-Hammerstein model based on a customized evolutionary algorithm (WH-EA). Unlike conventional procedures, WH-EA is able to look for the best BLA split capturing at the same time the process static nonlinearity with high precision, solving a single optimization problem. The algorithm is fed with the estimated BLA and its pole/zero locations are subtly modified within an adequate search space to allow its fine-tuning, while piecewise linear function is used for the nonlinear block. The performance of this approach has been evaluated through

TABLE 3: Performance indicators of the estimated Wiener-Hammerstein model. All values are shown in mV.

	BLA		Wiener-Hammerstein	
	Estimation	Test	Estimation	Test
μ	-35.825	-35.951	$5.6e^{-4}$	$1.1e^{-5}$
std	42.108	43.143	0.322	0.306
eRMS	55.286	56.159	0.322	0.306



(a)



(b)

FIGURE 13: (a) Poles (\times) and zeros (o) of the linear subsystem G_w (blue) and BLA (red). Zero in $z = -30.2553$ belonging to BLA and its adjusted final value in $z = -26.7609$ fall outside the plot. (b) Poles (\times) and zeros (o) of the linear subsystem G_h (green) and BLA (red).

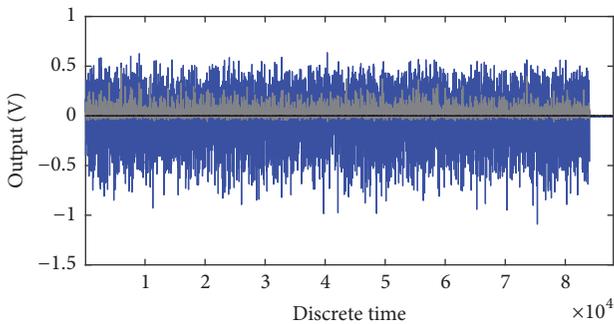


FIGURE 14: Model output (blue), simulation error of the BLA (grey), and simulation error of the estimated Wiener-Hammerstein model (black).

a numerical example with a complex static nonlinearity and through the well-known benchmark data (SYSID'09). The

results show that it is possible, using WH-EA, to identify a Wiener-Hammerstein system with a good precision in a parametric framework avoiding high user interaction and drawbacks involved in using the QBLA. Further research will be related to WH-EA extension for nonlinear multivariable systems by using a multiobjective optimization approach.

WH-EA Parameters

- nc : Number of pairs of complex conjugate zeros of the BLA
- nr : Number of real zeros of the BLA
- mc : Number of pairs of complex conjugate poles of the BLA
- mr : Number of real poles of the BLA
- n : Number of points to represent nonlinearity
- K_{NL} : Static gain of the BLA

TABLE 4: Nonlinearity coordinates ($n = 8, 14$ parameters) estimated by WH-EA from benchmark data.

i	1	2	3	4	5	6	7	8
v_i	-0.2168	0.1596	0.3819	0.4943	0.6047	0.7596	1.0811	1.3953
w_i	-0.1979	0.1440	0.3443	0.4276	0.4822	0.5248	0.5698	0.5901

TABLE 5: Performance measurements on benchmark data (SYSID'09). All the values are shown in mV. θ indicates the number of parameters used for the model.

Method/technique	eRMS (mV)	θ
Nonparametric BLA, QBLA [41]	0.278	44
Classification of poles and zeros using QBLA [42]	0.286	26
Fractional model parameterization [43]	0.295	26
Advanced method [44, 45]	0.30	64
WH-EA (this paper)	0.306	26
Brute force method [45]	0.31	30
Scanning technique [46]	0.370	-
Polynomial nonlinear state space [47]	0.42	797
Generalized Hammerstein-Wiener [48]	0.481	47
Incremental nonlinear optimization [49]	0.679	25
LS-SVMs [50]	4.070	-
Biosocial culture [51]	8.546	34

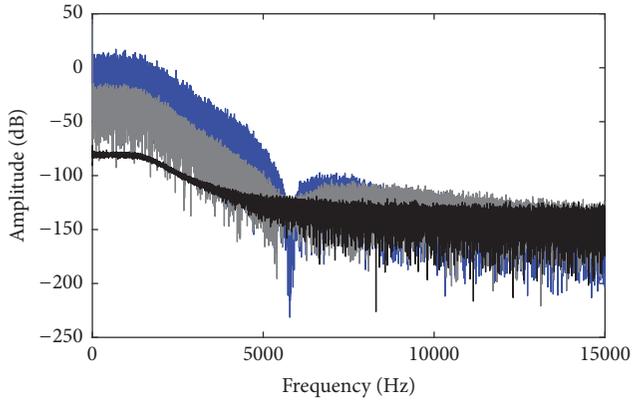
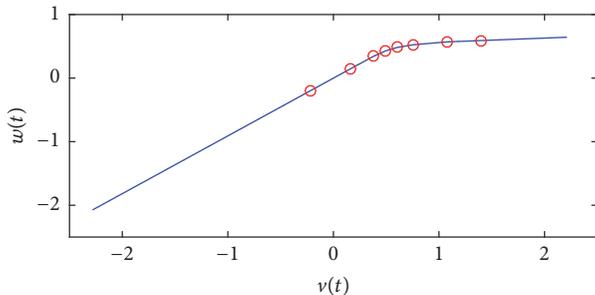


FIGURE 15: DFT spectra of the modelled output signal (blue), linear model error (grey), and nonlinear model error (black).

FIGURE 16: Captured nonlinearity as a piecewise linear function with $n = 8$ by WH-EA from the benchmark data. Notice that the nonlinear block characterization only needs 14 parameters since the first and last straight segments can be defined just with their angles.

$\sigma_{ini}^2, \sigma_{end}^2$: Initial and final standard deviations for control of the aggressiveness of mutations M.1 and M.2 ($\sigma_{ini}^2 > \sigma_{end}^2$)

σ_{ratio}^2 : Rate with which the standard deviation $\sigma(g)^2$ decreases from σ_{ini}^2 to σ_{end}^2

Δ_s : Space over which a gene can move

α : Minimum distance between two points on the abscissa axis for mutations M.2 and M.3

Ω : Scale factor to define the search space for static nonlinearity

MaxGen: Generations number

NP: Population size

δ_{zp} : Control parameter for selection between mutation M.1 or crossover C.1

δ_{ni} : Control parameter for selection between mutation (either M.2 or M.3) or crossover C.2

$\eta(g)$: Variable probability for selection between mutation M.2 or mutation M.3

η_{min} : Minimum probability for selection of mutation M.2. The maximum probability for this selection is 1

$\gamma(g)$: Variable probability to choose between modifying static nonlinearity or location of poles and zeros

γ_{rat} : Rate at which $\gamma(g)$ will decrease from initial probability $\gamma_{ini} = 1$ to final probability $\gamma_{end} = 0.5$

ξ : Probability to modify the genetic information related to the classification of poles and zeros.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the Spanish Ministry of Economy and Competitiveness (Project DPI2015-71443-R) and Salesian Polytechnic University of Ecuador through a Ph.D. scholarship granted to the first author.

References

- [1] J. Schoukens and R. Pintelon, *Identification of Linear Systems: A Practical Guideline to Accurate Modeling*, Elsevier, 2014.
- [2] L. A. Mora and J. E. Amaya, "A new identification method based on open loop step response of overdamped system," *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 14, no. 1, pp. 31–43, 2017.
- [3] T. Liu, Q.-G. Wang, and H.-P. Huang, "A tutorial review on process identification from step or relay feedback test," *Journal of Process Control*, vol. 23, no. 10, pp. 1597–1623, 2013.
- [4] P. van Overschee and B. de Moor, *Subspace Identification for Linear Systems: Theory Implementation Applications*, Springer Science & Business Media, Germany, 2012.
- [5] L. Ljung, *System Identification: Theory for the User*, *prentice Hall Information and System Sciences Series*, Prentice-Hall, NJ, USA, 1999.
- [6] R. S. Esfandiari and B. Lu, *Modeling and Analysis of Dynamic Systems*, CRC Press, 2014.
- [7] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*, John Wiley & Sons, 5th edition, 2012.
- [8] J. Bonilla, L. Roca, A. De La Calle, and S. Dormido, "Dynamic model of a molten salt-gas heat recovery system for a hybrid renewable solar thermal power plant," *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 14, no. 1, pp. 70–81, 2017.
- [9] J. Schoukens, J. Suykens, and L. Ljung, "Wiener-hammerstein benchmark," in *In 15th IFAC Symposium on System Identification, Saint-Malo, France, July, 2009*.
- [10] T. Wigren and J. Schoukens, "Three free data sets for development and benchmarking in nonlinear system identification," in *Proceedings of the 12th European Control Conference, ECC*, pp. 2933–2938, July 2013.
- [11] M. Schoukens, P. Mattson, T. Wigren, and J. Noël, Cascaded tanks benchmark combining soft and hard nonlinearities, workshop on nonlinear system identification benchmarks, Brussels, Belgium, 2016.
- [12] J. P. Noël and M. Schoukens, Hysteretic benchmark with a dynamic nonlinearity, workshop on nonlinear system identification benchmarks, pp. 7–14, Brussels, Belgium, 2016.
- [13] B. De Moor, P. De Gerssem, B. De Schutter, and W. Favoreel, "Daisy: a database for identification of systems," *Journal A 3*, 1997.
- [14] G. P. Liu, *Nonlinear Identification And Control: A Neural Network Approach*, Springer Science Business Media, 2012.
- [15] F. Giri and E. W. Bai, *Block-Oriented Nonlinear System Identification*, Springer, Berlin, Germany, 2010.
- [16] A. Janczak, *Identification of Nonlinear Systems Using Neural Networks and Polynomial Models: A Block-Oriented Approach*, vol. 310, Springer Science & Business Media, 2004.
- [17] F. J. Doyle, R. K. Pearson, and B. A. Ogunnaike, *Identification and Control Using Volterra Models*, Springer, New York, NY, USA, 2012.
- [18] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*, John Wiley & Sons, Chichester, UK, 2013.
- [19] J. J. E. Oviedo, J. P. Vandewalle, and V. Wertz, *Fuzzy logic, Identification and Predictive Control*, Springer Science & Business Media, 2006.
- [20] J. A. Suykens and J. P. Vandewalle, *Nonlinear Modeling: Advanced Black-Box Techniques*, Springer Science Business Media, New York, NY, USA, 2012.
- [21] S. A. Billings and S. Y. Fakhouri, "Identification of systems containing linear dynamic and static nonlinear elements," *Automatica*, vol. 18, no. 1, pp. 15–26, 1982.
- [22] P. Lopes dos Santos, J. A. Ramos, and J. L. Martins de Carvalho, "Identification of a benchmark wiener-hammerstein: a bilinear and hammerstein-bilinear model approach," *Control Engineering Practice*, vol. 20, no. 11, pp. 1156–1164, 2012.
- [23] A. Kalafatis, N. Arifin, L. Wang, and W. R. Cluett, "A new approach to the identification of pH processes based on the wiener model," *Chemical Engineering Science*, vol. 50, no. 23, pp. 3693–3701, 1995.
- [24] F. Jurado, "A method for the identification of solid oxide fuel cells using a hammerstein model," *Journal of Power Sources*, vol. 154, no. 1, pp. 145–152, 2006.
- [25] S. Boubaker, "Identification of nonlinear hammerstein system using mixed integer-real coded particle swarm optimization: application to the electric daily peak-load forecasting," *Nonlinear Dynamics*, vol. 90, no. 2, pp. 797–814, 2017.
- [26] M. S. Gaya, M. U. Zango, L. A. Yusuf et al., "Estimation of turbidity in water treatment plant using hammerstein-wiener and neural network technique," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 5, no. 3, pp. 666–672, 2017.
- [27] A. Buscarino, C. Corradino, L. Fortuna, M. L. Apicella, G. Mazzino, and M. G. Xibilia, "Temperature model identification on FTU liquid lithium limiter," in *Proceedings of the 42nd Conference of the Industrial Electronics Society, IECON 2016*, pp. 6380–6384, October 2016.
- [28] M. R. S. Mohammadi, R. Matos, and C. Rebelo, "Bobtail bolt preload loss in wind turbine tower prototype: Hammerstein-wiener identification model," *ce/papers*, vol. 10, pp. 175–184, 2017.
- [29] E.-W. Bai, Z. Cai, S. Dudley-Javoroski, and R. K. Shields, "Application of wiener-hammerstein system identification in electrically stimulated paralyzed skeletal muscle modeling," in *Proceedings of the 47th IEEE Conference on Decision and Control, CDC 2008*, pp. 3305–3310, Mexico, December 2008.
- [30] E.-W. Bai, Z. Cai, S. Dudley-Javoroski, and R. K. Shields, "Identification of a modified wiener-hammerstein system and its application in electrically stimulated paralyzed skeletal muscle modeling," *Automatica*, vol. 45, no. 3, pp. 736–743, 2009.
- [31] O. P. Dewhirst, D. M. Simpson, N. Angarita, R. Allen, and P. L. Newland, "Wiener-Hammerstein parameter estimation using differential evolution," in *International Conference on Bio-inspired Systems and Signal Processing*, pp. 271–276, January, 2010.

- [32] J. A. Oliver, R. Prieto, J. A. Cobos, O. García, and P. Alou, "Hybrid wiener-hammerstein structure for grey-box modeling of dc-dc converters," in *Proceedings of the 24th Annual IEEE Applied Power Electronics Conference and Exposition, APEC*, pp. 280–285, USA, February 2009.
- [33] A. Haryanto and K.-S. Hong, "Maximum likelihood identification of wiener-hammerstein models," *Mechanical Systems and Signal Processing*, vol. 41, no. 1-2, pp. 54–70, 2013.
- [34] I. Benyó, J. Kovács, J. Mononen, and U. Kortela, "Modelling of steam temperature dynamics of a superheater," *International Journal of Simulation: Systems, Science and Technology*, vol. 6, no. 6, pp. 3–9, 2005.
- [35] M. J. Korenberg and I. W. Hunter, "The identification of nonlinear biological systems: Lnl cascade models," *Biological Cybernetics*, vol. 55, no. 2-3, pp. 125–134, 1986.
- [36] J. C. Gómez, A. Jutan, and E. Baeyens, "Wiener model identification and predictive control of a pH neutralisation process," *IEE Proceedings Control Theory and Applications*, vol. 151, no. 3, pp. 329–338, 2004.
- [37] S. Li and Y. Li, "Model predictive control of an intensified continuous reactor using a neural network Wiener model," *Neurocomputing*, vol. 185, pp. 93–104, 2016.
- [38] M. Ławryńczuk, "Nonlinear predictive control for Hammerstein–Wiener systems," *ISA transactions*, vol. 55, pp. 49–62, 2015.
- [39] Q. Zhang, Q. Wang, and G. Li, "Nonlinear modeling and predictive functional control of hammerstein system with application to the turntable servo system," *Mechanical Systems and Signal Processing*, vol. 72-73, pp. 383–394, 2016.
- [40] M. Ławryńczuk, "Nonlinear predictive control of dynamic systems represented by wiener–hammerstein models," *Nonlinear Dynamics*, vol. 86, no. 2, pp. 1193–1214, 2016.
- [41] M. Schoukens, R. Pintelon, and Y. Rolain, "Identification of wiener-hammerstein systems by a nonparametric separation of the best linear approximation," *Automatica*, vol. 50, no. 2, pp. 628–634, 2014.
- [42] D. T. Westwick and J. Schoukens, "Classification of the poles and zeros of the best linear approximations of wiener-hammerstein systems," in *Proceedings of the Universite Libre de Bruxelles*, pp. 470–475, Belgium, July 2012.
- [43] L. Vanbeylen, "A fractional approach to identify wiener-hammerstein systems," *Automatica*, vol. 50, no. 3, pp. 903–909, 2014.
- [44] L. Lauwers, *Some practical applications of the best linear approximation in nonlinear block-oriented modelling*, Vrije Universiteit Brussel, 2011.
- [45] J. Sjöberg, L. Lauwers, and J. Schoukens, "Identification of Wiener-Hammerstein models: two algorithms based on the best split of a linear model applied to the SYSID'09 benchmark problem," *Control Engineering Practice*, vol. 20, no. 11, pp. 1119–1125, 2012.
- [46] D. T. Westwick and J. Schoukens, "Initial estimates of the linear subsystems of wiener-hammerstein models," *Automatica*, vol. 48, no. 11, pp. 2931–2936, 2012.
- [47] J. Paduart, L. Lauwers, R. Pintelon, and J. Schoukens, "Identification of a Wiener-Hammerstein system using the polynomial nonlinear state space approach," *Control Engineering Practice*, vol. 20, no. 11, pp. 1133–1139, 2012.
- [48] A. Wills and B. Ninness, "Generalised hammerstein-wiener system estimation and a benchmark application," *Control Engineering Practice*, vol. 20, no. 11, pp. 1097–1108, 2012.
- [49] A. H. Tan, H. K. Wong, and K. Godfrey, "Identification of a Wiener-Hammerstein system using an incremental nonlinear optimisation technique," *Control Engineering Practice*, vol. 20, no. 11, pp. 1140–1148, 2012.
- [50] T. Falck, K. Pelckmans, J. A. K. Suykens, and B. De Moor, "Identification of wiener-hammerstein systems using LS-SVMs," *IFAC Proceedings Volumes*, vol. 420, no. 10, pp. 820–825, 2009.
- [51] A. Naitali and F. Giri, "Wiener-hammerstein system identification-an evolutionary approach," *International Journal of Systems Science*, vol. 47, no. 1, pp. 45–61, 2016.
- [52] M. Schoukens and K. Tiels, "Identification of block-oriented nonlinear systems starting from linear approximations: a survey," *Automatica*, vol. 85, pp. 272–292, 2016.
- [53] L. Ljung, "Estimating Linear Time-Invariant models of nonlinear time-varying systems," *European Journal of Control*, vol. 7, no. 2-3, pp. 203–219, 2001.
- [54] M. Enqvist and L. Ljung, "Linear approximations of nonlinear FIR systems for separable input processes," *Automatica*, vol. 41, no. 3, pp. 459–473, 2005.
- [55] J. Schoukens, R. Pintelon, T. Dobrowiecki, and Y. Rolain, "Identification of linear systems with nonlinear distortions," *Automatica*, vol. 41, no. 3, pp. 491–504, 2005.
- [56] J. Schoukens, R. Pintelon, and Y. Rolain, *Mastering system identification in 100 exercises*, John Wiley & Sons, 2012.
- [57] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach, Second Edition*, John Wiley & Sons, 2012.
- [58] J. Schoukens, J. Lataire, R. Pintelon, G. Vandersteen, and T. Dobrowiecki, "Robustness issues of the best linear approximation of a nonlinear system," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 5, pp. 1737–1745, 2009.
- [59] J. Bussgang, *Crosscorrelation Functions of Amplitude-Distorted Gaussian Signals. Research Lab. Electron*, 1952.
- [60] L. Ljung, *System Identification Toolbox for Use with MATLAB*, 2007.
- [61] G. Vandersteen, *Identification of Linear and Nonlinear Systems in An Errors-In-Variables Least Squares and Total Least Squares Framework*, Vrije Universiteit Brussel, Belgium, 1997.

Research Article

A Novel Fuzzy Algorithm to Introduce New Variables in the Drug Supply Decision-Making Process in Medicine

Jose M. Gonzalez-Cava ¹, José Antonio Rebozo,² José Luis Casteleiro-Roca ^{1,3},
José Luis Calvo-Rolle ³ and Juan Albino Méndez Pérez ¹

¹Department of Computer Science and System Engineering, Universidad de La Laguna (ULL),
San Cristóbal de La Laguna, 38200 Tenerife, Spain

²Hospital Universitario de Canarias, San Cristóbal de La Laguna, Tenerife, Spain

³Department of Industrial Engineering, Universidade da Coruña, Coruña, Spain

Correspondence should be addressed to Jose M. Gonzalez-Cava; jgonzalc@ull.edu.es

Received 30 November 2017; Revised 16 January 2018; Accepted 23 January 2018; Published 18 February 2018

Academic Editor: José Manuel Andújar

Copyright © 2018 Jose M. Gonzalez-Cava et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One of the main challenges in medicine is to guarantee an appropriate drug supply according to the real needs of patients. Closed-loop strategies have been widely used to develop automatic solutions based on feedback variables. However, when the variable of interest cannot be directly measured or there is a lack of knowledge behind the process, it turns into a difficult issue to solve. In this research, a novel algorithm to approach this problem is presented. The main objective of this study is to provide a new general algorithm capable of determining the influence of a certain clinical variable in the decision making process for drug supply and then defining an automatic system able to guide the process considering this information. Thus, this new technique will provide a way to validate a given physiological signal as a feedback variable for drug titration. In addition, the result of the algorithm in terms of fuzzy rules and membership functions will define a fuzzy-based decision system for the drug delivery process. The method proposed is based on a Fuzzy Inference System whose structure is obtained through a decision tree algorithm. A four-step methodology is then developed: data collection, preprocessing, Fuzzy Inference System generation, and the validation of results. To test this methodology, the analgesia control scenario was analysed. Specifically, the viability of the Analgesia Nociception Index (ANI) as a guiding variable for the analgesic process during surgical interventions was studied. Real data was obtained from fifteen patients undergoing cholecystectomy surgery.

1. Introduction

Artificial Intelligence (AI) plays an important role in science and engineering. This methodology is able to make decisions after a training process based on learning from a dataset obtained through expertise. One of the possible definitions of Artificial Intelligence refers to cognitive process and, specifically, to reasoning. Consequently, there is a natural relationship between Artificial Intelligence and decision-making [1]. Great progress has been made in different fields as industrial engineering [2, 3], tourist sector [4, 5], or energy field [6].

Specifically, in medicine, AI techniques have been applied with different aims. It includes the capability of learning automatically from data to control the health management

systems, including an active guidance of clinicians in their treatment decisions. For clinical decision support, the key idea of the training process is extracting the expert knowledge from the information concerning medical records and the unstructured data including natural language [7]. E-health systems have become popular as they automatically evaluate the situation of patients without involvement from a physician [8]. Decision-making process in hospital management for prioritization of risks and assessment of failures has been also approached [9, 10]. Moreover, AI has been used to automatic diagnosis and classification of illness [11, 12] and also for medical sensors fault detection [13]. Specifically, in medicine, the classifiers proposed to support a decision-making process must be suitable for being understood and

evaluated from a clinician point of view [14]. Fuzzy rule-based systems have been widely used in medicine as they consist of simple linguistic rules that relate concepts in a natural manner [15, 16].

One of the main challenges in medicine is related to personalising the drug dose according to the real needs of patients. In most cases, the information obtained from the variable of interest leads to an increment or decrement of the drug infusion according to the medical criteria. AI has been also applied to automate the administration of drugs in medicine [17–20]. Important results have been reached in vasopressor administration [21, 22] or control of anaesthesia [23–26]. The key idea of these systems is a closed-loop scheme in which a controller decides the drug dose comparing the information of the measured variable to the proposed target. To design an appropriate control structure, it is necessary to deal with a well-known process. As a result, it is difficult to automate those processes in which the control variable cannot be easily measured or the relationship between the drug infusion and the effects on patient is not well established.

Nowadays, different clinical monitors are being developed in order to propose new variables to improve the decision-making process in medicine. However, trying to establish a strict criterion to correlate the new information with the physician's action based on traditional clinical variables is not an easy task. The main objective of this research was defining a novel general methodology capable of studying the feasibility of a new clinical variable (controlled variable) to guide the drug delivery process and then designing automatically a fuzzy-based decision system taking this new information into account. Firstly, the accuracy of the new monitor to guide the drug infusion should be analysed. Then, the relationship between the new measurement and the physician criteria based on their expertise can be automatically proposed. The resulting Fuzzy Inference System based on a set of rules and membership functions makes it possible to obtain an easily interpretable drug delivery protocol for the clinician. A four-step methodology was proposed.

- (i) Data collection for training process
- (ii) Preprocessing and analysis of data
- (iii) Designing the Fuzzy Inference System through a decision tree algorithm
- (iv) Validation of the results obtained

There are different possible scenarios in which our algorithm could be applied. Specifically, to test the methodology above, the analgesia drug delivery process was analysed. Although different commercial monitors have been proposed, the main problem for the analgesia control is the absence of a reliable monitor to measure pain in patients undergoing surgery [27, 28]. In this research, the suitability of the Analgesia Nociception Index (ANI) to guide the analgesic process under surgery was analysed. Training data were obtained from 15 patients undergoing cholecystectomy surgery. The paper is organised in the following way. The next section presents a detailed problem description as a starting point for this research. Section 3 provides a detailed explanation of the methodology proposed in this paper.

Section 4 presents the application of the methodology to the analgesia control field. Section 5 presents the results of the method. Section 6 includes the discussion of the results. Finally, in Section 7 we conclude the paper.

2. Problem Description

Delivering an appropriate amount of drug according to the real state of patient is such a hard task in medicine. Generally, physician evaluates the current state of the patient by means of specific monitors or using different clinical signs. Then, they decide whether it is necessary to change the drug dose. It is important to use the appropriate concentrations of medications to optimize clinical outcomes in patients in various clinical situations [29, 30]. However, finding the variable that can be directly related to the effect of drug is not a trivial problem.

As a matter of fact, a new trend has been based on the proposal and development of new variables, techniques, and monitors capable of offering new information that may be included in the decision-making process. However, are these new measures directly related to the process involved? How could we define a new drug supply protocol in order to include this new information? These are the questions that this research aims to answer. As a result, the main objectives of this paper are as follows:

- (1) Determining not only if the new controlled variable is able to guide the drug supply process but also which information should be specifically considered
- (2) Defining a rule-based decision system in order to guide the supply process taking the new information into account.

Actually, there are a lot of fields in which the development of this algorithm would result in a success: glucose monitoring [31, 32], anaesthesia [33], or therapeutic drug monitoring [34, 35]. In this research, we have focused on the control of analgesia. Optimizing the dose of opioid may limit the risk of overdosing and the risk of postoperative hyperalgesia and may reduce the time of recovery after surgical procedure [36]. However, the evaluation of analgesia and, therefore, the nociception-antinociception balance during surgery is a challenge to address due to the absence of an objective measure for monitoring analgesia. Traditional methods for supplying opioids use nonspecific and nonsensitive methods based on simple changes in vital signs such as movement, tachycardia, or lacrimation [28]. Recently, different monitors have been developed for measuring analgesia during clinical interventions proposing different information as nociception measures: heart rate information [37], electromyogram [38], electroencephalogram [39], or electrical skin conductance [40]. Nevertheless, the reliability of these monitors has not been deeply studied in clinical practice to assert that there exists a variable directly related to analgesia [27, 41].

Analgesia Nociception Index is a measure based on heart rate variability (HRV) analysis. HRV has been shown in several studies to measure Autonomic Nervous System tone, strongly influenced by anaesthetic drugs [42]. ANI has been

employed in several research in order to validate it as a device capable of measuring the nociception balance [43–46]. ANI seems more sensitive than other traditional measures based on hemodynamic response of patient under propofol interventions to moderate nociceptive stimuli [47]. Using the ANI monitor as a guidance variable for analgesic titration may reduce the time recovery after the intervention, as well as the consumption of the analgesic agent [48, 49]. Moreover, ANI may enable consistent reflection of stimulation during propofol-remifentanyl anaesthesia, improving detection of a possible inadequate nociception/antinociception balance [50].

For conscious patient, the sympathetic-parasympathetic balance is affected by psychological stress. Using ANI in this case does not exclusively detect nociception but may be modified by stress and emotion [51, 52]. In general, further research is needed to evaluate whether ANI is a tool able to provide beneficial effects to the patients during anaesthesia. Traditional studies tend to compare ANI information with postoperative patient's painful experience to validate the ANI monitor. Visual Analogue Scale (VAS) is a standard measurement tool in pain research and clinical practice [53, 54]. It is supposed that changes in VAS score represent a relative change in magnitude of pain sensation. However, trying to establish a correlation between the ANI index during surgery and the postoperative evaluation of pain through VAN is influenced by pain subjective experience of patients [55]. Another trend is studying the variation of the monitor's measure through the application of painful stimuli to the patient [56], a clinical practice that may damage the patient's health. As a result, more research is needed to find a feasible method to test the validity of the different alternatives proposed. Although no analytical relationship has been proposed between drug infusion and ANI index, very promising results have been reached when using ANI as guidance variable in analgesia [57, 58]. The algorithm developed in this research will be applied to propose a new solution to the analgesia problem from the AI point of view.

In light of the above, applying the novel algorithm proposed in this research to the control of analgesia will result in

- (i) Determining whether the information displayed by the Analgesia Nociception Index is suitable to guide the analgesic process
- (ii) Defining a Fuzzy Inference System considering the information displayed by the ANI capable of predicting the actions of the clinician.

3. Methods

In the present study, decision trees as well as fuzzy logic techniques were used. The basis of these algorithms is introduced in the following subsections. The main idea was using the information of a decision tree to design a Fuzzy Inference System (FIS). This structure will be capable of predicting the expert's decisions after a training step based on real data when a new monitor is involved in a drug supply process. Although more recent machine learning techniques have emerged with

the purpose of the automation of data analysis, fuzzy logic has been chosen in this study for several reasons. Firstly, most of medical decisions when changing the drug dose cannot be based on crisp values or strict predefined criteria. Thus, fuzzy values due to the presence of ambiguous concepts in the decision making process such as interpatient variability or the existence of a lack of knowledge behind the process are required. That is why using membership functions in order to define the different categories for the decision variables seems to be the most appropriate option for the decision-making process. On the other hand, fuzzy logic is a well-known method able to relate easily the heuristic knowledge to a set of rules in a natural manner. What is more, no complex mathematical modelling is needed as it is based on a linguistic characterisation of the quality of the controlled process. Obtaining a Fuzzy Inference System automatically through the algorithm we propose will result not only in the development of an automatic system for the drug supply trained with real data, but also in the definition of the basis of the process by means of a set of rules easily interpretable for clinicians.

The general scheme of the method proposed to design the decision-making system is shown in Figure 1.

One of the key steps in this methodology is the data collection. Data displayed by the new monitor must be recorded in parallel to the traditional drug supply process. It is important to note that the new monitor involved in this process should compute a numerical index in order to be able to apply the novel algorithm. To avoid conditioning the expert decision, the new information displayed must be hidden. Then, a preprocessing step is performed. Several proposals of the input data including different characteristics of the new measure must be considered. A decision tree algorithm is trained using the data recorded. The rules obtained will be the base to design the Fuzzy Inference System to predict the dose changes. On the one hand, it is possible to study the performance of the algorithm when trying to relate the new measure to the physician's actions. As a result, a first approach of the reliability of the new measure to guide the drug supply can be reached. Moreover, it is possible to determine which input proposal fits better to the decision-making process. On the other hand, the resulting Fuzzy Inference System consists of a set of rules whose interpretability improve the "user-friendliness" of the drug delivery protocol.

3.1. Decision Tree. Decision tree is a supervised machine learning algorithm able to build a model that makes predictions based on a known set of input data and known responses (output). The goal is to assign a class (categorical variable) from a finite set of classes to an observation. The decision tree consists of tests nodes linked to two or more subtrees and leafs or decision nodes labelled with a class which means the decision [59]. An instance is classified by starting at the root node of the tree. If the node is a test, the process continues with one of the subtrees. On the other hand, when a leaf is reached, the instance is classified with the correspondent label. An attribute node has exactly as many branches as its number of different value classes. Different algorithms to induce decision trees have been proposed [60, 61]. The

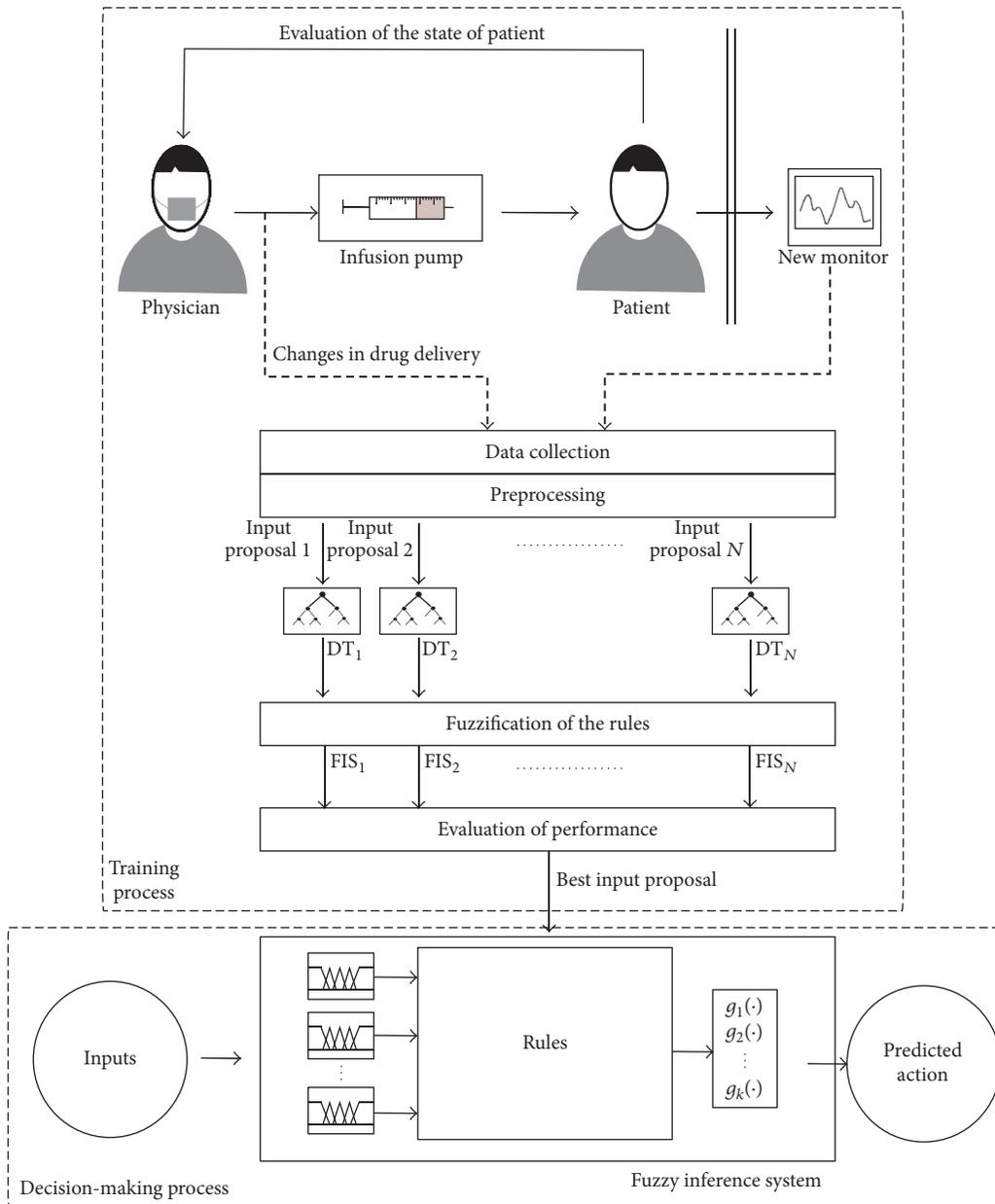


FIGURE 1: General scheme of the algorithm proposed in this study.

main idea relies on using statistical calculation of information gain from the attributes. As a result, attributes adding the most information about the decision are selected first in the decision tree construction.

For this research, a CART (Classification and Regression Trees) algorithm was proposed. This method introduced by Breiman et al. [62] is focused on minimising the relative sum of squared errors in the two partitions resulting from a split. Generally, a two-step process is developed: a preliminary induction of the model through a training set under the “divide and conquer” principle and a checking process of the accuracy from a testing set. The search for splits in CART is based on two main characteristics: the covariate to split on and splitting point within that covariate [63]. Firstly, trees are

grown to a maximal size stopping when no further splits are possible due to the lack of data [64]. Gini, similar to entropy criterion, is used as the splitting rule for classification. For a two-decision target the Gini measure of impurity of a node t is given by the expression below.

$$G(t) = 1 - p(t)^2 - (1 - p(t))^2, \quad (1)$$

where $p(t)$ is the relative frequency of one of the label in the node. Then, the tree is pruned back to the root based strictly on the training data according to a cost-complexity measure defined as

$$Ra(T) = R(T) + a|T|, \quad (2)$$

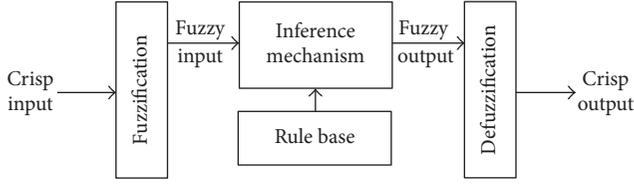


FIGURE 2: General structure of a Fuzzy Inference System.

where $R(T)$ is the training sample cost of the tree, $|T|$ is the number of terminal nodes, and a is a penalty imposed on each node increasing from 0 to a value sufficient to prune away all splits. As a consequence, the next split to be pruned is the one that decreases the total performance of the tree.

3.2. Fuzzy Inference System. Fuzzy Inference System (FIS) is a fuzzy logic based structure capable of making decisions in real time taking human expert knowledge into account. The main idea is based on mapping the inputs and the outputs through a set of predefined rules that involves the heuristic knowledge. According to fuzzy sets theory, each variable (input or output) is defined through a linguistic variable \tilde{u}_i whose value can be described through linguistic values \tilde{A}_i^j belonging to a universe of discourse U_i . Unlike crisp values, the values of the universe of discourse “belong to” a linguistic value in a certain degree $[0, 1]$ described by a membership function $\mu(u_i)$.

$$\mu_{A_i^j}(u_i) = X \rightarrow [0, 1]. \quad (3)$$

A value near 1 indicates that the value is almost fully in the set. A fuzzification process is necessary to turn crisp values to a fuzzy value. The singleton fuzzification is the most commonly used method. Then, the mapping of the inputs to the output is characterised by *if-then* rules. An inference step is needed to obtain conclusions from inputs and rule base. In this study, a Takagi-Sugeno inference system was developed.

$$\begin{aligned} \text{IF } u_1 \text{ is } A^1_i, u_2 \text{ is } A^2_i, \dots, u_n \text{ is } A^n_i, \\ \text{THEN } b_i = g_i(\cdot), \end{aligned} \quad (4)$$

where “ \cdot ” represents the argument of g_i function. As a result, the consequence of a Takagi-Sugeno inference is a function that may include the input terms u_i . Finally, a defuzzification method is needed to obtain a crisp value of the output:

$$y = \frac{\sum_{i=1}^R b_i \mu_i}{\sum_{i=1}^R \mu_i}. \quad (5)$$

The general structure of a Fuzzy Inference System is shown in Figure 2.

3.3. Fuzzification of the Decision Tree Rules. Generally, when a decision must be made in medicine, there is not a predefined universal criterion. It is mainly due to the different inter- and intravariability characteristics that the process involves. As a result, it does not make any sense to consider a crisp value as a

strict limit to make a decision. That is one of the main reasons why a FIS was introduced in this research. Furthermore, fuzzy logic is based on “categories” or membership functions easier to interpret for clinicians as it groups information with similar characteristics for the decision-making process.

One of the key steps when designing a FIS is related to the definition of the membership functions and the rule base. It is especially difficult when there is not a deep heuristic knowledge behind the process. To avoid this problem, a decision tree technique is proposed to obtain it automatically from real data. The limits of the membership functions will be defined through the conditions in the test nodes and the rules will inherit from it. However, while the limits of the decision tree are crisp values based on training data, fuzzy values are required for Fuzzy Inference Systems. In addition, when the number of training and testing data is limited, it can turn into a harder problem. In order to generalise our model and to take advantage of the fuzzy techniques, triangular and trapezoidal membership functions were used for intermedia and edge partitions of the universe of discourse, respectively. Moreover, we proposed to increase the limits of each membership function in 10% to get an overlap and avoid problems related to the limitation in the amount of data in the training step. The new limits for each membership function are calculated as shown below:

New lower limit = lower limit

$$- \frac{\text{upper limit} - \text{lower limit}}{2}$$

· 0.1

(6)

New upper limit = upper limit

$$+ \frac{\text{upper limit} - \text{lower limit}}{2}$$

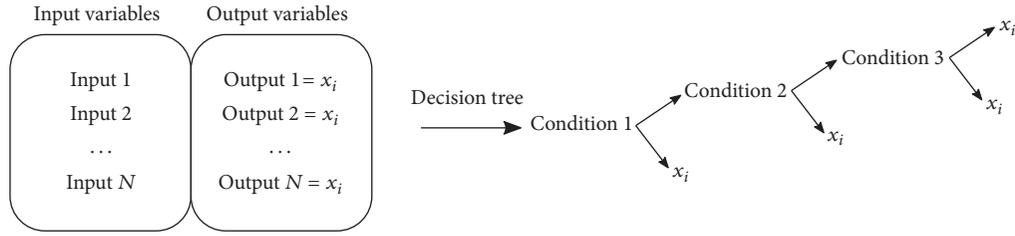
· 0.1.

The 2-step process for the fuzzification of the inputs is described in Figure 3.

Finally, the number of output functions matches the number of actions that the physician can handle. A constant function will be proposed for each action. For a general situation in which two decisions can be made (decreasing or increasing drug), a constant value of 0 and 100 could be associated with each action, respectively. In this case, the Fuzzy Inference System will calculate a number within the 0–100 range that could be considered as a percentage of action. A value of 50 could be regarded as the limit between both decisions.

3.4. Evaluation of the FIS Decision Maker. In order to evaluate the performance of the method proposed and the capability of the resulting Fuzzy Inference System to predict the decision-making process, a k -fold cross-validation must be performed [65]. The original sample is randomly divided into k equal sized subsamples. A single subsample is considered as the validation data for testing the Fuzzy Inference System, while the remaining $k-1$ subsamples are used as

1st step: application of decision tree



N : number of training data.

x_i : action to predict, $i = 1, \dots, K$, K = number of different actions to predict.

2nd step: fuzzification of the decision tree

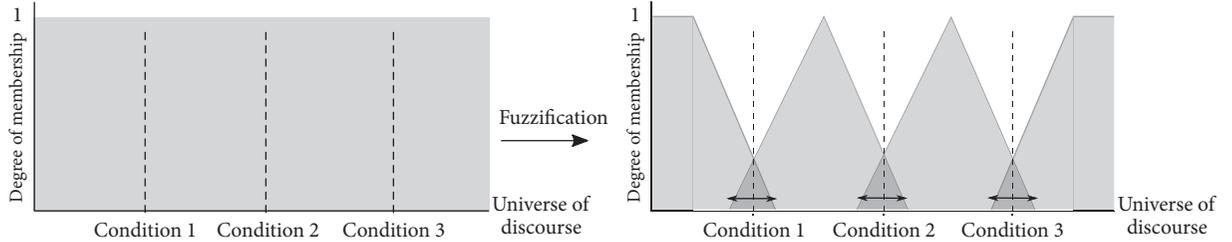


FIGURE 3: Graphic description of the Fuzzy Inference System definition process proposed in this research.

training data. The process is repeated k times varying the validation data and the results are averaged to obtain a single estimation. Different measures are calculated to study the performance of the classification [66]. The *accuracy* indicates the percentage of the dataset that are correctly classified by the proposed classifier. The *sensitivity* and *specificity* calculate the proportion of positive and negative records that are correctly classified, respectively. *Precision* refers to the fraction of relevant instances among the retrieved instances while *recall* is the fraction of relevant instances that have been retrieved over total relevant instances. The mathematical expressions to calculate the different measures are shown below.

$$\begin{aligned}
 \text{Accuracy} &= \frac{TP + TN}{TP + TN + FN + FP} \cdot 100 (\%) \\
 \text{Sensitivity} &= \frac{TP}{TP + FN} \\
 \text{Specificity} &= \frac{TN}{TN + FP} \\
 \text{Precision} &= \frac{TP}{TP + FP} \\
 \text{Recall} &= \frac{TP}{TP + FN}
 \end{aligned} \tag{7}$$

Given two classes, TP (true positives) refer to the positive records that have been correctly classified by the FIS, while TN (true negatives) are the negative records that have been correctly labelled by the classifier. On the other hand, FP (false positives) are the negative records that have been incorrectly labelled, while FN (false negatives) refers to the positive records incorrectly classified by the FIS. The

TABLE 1: Confusion matrix for positive and negatives records.

	Predicted	
	Positive	Negative
Observed		
Positive	TP	FN
Negative	FP	TN

confusion matrix to define the different measures is shown in Table 1.

Different input proposals based on different information from the new monitor are considered in the algorithm. In order to choose the best input proposal for the decision making process, the evaluation of the input variables resulting in the highest accuracy, sensitivity, specificity, precision, and recall will be considered for the final FIS. To determine whether the information provided by a new monitor is relevant for a specific decision-making process, the measures of performance should be compared with the results obtained in similar previous research based on traditional decision methods or, if it was not possible, being evaluated by an expert.

4. Analgesia Assessment Application

This study has been approved by the Ethics Committee for the Clinical Research of the Hospital Universitario de Canarias (2014-97 (760954923-54923-4-14)). After obtaining written informed consent from patient, fifteen patients undergoing cholecystectomy surgery were enrolled in this study. A total intravenous anaesthesia (TIVA) with propofol (hypnotic) and remifentanyl (analgesic) was performed for induction and maintenance of general anaesthesia. A Bispectral Index (BIS)

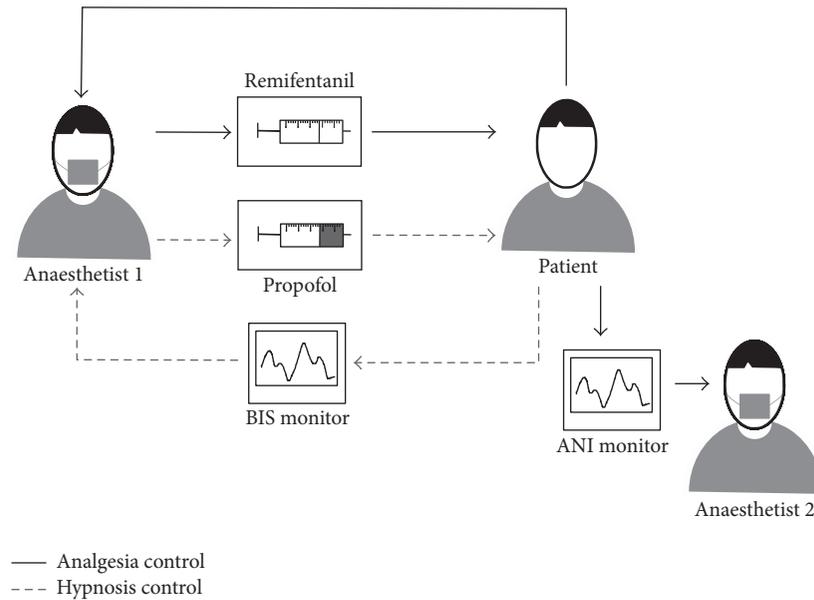


FIGURE 4: Scheme of the data collection process during the surgical interventions.

monitor (Aspect Medical Systems Inc., Newton, MA, USA) was used as guidance variable for propofol titration. The propofol dose was changed during the surgery to maintain BIS values between 40 and 60, with a target of 50. The target dose of remifentanyl was adjusted at the discretion of the anaesthesiologist, according to clinical practice parameters, anticipation to surgical stimuli, reactivity, or hemodynamic events. The dose of remifentanyl was adjusted in steps of $0.05\text{--}0.1\text{ mcg kg}^{-1}\text{ min}^{-1}$.

4.1. The Analgesia Nociception Index. The Analgesia Nociception Index (ANI), developed by Mdloris Medical System, is a noninvasive system that displays a continuous index related to the Autonomic Nervous System (ANS) through the heart rate variability. ANI is supposed to be a monitoring system of the parasympathetic activity that displays information about the level of pain or stress in patients undergoing surgery. ANI index computation is based on a frequency domain analysis of the ECG signal. The main idea leads to studying the spectral content of RR waves series after a preprocessing step focusing on the high frequency range only influenced by the parasympathetic tone. Specific ECG electrodes are placed on the chest or back of patient to collect the heart rate variability. Every second two measures ranging from 0 to 100 are displayed: instant ANI and mean ANI. Instant ANI is directly related to the reactions of the patient to painful stimuli while mean ANI, computed after two minutes of averaging instant ANI, is related to the effects of analgesia on a patient. As a result, instant ANI may detect the actions of the surgeon and mean ANI could be useful for the titration of analgesia. Target values between 50 and 70 for mean ANI have been proposed to avoid unwanted hemodynamic events. Values under 50 increase the possibility of hypertension, tachycardia, or bradycardia events.

4.2. Data Collection and Preprocessing. According to the process described in *Methods*, the data collection is the first step of the proposed methodology. In this case, two anaesthetists took part in each surgery. One of them was in charge of the drug supply, while the second one supervised the data recording process in a computer. A software in Matlab was developed in order to collect the data automatically. The scheme of the process is shown in Figure 4. Information displayed by the ANI monitor was hidden in order to avoid conditioning the decisions of the first anaesthetist. Instant ANI, mean ANI, and remifentanyl dose changes ($\text{mcg kg}^{-1}\text{ min}^{-1}$) were recorded every five seconds. Predefined surgical stimuli were also registered: nasogastric tube, laryngoscopy, incision, trocars, and the creation of pneumoperitoneum.

A postoperative offline study was made to try to correlate the rate changes of remifentanyl with the ANI values. Before the analysis, a data preprocessing was necessary. On the one hand, zero-index value because of poor signal or external disturbances was corrected through a linear interpolation algorithm. On the other hand, only changes of remifentanyl due to the analgesic state of patients were considered in this study. Changes of remifentanyl rate during surgery were based on two criteria: the anticipation to predefined painful surgical stimuli and the analgesic state of patient. As far as ANI is not able to predict the changes based on the anticipation to surgical stimuli, these values were not considered in this study.

4.3. Decision Variables. The accuracy of the algorithm prediction will be directly related to the combination of the input-output variables proposed. As a result, different information obtained through the ANI index was tried to establish a relationship with the action of the anaesthetist. Firstly, a categorical variable was considered for the output. Consequently,

TABLE 2: Description of the input variables proposed for the study.

Variables	Description
ANI_{i20}	Instant ANI of last 20 samples (100 s)
ANI_m and ANI_{i5}	Last mean ANI and last 5 samples of instant ANI (25 s)
ANI_{m5} and ANI_{i10}	Mean ANI of last 5 samples (25 s) and instant ANI of last 10 samples (50 s)
ANI_{m10} and ANI_{i20}	Mean ANI of last 10 samples (50 s) and instant ANI of last 20 samples (100 s)
ANI_{m10} and ANI_{i30}	Mean ANI of last 10 samples (50 s) and instant ANI of last 30 samples (150 s)
Average ANI_{m5} and ANI_{i5}	Average of last 5 samples of Mean ANI (25 s) last 5 samples of instant ANI (25 s)
Average ANI_{m5} and ANI_{i20}	Average of last 5 samples of Mean ANI (25 s) last 20 samples of instant ANI (100 s)
Average ANI_{m5} and ΔANI_{i5}	Average of last 5 samples of Mean ANI (25 s) the increment of last 5 samples of instant ANI (25 s)
Average ANI_{m5} and ΔANI_{i20}	Average of last 5 samples of Mean ANI (25 s) the increment of last 20 samples of instant ANI (100 s)

“increasing drug” or “decreasing drug” labels were defined as it fully considers the anaesthetist’s actions. Moreover, nonquantitative values of changes in remifentanyl dose were analysed as the rate changes were limited by the clinical protocol (steps of $0.05\text{--}0.1\text{ mcg kg}^{-1}\text{ min}^{-1}$).

For the input, different variables computed from instant ANI and mean ANI were taken into account. In addition, the effects of considering different time intervals for both variables were also analysed. Finally, to study the evolution of ANI, the increment of instant ANI was computed through the slope of the regression line that best matched the last values for a time interval. The variables proposed and their description are shown in Table 2.

5. Results

The decision-making methodology proposed was applied to the analgesic drug supply scenario. Fifteen patients undergoing cholecystectomy surgery were enrolled in this study. An example of the data collected during the interventions is shown in Figure 5. A total of 91 increasing/decreasing events were registered during the 15 surgeries. After discarding the changes due to the anticipation to painful stimuli, 53 events were finally considered for this study (32 increasing versus 21 decreasing).

5.1. Evaluation of the Proposed Variables. The performance of the resulting Fuzzy Inference Systems for both “increasing drug” and “decreasing drug” actions considering the different variables proposed in Section 4.3 is shown in Tables 3 and 4. A 4-fold cross-validation was applied for each combination according to the total number of training data.

In light of the results of the cross-validation, accuracy was over 60% for all the inputs considered. Specifically, accuracy was over 70% in most of the combinations proposed. Regarding the analgesia scenario, it was highly important to note that a low value of analgesia in patients can lead to complications and prolonged rehabilitation as well as the development of chronic pain with reduction in quality of life [67, 68]. Consequently, accurate increments of remifentanyl dose were desirable. That was why sensitivity and recall measures in Table 3 were specifically considered to choose the appropriate FIS. In this sense, there were up to four input-output combinations that resulted in sensitivity and recall

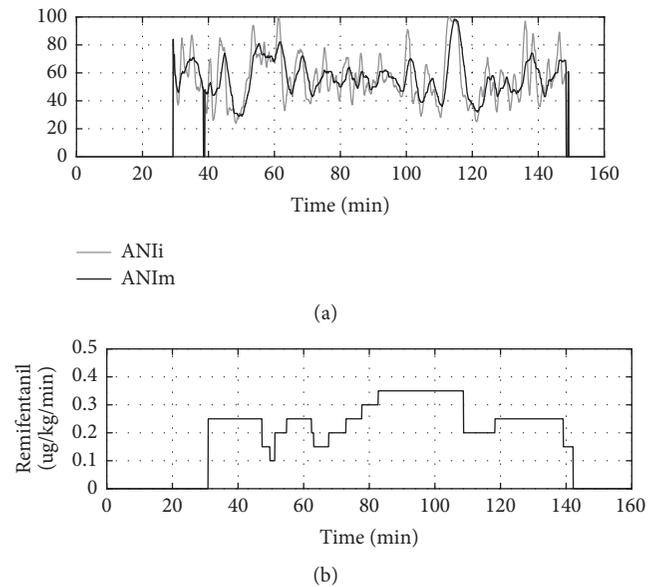


FIGURE 5: Example of the data collected for a patient undergoing cholecystectomy surgery. ANI registered (a) and remifentanyl infusion rate (b). ANIi: instant ANI. ANIm: mean ANI.

values over 0.8. Taking these four combinations into account as well as specificity and precision values, it was concluded that the best performance was reached when considering the average of the last 5 samples of mean ANI and the increment of the last 20 samples of instant ANI. Similar results were reached when analysing Table 4, as this input-output proposal resulted not only in the highest sensitivity-specificity combination but also in the highest precision value.

As a result, the last 5 samples of mean ANI and the increment of the last 20 samples of instant ANI were chosen for the input of our decision-making system. Regarding the performance reached, it was possible to affirm that there existed a relationship between the actions of the anaesthesiologist during surgery and the values displayed by the Analgesia Nociception Index.

5.2. Structure of the Fuzzy Inference System. Taking into account the comparison of performances in Section 5.1, the

TABLE 3: Comparison of performance for the different input combinations for when applying the proposed algorithm for increasing drug action.

Input	Accuracy (%)	Sensitivity	Specificity	Precision	Recall
ANI_{i20}	71.57	0.83	0.65	0.80	0.83
ANI_m and ANI_{i5}	71.57	0.77	0.74	0.83	0.77
ANI_{m5} and ANI_{i10}	66.07	0.69	0.71	0.80	0.69
ANI_{m10} and ANI_{i20}	69.64	0.81	0.62	0.80	0.81
ANI_{m10} and ANI_{i30}	71.57	0.76	0.72	0.83	0.76
Average ANI_{m5} and ANI_{i5}	71.57	0.77	0.72	0.81	0.77
Average ANI_{m5} and ANI_{i20}	71.57	0.83	0.64	0.80	0.83
Average ANI_{m5} and ΔANI_{i5}	62.09	0.71	0.56	0.73	0.71
Average ANI_{m5} and ΔANI_{i20}	75.41	0.82	0.71	0.80	0.82

TABLE 4: Comparison of performance for the different input combinations for when applying the proposed algorithm for decreasing drug action.

Input	Accuracy (%)	Sensitivity	Specificity	Precision	Recall
ANI_{i20}	71.57	0.65	0.83	0.5	0.65
ANI_m and ANI_{i5}	71.57	0.74	0.77	0.72	0.74
ANI_{m5} and ANI_{i10}	66.07	0.71	0.69	0.62	0.71
ANI_{m10} and ANI_{i20}	69.64	0.62	0.81	0.46	0.63
ANI_{m10} and ANI_{i30}	71.57	0.72	0.76	0.60	0.72
Average ANI_{m5} and ANI_{i5}	71.57	0.72	0.77	0.70	0.72
Average ANI_{m5} and ANI_{i20}	71.57	0.64	0.83	0.51	0.64
Average ANI_{m5} and ΔANI_{i5}	62.09	0.56	0.71	0.62	0.56
Average ANI_{m5} and ΔANI_{i20}	75.41	0.71	0.82	0.74	0.71

analysis and the result of the final FIS regarding the input-output proposal with the highest prediction score (the last 5 samples of mean ANI and the increment of the last 20 samples of instant ANI) are studied in this section. On the one hand, the decision tree obtained is shown in Figure 6. The value of the input variables proposed for the 53 registered events are shown in Figure 7.

Triangular as well as trapezoidal membership functions were used for both inputs. The number of the membership functions were defined by the total number of test nodes associated to each input. The limits, inherited from the test node conditions, were fuzzified according to the criteria in Section 3.3. The results are shown in Figure 8. A number of four and two membership functions were defined for the average ANI_{m5} and ΔANI_{i20} inputs, respectively.

Finally, the output functions of the FIS were defined. In this case, a two-decision system was needed: “increasing drug” and “decreasing drug” actions. Two constant output functions were proposed: “0” and “100” referred to the decreasing and increasing actions, respectively. As a result, the output of the FIS was a number within 0–100 range which could be regarded as a percentage of action. In this study, the results over 50 were considered as an “increasing drug” prediction while the values under 50 were considered as “decreasing drug” prediction. The fuzzy decision surface

obtained and the comparison with the nonfuzzified decision tree surface are shown in Figure 9.

To evaluate the performance of the FIS, the decision system was evaluated through a receiver-operating characteristic (ROC) curve by plotting the sensitivity, or true positive rate as a function of the false-positive rate. The ROC curves for both increasing and decreasing predictions are shown in Figure 10. An Area Under the Curve (AUC) of 0.8557 was reached for the predictive model proposed.

Finally, the prediction of the Fuzzy Inference System developed for the 53 training data is shown in Figure 11.

6. Discussion

A new algorithm for the design of a computer-based decision system in medicine has been presented. Specifically, the application of the proposed methodology in this research has resulted in the development of a Fuzzy Inference System as a computer-assisted medical decision-making for the analgesia scenario. On the one hand, it was possible to determine that there existed a relationship between the Analgesia Nociception Index and the remifentanyl supply during surgery. Particularly, the average of last five samples of mean ANI and the increment of last twenty samples of instant ANI reached an accuracy of 75.41%. Moreover, sensitivity and recall values

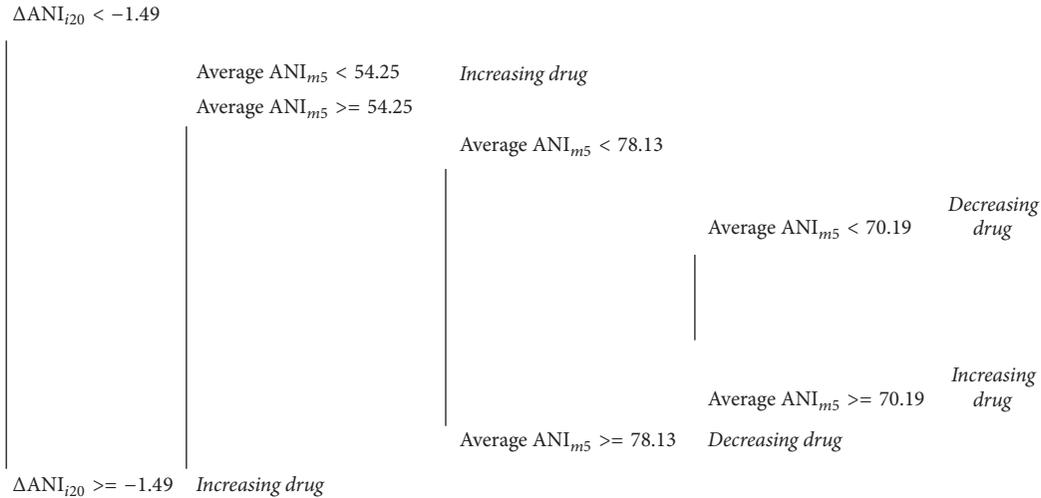


FIGURE 6: Decision tree obtained for the best input-output proposal (the last 5 samples of mean ANI and the increment of the last 20 samples of instant ANI) when applying the algorithm.

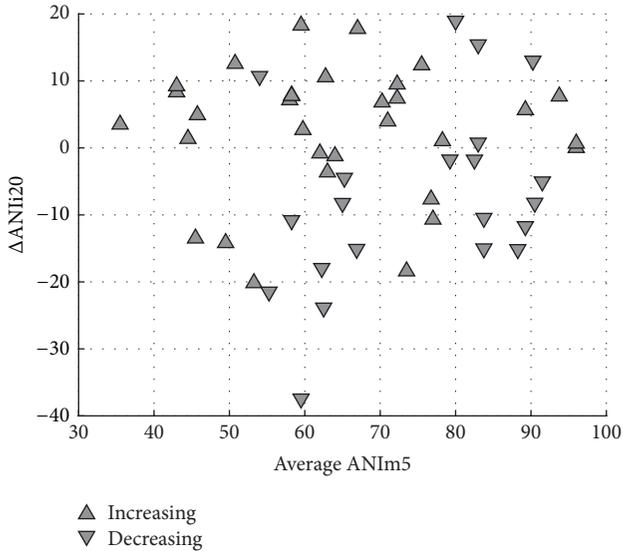


FIGURE 7: Value of average ANI_{m5} and ΔANI_{i20} computed for every remifentanyl rate change (increasing drug or decreasing drug).

over 0.8 were reached when predicting the increments of remifentanyl. Similar performance has been reached when applying different machine learning algorithms to optimize drug supply in medicine. Specifically, an accuracy ranging from 75% to 88% was obtained after a leave-one-out study when predicting the discrepancies between planned and delivered dose in proton therapy [69]. In [70], the overall predictive accuracy of the presented models for prediction of optimal cancer drug therapies was 80%. In the analgesia field, the development of models to predict the postoperative pain treatment reached an accuracy of 65% [71]. According to previous research, a clinically acceptable accuracy level is reached when applying our proposal. A two-input one-output FIS based on Takagi-Sugeno inference was developed. The resulting model reached an AUC of 0.8557 for both increasing

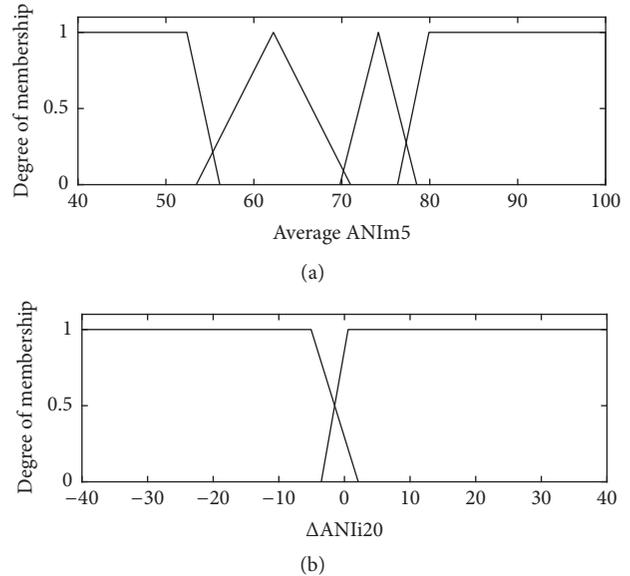


FIGURE 8: Input fuzzy partitions of the Fuzzy Inferences System. (a) Membership functions for average ANI_{m5} input. (b) Membership functions for the ΔANI_{i20} input.

and decreasing drug actions. As a result, the performance of the model to predict the actions of the anaesthesiologist may be classified as good [72].

From the Artificial Intelligence point of view, different approaches have been also proposed to fuzzy rule extraction from numerical data for classification [73–75]. However, the main novelty of this research is the definition not only of an automatic algorithm but also of the whole process in order to evaluate the reliability of a monitor involved in a decision-making system. As far as we know, this is the first study that tries to establish a correlation between the information displayed by a monitor and the expertise-based decision-making process during surgical interventions. Consequently,

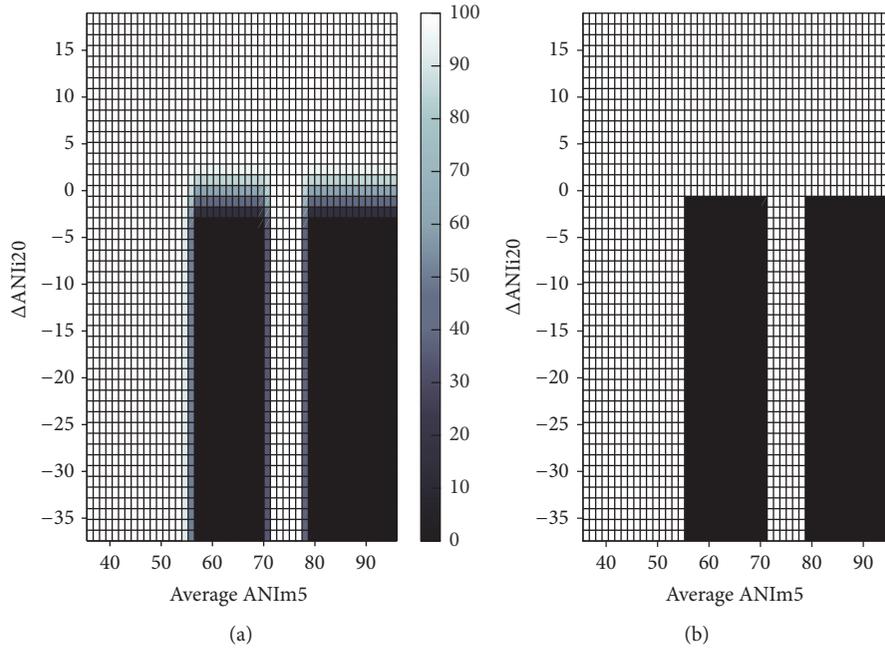


FIGURE 9: Comparison of the response surface for the Fuzzy Inference System (a) and for the decision tree (b). Output ranging from 0 (decreasing action) to 100 (increasing action).

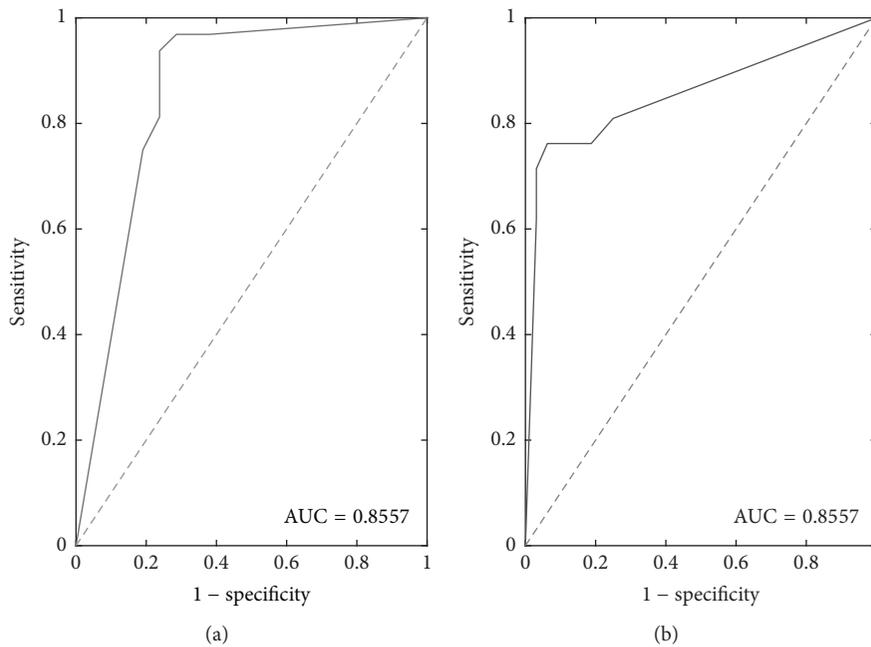


FIGURE 10: ROC curve showing the relationship between sensitivity (true positive rate) and 1 - specificity (true negative rate) determining the performance of the FIS model proposed to predict the increments (a) and decrements (b) of remifentanyl rate. AUC, Area Under the Curve.

it is possible to validate the accuracy of the device to drug assessment as well as to define a new drug delivery protocol. As a result, the knowledge behind the process is automatically built as a set of rules and categories or membership functions. On the one hand, this structure makes it easy to translate the knowledge into an interpretable language for clinicians. What is more, the output expressed as a percentage gives information about the decision and its reliability.

Merging decision trees with fuzzy logic has been previously proposed in order to handle uncertainty, ambiguity, and indeterminacy in the store information. However, unlike our method which results in a Fuzzy Inference System, the previous research has been based on fuzzy decision trees [76, 77]. They are mainly based on the use of decision tree whose nodes are not crisp values but membership functions. Bockstaller et al. proposed a new fuzzy decision tree for

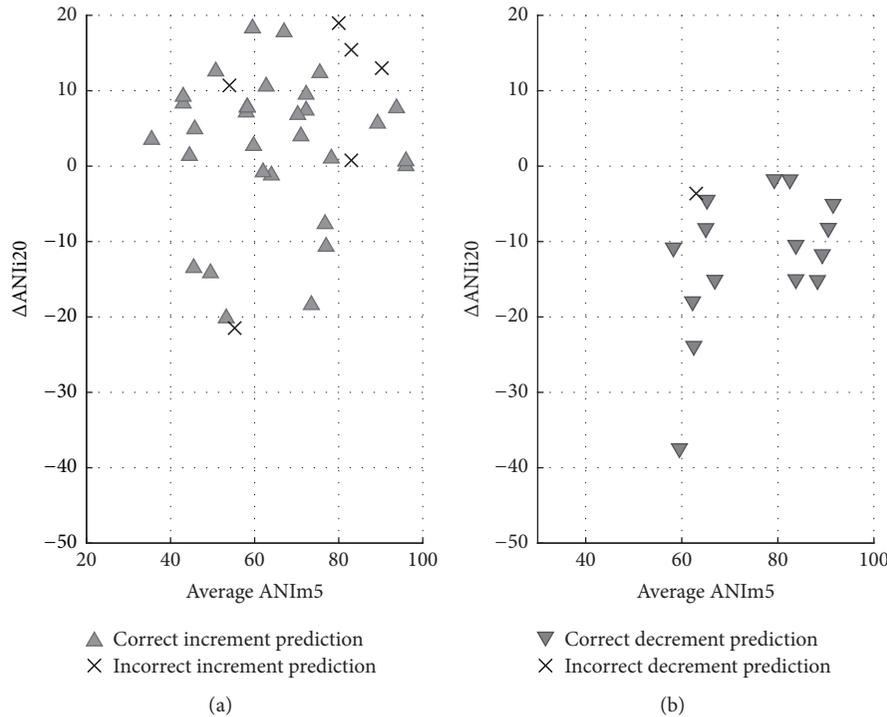


FIGURE 11: Prediction of the Fuzzy Inference System for increments (a) and decrements (b) of remifentanyl rate.

sustainability assessment [78]. They developed CONTRA tool to support the design of fuzzy decision tree. When using CONTRA, the user has to define the threshold and the limit values of each membership function in a previous step. Moreover, the choice of a weight or rank must be assigned to the different input variables in order to compute the output. Unlike our proposal, a previous heuristic knowledge is necessary in order to design the model. What is more, CONTRA tool limited the input variables to be aggregated between two and five in order to limit a maximum of thirty-two-decision rules. However, our method was able to work with an unlimited number of inputs.

One of the main limitations to this work is that the presented algorithm has been only applied to a two-decision system (increasing and decreasing drug) in analgesia. Further studies should be considered when applying this algorithm to another similar scenarios such as hypnosis, neuromuscular blockade or glucose control. In addition, including this information to an automatic control system would be the first step in order to automate the analgesic process through a closed-loop strategy.

7. Conclusion

This paper introduced a new methodology in order to design a fuzzy-based decision system to improve the drug delivery process when a new guiding variable is involved. Furthermore, the capability of a new monitor to guide the drug titration can be analysed. This methodology involved the whole process: from the recording of numerical data computed by the new monitor to the design of the Fuzzy

Inference System from real data. Fuzzy logic was used as it provides a well-understood mechanism for inducing classification rules from data and avoid possible problems related to the limitation of the number of training data. Rules as well as membership functions were extracted from a decision tree algorithm in order to automate the process. The algorithm proposed was tested in the analgesia scenario. In light of the results, it can be concluded that our method can be used to develop a decision-making system from real data in the medicine field although there exists a lack of knowledge behind the process.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This study was funded by the Spanish Ministry of Education, Culture and Sport (<http://www.mecd.gob.es>) under the "Formación de Profesorado" Grant FPU15/03347 to Jose M. Gonzalez-Cava.

References

- [1] J.-C. Pomerol, "Artificial intelligence and human decision making," *European Journal of Operational Research*, vol. 99, no. 1, pp. 3–25, 1997.
- [2] J. Lee, J. Ni, D. Djurdjanovic, H. Qiu, and H. Liao, "Intelligent prognostics tools and e-maintenance," *Computers in Industry*, vol. 57, no. 6, pp. 476–489, 2006.

- [3] P. Holimchayachotikul and K. Leksakul, "Predictive performance measurement system for retail industry using neuro-fuzzy system based on swarm intelligence," *Soft Computing*, vol. 21, no. 7, pp. 1895–1912, 2017.
- [4] I. Cenamor, T. de la Rosa, S. Núñez, and D. Borrajo, "Planning for tourism routes using social networks," *Expert Systems with Applications*, vol. 69, pp. 1–9, 2017.
- [5] P. Garrido, J. Barrachina, F. J. Martinez, and F. J. Seron, "Smart tourist information points by combining agents, semantics and AI techniques," *Computer Science and Information Systems*, vol. 14, no. 1, pp. 1–23, 2017.
- [6] Z. Wang and R. S. Srinivasan, "A review of artificial intelligence based building energy use prediction: contrasting the capabilities of single and ensemble prediction models," *Renewable and Sustainable Energy Reviews*, vol. 75, pp. 796–808, 2017.
- [7] B. Robson and S. Boray, "Data-mining to build a knowledge representation store for clinical decision support. Studies on curation and validation based on machine performance in multiple choice medical licensing examinations," *Computers in Biology and Medicine*, vol. 73, pp. 71–93, 2016.
- [8] L. Sarangi, M. N. Mohanty, and S. Patnaik, "Design of ANFIS based e-health care system for cardio vascular disease detection," in *Recent Developments in Intelligent Systems and Interactive Applications*, vol. 541 of *Advances in Intelligent Systems and Computing*, pp. 445–453, Springer International Publishing, Cham, 2017.
- [9] N. Chanamool and T. Naenna, "Fuzzy FMEA application to improve decision-making process in an emergency department," *Applied Soft Computing*, vol. 43, pp. 441–453, 2016.
- [10] L. A. G. Celi, R. J. Tang, M. C. Villarreal, G. A. Davidzon, W. T. Lester, and H. C. Chueh, "A clinical database-driven approach to decision support: predicting mortality among patients with acute kidney injury," *Journal of Healthcare Engineering*, vol. 2, no. 1, pp. 97–109, 2011.
- [11] J. S. de Bruin, K.-P. Adlassnig, A. Blacky, and W. Koller, "Detecting borderline infection in an automated monitoring system for healthcare-associated infection using fuzzy logic," *Artificial Intelligence in Medicine*, vol. 69, pp. 33–41, 2016.
- [12] R. Palaniappan, K. Sundaraj, S. Sundaraj, N. Hularaj, and S. S. Revadi, "A telemedicine tool to detect pulmonary pathology using computerized pulmonary acoustic signal analysis," *Applied Soft Computing*, vol. 37, pp. 952–959, 2015.
- [13] J.-L. Casteleiro-Roca, J. L. Calvo-Rolle, J. A. M. Pérez, N. R. Gutiérrez, and F. J. de Cos Juez, "Hybrid intelligent system to perform fault detection on BIS sensor during surgeries," *Sensors*, vol. 17, no. 1, article 179, 2017.
- [14] D. Nauck and R. Kruse, "Obtaining interpretable fuzzy classification rules from medical data," *Artificial Intelligence in Medicine*, vol. 16, no. 2, pp. 149–169, 1999.
- [15] M. Jakubczyk and B. Kaminski, "Fuzzy approach to decision analysis with multiple criteria and uncertainty in health technology assessment," *Annals of Operations Research*, vol. 251, no. 1–2, pp. 301–324, 2017.
- [16] J. A. Hazelzet, "Can fuzzy logic make things more clear?" *Critical Care*, vol. 13, no. 1, 2009.
- [17] M. J. Gangeh, A. Hashim, A. Giles, L. Sannachi, and G. J. Czarnota, "Computer aided prognosis for cell death categorization and prediction in vivo using quantitative ultrasound and machine learning techniques," *Medical Physics*, vol. 43, no. 12, pp. 6439–6454, 2016.
- [18] Y. Wang, P. Wu, Y. Liu, C. Weng, and D. Zeng, "Learning optimal individualized treatment rules from electronic health record data," in *Proceedings of the 2016 IEEE International Conference on Healthcare Informatics, ICHI '16*, pp. 65–71, 2016.
- [19] M. Dojat, A. Harf, D. Touchard, M. Laforest, F. Lemaire, and L. Brochard, "Evaluation of a knowledge-based system providing ventilatory management and decision for extubation," *American Journal of Respiratory and Critical Care Medicine*, vol. 153, no. 3, pp. 997–1004, 1996.
- [20] H. Ying et al., "A fuzzy discrete event system approach to determining optimal HIV/AIDS treatment regimens," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 4, pp. 663–676, 2006.
- [21] C. M. Salgado, S. M. Vieira, L. F. Mendonça, S. Finkelstein, and J. M. C. Sousa, "Ensemble fuzzy models in personalized medicine: application to vasopressors administration," *Engineering Applications of Artificial Intelligence*, vol. 49, pp. 141–148, 2016.
- [22] A. S. Fialho, F. Cismondi, S. M. Vieira et al., "Fuzzy modeling to predict administration of vasopressors in intensive care unit patients," in *Proceedings of the 2011 IEEE International Conference on Fuzzy Systems, FUZZ '11*, pp. 2296–2303, June 2011.
- [23] A. Marrero, J. A. Méndez, J. A. Reboso, I. Martín, and J. L. Calvo, "Adaptive fuzzy modeling of the hypnotic process in anesthesia," *Journal of Clinical Monitoring and Computing*, vol. 31, no. 2, pp. 319–330, 2017.
- [24] J. Schäublin, M. Derighetti, P. Feigenwinter, S. Petersen-Felix, and A. M. Zbinden, "Fuzzy logic control of mechanical ventilation during anaesthesia," *British Journal of Anaesthesia*, vol. 77, no. 5, pp. 636–641, 1996.
- [25] D. G. Mason, J. J. Ross, N. D. Edwards, D. A. Linkens, and C. S. Reilly, "Self-learning fuzzy control of atracurium-induced neuromuscular block during surgery," *Medical & Biological Engineering & Computing*, vol. 35, no. 5, pp. 498–503, 1997.
- [26] J. A. Méndez, A. Marrero, J. A. Reboso, and A. León, "Adaptive fuzzy predictive controller for anesthesia delivery," *Control Engineering Practice*, vol. 46, pp. 1–9, 2016.
- [27] A. Abad-Gurumeta, J. Ripollés-Melchor, R. Casans-Francés, and J. Calvo-Vecino, "Monitorización de la nocicepción, realidad o ficción?" *Revista Española de Anestesiología y Reanimación*, vol. 64, no. 7, pp. 406–414, 2017.
- [28] M. Gruenewald and C. Ilies, "Monitoring the nociception-anti-nociception balance," *Best Practice & Research Clinical Anaesthesiology*, vol. 27, no. 2, pp. 235–247, 2013.
- [29] J.-S. Kang and M.-H. Lee, "Overview of therapeutic drug monitoring," *Korean Journal of Internal Medicine*, vol. 24, no. 1, pp. 1–10, 2009.
- [30] A. S. Gross, "Best practice in therapeutic drug monitoring," *British Journal of Clinical Pharmacology*, vol. 52, pp. 5–9.
- [31] W. V. Tamborlane et al., "Continuous glucose monitoring and intensive treatment of type 1 diabetes," *The New England Journal of Medicine*, vol. 359, no. 14, pp. 1464–1476, 2008.
- [32] B. W. Bode, H. Sabbah, and P. C. Davidson, "What's ahead in glucose monitoring? New techniques hold promise for improved ease and accuracy," *Postgraduate Medical Journal*, vol. 109, no. 4, pp. 41–49, 2001.
- [33] T. Musialowicz and P. Lahtinen, "Current status of EEG-based depth-of-consciousness monitoring during general anesthesia," *Current Anesthesiology Reports*, vol. 4, no. 3, pp. 251–260, 2014.
- [34] M. Hallworth, "Therapeutic drug monitoring," in *Clarke's Analysis and Poisons*, p. 59, 2011.
- [35] M. C. Milone, "Analytical techniques used in therapeutic drug monitoring," *Therapeutic Drug Monitoring*, pp. 49–73, 2012.

- [36] K. Kuusniemi and R. Pöyhä, "Present-day challenges and future solutions in postoperative pain management: results from PainForum 2014," *Journal of Pain Research*, vol. 9, pp. 25–36, 2016.
- [37] M. M. R. F. Struys, C. Vanpeteghem, M. Huiku, K. Uutela, N. B. K. Blyaert, and E. P. Mortier, "Changes in a surgical stress index in response to standardized pain stimuli during propofol-remifentanyl infusion," *British Journal of Anaesthesia*, vol. 99, no. 3, pp. 359–367, 2007.
- [38] F. Von Dincklage, H. Velten, B. Rehberg, and J. H. Baars, "Monitoring of the responsiveness to noxious stimuli during sevoflurane mono-anaesthesia by using RIII reflex threshold and bispectral index," *British Journal of Anaesthesia*, vol. 104, no. 6, pp. 740–745, 2010.
- [39] R. K. Ellerkmann, A. Grass, A. Hoeft, and M. Soehle, "The response of the Composite Variability Index to a standardized noxious stimulus during propofol-remifentanyl anaesthesia," *Anesthesia & Analgesia*, vol. 116, no. 3, pp. 580–588, 2013.
- [40] B. Hullett, N. Chambers, J. Preuss et al., "Monitoring electrical skin conductance: a tool for the assessment of postoperative pain in children?" *Anesthesiology*, vol. 111, no. 3, pp. 513–517, 2009.
- [41] R. Cowen, M. K. Stasiowska, H. Laycock, and C. Bantel, "Assessing pain objectively: The use of physiological markers," *Anaesthesia*, vol. 70, no. 7, pp. 828–847, 2015.
- [42] R. Logier, M. Jeanne, J. De Jonckheere, A. Dassonneville, M. Delecroix, and B. Tavernier, "PhysioDoloris: a monitoring device for analgesia/nociception balance evaluation using heart rate variability analysis," in *Proceedings of the 2010 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC '10*, pp. 1194–1197, September 2010.
- [43] A. Gritsan, N. Dovich, D. Kurnosov, and E. Gritsan, "Control of the adequacy of analgesia during general anesthesia with the use of the monitor analgesia nociception index," *Anesthesia & Analgesia*, vol. 123, 2016.
- [44] E. Boselli, M. Daniela-Ionescu, G. Béguot et al., "Prospective observational study of the non-invasive assessment of immediate postoperative pain using the analgesia/nociception index (ANI)," *British Journal of Anaesthesia*, vol. 111, no. 3, pp. 453–459, 2013.
- [45] M. Gruenewald, T. Schoenherr, J. Herz, C. Ilies, A. Fudickar, and B. Bein, "Analgesia Nociception Index (ANI) for detection of noxious stimulation during sevofluran-remifentanyl anaesthesia," *European Journal of Anaesthesiology*, vol. 30, pp. 223–223, 2013.
- [46] L. Ursulet, J. Cros, J. De Jonckheere, P. Senges, A. Vincelot, and N. Nathan, "Bedside analysis of heart rate variability by Analgesia Nociception Index (ANI) predicts hypotension after spinal anesthesia for elective Caesarean delivery," *European Journal of Anaesthesiology*, vol. 29, p. 5, 2012.
- [47] M. Jeanne, C. Clément, J. De Jonckheere, R. Logier, and B. Tavernier, "Variations of the analgesia nociception index during general anaesthesia for laparoscopic abdominal surgery," *Journal of Clinical Monitoring and Computing*, vol. 26, no. 4, pp. 289–294, 2012.
- [48] M. Sesay, A. Mainchain, M. Biais, D. Liguoro, and K. Nouette-Gaulain, "Impact of the Anesthesia Nociception Index on remifentanyl consumption during anterior cervical discectomy," *Anesthesia and Analgesia*, vol. 123, p. 428, 2016.
- [49] G. Daccache, S. Goursaud, E. Lemasson, L. Berger, J. Fellahi, and J. Hanouz, "Target-controlled dosing of remifentanyl guided by the analgesia nociception index: a feasibility study," *European Journal of Anaesthesiology*, vol. 31, p. 35, 2014.
- [50] M. Gruenewald, C. Ilies, J. Herz et al., "Influence of nociceptive stimulation on analgesia nociception index (ANI) during propofol-remifentanyl anaesthesia," *British Journal of Anaesthesia*, vol. 110, no. 6, pp. 1024–1030, 2013.
- [51] G. Jess, E. M. Pogatzki-Zahn, P. K. Zahn, and C. H. Meyer-Frieem, "Monitoring heart rate variability to assess experimentally induced pain using the analgesia nociception index," *European Journal of Anaesthesiology*, vol. 33, no. 2, pp. 118–125, 2016.
- [52] Q. Yan, H. Y. An, and Y. Feng, "Pain assessment in conscious healthy volunteers: a crossover study evaluating the analgesia/nociception index," *British Journal of Anaesthesia*, vol. 118, no. 4, pp. 635–636, 2017.
- [53] P. S. Myles and N. Urquhart, "The linearity of the visual analogue scale in patients with severe acute pain," *Anaesthesia and Intensive Care Journal*, vol. 33, no. 1, pp. 54–58, 2005.
- [54] C. A. Bodian, G. Freedman, S. Hossain, J. B. Eisenkraft, and Y. Beilin, "The visual analog scale for pain: clinical significance in postoperative patients," *Anesthesiology*, vol. 95, no. 6, pp. 1356–1361, 2001.
- [55] J. A. Szentl, A. Webb, C. Weeraratne, A. Campbell, H. Sivakumar, and S. Leong, "Postoperative pain after laparoscopic cholecystectomy is not reduced by intraoperative analgesia guided by analgesia nociception index (ANI®) monitoring: a randomized clinical trial," *British Journal of Anaesthesia*, vol. 114, no. 4, pp. 640–645, 2015.
- [56] A. Castro, F. G. de Almeida, P. Amorim, and C. S. Nunes, "A novel multivariate STeady-state index during general ANesthesia (STAN)," *Journal of Clinical Monitoring and Computing*, vol. 31, no. 4, pp. 851–860, 2017.
- [57] M. Jeanne, M. Delecroix, J. De Jonckheere, A. Keribedj, R. Logier, and B. Tavernier, "Variations of the analgesia nociception index during propofol anesthesia for total knee replacement," *The Clinical Journal of Pain*, vol. 30, no. 12, pp. 1084–1088, 2014.
- [58] E. Boselli, R. Logier, L. Bouvet, and B. Allaouchiche, "Prediction of hemodynamic reactivity using dynamic variations of Analgesia/Nociception Index (Delta ANI)," *Journal of Clinical Monitoring and Computing*, vol. 30, no. 6, pp. 977–984, 2016.
- [59] V. Podgorelec, P. Kokol, B. Stiglic, and I. Rozman, "Decision trees: an overview and their use in medicine," *Journal of Medical Systems*, vol. 26, no. 5, pp. 445–463, 2002.
- [60] T. G. Dietterich, "Experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization," *Machine Learning*, vol. 40, no. 2, pp. 139–157, 2000.
- [61] A. Colin, "Building decision trees with the ID3 algorithm," *Dr. Dobb's Journal*, vol. 21, no. 6, pp. 107–109, 1996.
- [62] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, vol. 19, 1984.
- [63] A. Venkatasubramaniam, J. Wolfson, N. Mitchell, T. Barnes, M. Jaka, and S. French, "Decision trees in epidemiological research," *Emerging Themes in Epidemiology*, vol. 14, no. 1, article 11, 2017.
- [64] X. Wu and V. Kumar, "The Top Ten Algorithms in Data Mining," 2009.
- [65] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. 2, pp. 1137–1143, 1995.

- [66] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Addison-Wesley Longman Publishing Co., Inc., Massachusetts, Mass, USA, 5th edition, 2005.
- [67] H. Kehlet and K. Holte, "Effect of postoperative analgesia on surgical outcome," *British Journal of Anaesthesia*, vol. 87, no. 1, pp. 62–72, 2001.
- [68] H. Kehlet, T. S. Jensen, and C. J. Woolf, "Persistent postsurgical pain: risk factors and prevention," *The Lancet*, vol. 367, no. 9522, pp. 1618–1625, 2006.
- [69] P. Gueth, D. Dauvergne, N. Freud et al., "Machine learning-based patient specific prompt-gamma dose monitoring in proton therapy," *Physics in Medicine and Biology*, vol. 58, no. 13, pp. 4563–4577, 2013.
- [70] C. Huang, R. Mezenzev, J. F. McDonald, F. Vannberg, and B. Liu, "Open source machine-learning algorithms for the prediction of optimal cancer drug therapies," *PLoS ONE*, vol. 12, no. 10, 2017.
- [71] M. Gram, J. Erlenwein, F. Petzke et al., "Prediction of postoperative opioid analgesia using clinical-experimental parameters and electroencephalography," *European Journal of Pain*, vol. 21, no. 2, pp. 264–277, 2017.
- [72] H. F. Galley, "II. Solid as a ROC," *British Journal of Anaesthesia*, vol. 93, no. 5, pp. 623–626, 2004.
- [73] N. R. Pal and S. Chakraborty, "Fuzzy rule extraction from ID3-type decision trees for real data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 31, no. 5, pp. 745–754, 2001.
- [74] K. Nozaki, H. Ishibuchi, and H. Tanaka, "Adaptive fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 238–250, 1996.
- [75] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [76] C. Olaru and L. Wehenkel, "A complete fuzzy decision tree technique," *Fuzzy Sets and Systems*, vol. 138, no. 2, pp. 221–254, 2003.
- [77] X. Wang, X. Liu, W. Pedrycz, and L. Zhang, "Fuzzy rule based decision trees," *Pattern Recognition*, vol. 48, no. 1, pp. 50–59, 2015.
- [78] C. Bockstaller, S. Beauchet, V. Manneville, B. Amiaud, and R. Botreau, "A tool to design fuzzy decision trees for sustainability assessment," *Environmental Modeling and Software*, vol. 97, pp. 130–144, 2017.