



**HAL**  
open science

## Autonomous Car Fuzzy Control Modeled by Iterative Genetic Algorithms

Enrique Onieva, Javier Alonso, Joshué Pérez Rastelli, Vicente Milanés, Teresa de Pedro

► **To cite this version:**

Enrique Onieva, Javier Alonso, Joshué Pérez Rastelli, Vicente Milanés, Teresa de Pedro. Autonomous Car Fuzzy Control Modeled by Iterative Genetic Algorithms. . IEEE International Conference on Fuzzy Systems, Aug 2009, Jeju, South Korea. hal-00738079

**HAL Id: hal-00738079**

**<https://inria.hal.science/hal-00738079>**

Submitted on 3 Oct 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Autonomous Car Fuzzy Control Modeled by Iterative Genetic Algorithms

Enrique Onieva, Javier Alonso, Joshué Pérez, Vicente Milanés, Teresa de Pedro

**Abstract**—The techniques of Soft Computing are recognized as having a strong learning and cognition capability as well as good tolerance to uncertainty and imprecision. These properties allow them to be applied successfully to Intelligent Transportation Systems (ITS), a broad range of diverse technologies that designed to answer many transportation problems. The unmanned control of the steering wheel is one of the most important challenges faced by researchers in this area. This paper presents a method of automatically adjusting a fuzzy controller to manage the steering wheel of a mass-produced vehicle. Information about the state of the car while a human driver is handling it is captured and used to search, via genetic algorithms, for the best fit of an appropriate fuzzy controller. Evaluation of the fuzzy controller will take into account its adjustment to the human driver's actions and the absence of abrupt changes in its control surface, so that not only is the route tracking good, but the drive is smooth and comfortable for the vehicle's occupants.

## I. INTRODUCTION

The main interest in Intelligent Transport Systems (ITS) is the development of Advanced Driving Assistance Systems (ADAS) aimed at greatly reducing the number of road accidents [1]. In this line, research into autonomous vehicles is an investment for the future because its spin-offs include techniques and devices that can be used in ADAS. Also, there are specific situations in which autonomous vehicles can already be used without a human operator, such as docks, airports, etc.

The AUTOPIA research group has been used fuzzy logic as the inference motor for its experimental autonomous cars since 1996. The use of fuzzy logic was a must because the group's goal is to make the cars move as if a human were driving. Fuzzy logic allows the use of vague terms such as deviated to the right or too fast, so that the rule base needed to control the car is quite simple and readily understood.

The trade-off between performance and complexity is the main factor in ITS design. The use of artificial intelligence techniques is especially indicated when the aim is to emulate human control actions such as driving a car. Fuzzy logic [2] has become a particularly widely used methodological approach to these tasks [3] since Sugeno's work on vehicle control in the early 1990s. Fuzzy systems arose from the desire to describe complex systems linguistically, and fuzzy controllers allow a human approach to control design without the demand for knowledge of mathematical modeling of more conventional control design methods.

The authors are with the Industrial Computer Science Department, Instituto de Automática Industrial (CSIC), La Poveda-Arganda del Rey, 28500 Madrid, Spain (e-mail: onieva@iai.csic.es; jalonso@iai.csic.es; jperez@iai.csic.es; vmlanes@iai.csic.es; tere@iai.csic.es).

The fuzzy logic membership functions of the car's controllers were originally adjusted using a heuristic procedure. This intensively time-consuming method can now be replaced with an automated method of optimization using a genetic algorithm (GA). Our research group has previously used GAs to reduce computation time in new map design [4]. Indeed, they are general-purpose search algorithms, using principles inspired by natural genetic populations to evolve solutions to problems [5][6]. The basic idea is to follow the evolution over time of a population of chromosomes representing candidate solutions to the specific problem, with the chromosomes being subject to a process of competition and controlled variation. Each chromosome in the population has an associated fitness which will determine whether it will be used to form new chromosomes in the competition process, which is termed selection. The new chromosomes are created using genetic operators such as crossover and mutation.

This paper describes the use of genetic algorithms in the design of fuzzy logic controllers for steering wheel management with the following main objective: to find fuzzy coherent controllers that can reproduce human driver behaviour. The vehicle used to capture and process the human driver and experimentation information was a common car provided with automatic actuators operating on the controls. First, information about human driving behaviour was recorded. A two-phase GA-based system was then used to obtain a fuzzy controller able to mimic the human driver's behaviour and provide smooth control actions, so that the ride is comfortable and safe for the vehicle's occupants. Finally, the controller was tested in an experimental area to verify its good performance, comparing the controller's results with the measurements made for the human driver.

## II. EQUIPMENT AND INFRASTRUCTURE USED

This section describes the system as mounted on a car in order to conduct the trials. The work was done at the Industrial Automation Institute (IAI) research centre as part of the AUTOPIA program. AUTOPIA was created to build on the extensive experience of IAI in the development of autonomous robots and fuzzy control, with the goal of transferring autonomous mobile robot control technologies to computer-aided vehicle driving.

The system is mounted on a mass-produced car propelled by a petrol motor *Citroën C3 Pluriel* which is equipped with the necessary devices to perform autonomous driving [7]. Figure 1 shows the car used to carry out the experimentation. The input equipment consists essentially of:

- A double-frequency GPS receiver running in RTK carrier phase differential mode that supplies 2 cm resolution positioning at a refresh rate of 5 Hz. An autonomous uncorrected GPS receiver that calculates its position with accuracy between 10 and 20 m. When it receives positioning error correction data from a ground GPS base station, this precision can be increased to 1 or 2 cm. This correction signal is defined in [8], as are the update rate requirements which were always satisfied in the present work.
- Wireless LAN (IEEE 802.11) support that will allow the GPS to receive positioning error corrections from the GPS base station. Because of this, the positioning error will always be between 1 and 2 cm.
- An inertial measurement unit (IMU) Crossbow IMU300CC placed close to the centre of the vehicle.
- Car odometry supplied by a set of built-in sensors in the wheels, whose measurements can be read by accessing the Controller Area Network (CAN) bus of the vehicle. This was implemented by means of a CANcard2.6. Specifically, we read the car's *wheel-speed* data [9].



Fig. 1. Vehicle used to carry out the experiments.

An on-board computer is able to request values from each of the input devices with which to compute the fuzzy controller's input values. This then performs the inference process to yield output values to apply to move the steering wheel. The output is the reference position sent to a low-level control layer consisting of a PID controller that manages the DC motor attached to the steering bar to take it to the reference position.

The experiments were carried out in a private driving circuit at the IAI facilities denominated ZOCO (Spanish acronym of *driving zone*), which represents an inner city area, with a combination of straight-road segments, curves, 90 crossings, and slopes of up to 3%. Figure 2 shows an aerial view of ZOCO.

### III. ORBEX FUZZY COPROCESSOR

The controller is based on a computational model of a fuzzy coprocessor named ORBEX (Spanish acronym for Fuzzy Experimental Computer), which had been developed previously at the IAI [10][11]. ORBEX allows several forms of driving to be defined to emulate different types of driver (calm, quick, brusque, etc.), or to adapt the driving to traffic



Fig. 2. Aerial view of the experimentation zone (ZOCO).

conditions (platoons, overtaking, etc.). These strategies can be defined and implemented by means of *if ... then ...* rules in quasi-natural language, for instance:

**If lateral-deviation is high-right or angular-deviation is high-right then steering is left**

The t-norm *minimum* and the t-conorm *maximum* are used to implement *and* and *or* operators, respectively. Mamdani-type inference [12] is used, and the defuzzification operator is the centre of mass.

Sugeno et al. [13] proved that a fuzzy system modeled with singleton consequents is a special case of a fuzzy system modeled with trapezoidal consequents and can do almost everything the latter can. Singletons are very commonly used in practical control system applications [14][15][16]. For this reason, ORBEX and our present research use singletons as consequents of their fuzzy rules.

### IV. INFORMATION CAPTURE AND PROCESSING

The fuzzy controllers use three different inputs. Two of them are obtained from the match between the GPS positioning information and the reference route stored in the on-board computer: the *lateral* and *angular* errors. The lateral error represents the distance of the current car position from the theoretical car position if it were on the desired trajectory (the reference line). While theoretically it is defined in the complete interval  $\pm\infty$ , for practical purposes its value is restricted to  $\pm 5$  meters. The angular error is the angle formed by the reference line and the car's velocity vector. Its values are restricted to  $\pm 180$  degrees. Figure 3 shows a graphical representation of these variables. The third input variable represents the *actual steering wheel position*. The reason for using this is to provide feedback to the control loop in order to obtain fuzzy controllers that infer smooth control actions, and thus ensure a safe and comfortable ride for the vehicle's occupants. Due to the car used, the maximum steering turn is  $\pm 540$  degrees.

All the input values are normalized to the interval  $[-1, 1]$  to simplify their representation and processing. The lateral error is normalized using  $\pm 5$  meters as minimum-maximum value, the angular error using  $\pm 100$  degrees, and the actual steering wheel position using  $\pm 540$  degrees.

#### A. Information Capture

The GPS map that specifies the desired route is constructed automatically by tracking the route with a GPS-equipped car,

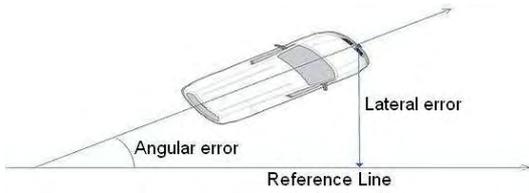


Fig. 3. Graphical representation of the lateral and angular error.

and, at the end of the run, a GA system selects the most significant waypoints that will be used to define the reference line [4]. The reference line created to capture information on the human driver's behaviour is shown in figure 4. The route consists of three curves to the left followed by three curves to the right. Once the desired route had been constructed, a human driver followed it while the on-board computer saved data on the lateral and angular errors and the steering wheel position at a 5Hz frequency. The normalized data obtained by monitoring the human driver's behaviour is shown in figure 5.

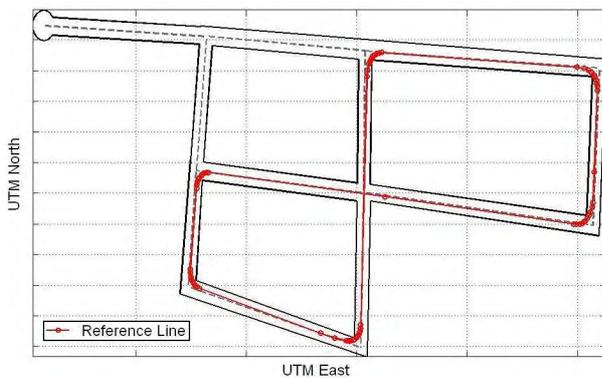


Fig. 4. GPS reference route.

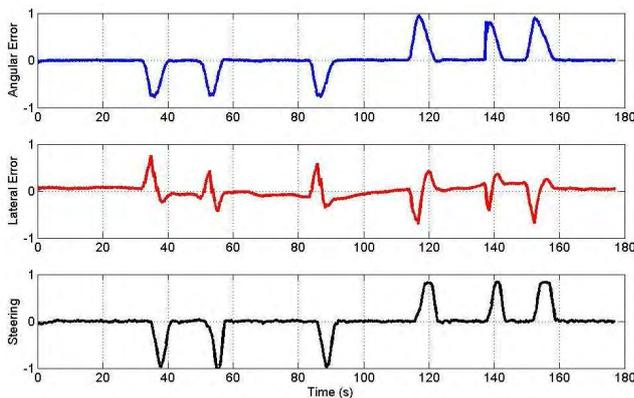


Fig. 5. Input/output values monitored by the computer while the human is driving.

### B. Information Processing

After capturing the information on the human driver's behaviour, a point swarm was obtained by sampling the data

at a 5Hz frequency. This point swarm is shown in figure 6(left), where the Z axis represents the normalized position of the steering adopted by the human driver, and the X-Y axes represent the observed normalized input values (lateral and angular errors or actual steering wheel position). To avoid having too many points in the space that represent common situations (such driving in a straight line), a  $21 \times 21 \times 21$  grid is defined, and at each point in the grid the mean output of the closest points in the real input is taken. There are zones in the grid such as extreme situations (extreme input values) which have no associated value. To ensure that some well-known cases are covered, the following points deduced from common sense were added to the swarm:

- $(x, y, z, 1) \forall x, y, z = 0.7, 0.8, 0.9, 1$ . This means: *if angular error is greater than 70 degrees and lateral error is greater than 3.5m and actual steering angle is greater than 378 degrees, all of them to the left, then turn the steering maximum to the right.*
- $(x, y, z, -1) \forall x, y, z = -0.7, -0.8, -0.9, -1$ . This is the mirror image situation if the previous example.

The point swarm after this information processing is shown in figure 6, and will be used by the GA method as a training set to ensure that the controller's output will be similar to the observed human behaviour, and that the fuzzy controllers generated have a smooth surface, i.e., that the car will be steered without any brusque manoeuvres.

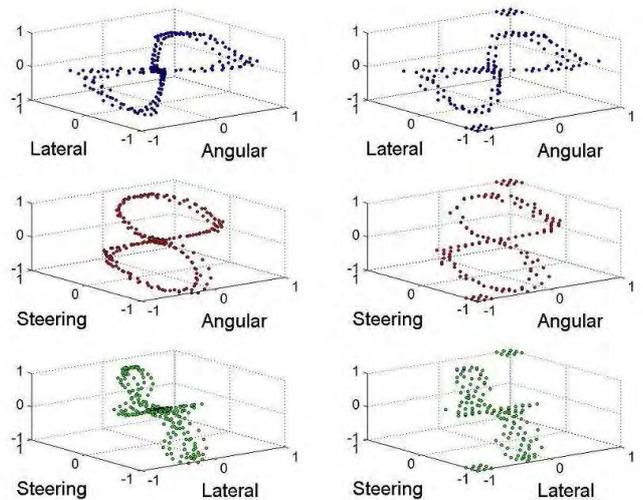


Fig. 6. Initial point swarm (left) and point swarm after processing (right).

## V. FUZZY CONTROLLER MODELING

A Sugeno fuzzy system is used to model the controller. The controller will have three fuzzy input variables called *Lateral-Error*, *Angular-Error*, and *Actual-Steering*. The fuzzification of each input variable is done by means of trapezoidal membership functions (MFs), with 3 symmetric MFs each used to codify the *Lateral-Error* and the *Angular-Error*, and 7 symmetric MFs to codify the *Actual-Steering* (13 MFs in total). *Lateral-Error* and *Angular-Error* have

labels  $\in [left, right, no]$  depending of the sense of the deviation or no deviation, respectively, and *Actual-Steering* has the labels  $\in [high - left, medium - left, low - left, centre, low - right, medium - right, high - right]$ .

There is only one fuzzy output variable, called *Steering*, with 9 possible singletons uniformly distributed in  $\pm 1$  which are labeled as:

- *R4* in -1: steering turn of 100% towards the right.
- *R3* in -0.75: steering turn of 75% towards the right.
- *R2* in -0.5: steering turn of 50% towards the right.
- *R1* in -0.25: steering turn of 25% towards the right.
- *NO* in 0: no steering turn.
- *L1* in 0.25: steering turn of 25% towards the left.
- *L2* in 0.5: steering turn of 50% towards the left.
- *L3* in 0.75: steering turn of 75% towards the left.
- *L4* in 1: steering turn of 100% towards the left.

The rule base consists of 63 rules ( $3 \times 3 \times 7$ ) with the format:

If (*Angular-Error* is *ANG*) and (*Lateral-Error* is *LAT*) and (*Actual-Steering* is *ACT - ST*) then *Steering* is *ST*, where:

- $[ANG, LAT] \in [left, no, right]$
- $[ACT - ST] \in [high - left, medium - left, low - left, centre, low - right, medium - right, high - right]$
- $[ST] \in [R4, R3, R2, R1, NO, L1, L2, L3, L4]$

In this way it is ensured that all the possible combinations of input labels are covered by one rule.

#### A. Membership Function Adjustment

In the first stage of the method, the membership functions that define the fuzzy controller's input variables will be tuned by means of a genetic algorithm. A membership function (MF) will be represented by four values (a,b,c,d). To generate suitable interpretable fuzzy controllers, the MFs are constrained to satisfy certain semantic restrictions [17]. Due to the nature of the problem, the MFs will be symmetric with respect to 0. For instance, if the car has drifted 4m to the right and this is taken to be highly deviated, then if it has drifted 4m to the left this will also be highly deviated. Additional constraints used are: the membership functions have to cover the entire range of a variable; there has to be a membership function that covers the extremes of the range of the variable with degree 1; every value of the range of the variable must be covered with 1 or 2 membership functions; and a value may not be covered with degree 1 by more than one membership function.

To represent a set of membership functions for a GA to work with, a list of real values is used, as shown in figure 7. One needs 4 values to codify 3 MFs, and 12 to codify 7 MFs, so that the present GA will need a list of 20 real values.

*Genetic Algorithm on MFs* A steady-state genetic algorithm is used to adjust the membership functions. It has the following parameters and operators:

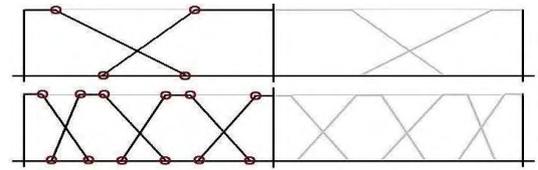


Fig. 7. Codification of 3 MFs with 4 real values (top), and of 7 MFs with 12 real values (bottom).

- Initialization: Each value  $v_i$  in a chromosome will be initialized to a uniformly distributed value in the interval  $[m_i \pm \xi (= 0.2)]$  with probability  $\rho (= 0.5)$ , otherwise,  $v_i = m_i$ , where  $m$  is the best chromosome found during the run. For the first iteration, a uniform set of MFs will be used as  $m$ .
- The population size is 15, and the genetic algorithm will be run for 25 generations in each iteration.
- Selection: Binary tournament.
- Crossover Operator: BLX- $\alpha (= 0.25)$  [18] is used to generate two offspring from two chromosomes selected as parents.
- Mutation Operator: Random change under a mutation rate,  $p_m (= 0.1)$  [5], [6].
- Replacement Method: Each generated chromosome will replace the worst chromosome in the population if it obtains a better fitness value than the latter.
- Fitness Function: To measure the quality of a certain controller two factors must be considered: the fit to the actions taken by the human driver, and the requirement of having a smooth surface for the responses not to be abrupt. The fit is evaluated using the mean squared error (MSE) between the controller's output and the actions taken by the human driver (training set described in section IV-B). The surface is evaluated using the greatest difference  $D$  between two adjacent points on the grid  $(i,j)$ ,  $i,j = [-1, -0.9, \dots, 0, \dots, 0.9, 1]$ . The fitness function used is a weighted aggregate of  $MSE$  and  $D$  ( $F = 0.75MSE + 0.25D$ ).

The parameter values above were assigned on the basis of empirical results.

#### B. Rule Base Adjustment

The second stage of the iterative method consists of adjusting the rule base (RB). To do that the GA will be responsible for associating each rule with one of the 9 possible singleton consequents explained above. To represent a rule base for the GA to work with, a list of 63 (recall that there are 63 rules) integer values  $\in [1, 9]$  will be used, where a value of 1 represents associating the consequent *R4* with the rule, 2 represents associating the consequent *R3*, etc. The GA will thus be working with a list of integer values of size 63.

*Genetic Algorithm on RBs* The steady-state genetic algorithm used to adjust the rule base uses the same fitness function, selection procedure, population size, and number

of generations and mutation operator as the previous genetic algorithm. The differences are in the following aspects:

- Initialization: Each value  $v_i$  in a chromosome will be initialized to a uniformly distributed value in the interval  $[m_i \pm \xi(= 2)]$  with probability  $\rho(= 0.75)$ , otherwise,  $v_i = m_i$ , where  $m$  is the best chromosome found during the run. For the first iteration, a random RB will be used as  $m$ .
- Crossover Operator: A two-point crossover [19] is used to generate one offspring from two chromosomes selected as parents.

C. Method Schema

The method is implemented in two different phases, which are repeated a given number of times, and which represent the two different genetic algorithms presented above, one of which is responsible for improving the membership functions, and the other for improving the rule base. The general schema used is shown in figure 8.

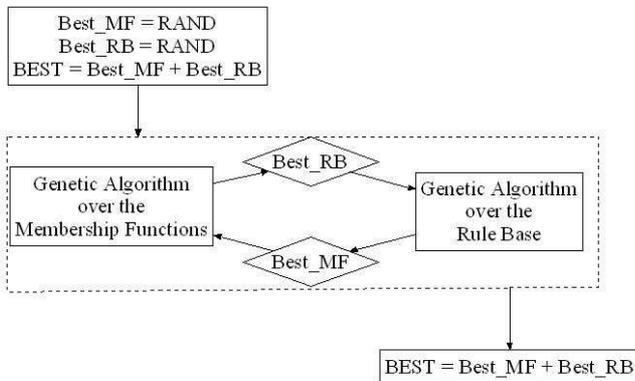


Fig. 8. The Two-Phase Schema Used.

In the first phase, the GA on the MFs, the chromosomes will be combined with the best RB obtained from the second phase to evaluate the fuzzy controller obtained. In the second phase, the GA on the RB, the chromosomes will be combined with the best MFs obtained in the previous phase. This procedure will be repeated 200 times, thus obtaining a good MF-RB combination at the same time as improving both.

VI. THE RESULTING FUZZY CONTROLLER

Once the input data has been processed as described in section IV, the method described in section V is applied, obtaining the fuzzy controller to be described below. The MFs obtained to codify the input values are shown in figure 9.

The RB obtained is represented by means of seven tables in which each cell represents the selected output consequent for a given value of *Angular-Error* (AE), *Lateral-Error* (LE), and *Actual-Steering* (AS) as presented in table I.

VII. EXPERIMENTATION

Once the iterative genetic process was completed, the controller presented in the previous section was tested in the

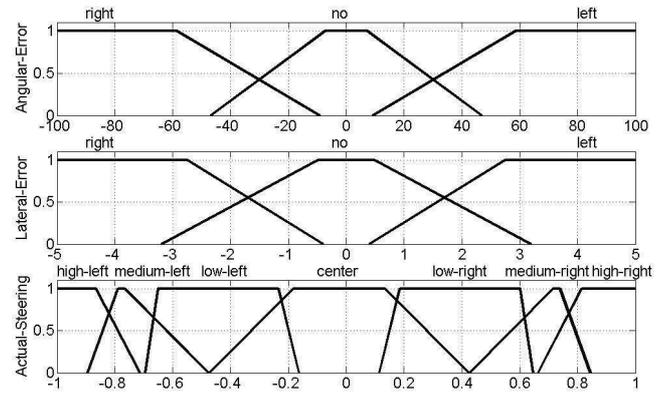


Fig. 9. MFs obtained for *Angular-Error* (top), *Lateral-Error* (centre), and *Actual-Steering* (bottom).

LE \ AE	right	no	left
right	L4	L4	NO
no	L4	L2	R2
left	L2	NO	R3
AS = high-left			

LE \ AE	right	no	left
right	NO	NO	R3
no	R2	R2	R3
left	R1	R4	R4
AS = high-right			

LE \ AE	right	no	left
right	L4	L2	L1
no	L3	L1	NO
left	R1	R2	R3
AS = medium-left			

LE \ AE	right	no	left
right	L3	NO	R1
no	L1	R1	R3
left	L1	R3	R4
AS = medium-right			

LE \ AE	right	no	left
right	L4	L1	L2
no	L4	NO	R3
left	R2	R1	R4
AS = low-left			

LE \ AE	right	no	left
right	L2	L2	L1
no	NO	NO	R3
left	R2	R1	R4
AS = low-right			

LE \ AE	right	no	left
right	L2	L2	NO
no	L1	NO	R1
left	NO	R1	R4
AS = centre			

TABLE I  
RULE BASE OBTAINED.

Driving Zone using the GPS reference route shown in figure 4. The behaviour followed by the resulting fuzzy controller will be described in the following. The route was driven at a speed of between 15km/h on the curved sections and 25km/h on the straight sections. The experiment is shown in terms of north-east UTM coordinates in figure 10, representing the reference route and the route followed by the autonomous car with the final controller obtained.

Table II presents the comparison of the measured average values of the angular and lateral errors and their time derivatives (with time in seconds) for the human driver and the GA-obtained controller along the test route. One observes that the performance of the controller was good, even improving over the measurements corresponding to the human driver. The low values of the derivatives indicate that the control actions were implemented smoothly, thus providing a comfortable ride for the vehicle's occupants.

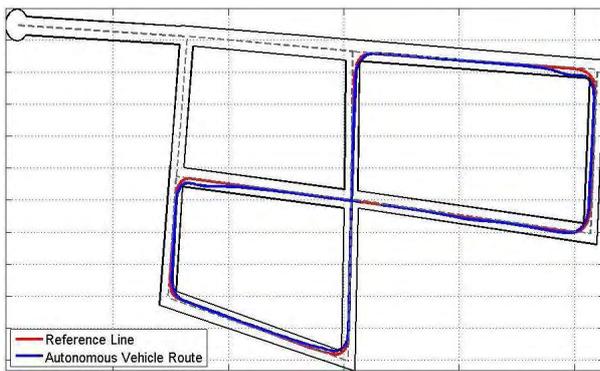


Fig. 10. Route followed by the autonomous car managed by the controller obtained using the GA.

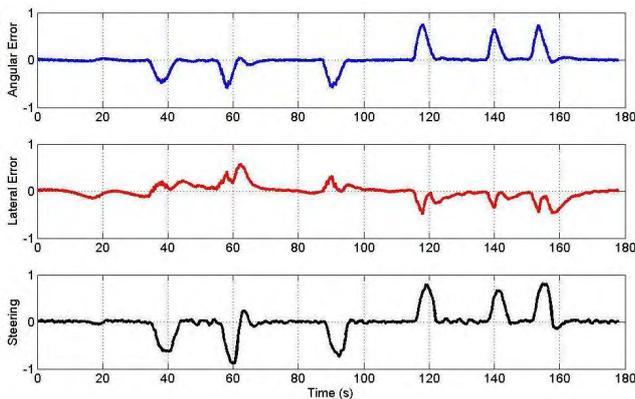


Fig. 11. Input/output values recorded by the computer during the autonomous driving trial.

## VIII. CONCLUSIONS

This work has presented an automatic adjustment of a fuzzy controller designed to automatically manage the steering wheel of a mass-produced vehicle which was equipped with the necessary instrumentation and software. To achieve this, an iterative genetic algorithm based method was implemented, capable of iteratively adjusting the membership functions and rule base which define a fuzzy controller. The method applied genetic algorithms with some constraints applied to the controllers to guarantee that the results could perform the automatic driving of an unmanned car. The controller also has smooth control actions to guarantee a safe and comfortable ride for the car's occupants.

The controller was tested in a private experimentation zone, and showed good, smooth, driving behaviour, even when moving at relatively high speeds. This represents a

	AE	LE	$\partial AE/\partial t$	$\partial LE/\partial t$
Human	11.72	0.46	9.02	0.46
Fuzzy Controller	8.87	0.40	8.32	0.40

TABLE II

COMPARISON OF THE DRIVING RESULTS OF THE HUMAN AND THE CONTROLLER.

good starting point for future ITS research.

Subsequent work will focus on improving the models obtained here in order to guarantee precise, comfortable, and smooth driving along a reference line. Furthermore, the consequences of using a determined number of membership functions will be studied, together with other car state variables or rule bases. Neither must safety and efficiency be forgotten. All these important features will be considered in future studies.

## ACKNOWLEDGEMENT

This work was supported by the Ministerio de Fomento project GUIADE (P9/08), the CENIT project MARTA 2007/06, and the TRANSITO project TRA2008-06602-C03-01.

## REFERENCES

- [1] Willie D. Jones. Keeping Cars from Crashing. *IEEE SPECTRUM*, September 2001, pp 40-45
- [2] L. A. Zadeh, "Fuzzy sets", *Inform. and Contr.*, **8**, 338-353 (1965).
- [3] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control", *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-15**(1), 116-132 (1985).
- [4] J.I. Serrano, J. Alonso, M.D. del Castillo and J.E. Naranjo. "Evolutionary optimization of autonomous vehicle tracks", *IEEE Congress on Evolutionary Computation*, **2**(2), 1332 - 1339 (2005).
- [5] J.H. Holland. "Adaptation in Natural and Artificial Systems". *The University of Michigan Press*. (1975).
- [6] D.E. Goldberg. "Genetic Algorithms in Search, Optimization, and Machine Learning". *Addison Wesley, New York*. (1989).
- [7] J.E. Naranjo, C. González, J. Reviejo, R. García, T. de Pedro, and M.A. Sotelo, "Using Fuzzy Logic in Automated Vehicle Control," *IEEE Intelligent Systems*. **22**(1), 26-45 (2007).
- [8] RTCM Special Committee no. 104, RTCM Recommended Standards for Differential NAVSTAR GPS Service, 1994, Arlington, VA: Radio Technical Commission Maritime Services. RTCM paper 194-93/SC104-STD.
- [9] V. Milanés, J. E. Naranjo, C. González, J. Alonso and T. de Pedro. "Autonomous vehicle based on cooperative GPS and inertial systems," *Robotica*, **26**, 627-633 (2008).
- [10] R. García and T. De Pedro. "First Application of the ORBEX Coprocessor: Control of Unmanned Vehicles," *EUSFLAT-ESTYLF Joint Conference. Mathware and Soft Computing*, **7**(2-3), 265-273 (2000).
- [11] R. García Rosa and T. De Pedro. "Modeling a fuzzy coprocessor and its programming language," *Mathware and Soft Computing*, **5**(2-3), 167-174 (1998).
- [12] E. H. Mamdani, "Applications of fuzzy algorithms for simple dynamic plant," *Proc. IEEE*, **62**(12), 1585-1588 (1974).
- [13] M. Sugeno, "On stability of fuzzy systems expressed by fuzzy rules," *IEEE Trans. Fuzzy Syst.*, **7**(2), 201-224 (1999).
- [14] J. E. Naranjo, C. González, R. García, T. de Pedro, and R. E. Haber, "Power steering control architecture for automatic driving," *IEEE Trans. Intell. Transp. Syst.*, **6**(4), 406-415 (2005).
- [15] P. Salgado and J. B. Cunha, "Greenhouse climate hierarchical fuzzy modeling," *Control Eng. Pract.*, **13**(5), 613-628 (2005).
- [16] G. Vachkov and T. Fukuda, "Structured learning and decomposition of fuzzy models for robotic control applications," *J. Intell. Robot. Syst.*, **32**(1), 1-21 (2001).
- [17] C. Mencar and A. M. Fanelli. "Interpretability Constraints for Fuzzy Information Granulation". *Information Sciences*, **178**, 4585-4618 (2008).
- [18] L.J. Eshelman, J.D. Schaffer. "Real Coded Genetic Algorithms and Interval Schemata". *Foundation of Genetic Algorithms 2*, L. Darrell Whitley (Ed.) (Morgan Kaufmann Publishers, San Mateo). 187-202 (1993).
- [19] L.J. Eshelman, R.A. Caruana, J.D. Schaffer. "Biases in the crossover landscape". *Proceedings of the 3rd international conference on Genetic algorithms*. 10 - 19 (1989).