HAL
open science

Genetic Fuzzy-based Steering Wheel Controller using a Mass-Produced Car

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

To cite this version:

Enrique Onieva, Vicente Milanés, Joshué Pérez Rastelli, Teresa de Pedro. Genetic Fuzzy-based Steering Wheel Controller using a Mass-Produced Car. International Journal of Innovative Computing, Information and Control, 2012, 8 (5 (B)), pp.3477-3494. hal-00732880

HAL Id: hal-00732880

https://inria.hal.science/hal-00732880

Submitted on 1 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.

International Journal of Innovative Computing, Information and Control
Volume **8**, Number **5(B)**, May **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 3477–3494

GENETIC FUZZY-BASED STEERING WHEEL CONTROLLER USING A MASS-PRODUCED CAR

E. Onieva, V. Milanés, J. Pérez and T. de Pedro

AUTOPIA Program at the Robotics and Automation Center (UPM-CSIC)
La Poveda-Arganda del Rey, Madrid 28500, Spain
{enrique.onieva; vicente.milanes; joshue.perez; teresa.pedro}@car.upm-csic.es

Received January 2011; revised May 2011

Abstract. *Intelligent Transportation Systems (ITS) cover a broad range of methods and technologies that provide answers to many problems of transportation. Unmanned control of the steering wheel is one of the most important challenges facing researchers in this area. This paper presents a method to adjust automatically a fuzzy controller to manage the steering wheel of a mass-produced vehicle to reproduce the steering of a human driver. To this end, information is recorded about the car's state while being driven by human drivers and used to obtain, via genetic algorithms, appropriate fuzzy controllers that can drive the car in the way that humans do. These controllers have satisfy two main objectives: to reproduce the human behavior, and to provide smooth actions to ensure comfortable driving. Finally, the results of automated driving on a test circuit are presented, showing both good route tracking (similar to the performance obtained by persons in the same task) and smooth driving.*
Keywords: Autonomous vehicles, Genetic algorithms, Fuzzy control, Fuzzy logic, Intelligent vehicles, Intelligent transportation systems (ITS), Lateral control

1. **Introduction.** Autonomous vehicle control is an open field of research in the area of Intelligent Transportation Systems (ITS). ITS apply information and communication techniques so as to achieve both safer and more efficient driving. In this sense, automotive manufacturers have introduced Advanced Driver Assistance Systems (ADAS) in their commercial car as an attempt to prevent collisions and minimize injuries [1]. These systems are based on different sensors capable of receiving the information coming from vehicle's surroundings. This information is used to warn the driver in case of unexpected traffic circumstances. The next step is to provide the vehicle with autonomous systems capable of not only warning the driver but also acting so as avoiding traffic accidents.

Different research groups have faced ITS problems worldwide, like California PATH (Partners for Advanced Transit and Highways) [2, 3], VaMoRs-P [4], VITA (VIsion Technology Application) [5], Ohio State University [6] and Carnegie Mellon University [7] in the United States; PROMETHEUS (Program for a European Traffic with Highest Efficiency and Unprecedented Safety) [8], the ARGO vehicle [9] and CyberCars project [10] in Europe; and several research programs under ITS Japan [11, 12].

AUTOPIA program at Center of Automation of Robotics (CAR) of the Spanish National Research Council (CSIC) has been working in autonomous vehicles during last 15 years. A fully-autonomous vehicle is based on managing vehicle's actuators – i.e., steering wheel and brake and throttle pedals. To this end, vehicles are equipped with the instrumentation and software necessary to perform driving-related tasks autonomously. In recent work [13], longitudinal control – i.e., brake and throttle pedals – was automated

using fuzzy logic in a mass-produced vehicle – a convertible Citron C3 Pluriel. This communication deals with the automatic control of the steering wheel (also known as lateral control) of this gas-propelled vehicle.

The actual implementation of full automatic-steering control is one of the hardest disciplines in the intelligent-vehicles field, and perhaps has a long way to go before it comes to market [14]. However, this too may be because it is receiving much research attention, as is reflected in the DARPA (Defense Advanced Research Projects Agency) Grand Challenge [15, 16, 17] which is a prize competition for driverless cars, sponsored by the the central research organization of the United States. DARPA has sponsored three competitions, all in the area of autonomous vehicles. Stanford University's winning entry in the second DARPA Grand Challenge exhibited lane following behavior by evaluating a set of candidate trajectories that tracked the desired path [18]. The last competition, the DARPA Urban Challenge, was won by Boss from the Tartan Racing Team, finishing the four-hour race 19 minutes and 8 seconds ahead of its closest rival [19, 20].

This work is motivated for being able to mimic the human behavior at driving. Persons are able to drive in a very proper way with making only use of vague information about the state of the vehicle; it is important to note that we are not able to measure distances with respect to the center of the lane, but we are able to drive in a satisfactory way with few more that the knowledge of *the vehicle is deriver thought a side of the road.* To emulate this simple but more accurate results obtained by persons when driving is the main motivation of this work.

The trade-off between performance and complexity is a main factor in steering control systems design. The use of artificial intelligence techniques is especially indicated when the aim is to emulate human control actions such as driving a car. Different artificial intelligence techniques have been described in the literature for steering wheel control. In [21] parameters of steer-by-wire are modeled by means of a genetic algorithm. [22] shows how to learn to manage a steering of a simulated vehicle by means of reinforcement learning. Fuzzy logic is used in [23] as approach, and neural networks [24], with artificial vision [25], positioning systems [26] or a fusion of both [27] being the usual forms of getting information about the road. In [28] it is presented an adaptive steering controller for vehicle tracking with independence of the dynamics.

Fuzzy logic [29] has become a particularly widely used methodological approach on real world applications control. A vehicle is a very complex nonlinear system. Therefore, it is very difficult to accurately formalize mathematically. Methods described in the literature to guarantee asymptotic global stability can be applied for this purpose. Fuzzy logic based systems do not require models, which makes them especially appropriate for processes such as driving, whose mathematical formalism is not clear or global for all the cases. In such systems and especially in fuzzy control systems, extensive experimental validation is often considered proof enough of stability. Fuzzy systems arose from the desire to describe complex systems linguistically [30], and fuzzy controllers allow a human approach to control design without the demand for knowledge of mathematical modeling of more conventional control design methods [31, 32, 33].

The membership functions and the rule base of a fuzzy controller are usually generated either iteratively, by trial-and-error, or by human experts. A task such as this is a natural candidate for an automatic optimization method since it will search for membership functions that will cause the controller to perform optimally. In much the same way, a Genetic Algorithm (GA) can be used to generate the rules which use these membership functions [34, 35]. Genetic Algorithms are general purpose search algorithms which use principles inspired by natural genetic populations to evolve solutions to problems [36, 37]. The basic idea is to allow a population of solutions to the problem to evolve over time

through a process of competition and controlled variation termed selection. GAs have been extensively applied in the field of control engineering [38, 39]. The application of GA is one way of optimizing the performance of the controllers by automatically tuning those parameters [40, 41].

The contribution of this work is an autonomous steering control for a mass produced vehicle based on fuzzy logic and tuned by genetic algorithms. The main innovative issue with respect to the systems found in bibliography lies in the fact that the system is not looking for a perfect control of the vehicle (0 error) but for controllers able to obtain precision similar to the ones obtained by human drivers, as well as to provide smooth control actions, in order to carry out a comfortable drive for possible occupiers; finally, controllers must be easily understandable and interpretable. To achieve that, controllers are evaluated in terms of performance and smoothness, as well as the genetic optimization process is coded to satisfy imposed restrictions about the interpretability of the controllers.

As commented previously, the principal aim is to find fuzzy controllers that are coherent and capable of emulating the behavior of a human driver. The vehicles used for the capture and processing of information of the human driver and for the experiments were common mass-produced cars, provided with automatic actuators operating on the controls (power-assisted steering and accelerator). In the first stage, data were captured about the human drivers' handling the car following a reference circuit while, apart from the visual information, the computer shown to him the deviation values with the aim of minimizing errors. Data were then processed to extract the relevant information on the driver's steering. A system was then created that is capable of being fed that information and returning an appropriate fuzzy controller via the application of a genetic algorithm. The genetic algorithm is applied in an iterative two-phase procedure, with the first attempting to improve the fuzzy controller's membership functions and the second to improve its rule base. These two phases are executed in a loop over several repetitions. Finally the controllers were tested in a private experimental area to verify that their behavior is similar to that of the human driver.

2. **Equipment and Infrastructure.** The work described in this paper was carried out at the Centre for Automation and Robotics (CAR). Based on the extensive experience of CAR in the development of autonomous robots and fuzzy control. The goal of AUTOPIA is to transfer autonomous mobile robot control technologies to computer-aided vehicle driving by developing a test-bed infrastructure for experimenting with control systems, strategies, and sensors applied to automatic vehicle driving [42] that is open to groups interested in this field of research.

The experiments to be described below employed the following resources:

- An asphalted test circuit denominated ZOCO (Spanish acronym of Driving Zone) whose aerial view is shown in Figure 1. ZOCO was initially designed as an inner city area, with a combination of straight-road segments, curves, 90 crossings, and slopes of up to 3%. The out of road zones of ZOCO are approximately at the same level that the asphalted ones, so, it is easy to use the entire surface in order to implement curves of different curvature radius in order to test the performance of the steering controllers in as many situations as possible.
- A double-frequency GPS (Global Positioning System) 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.

- Coverage of a Differential Global Positioning System (DGPS) supplied by a high precision GPS base station.
- 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.
- To prevent errors when the GPS signal is lost, car odometry is 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.
- In addition, an Inertial Measurement Unit (IMU) Crossbow IMU300CC is placed close to the centre of the vehicle.
- All the equipment is mounted on a *Citroën* C3 Pluriel equipped with a actuators for the steering wheel and pedals, and a computer connected to the communications network. The vehicle is shown in Figure 2.

FIGURE 1. Aerial view of ZOCO

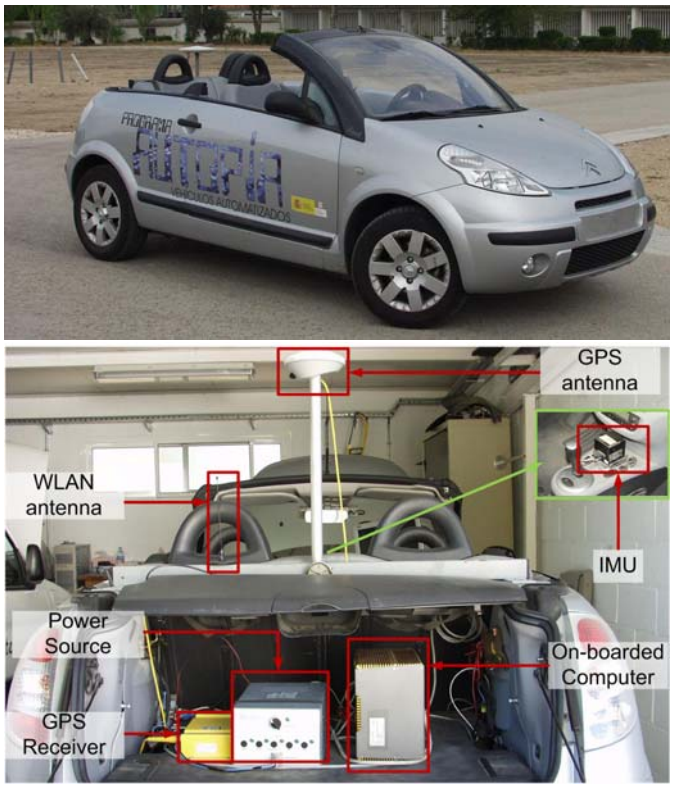


FIGURE 2. Vehicle used in the present work

2.1. **Fuzzy inference system.** The ORBEX [43] (Spanish acronym of Fuzzy Experimental Computer) fuzzy development environment was used to construct the controllers. With ORBEX, various forms of driving can be defined to emulate different types of drivers (quiet, quick, thrifty, etc.), or to adapt the driving to traffic conditions (platoons, overtaking, etc.). These strategies can be defined and implemented by means of *if ... then ...* rules in an almost natural language.

ORBEX works with fuzzy controllers that use trapezoidal membership functions to codify the input variables and singletons, that are punctual values, to codify the output variables. This makes the fuzzy controller to be equivalent to a zero-order TSK fuzzy controller [44]. Rules are defined as, for example:

$$\text{IF } x_1 \text{ is } X_1^{i_1}\ (AND|OR)\ x_2 \text{ is } X_2^{i_2}\ \ldots (AND|OR)\ x_N \text{ is } X_N^{i_N}$$
$$\text{THEN } out = R_{i_1,i_2,\ldots,i_N}$$

where $X_v^{i_v} \in \{X_v^1, X_v^2, \ldots, X_v^{n_v}\}$ are the membership functions used to codify the input x_v, there are n_v different membership function for that variable and $R_{i_1,i_2,\ldots,i_N}$ is a numerical value representing the location of the singleton that acts as rule consequent.

Binary operators *AND/OR* are implemented by t-norm *minimum* and t-conorm *maximum*. Mamdani-type inference [45] is used, and the defuzzification operator is the centre of mass. Therefore, the crisp value of y_{out} of an output variable is calculated as:

$$y_{out} = \frac{\sum y_i \cdot w_i}{\sum w_i} \tag{1}$$

where w_i represents the weight of the i-th rule and y_i is the value of the output y inferred by the i-th rule. The weight of a rule represents its contribution to the global control action (calculated as the minimal degree of current crisp input value membership of its respective fuzzy partitions).

Controllers in ORBEX are written in general text files which can be read and executed from external programs by simple C++ instructions ($fzzread('controller.txt')$ or $output = fzzexecute(input_1, input_2, \ldots)$).

The use of trapezoids and singletons for codifying the input and output variables guaranties that the execution of the control stage will be fast, since they are only needed sums and products to infer the value of membership of a value to a fuzzy partition, this quality if more than desirable for implementing real time systems such the ones involved in managing vehicles.

3. **Information Capture and Processing.** Fuzzy controllers use two input variables, the *lateral* and *angular* error, they are obtained from the match between GPS positioning information and the reference route defined as GPS digital cartography. The lateral error is the distance from the current position of the car to the theoretical position if it was on the the reference line of its desired trajectory. The angular error is the angle (in degrees) between the reference line and the car's velocity vector. Figure 3 shows a graphical representation of these variables. The output is the desired position of the steering wheel. This output will be the reference position sent to a low-level control layer consisting of a proportional integral derivative (PID) controller that manages a motor attached to the steering bar to take it to the reference position described in detail in [46].

Reference line constructed to capture data about the behavior of the human drivers when follow it is shown in Figure 4 over a schematic representation of *ZOCO*. GPS points are obtained automatically by tracking the route with a GPS equipped car, and, after that, a computer system selects the most significant way points that will be define the reference line. The full selection process is described in [47].

FIGURE 3. Graphical representation of the input variables (lateral and angular error) with respect to the reference line

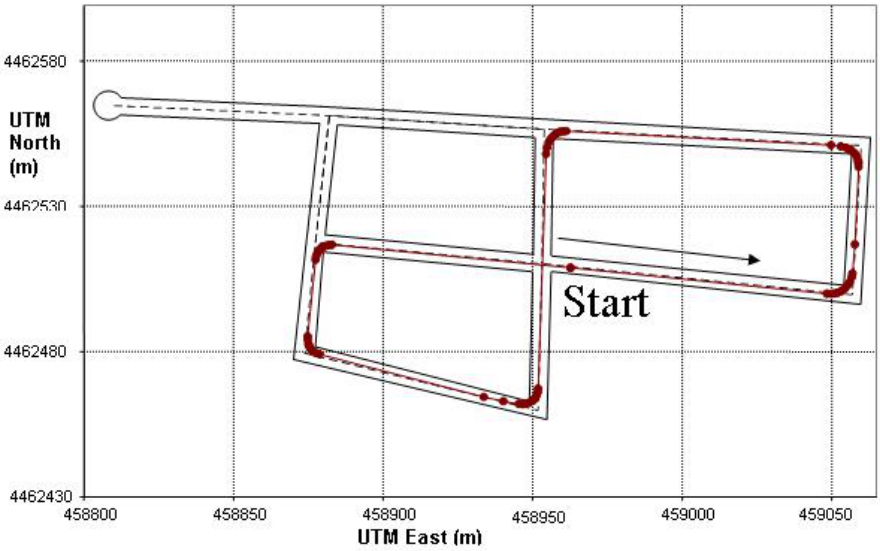


FIGURE 4. GPS reference route used in the data capture and the experiments

Two human drivers followed the route at a constant speed of around 15 km/h with the on-board computer showing them the actual deviation from the reference line in order to minimize errors. This provided a large data set ready for processing in order to obtain the input and output values for the genetic algorithm. In particular, this data set consists of the information about the car's lateral and angular errors, and the corresponding action taken by the driver. Data set is shown in Figure 5(a), where steering actions are represented between -1 (maximum turn to the right) and 1 (maximum turn to the left).

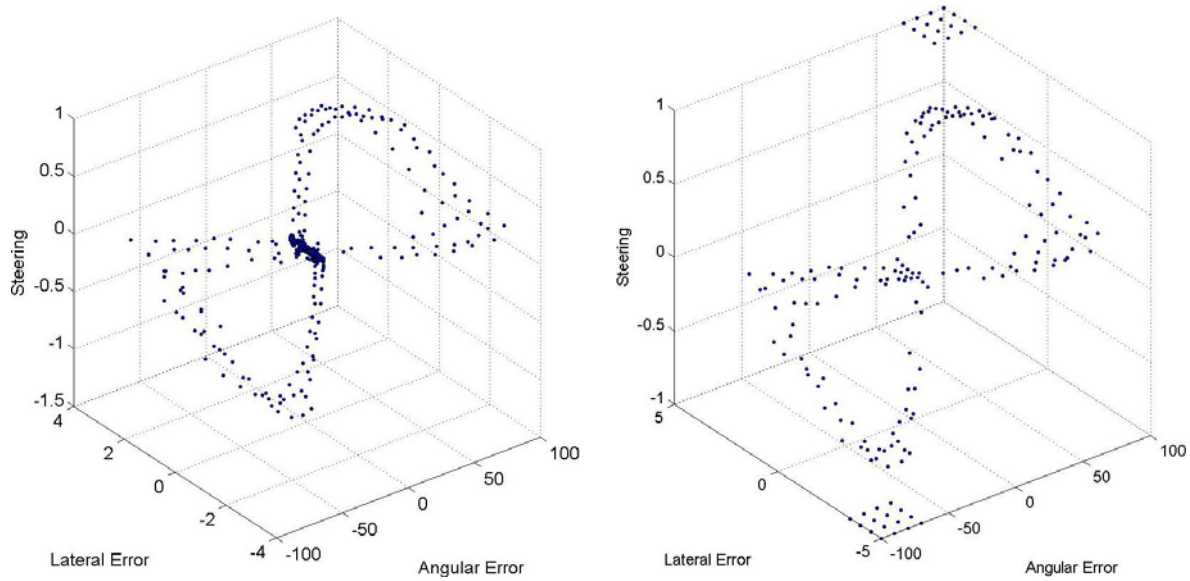
While such a great quantity of points almost determines a complete control surface, it will also slow down the process of evaluating the fuzzy controller in the comparison of its response with that of the human. Also, if one wants to add some empirical value to the data set, its effect will be negligible. One therefore needs to have some way to speed up the process, and to ensure that the design of the controller allows it a certain freedom not to be limited to these points.

To this end, values are first normalized to the interval $[-1, 1]$, assuming the limit for the lateral error is ± 5 meters. Then a $\{-1, 0.9, \ldots, 0, 0.1, \ldots, 0.9, 1\}^2$ grid is defined over the XY plane. Output value for each grid point is defined as the mean output of the closest points in the real input. After that, it has been reduced the number of points without modifying too much their distribution. To ensure coverage of extreme cases, the following common-sense points are added to the point swarm in order of to assure that some well known cases are covered:

- $(x, y, 1)$ $\forall x, y = 0.7, 0.8, 0.9, 1$. That means: *if angular error is greater than 70 degrees and lateral error is greater than 3.5 meters, all of them thought the left then turn the steering at maximum to the right.*

- $(x, y, -1)$ $\forall x, y = -0.7, -0.8, -0.9, -1$. That is the mirror situations to the previous example.

Figure 5(b) shows the final set of points used as training data by the genetic algorithm to ensure similarity with the observed human behavior, and the liberty to generate fuzzy controllers with a smooth surface so that the car will maintain the desired orientation without harsh steering movements.



(a) (b)

FIGURE 5. (a) Data set obtained by monitoring the human's actions; (b) training set obtained after processing the raw information

4. **Fuzzy Controller Representation.** A zero-order TSK fuzzy system with trapezoidal membership functions for input variables and singletons for output variables is used to model the controller. This model allows fast calculations [48].

Controllers have two fuzzy input variables, denominated *Lateral* and *Angular*. Each input variable is fuzzified by means of three or five membership functions, as will be explained below in Subsection 4.1.

There is only one fuzzy output variable, denominated *Steering*. It has twenty one linguistic labels whose membership functions are defined by singletons. We shall distinguish three rule bases of differing complexity, as will be explained in Subsection 4.2.

The representation of the fuzzy controllers with which the genetic algorithm will operate is divided into two parts: the representation of the membership functions (MF) and the representation of the rule base (RB). This allows us to work on and improve these two aspects of the fuzzy controller independently.

4.1. **Membership functions.** Two kind of controllers are implemented depending on the number of membership function used to codify the input variables. Controller can use three or five symmetrical MF.

To represent 3 MF, 4 values are used (x_1, x_2, x_3, x_4). The first two will represent a central membership function for the linguistic variables no deviation (*ND*) and the last two represent two membership functions representing the left/right deviation (*LD* and *RD*), as is shown in Figure 6.

To represent 5 membership functions, 8 values are used: $(x_1, x_2, \ldots, x_8)$. The first two represent the central membership function for no deviation (*ND*), the next four represent

the membership function low left/right deviation (LLD and LRD), and the last two represent two membership functions for the high left/right deviation (HLD and HRD) as is shown in Figure 7.

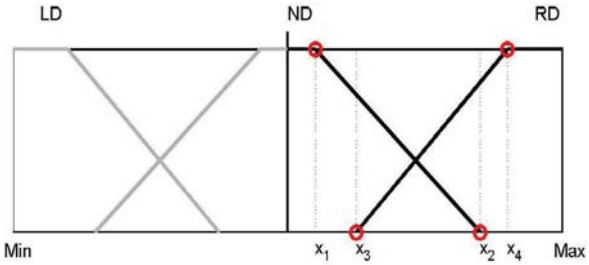


FIGURE 6. The encoding of 3 trapezoidal MF by means of 4 real values

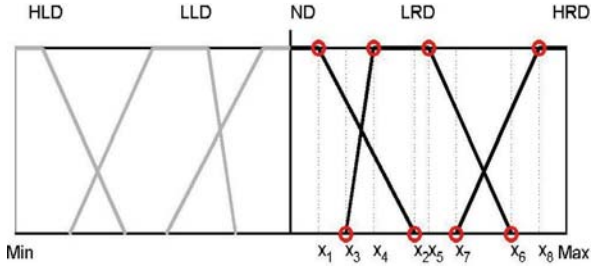


FIGURE 7. The encoding of 5 trapezoidal MF by means of 8 real values

To ensure that the membership functions have coherent values, constraints are imposed. They are designed to guaranteeing that: (i) error towards the left is represented by negative values and towards the right by positive values; (ii) each input value is covered with maximum degree by at most one membership function; and (iii) that each input value is covered (with any degree) by at least one membership function.

Output variable is encoded by means of 21 singletons situated at $\{-1, 0.9, \ldots, 0, 0.1, \ldots, 0.9, 1\}$. Values less than 0 denote steering to the right, and values greater than 0 denote steering to the left. Singletons are named as:

- Rx, $x = 1 \ldots 10$: denote steering movements through the right (negative). So, $R7$ denotes the singleton at -0.7, that is, to move the steering at 70% in clockwise sense.
- Lx, $x = 1 \ldots 10$: denote steering movements through the left (positive). $L5$ denotes the singleton at 0.5, that is the same that moving the steering at 50% in anti clockwise sense.
- NO denotes the singleton situated at 0.0, and represents the action of maintaining the steering in a central position.

The singleton values are unchanged by the genetic process which instead is responsible for assigning the singletons as consequents of a certain rule of the fuzzy rule base.

4.2. **Rule base.** The genetic algorithm changes the consequent assigned to a rule to obtain the best configuration of rules for the controller. There will be a rule for each possible antecedent. Three types of rule bases (RB) were defined, identified as *Marginal*, *Central* and *Total*.

The *Marginal* rule base is composed by six or ten rules generated as:

- If (Lateral is LAT) then (Steering = ST).
- If (Angular is ANG) then (Steering = ST).

The *Central* rule base is composed by nine or twenty-five rules generated as:

- If (Lateral is LAT) and (Angular is ANG) then (Steering = ST).

The *Total* rule base is composed by the junction of the rules of *central* and *marginal* rule bases.

In descriptions of the rule bases, $\{LAT, ANG\} \in [LD, ND, RD]$ (codification of membership functions with three labels) or $[HLD, LLD, ND, LRD, HRD]$ (codification of membership functions with five labels). The output variable ST can be assigned to one of the 21 singletons $[L10, L9, \ldots, NO, R1, \ldots, R9, R10]$ presented previously.

The total number of rules for each configuration membership functions codification/type of rule base is:

- 3 MFs and Marginal RB: 6 rules
- 3 MFs and Central RB: 9 rules
- 3 MFs and Total RB: 15 rules
- 5 MFs and Marginal RB: 10 rules
- 5 MFs and Central RB: 25 rules
- 5 MFs and Total RB: 35 rules

As in the case of the optimization of the membership functions, the rule base also includes specific constraints in order to improve the application of the method to the problem and to maintain the interpretability of the controller. In particular, the constraint is that, for each pair of rules $Rule_1$ and $Rule_2$, if the antecedent of $Rule_1$ is greater than or equal to the antecedent of $Rule_2$, the singleton used for $Rule_1$ must be greater than or equal to that of the antecedent used for $Rule_2$. For example, if there exists a rule of the form *if (lateral is LLD) then (steering is R2)*, rule *if (lateral is HLD) then (steering is St)* must have $St \in \{R10, \ldots, R2\}$.

To represent the rule base in a way that can be used by the genetic algorithm, an integer encoding is used to represent the singleton assigned to the rule. One thus needs a vector of *Number of rules* integers between 1 and 21. So, a value i in the j-th cell of the vector means: to assign the i-th singleton as consequent of the j-th rule.

5. **Genetic Algorithm.** A two-row matrix of 4 or 8 real values (in $[0, 1]$) is used to represent the membership functions of the controllers (one row per input variable), and a vector of 6, 9, 15, 10, 25, or 35 integer elements (in $[1, 21]$) to represent the rule base. The method is implemented in an iterative repetition of two phases each with its own genetic algorithm. One is responsible for improving the membership functions, and the other for improving the rule base.

The general schema is shown in Figure 8. To begin, $Best_{MF}$ and $Best_{RB}$ are randomly initialized. Then, the iterative process consists of generating a random population of $(N_{MF} - 1)$ individual plus the $Best_{MF}$ that has been found so far. A GA is run on the MF population to update the $Best_{MF}$ configuration. The GA will evaluate the MF chromosomes by using the $Best_{RB}$ that has been found so far. The same procedure is then performed on the RB: generate $(N_{RB} - 1)$ random RBs plus the $Best_{RB}$ and run a GA on the population to update the $Best_{RB}$ configuration. The RB chromosomes will be evaluated using the $Best_{MF}$ configuration found so far. This process is repeated It times. The values $N_{MF,RB}$ represent the population sizes for the two genetic algorithms.

The genetic algorithms applied to the RB and MF follow the same basic stationary genetic algorithm schema. They consist in repeating the following sequence of actions for $G_{MF,RB}$ generations in each iteration: select two parents, combine them to generate two offspring, mutate them, and, if they are improvements, substitute them for the worst chromosomes in the population. If any chromosome improves the fitness obtained by Best, it will replace Best.

The following genetic operators [49] are used: The *selection* of parents to be crossed is done by binary tournament. $BLX - \alpha$ [50] is used in the optimization of the MF, and one-point crossover [37] for the optimization of the RB as *crossover operator*. Each component of every chromosome generated is *mutated* by a random change according to a probability defined by the mutation probability, p_m [36, 37]. Each new chromosome generated will *replace* the worst chromosome in the population if its fitness is better.

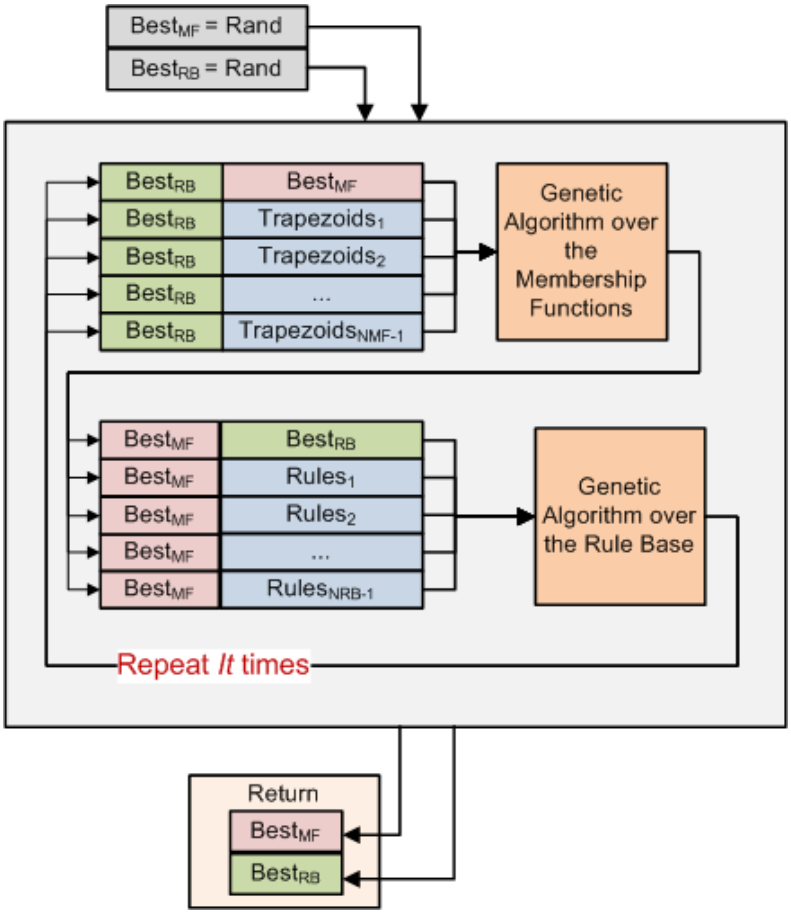


FIGURE 8. Iterative genetic based schema

5.1. **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.
- The requirement of having a smooth surface in its central zone since this is the zone of the surface that is most frequently used.

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).

$$MSE = \frac{1}{2N} \sum_{(i=1\ldots N)} \left(F\left(x^i_{lateral}, x^i_{angular}\right) - y^i\right)^2 \tag{2}$$

where $F(x^i_{lateral}, x^i_{angular})$ is the known output of the controller when executed over the l-th example, with y^i as known output (given by human drivers). The smallest value of this measure means the best performance of the controller.

The smoothness of the surface is evaluated as the greatest difference D between two adjacent points on the grid using a grid (i, j), $i, j = [-1, -0.9, \ldots, 0, \ldots, 0.9, 1]$.

The fitness function used is a weighted aggregate of MSE and D ($F = v \cdot MSE + (1 - v) \cdot D$), with $v = 0.75$ (to give more importance to the similitude with the human behavior than to the smoothness of the surface). The genetic algorithm seeks a minimum of this value.

6. **Results and Experimentation.** The following aspects have to be taken into account in configuring the parameters used in the experiments: there is always a copy of the population's best chromosome; a new population is created before the improvement of each part of the controller (MFs and RBs); and each part of the fuzzy controller is improved separately. Given the characteristics of the method, the parameters must therefore satisfy the following premises.

$N_{MF,RB}$ must be small since there will always be a copy of the best chromosome found so that the selection of that chromosome will be more likely, in order to create offspring that are close to it. $G_{MF,RB}$ must also be small since the two parts of the controller are optimized separately, and it is preferable to have many iterations of the method with few generations than few iterations with many generations. It must be large because of the foregoing comment. The α parameter used in the $BLX-\alpha$ crossover must be small ($\alpha < 0.5$) so that more importance is given to exploiting the good chromosomes in the population than to exploring new chromosome zones (this is achieved by the separation of the genetic algorithms). The mutation probability p_m must be high, in order to change a large number of components in a chromosome.

With these considerations, the parameters chosen to execute the method were: $\alpha = 0.2$, $p_m = 0.25$, $It = 100$, $N_{MF} = N_{RB} = 10$ and $G_{MF} = G_{RB} = 20$.

All the process has been implemented in a graphical environment coded with MATLAB ©. The application works in two phases; at first, the training data is loaded and processed by the application according with the process presented in Section 3. After that, the iterative genetic process described in Section 5 is executed with parameters commented before; for the evaluation of a fuzzy controller, the program writes the controller in the ORBEX (Section 2) format, since it is the one used by the system mounted in the car, and evaluates the controller giving input values corresponding with the training set and comparing the output values with the ones given by humans, as well as measures the control surface to get the higher jump in it (Section 5.1). The iterative process has to evaluate 10.000 controllers (100 iterations · 2 genetic algorithms · (10 initial individuals + 20 generations · 2 offsprings)), and the averaged execution time is about 2 hours each.

6.1. **Obtained controllers.** After processing the input data as described in Section 3, the method described in Section 5 was applied in order to obtain a set of fuzzy controllers that represents the behavior shown by the human drivers. Figures 9 and 10 show the membership functions of the controllers that were obtained. Tables 1 and 2 show the different rule bases.

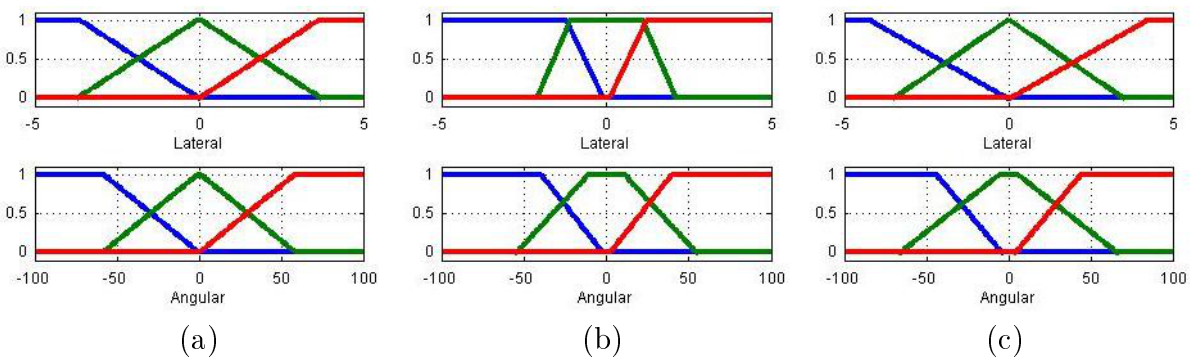


FIGURE 9. Membership functions obtained for controllers with 3 MF codification: lateral error (top) and angular error (bottom); (a) marginal RB, (b) central RB and (c) total RB

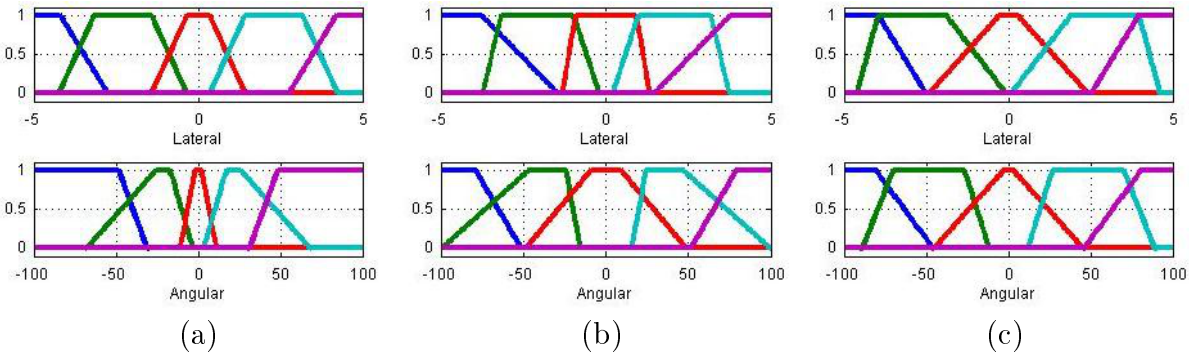


FIGURE 10. Membership functions obtained for controllers with 5 MF codification: lateral error (top) and angular error (bottom); (a) marginal RB, (b) central RB and (c) total RB

TABLE 1. Tabular RB representation of controllers with 3 MF: marginal (top), central (center) and total (bottom)

Lat\Ang	∅	LD	ND	RD
∅		R8	NO	L8
LD	R10			
ND	NO			
RD	L10			

Lat\Ang	∅	LD	ND	RD
∅				
LD		R10	R4	NO
ND		R5	NO	L5
RD		NO	L4	L10

Lat\Ang	∅	LD	ND	RD
∅		R10	NO	L10
LD	R10	R10	R8	R6
ND	NO	R3	NO	L4
RD	L9	L8	L8	L9

These controllers will henceforth be denoted as $3M$ (3 MFs and Marginal RB), $3C$ (3 MFs and Central RB), $3T$ (3 MFs and Total RB), $5M$ (5 MFs and Marginal RB), $5C$ (5 MFs and Central RB) and $5T$ (5 MFs and Total RB). The next step was to test the controllers on the ZOCO test circuit, as will be described in the following subsection.

6.2. **Experimentation.** The fuzzy controllers presented in the previous section were tested on the ZOCO test circuit using the GPS reference route shown in Figure 3. For safety reasons, the pedals of the car were managed manually, the average speeds were maintained between 10 and 15 km/h with averaged maxima of about 20 km/h (32 km/h in the last experiment). The results of the experiments are presented in terms of North-East UTM coordinates, giving the reference route and the route followed by the autonomous car with each controller. The corresponding routes that were followed together with the speeds used are shown in Figure 11. Positions are shown with respect with UTM point (North: 458869.75; East: 4462455.98) to a better understand of the distances. The circled points are given as reference points to better understand the speed plots.

TABLE 2. Tabular RB representation of controllers with 5 MF: marginal (top), central (center) and total (bottom)

Lat\Ang	∅	HLD	LLD	NO	LRD	HRD
∅		R10	R7	R4	R1	L5
HLD	R5					
LLD	R1					
NO	L4					
LRD	L9					
HRD	L10					

Lat\Ang	∅	HLD	LLD	NO	LRD	HRD
∅						
HLD		R10	R8	R3	L1	L2
LLD		R7	R5	R1	L2	L2
NO		R4	R4	NO	L2	L5
LRD		R1	NO	L2	L5	L5
HRD		L1	R2	L3	L6	L8

Lat\Ang	∅	HLD	LLD	NO	LRD	HRD
∅		R10	R8	R1	L3	L5
HLD	R10	R8	R3	L2	L6	L7
LLD	R3	R8	R8	R4	NO	L3
NO	L2	R4	R2	NO	L6	L7
LRD	L7	R5	R4	R1	L2	L10
HRD	L10	R1	L2	L3	L5	L9

Controller $3M$ (Figure 11 top left) shows worse behavior in the first two curves, while it performs a good control in the rest of the circuit, performing very good behavior in the straight segments. The case is very similar for $3C$ (Figure 11 middle right), with the exception of the two first curves. The two will be compared below in terms of their angular and lateral errors. For $3T$ (Figure 11 bottom left) the behavior in the curves seems to be more precise than the previous two; it does not happen in straight segments. $5M$ (Figure 11 top right) shows a behavior that is perhaps very similar to the previous ones. The behavior shown by $5C$ (Figure 11 middle right) seems to be better than of those seen so far: one can appreciate a reduction of the error in the curves and good tracking in the straight segments. Finally, Trajectory shown by $5T$ (Figure 11 bottom right) seems more accurate than the rest.

One observes from Figure 11 that, qualitatively, the controllers all present relatively good behavior, with exception of some punctual track segments for some controllers. Moreover, most of them are able to track a straight segment of road, but they all seem to present the same defect in that, after a curved section, the car deviates from the reference line. While this may in part be a product of the relatively high speed used for the test circuit (see the summaries of the speed data of the experiments in Table 3), it is more probably a result of the non-predictive model of the controllers. Once the car has navigated the curved segment, it has no knowledge of where the next target point is, so that it maintains the amount of steering wheel turn until the next GPS target position is processed after the curve, thus deviating away from the reference line.

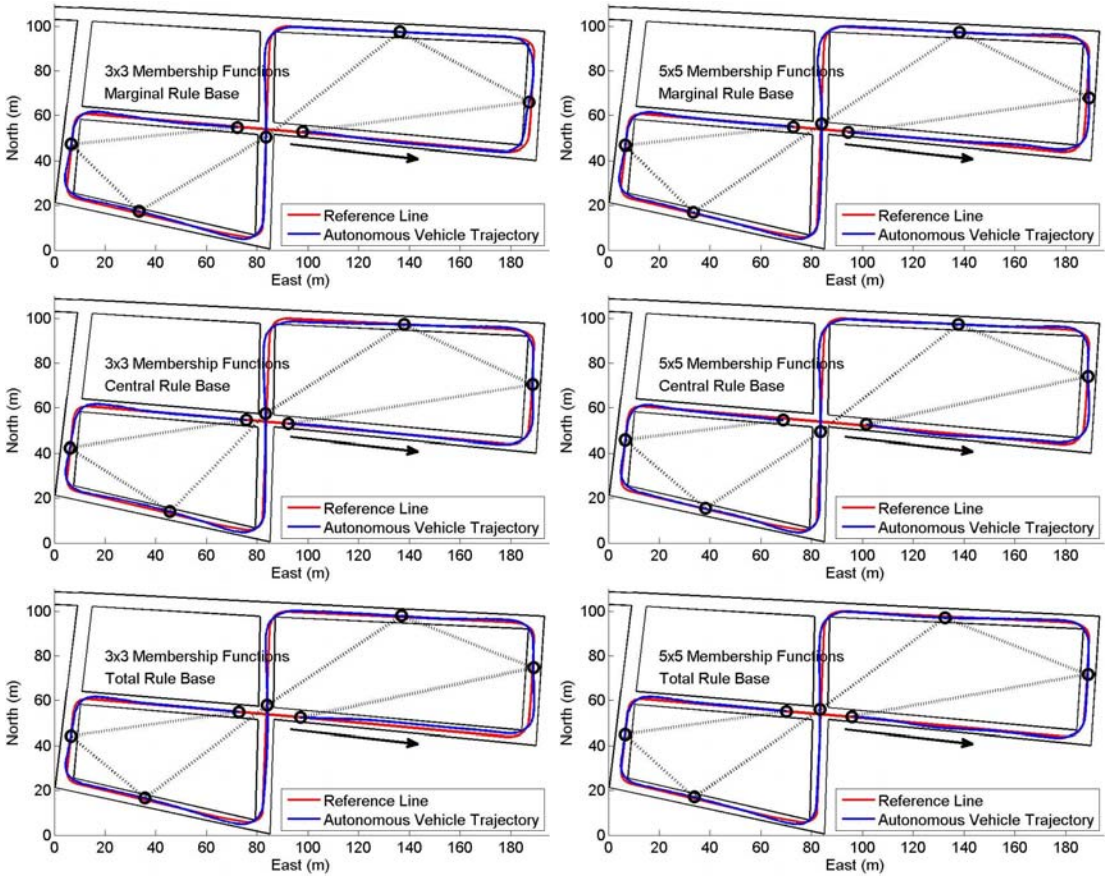


FIGURE 11. Experimental trajectories. Controllers with 3 (left) and 5 (right) membership functions; marginal (top), central (middle) and total (bottom) rule base.

TABLE 3. Experimental distance, time, and speed data for controllers

	3M	3C	3T	5M	5C	5T
Distance (m)	433.9	444.7	437.3	441.0	428.6	434.6
Time (s)	139.8	129.8	138.0	126.2	122.8	126.0
Mean Speed (km/h)	12.8	14.6	13.5	14.9	14.8	14.8
Max Speed (km/h)	22.4	22.1	24.0	22.1	28.6	27.9

Black circles shown in Figure 11 represents mark points to separate the map in six different parts, the first three denoting left turns and the other ones denoting right turns. Averaged lateral and angular errors obtained by each controller in each one of the segments are summarized in Tables 4 and 5 they are shown the averaged absolute lateral and angular errors obtained by each one of the controllers in each one of the defined segments. From the tables, it is interesting to see how controllers with more granularity (five membership functions) get better results in almost all the segments in what respect to the lateral error, since controllers with only three membership functions the better results in angular error. In other hand, controllers with five membership functions get lower deviation what indicates that the committed error is more regular in these cases.

Finally, Figures 12 and 13 present the quantitative results of the experiments, showing the averaged lateral and angular errors committed by the human drivers (*H1* and *H2*)

TABLE 4. Averaged absolute lateral error obtained by each controller in each circuit segment

Controller	Seg_1	Seg_2	Seg_3	Seg_4	Seg_5	Seg_6	$Mean$	Std
3M	0.84	**0.51**	**0.81**	0.69	1.00	0.74	0.76	0.16
3C	**0.50**	0.67	1.44	0.75	0.90	0.68	0.81	0.33
3T	1.33	0.60	0.84	0.85	0.83	0.66	0.89	0.25
5M	0.87	0.66	**0.81**	0.79	0.97	**0.59**	0.78	0.14
5C	0.88	0.72	0.94	0.76	**0.65**	0.73	0.81	0.11
5T	0.59	0.74	0.96	**0.66**	0.69	0.77	**0.72**	0.13

TABLE 5. Averaged absolute angular error obtained by each controller in each circuit segment

Controller	Seg_1	Seg_2	Seg_3	Seg_4	Seg_5	Seg_6	$Mean$	Std
3M	**7.45**	11.89	12.63	14.08	17.17	14.24	**11.89**	3.23
3C	8.40	14.62	15.48	15.06	**15.52**	14.26	13.34	2.73
3T	10.37	**10.71**	**11.41**	14.00	18.37	**13.18**	12.37	2.98
5M	12.25	12.97	12.66	**13.90**	19.19	14.35	13.78	2.56
5C	12.13	15.62	15.33	15.73	16.35	14.72	14.72	1.49
5T	8.92	15.21	15.31	14.11	16.96	14.39	13.50	2.75

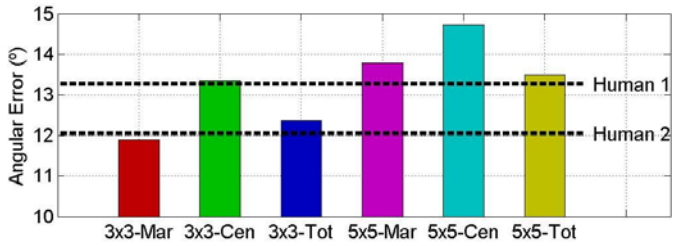


FIGURE 12. Comparative results (averaged angular error) of the human drivers and the fuzzy controllers

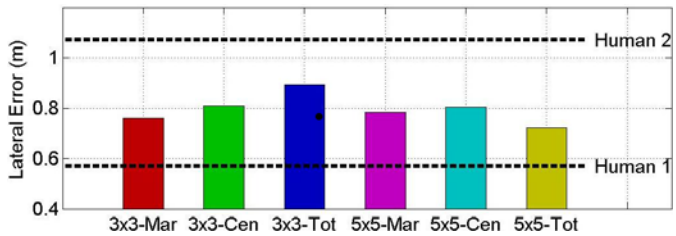


FIGURE 13. Comparative results (averaged lateral error) of the human drivers and the fuzzy controllers

and by the different controllers. No significant differences can be appreciated between the results of the controllers and the human drivers.

Controller $3M$ was able to improve on the averaged angular errors committed by both human drivers. All the generated controllers outperform the lateral error obtained by the first human driver while no one is able to improve the lateral error obtained by the second driver. No significant differences can be appreciated when comparing controllers with 5 MF between them; the one with total rule base ($5T$) obtain better values that the ones with marginal and central rule base ($5M$ and $5C$) in both aspects. $5T$ obtains the

best lateral error value (of the automatic controllers), but no significant differences are found in comparison with, for example, $3M$ or $5M$. Qualitative results show that the best controller of all the generated ones is the one with 3 membership functions and marginal rule base, that is also, the simplest one.

7. **Conclusions and Future Work.** This work has presented a method of obtaining fuzzy controllers capable of automatically steering a mass-produced vehicle which had previously been equipped with the instrumentation and software necessary to achieve automatic driving. To this end, an iterative genetic algorithm was implemented which iteratively adjusts the membership functions and rule bases of the controllers by imposing certain constraints on the controllers in order to ensure that those which result have the capability of automatically steering an unmanned car. The controllers were also required to have a smooth control surface to guarantee safe and comfortable driving for the occupants of the car.

The resulting controllers were tested on a private asphalted test circuit, showing good behavior on straight segments, but less precise on coming out of curved segments. This was caused by the non-predictive model used in the control, which only included analysis of the immediate reference points of the desired route. Future research will consider the possibility of using different fuzzy models in order to resolve this problem, as well as to allow faster tests. Some options to be considered will be to add estimations of future positions as inputs to the fuzzy controller or the use of the actual steering position, to be able to generate additional feedback on the controllers. But the solution of this problem passes through the incorporation of additional input variables that give to the controllers more sophisticated information about the state of the vehicle and may affect to the lateral control: speed, actual position of the steering, curvature of the current road segment, width of the road, etcetera.

All the controllers gave very smooth driving, even when the speeds were relatively high. This represents a good starting point for our subsequent work in the field of Intelligent Transportation Systems. Our next studies will be focused on improving the presented models in order to guarantee precision driving along a reference line, without losing sight of the comfortable and smooth driving currently achieved. We shall also investigate the consequences of using a given number of membership functions, the use of some other car state variable or the rule base. Neither can safety and efficiency be forgotten at this point of the project, and these important features will be considered in our subsequent work.

This work present a starting point to a long future research in mimicking the human behavior at driving. Future works should include a deeper data acquisition in order to capture actions taken by drivers in as many situations as possible, as well as the incorporation of more sources of information with the aim of modeling more complex behavior to face cooperative situations, in which two or more vehicles must cooperate with the aim of carrying out maneuvers without risk (crossroads, roundabouts, overtakes, etcetera).

Acknowledgment. This work was supported by the Plan Nacional, under the project TRÁNSITO (TRA2008-06602-C03-01) and by the Comisión Interministerial de Ciencia y Tecnología under the project GUIADE (Ministerio de Fomento T9/08).

REFERENCES

[1] W. D. Jones, Keeping cars from crashing, *IEEE Spectrum*, vol.38, pp.40-45, 2001.
[2] J. Farrell, M. Barth, R. Galijan and J. Sinko, *GPS/INS Based Lateral and Longitudinal Control Demonstration: Final Report*, Univ. California, Berkeley, CA, USA, 1998.

[3] J. Guldner, W. Sienel, H. S. Tan, J. Ackermann, S. Patwardhan and T. Bunte, Robust automatic steering control for look-down reference systems with front and rear sensors, *IEEE Transactions on Control Systems Technology*, vol.7, pp.2-11, 1999.

[4] E. D. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrandt, M. Maurer, F. Thomanek and J. Schiehlen, The seeing passenger car 'vamors-p', *Intelligent Vehicles Symposium*, pp.68-73, 1994.

[5] B. Ulmer, Vita ii-active collision avoidance in real traffic, *Intelligent Vehicles Symposium*, pp.1-6, 1994.

[6] R. Fenton, Ivhs/ahs: Driving into the future, *IEEE Control Systems Magazine*, vol.14, no.6, pp.13-20, 1994.

[7] C. Urmson, D. Duggins, T. Jochem, D. Pomerleau and C. Thorpe, From automated highways to urban challenges, *IEEE International Conference on Vehicular Electronics and Safety*, pp.6-10, 2008.

[8] M. Williams, The prometheus programme, *IEEE Colloquium on Towards Safer Road Transport – Engineering Solutions*, pp.41-43, 1992.

[9] A. Broggi, M. Bertozzi and A. Fascioli, Argo and the millemiglia in automatico tour, *IEEE Intelligent Systems and Their Applications*, vol.14, no.1, pp.55-64, 1999.

[10] M. Parent, Automated urban vehicles: State of the art and future directions, *The 8th Control, Automation, Robotics and Vision Conference*, vol.1, pp.138-142, 2004.

[11] D. N. Godbole and J. Lygeros, Longitudinal control of the lead car of a platoon, *American Control Conference*, vol.1, pp.398-402, 1994.

[12] R. Jurgen, Smart cars and highways go global, *IEEE Spectrum*, vol.28, no.5, pp.26-36, 1991.

[13] E. Onieva, V. Milanés, C. González, T. de Pedro, J. Pérez and J. Alonso, Throttle and brake pedals automation for populated areas, *Robotica*, vol.28, no.4, pp.509-516, 2010.

[14] E. D. Dickmanns, The development of machine vision for road vehicles in the last decade, *IEEE Intelligent Vehicles Symposium*, vol.1, pp.268-281, 2002.

[15] D. Armstrong, G. Carr, U. Ozguner and R. Behringer, Darpa grand challenge on autonomous vehicles, *The 7th International IEEE Conference on Intelligent Transportation Systems*, pp.847, 2004.

[16] C. D. Crane, The 2005 darpa grand challenge, *International Symposium on Computational Intelligence in Robotics and Automation*, 2007.

[17] D. Wooden, M. Powers, M. Egerstedt, H. Christensen and T. Balch, A modular, hybrid system architecture for autonomous, urban driving, *Journal of Aerospace Computing, Information, and Communication*, vol.4, no.12, pp.1047-1058, 2007.

[18] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian and P. Mahoney, Stanley: The robot that won the darpa grand challenge, *Journal of Field Robotics*, vol.23, no.9, pp.661-692, 2006.

[19] C. Urmson, J. Anhalt, H. Bae, J. Bagnell, C. Baker, R. Bittner, T. Brown, M. Clark, M. Darms, D. Demitrish, J. Dolan, D. Duggins, D. Ferguson, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. Howard, S. Kolski, M. Likhachev, B. Litkouhi, A. Kelly, M. McNaughton, N. Miller, J. Nickolaou, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, V. Sadekar, B. Salesky, Y. Seo, S. Singh, J. Snider, J. Struble, A. Stentz, M. Taylor, W. Whittaker, Z. Wolkowicki, W. Zhang and J. Ziglar, Autonomous driving in urban environments: Boss and the urban challenge, *Journal of Field Robotics*, vol.25, no.8, pp.425-466, 2008.

[20] DARPA, *Darpa Urban Challenge Official Results*, http://www.darpa.mil/ GRANDCHALLENGE/ mediafaq.asp, 2008.

[21] L. Yu, Y. Qi and F. Liu, Research on control strategy and bench test of automobile steer-by-wire system, *IEEE Vehicle Power and Propulsion Conference*, pp.1-6, 2008.

[22] A. El Hajjaji, A. Ciocan and D. Hamad, Learning to drive a real car in 20 minutes, *Journal of Intelligent and Robotic Systems*, vol.41, no.2-3, pp.141-156, 2005.

[23] X. Li, X. P. Zhao and J. Chen, Controller design for electric power steering system using t-s fuzzy model approach, *International Journal of Automation and Computing*, vol.6, no.2, pp.198-203, 2009.

[24] A. Zhu and S. Yang, Neurofuzzy-based approach to mobile robot navigation in unknown environments, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol.37, pp.610-621, 2007.

[25] S.-C. Hsu, S.-F. Liang, K.-W. Fan and C.-T. Lin, A robust in-car digital image stabilization technique, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol.37, pp.234-247, 2007.

[26] J. Wang and B. Cai, A low-cost integrated gps/ins navigation system for the land vehicle, *Proc. of IEEE Intelligent Transportation Systems*, vol.2, pp.1022-1026, 2003.
[27] M. A. Sotelo, S. Alcalde, J. Reviejo, J. E. Naranjo, R. Garcia, T. de Pedro and C. Gonzalez, Vehicle fuzzy driving based on dgps and vision, *Proc. of Joint the 9th IFSA World Congress and the 20th NAFIPS International Conference*, vol.3, pp.1472-1477, 2001.
[28] Q. Wang, M. Oya, T. Kobayashi and K. Okumura, Adaptive steering controller to improve handling stability of combined vehicles, *International Journal of Innovative Computing, Information and Control*, vol.4, no.12, pp.3273-3283, 2008.
[29] L. Zadeh, Fuzzy sets, *Information and Control*, vol.8, no.3, pp.338-353, 1965.
[30] L. Zadeh, Fuzzy logic = computing with words, *IEEE Transactions on Fuzzy Systems*, vol.4, no.2, pp.103-111, 1996.
[31] Y. Jia, Robust control with decoupling performance for steering and traction of 4ws vehicles under velocity-varying motion, *IEEE Transactions on Control Systems Technology*, vol.8, no.3, pp.554-569, 2000.
[32] J. Ackermann, J. Guldner, W. Sienel, R. Steinhauser and V. I. Utkin, Linear and nonlinear controller design for robust automatic steering, *IEEE Transactions on Control Systems Technology*, vol.3, pp.132-143, 1995.
[33] R. Sharp, D. Casanova and P. Symonds, A mathematical model for driver steering control, with design, tuning and performance results, *Vehicle System Dynamics*, vol.33, no.5, pp.289-326, 2000.
[34] F. Herrera, M. Lozano and J. Verdegay, Tuning fuzzy logic controllers by genetic algorithms, *International Journal of Approximate Reasoning*, vol.12, pp.299-315, 1995.
[35] A. Homaifar and E. McCormick, Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms, *IEEE Transactions on Fuzzy Systems*, vol.3, pp.129-139, 1995.
[36] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, New York, 1989.
[37] J. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, 1975.
[38] K. F. Man, K. S. Tang and S. Kwong, Genetic algorithms: Concepts and applications [in engineering design], *IEEE Transactions on Industrial Electronics*, vol.43, pp.519-534, 1996.
[39] H. S. Kim and D. H. Kim, Genetic algorithm-based dynamic channel allocation to minimize the intercell interference in downlink wireless communication systems, *International Journal of Innovative Computing, Information and Control*, vol.6, no.11, pp.5179-5190, 2010.
[40] A. Chipperfield and P. Fleming, Genetic algorithms in control systems engineering, *Control Computation*, vol.23, pp.88-94, 1995.
[41] F.-T. Lin and T.-R. Tsai, A two-stage genetic algorithm for solving the transportation problem with fuzzy demands and fuzzy supplies, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.4775-4785, 2009.
[42] R. García and T. de Pedro, Modeling a fuzzy coprocessor and its programming language, *Mathware and Soft Computing*, vol.5, no.2-3, pp.167-174, 1998.
[43] R. García and T. de Pedro, First application of the orbex coprocessor: Control of unmanned vehicles, *Mathware and Soft Computing*, vol.7, no.2-3, pp.265-273, 2000.
[44] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Transactions on Systems, Man and Cybernetics*, vol.15, no.1, pp.116-132, 1996.
[45] E. Mamdani, Application of fuzzy algorithms for control of simple dynamic plant, *Proc. of the Institution of Electrical Engineers*, vol.121, pp.1585-1588, 1974.
[46] J. E. Naranjo, C. Gonzalez, R. Garcia, T. de Pedro and R. E. Haber, Power-steering control architecture for automatic driving, *IEEE Transactions on Intelligent Transportation Systems*, vol.6, pp.406-415, 2005.
[47] J. I. Serrano, J. Alonso, M. D. del Castillo and J. E. Naranjo, Evolutionary optimization of autonomous vehicle tracks, *IEEE Congress on Evolutionary Computation*, vol.2, pp.1332-1339, 2005.
[48] M. Sugeno, On stability of fuzzy systems expressed by fuzzy rules expressed with singletons consequents, *IEEE Trans. on Fuzzy Systems*, vol.7, no.2, pp.201-224, 1999.
[49] F. Herrera, M. Lozano and J. Verdegay, Tackling real-coded genetic algorithms: Operators and tools for the behaviour analysis, *Artificial Intelligence Review*, vol.12, pp.265-319, 1998.
[50] L. Eshelman and J. Schaffer, *Real Coded Genetic Algorithms and Interval Schemata. Foundation of Genetic Algorithms 2*, Morgan Kaufmann Publishers, 1993.