Generation of an Autonomous Car Fuzzy Controller via Genetic Algorithms

Enrique Onieva

Instituto de Automtica Industrial. Consejo Superior de Investigaciones Cientficas.
Ctra. Campo Real Km.0,200. 28500 La Poveda, Arganda del Rey. Madrid, Espaa.
onieva@iai.csic.es

There is a broad range of diverse technologies that hold answers many of the transportation problems under the topic of Intelligent Transportation Systems (ITSs); the automatic control of the steering wheel is one of the most important challenges facing researchers in this area, but it is not the only. This paper presents a method for the automatic adjust of fuzzy controllers to manage automatically a mass-produced vehicle to reproduce the attitudes of a human driver in different situations. To do that, information about the car state while the human driver is handling the car is taken and used to obtain, via genetic algorithms an appropriated fuzzy controller. The fuzzy controllers must carry out two main objectives: to reply the human driver's behavior and to provide smooth actions to guarantee a soft driving.

1 Introduction

Intelligent Transportation Systems (ITS) apply information and communication techniques in order to achieve safe and efficient driving. In automotive industry sensors are mainly used to give information to the driver and, in some cases, they are connected to a computer that performs some control actions, attempting to minimize injuries and to prevent collisions [1].

The actual implementation of full automatic-steering control is one of the disciplines in the intelligent-vehicles field that is receiving less attention now than it did some years ago. The reason for this is that perhaps it has a long way to go before it comes on the market [5], and vehicle manufacturers focus on more mature systems, especially for speed automation, some of which are already available on the market. But it does not receive so few attentions in the area of the research, as is showed in DARPA Grand Challenge, which is a prize competition for driverless cars, sponsored by the Defense Advanced Research Projects Agency, the central research organization of the United States Department of Defense. DARPA has sponsored three competitions, all in the area of autonomous vehicles.

The work carried out at the AUTOPIA Program of the Industrial Automatic Institute, a part of the Spanish Council for Scientific Research (CSIC), is focused on the area of autonomous vehicles. More concretely, the aim of the present work is the automatic control of the steering wheel (also known as lateral control) of a mass-produced vehicle, although another area of the research carried out by the AUTOPIA Program is longitudinal control, the control of the vehicle's speed and its adaptation to road features, using the throttle and the brake pedal as needed [3],[4].

This work expects to model the behavior of a human driver by means of fuzzy controller adjusted with a genetic algorithm. In the section 2 it is presented the AUTOPIA program, with the purpose of to give an idea of the instrumentation and infrastructure used to carry out this work; in section 3 it is presented an software application designed to generate automatically fuzzy controllers, taken as input, a set of input-output tuples and applying an iterative method, consisting in a separately optimization of the membership functions and the rule base of the controllers by means of the use of genetic algorithms. And finally, in section 4, the method is applied to model the manage of the steering wheel of a car, perform the automatic driving along a test track and will be proved in a private and experimental area to verify that they show a similar behavior to the shown by human driver.

2 The AUTOPIA Program

AUTOPIA tries to transfer the techniques developed for the independent robots controls to the vehicles controls, modifying as less as possible the surroundings in which these evolve.

The Car and the COnduction ZOne: The vehicle used in this work is a Citroën C3 Pluriel. ZOCO (acronym of COnduction ZOne) is a track for automatic vehicles testing. This track looks like some blocks of city, with some irregularities. A station of global positioning differential based on geographic information via satellite, DGPS, was also installed in ZOCO. It can be used by embarked movable systems in the cars, to obtain its position with a precision near the centimeter. A representation of ZOCO can be seen in section 4.

Control Architecture: The general architecture uses a schema where a set of independent autonomous vehicles are linked among them and with a central monitoring station, sharing the necessary information to cooperate and perform human-like maneuvers. The individual architecture installed in each vehicle is a three layer classic hierarchy: a sensorial layer, a control and an actuation layer. The control layer is divided into three sequential stages: planner, co-pilot and pilot. In this work the interest is centered in the pilot, because is where the fuzzy controllers are located and executed to perform basic human driver maneuvers. It receives a set of input parameters and a low level maneuver selection (to follow the precedent car, take a curve segment, Stop...) from the co-pit and it will be able to generate an output signal that can be applied to the vehicle actuators.

ORBEX: The ORBEX [6],[7] (acronym of Fuzzy Experimental Computer) fuzzy development environment is used for the building of the fuzzy controllers used by the pilot. ORBEX works with TSK fuzzy controllers [9], with consequents of type Singleton, which allows us to take decisions of control in a very short period of time and with very good precision, qualities more than desirable for real time systems, where inference time is a vital aspect, like the automatic driving of a car [8],[10].

3 Iterative Method to Adjust the Fuzzy Controllers

The rule base and the membership functions, which define a fuzzy controller will be tuning in order to adapt to the attitudes showed by a human driver while is driving a car. The method is divided in three phases; at first, there is applied a phase of information capture and processing, in order to define the desired surface that will have the controller. After it, iteratively, the membership functions and the rule base is adjusted by means of two different genetic algorithms. The next four subsections describe the information treatment phase, the two genetic algorithms and the global process to follow to adjust the fuzzy controller.

3.1 Information Capture and Processing

The human driver will manage the car while the on-boarder computer will be collecting data about him. This data are referred to the car state in a determined instant of time, such as, the position, speed, acceleration, steering angle, pressure applied to the pedals...

Once done it, a big data set is obtained, which is subjected to a process of normalization in the [-1,1] interval and, after it, a grid whose dimensions are selected by the user, will be applied to the data set and, in each point the output will be the average of the nearest points in the original data set. Finally, there is the possibility of add some points not included in the original data set, because of they can represent extreme zones of the input space where it is not possible take information. It allows the user to give information over its own knowledge, like *if the car is deviated the maximum to the left, turn the steering the maximum to the right.*

3.2 Membership Functions Adjustment

In the first stage of the method, the membership functions (MF's) that define the variables of the fuzzy controller will be tuning. A membership function is defined by four values (a,b,c,d), that allows to the program work with triangular (if b=c) or trapezoidal membership functions. To generate suitable fuzzy controller, the membership functions must carry out semantic restrictions, these restrictions are: the membership functions must be ordered; there have to be a membership function that covers the extremes of the range of the variable with degree 1; a value have to be covered with degree 1 by, at maximum, a membership function; every value of the range of the variable must be covered with

1 or 2 membership functions. Also there are optional restrictions, such as: use only triangular membership functions or use symmetric membership functions with reward to the centre of the range.

Thus, to represent a set of fuzzy trapezoidal membership function in a way that can be processed by a genetic algorithm, it will be used 4N values, where N is the total number of membership functions for the input variables. The output variables will not be modified. The stationary genetic algorithm used to adjust the membership functions has the following parameters:

- Initialization: each value v_i in a chromosome will be initialized to a uniformly distributed value in the interval $[m_i \pm \xi]$ under a probability ρ, in other case, $v_i = m_i$. Where m is the best chromosome found along the execution.
- Selection: Binary tournament.
- Crossover Operator: BLX-α [13].
- Mutation Operator: Random change under a mutation rate, p_m [11], [12].
- Replacing Method: each chromosome generated will replace the worst chromosome in the population if it obtains better fitness than it.
- Fitness Function: To measure the quality of a certain controller two factors must be consider: the adjustment to the actions taken by the human driver and to have a smooth surface for not to have abrupt responses. To evaluate the first one, it is used the mean squared error between the controller's output and the actions taken by the human driver (MSE). To evaluate the second one, the generated surface is evaluated in a grid (i,j), i,j = [-1,-0.9,...0,...0.9,1] and it is taken D like the biggest difference between two adjacent points in the grid. The fitness function used will be the weighted aggregation of MSE and D (F = vMSE + (1-v)D), where v is selected by the user in order to give more importance to the desired value.

3.3 Rule Base Adjustment

The second stage of the method consists in the adjustment of the rule base (RB). To do that, the consequents of a set of defined rule bases are modified. The process starts with the selection by the user of the number different possible Sugeno's singleton, which can be selected as consequent of a rule. Once done this, a set of possible singleton distributed uniformly in the range of action is created and the genetic algorithm will be in charge of associate each rule with one of the possible consequent. A set of three different rule bases are available; the genetic algorithm will work with the consequents of the rules, changing the consequent assigned to a rule to obtain the best configuration of rules for the controller. Independently the number of membership functions of the input variables, there is 3 different rule bases that can named $Marginal$, $Central$ and $Total$ respectively. The Marginal rule base works only with rules with a simple antecedent; the Central rule bases represent the rules with AND-composed antecedents; the Total rule bases consist in a union of the Central and Marginal rule bases.

Thus, to represent the rule base in a way that can be processed by a genetic algorithm, it will be used N integer values, where N is the total number of rules of the selected rule base, and each value in the position i of the vector represents the index of the consequent associated to the rule i.

The stationary genetic algorithm used to adjust the rule base uses the same initialization, selection, replacing and mutation method, but in this case, the values are integers and not real values like in the adjust of the membership functions, and the same fitness function, the different aspect is the crossover operator. In this case, a simple crossover [11] is used.

3.4 Global Schema

The implementation of the method it is made in two different phases, that are repeated a certain number of times and represent two different genetic algorithms, which have been commented before; one of them is in charge of improve the membership functions and the other is in charge of improve the rule base. To evaluate a rule base will be used the best membership functions set and to evaluate a membership functions set, the best rule base will be used, thus, there is always a copy of the best controller found in the actual population, the process will return the best combination MF's - RB found.

Previous to the application of the method described here, the stage of information capture and processing has been realized. The genetic algorithm, both over the rule base and the membership functions consists in the application of the both genetic algorithms described previously.

4 First Experiment: Lateral Control Modeling

The aim of the present experiment is the automatic control of the steering wheel (also known as lateral control) of a mass-produced vehicle. The automatic control of the speed and the steering of a vehicle are two of the main steps in order to develop autonomous intelligent vehicles. In this experiment, a development of steering control for automated cars based on fuzzy logic and its related field tests are presented. In first time, a human driver has followed the reference route, while the on-board computer of the car has registered his actions.

To model the fuzzy controller for the manage of the steering wheel, two input variables will be used, which are obtained from the match between GPS positioning information and the reference route defined as GPS digital cartography. The two input variables that have been defined are: the lateral error and the angular error. The lateral error represents the distance of the current car position to the theoretical car position if it was on the desired trajectory, the reference line, its values can to take any value ($-\infty$, ∞). The angular error is the angle shaped by the reference line and the car velocity vector, its values are restricted to the interval (-180, 180). The graphical representation of these values is:

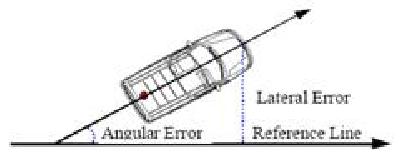

Fig. 1. Input variables graphical representation.

The output will be the reference position sent to a low-level control layer that manages the motor attached to the steering bar to take it to the reference position.

To proceed to the normalization of the values of the points sets in the [-1,1] interval, it has been assumed that the limit for the lateral error is ±5 meters and ±180 for the angular error. After it, a 21x21 grid is defined in the X-Y plane, and, in each point in the grid is taken the mean output of the closest points in the real input. As it can be seen at the graphics, there is zones in the grid without an associated value, so the actions taken by the human driver in extremes situations, like angular error of ±180, ... is not reflected in the graphic. To assure that some extremes cases are covered, the following points deduced of the common sense are added to the point swarms:

- (1,1,1): if lateral error is maximum to the right and angular error is maximum to the right, the steering turn must be maximum to the left
- (-1,-1,-1): if lateral error is maximum to the left and angular error is maximum to the left, the steering turn must be maximum to the right

Once done this, the data set obtained, and the processing done can be seen in the next figure:

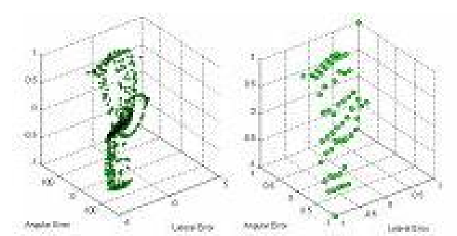

Fig. 2. Initial data set (left) and data set obtained after the data processing.

The process has been executed with the following parameter configuration. In this case, the commons parameters to the membership functions and rule base optimization are equal, in both phases, except ξ. The parameters used are: 100 iterations of the method, 15 individuals and 20 generations for each genetic optimization, with mutation probability of 0.1; α parameter (BLX crossover) is 0.25; $\xi = 0.25$ (for MF optimization) and $\xi = 3$ (for RB optimization).

Modification range of the best individual to initialize (ρ) is 0.5. There will be used 21 possible consequents and 5 membership functions for each input. And, finally, v = 0.75 (so it is given more important the similitude with the human behavior than the smooth surface).

Once applied the process, 3 fuzzy controllers have been obtained, they will be denoted $5M$ (5 membership functions and marginal rule base), $5C$ (5 membership functions and central rule base) and $5T$ (5 membership functions and total rule base). The control surfaces are showed in the next figure:

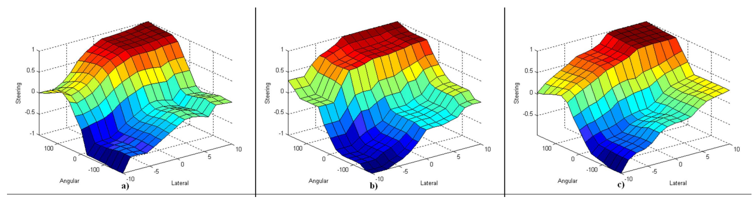

Fig. 3. Control surfaces of the obtained controllers: $5M$ (left), $5C$ (centre), $5T$ (right).

Once realized the process, the controller presented have been tested in the Conduction Zone using a GPS reference route. Notice that all the routes have been done in a velocity between 15km/h in curved lines and 25km/h in straight lines.

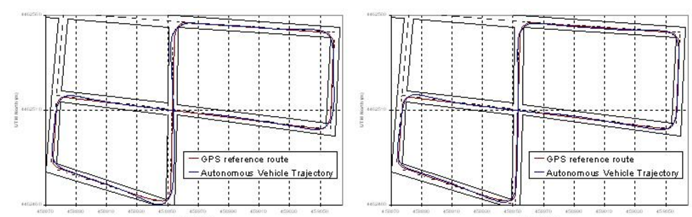

Fig. 4. Reference and trajectory followed by the controllers: $5M$(left) and $5C$ (rigth).

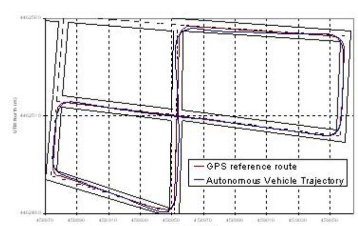

Fig. 5. Reference and trajectory followed by the controller: $5T$.

In the figures it can be observed a relative good behavior of the controllers; most of them are able to track a straight segment of road but all presents the same failure, that is, after a curve straight, the car is derived of the reference line. It can be produced by the relative high speed used to circulate around the test zone; or probably, because of the non predictive model of the controllers, it produces that once leaved the curve straight, the car doesn't know where is the next target point, for what the car keeps the steering wheel turn until the next GPS target position is processed post curve movement out of the reference line. Next figure shows the normalization of the average of the lateral and angular error for each controller.

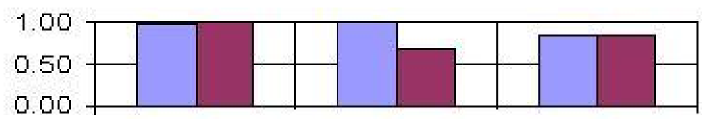

Fig. 6. Angular/Lateral error comparison; 5M(left), 5C(center), 5T(right).

The figure shows that the worse behavior with respect to the angular error has been obtained by the controller $5C$ and respect to the lateral error by the $5M$; the angular error of the $5M$ is very near to the obtained to the $5C$, so, it can be concluded that $5M$ presents the worse behavior. Furthermore, the controller $5C$ presents the minimum average of lateral error. The minimum value of angular error can be seen in the controller $5T$, which also has a good value of lateral error average.

5 Conclusions

In this work has been presented a method to obtain fuzzy controllers able to manage automatically a mass produced vehicle, which previously has been equipped with the instrumentation and software necessary to carry out automatic driving. To reach this, there has been implemented a iterative genetic algorithm, which is able to adjust iteratively the membership functions and rule base of the controllers, by means of apply some restrictions to the controllers in order to guarantee that the obtained ones has capability of carry out the automatic driving of a unmanned car. Also is necessary for the controllers to have a smooth surface in order to guarantee a safe and comfortable driving for the occupiers of the car. The controllers have been tested in a private asphalt circuit, showing a good behavior in straight lines for most of them, but also a bad behavior in curve lines; this is caused by the non predictive model used in the control, without analysis of the not immediate reference points of the desired route. The controllers has shown a very smooth driving, even, when they are circulating at relative high speeds, that a good starting point for our next researches in the field of Intelligent Transportation Systems.

References

1. Willie D. Jones. "Keeping Cars from, Crashing". IEEE SPECTRUM, September 2001, pp 40-45
2. J.E. Naranjo, C. Gonzlez, J. Reviejo, R. Garca, T. de Pedro, and M.A. Sotelo, "Using Fuzzy Logic in Automated Vehicle Control"IEEE Intelligent Systems. Jan/Feb 2007.
3. J.E. Naranjo, C. Gonzlez, J. Reviejo, R. Garca, and T. de Pedro, "A throttle & brake fuzzy controller: Towards the automatic car", in Lecture Notes on Computer Science, vol. 2809. Berlin, Germany: Springer-Verlag, Jul. 2003, pp. 291-301.
4. J. E. Naranjo, C. Gonzlez, J. Reviejo, R. Garca, and T. de Pedro, "Adaptive fuzzy control for inter-vehicle gap keeping," Special Issue on Adaptive Cruise Control, IEEE Trans. Intell. Transp. Syst., vol. 4, no. 3, pp. 132-142, Sep. 2003.
5. E. Dickmanns, "The development of machine vision for road vehicles in the last decade," in Proc. IEEE Intelligent Vehicles Symposium., Versailles, France, 2002, pp. 268-281.
6. R. Garca Rosa, T. De Pedro. Modeling a fuzzy coprocessor and its programming language. Mathware and Soft Computing, vol V, n. 2-3, pp 167-174, 1998.
7. R. Garcia, T. De Pedro. "First Application of the ORBEX Coprocessor: Control of Unmanned Vehicles". 1999 EUSFLAT-ESTYLF Joint Conference. Mathware and Soft Computing, n. 7, vo12-3, 2000, pp. 265-273.
8. Huang, S. and W. Ren (1999). Use of neural fuzzy networks with mixed genetic/gradient algorithm in automated vehicle control. IEEE Transactions on Industrial Electronics, 46, 1090-1102
9. Fuzzy Identification of Systems and its Application to Modelling and Control. IEEE Trans. on Syst. Man and Cybernetics, 15:166-132
10. M. Sugeno, "On stability of fuzzy systems expressed by fuzzy rules with singleton consequents," IEEE Transactions Fuzzy Systems., vol. 7, pp. 201-224, Apr. 1999.
11. Holland, J.H. (1975).Adaptation in Natural and Artificial Systems. The University of Michigan Press.
12. Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, New York.
13. Eshelman L.J. & Schaffer J.D. (1993). Real Coded Genetic Algorithms and Interval Schemata. Foundation of Genetic Algorithms 2,L. Darrell Whitley (Ed.) (Morgan Kaufmann Publishers, San Mateo), 187-202.