Adaptive Fuzzy Control for Inter-Vehicle Gap Keeping

José E. Naranjo, Carlos González, *Member, IEEE*, Jesús Reviejo, Ricardo García, and Teresa de Pedro

***Abstract*—There is a broad range of diverse technologies under the generic topic of intelligent transportation systems (ITS) that holds the answer to many of the transportation problems. In this paper, one approach to ITS is presented. One of the most important research topics in this field is adaptive cruise control (ACC). The main features of this kind of controller are the adaptation of the speed of the car to a predefined one and the keeping of a safe gap between the controlled car and the preceding vehicle on the road. We present an ACC controller based on fuzzy logic, which assists the speed and distance vehicle control, offering driving strategies and actuation over the throttle of a car. The driving information is supplied by the car tachometer and a RTK differential GPS, and the actuation over the car is made through an electronic interface that simulates the electrical signal of the accelerator pedal directly to the onboard computer. This control is embedded in an automatic driving system installed in two testbed mass-produced cars instrumented for testing the work of these controllers in a real environment. The results obtained in these experiments show a very good performance of the gap controller, which is adaptable to all the speeds and safe gap selections.**

***Index Terms*—Autonomous vehicles, longitudinal control, intelligent vehicles, field experiments, fuzzy logic, adaptive cruise control (ACC), safe gap, Stop&Go, platoon driving, wireless communications, intelligent transportation systems (ITS).**

I. Introduction

INTELLIGENT transportation systems (ITS) apply robotic techniques to achieve safe and efficient driving. In the 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 guiding actions, attempting to minimize injuries and to prevent collisions [1]. One of the applications of ITS is the providing of assistance to the control of some of the vehicle elements, like the throttle pedal and consequently, the speed-control assistance. A cruise control (CC) system is a common application of these techniques. It consists of maintaining the vehicle speed at a user (driver) pre-set speed. These kind of systems are already mass installed in top of the line end vehicles. A second step in the development of the speed assistances is adaptive cruise control (ACC) [2].

ACC is similar to conventional cruise control in that it keeps the vehicle pre-set speed. However, unlike conventional cruise control, this new system can automatically adjust speed in order to maintain a proper headway distance (gap) between vehicles in the same lane [3]. This is achieved through a radar headway sensor, digital signal processor and a speed controller. If the lead vehicle slows down, or if another object is detected, the system sends a signal to the engine or braking system to decelerate. Then, when the road is clear, the system will re-accelerate the vehicle back to the set speed [4]. Previous research has shown that ACC can improve traffic conditions significantly [5], [6]. ACC systems have been in market since Mitsubishi launched the "Preview Distance Control" for its Diamante model car in 1995. Toyota, Nissan, Jaguar, Mercedes, Lexus, BMW [7], and some car component industries have introduced an ACC system, although only as an optional device for luxury vehicles. One limitation of these commercial systems is that they control the speed of the car only at speeds above 30–40 km/h and they fail at lower speeds in heavy traffic, where if the preceding car stops, the equipped car must stop too at a safe headway. Stop&Go systems automate the throttle control in this kind of situation.

There are a lot of techniques to perform ACC. Conventional methods based on analytical control generate good results but with high design and computational costs since the application object, a car, is a nonlinear element and its full mathematical representation is impossible [8]–[11]. Other ways to reach a human-like speed control is the application of artificial intelligence techniques [12], [13]. One of these techniques is the fuzzy control that allows an approximate human reasoning and an intuitive control structure [14].

Fuzzy logic is a powerful albeit somehow controversial technique. It permits control without extensive knowledge of the equations of the process and it represents in a very effective way the human reasoning methods [15]. It is L. A. Zadeh, in Berkeley who originated fuzzy logic [16], with the first real application of a fuzzy control proposed by E. H. Mamdani [17]; however, it is in Japan where more fuzzy control developments have been made. This technique has come of age long ago and it has been used it to control cars [18], aircrafts [19], and railways [20].

Stop to go is one of the most tedious and tiring operations human drivers have to bear, and it is also one of the most common because there are traffic jams in most of the cities of the world. Accidents of rear-end collisions are common in this kind of situations. Stop&Go systems are being developed in order to automate this maneuver. The combination of ACC and Stop&Go increases driving comfort, smooths traffic speed and

Manuscript received October 3, 2002; revised September 22, 2003. This work was supported in part by ORBEX, Experimental Fuzzy Computer Project, TIC 961393-C06 of the CICYT, ZOCO, basic infrastructure creation, IN 96-0118 of the CICYT, COVAN, Automatic Vehicle Control in Natural Environment, CAM 06T/042/9 of the Comunidad Autónoma de Madrid, GLOBO, GPS and Fuzzy Logic Vehicle Control, TAP 98-0781 of the CICYT. This work has been funded by the Projects: COVAN CAM 06T/042/96; ORBEX CICYT TIC 96/1393.c06-03; ZOCO CICYT IN 96-0118; GLOBO CICYT TAP 98-0813. The Guest Editor for this paper was P. A. Ioannou.

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

Digital Object Identifier 10.1109/TITS.2003.821294

allows queues to discharge faster from bottlenecks [3], [21], [22].

This paper addresses the integration of mechatronics and fuzzy control techniques [23] in order to get robotic aids to drive cars, and presents the experiments done to test this integration. The present application is for a car computer throttle control powered by a fuzzy logic controller, with the capability of performing an Adaptive Cruise Control in an unmanned/manual driving. The speed tracking experiments have been made in a private test circuit and the testbed cars are two automated mass-produced vehicles.

II. Autopia Program

The work described in this paper was done at the Instituto de Automática Industrial (IAI), a part of the Spanish Council for Scientific Research (CSIC). Building on the extensive experience of IAI in the development of autonomous robots and fuzzy control, we set up the AUTOPIA Program, a set of national research projects similar to other countries' programs [24]–[27]. The goal of AUTOPIA is to transfer autonomous mobile robot control technologies to computer-aided vehicle driving. The aim of this program is the developing of a testbed infrastructure for experimentation in control systems, strategies and sensors applied to vehicle driving, open to groups interested in our research field [28]. Our research objective is automatic driving [29]. This objective may be called "utopia" for the moment, since full automatic driving will not be a reality for at least twenty years, but this is a great starting point to explore the future. Derived from this we define the second aim of AUTOPIA: the development of driving aids. The full automation is not yet possible, but the modular components of these automatic driving may be applied now to the automotive industry. A lot of applications can be developed from the experience gained using the systems of AUTOPIA. As an example of this philosophy, we might mention our CEPAS Project, a semi-automated high precision vineyard planting system, already implemented in Spain, which will be discussed in soon to be published articles [30].

In an overview of AUTOPIA, two testbed cars (Fig. 1) have been automated in order to perform lateral and longitudinal control from a computer, where a fuzzy logic based control system is embedded.

The main sensor is a high precision GPS and the steering and the accelerator pedals of the cars have been automated [31]. Braking is also automated, but it is not used in this work.

A. Application of a Fuzzy Coprocessor to the Automatic Vehicle Driving

Fuzzy control is abstract and generalist, or, in other words nonspecialized. It deals only with input and output variables and it does not require any knowledge of the mathematical model of the processes involved. The relationship between input and output variables is expressed in sentences that mimic closely human thinking. Furthermore, the variables, especially the output variables, are normalized and their values can be applied to low level controllers easily. These low level controllers act directly upon the physical actuators. In our case of the control of cars, if a car is substituted no changes are to be made to the model for it to work on it. Only low level controllers are to be modified when the dynamic environment changes.

Fig. 1. Citroën Berlingo Commercial Prototypes.

In order to emulate the human driving, the first step is to represent the human approximate reasoning [32]–[35]. In [34], approximate reasoning or fuzzy reasoning is defined as the process or processes by which a possibly imprecise conclusion is deduced from a collection of imprecise premises. One of the most powerful tools available in intelligent control is the fuzzy controller. Starting from previous knowledge of the driver behavior it is possible to extract the fuzzy control inputs, outputs, membership functions and fuzzy rules as a first approximation [35].

In the second step, we determined the fuzzy values, according to the main car parameters. Simulators were also used to do that [36].

In the third step, two mass-produced cars were instrumented in order to permit an automatic fuzzy control on the steering and the accelerator-brake set.

The software that supports the fuzzy controllers is ORBEX (experimental fuzzy coprocessor) [37] which allows writing fuzzy rules as sentences in almost natural language. This way, we have automated the fuzzy inference processes, and can represent in real time environments with only the fuzzy variables as input parameters, without any numerical complexity. The driving strategies to cope with different traffic situations are described in an almost natural language, the input language to ORBEX [38]. ORBEX allows the user to define variables and their fuzzy-granules and to combine several variables in rules to get out crisp signals to controllers. The rules are IF ... THEN fuzzy sentences where the original input granules, or others derived from them, are fuzzy AND/OR combined in the IF part of the rule to produce a fuzzy output, for instance:

IF crossing VERY near OR lane occupied THEN braking strong.

Note, that near is an original fuzzy-granule, while VERY near is a derived one. To simplify the defuzzification process the input variables have triangular or trapezoidal shapes and the outputs are singletons, as Sugeno suggested [15].

Two independent controllers have been designed in order to perform automatic driving: lateral control and longitudinal control or, in other words, steering control and speed control respectively [39].

We will describe the longitudinal control and the actual results when it is installed in an automated car.

B. Longitudinal Control

Conceptually, in human driving there are three control levels for managing the accelerator of a traditional mass-produced car. A mechanical layer (the accelerator pedal and all the associated hydraulic mechanisms), an actuation layer (the human foot that steps on the pedal) and a knowledge reasoning layer (the human brain).

The testbed cars used in our study are Citroën Berlingo electrical vans, equipped with automatic gearbox in order to simplify the gear change. This is a new generation car with a difference between it and the older models: the accelerator pedal constitutive elements are not mechanical but electronic. A potentiometer is installed in this pedal and, when the driver steps on it, an electrical signal is sent to the car's on-board computer that moves the wheels at a proportional speed.

In a similar way, when we automate this element of the car we also divide the control architecture into three layers:

A mechanical layer, identical to the human one, because it is forbidden to change the car components. The difference is that the input is not a mechanical pressure over the pedal, but an electrical signal simulating the signal output of the pedal potentiometer.

An electronic layer that, similar to the driver's foot, generates a signal proportional to a desired speed.

A control layer made up of a fuzzy control system and a knowledge base.

1) Mechanical Layer: The accelerator pedal of the Citroën Berlingo van has a signal potentiometer associated with it. The potentiometer output signal is sent to the car's internal on-board computer through an electronic interface.

2) Electronic Layer: Two components realize the task of the human foot. The first component is an industrial computer in which the control software is executed. The motherboard of this computer is an ICP Robo-505, with a Pentium 166 MHz processor, embedded in an industrial PC chassis. The second component is an interface between the PC and the car's on-board computer. The input to this interface is the control system speed command, transmitted through a DB9 RS-232 port, and the output is the simulated accelerator potentiometer signal. Therefore we have two accelerator signals: the manual signal and the automatic signal. A switch is used for selecting one of them when necessary. A second functionality is added to the electronic layer: the reading of the vehicle speed from the car's control panel speedometer. This information is generated by the car's internal on-board computer, acquired by the electronic interface and sent to the computer through the RS-232 bidirectional link. This speed measurement has an absolute error of ± 1 wheel turn/sampling interval, which is low enough to achieve good control.

The car speedometer works with a digital signal the wheel sends. Actually a signal is sent whenever the wheel performs $1/4$ of a turn. We measure the velocity counting this time, with a clock of 10 MHz. The error is negligible unless the speed is too low. The sampling is performed once every $1/10$ of a second (the loop refresh rate) thus the speed error is $1/4$ turn divided by $1/10$ second the absolute error is ± 0.9 km/hr.

Three commands are available for the interface card:

Speed Request: The control system in the PC sends the hexadecimal value 01. The card returns a 4 byte value with the instant speed.

Set Watchdog: The control system in the PC sends the hexadecimal value 02, followed by a value between 0 to 7 that represents the duration of the watchdog. This command is used for maintaining the card under control when the communication link fails. The card answers an echo as acknowledgment.

Set Desired Speed: The control system in the PC sends any hexadecimal value between 00 H and FFH, except 01 H and 02 H. 00 H represents a signal of 1 V at the accelerator (pedal max. up), and FFH represents 4 V (pedal max. down).

3) Control Layer: Once the system actuators and the electronic environment are defined, we proceed with the definition of the intelligent control system used for longitudinally controlling the testbed cars. This system is based on fuzzy logic embedded in a hierarchic architecture and it is associated with a powerful world model. In this paper we emphasize the description of the fuzzy controller, which carries out the tasks of maintenance of the speed (CC) and adapting to the lane speed (ACC).

In a short description of our fuzzy controller, it uses the t-norm minimum in order to implement AND operator. The t-conorm maximum is used in order to implement OR operator. The syntax of the fuzzy rules is defined as:

IF *X* [OR/AND *Z*] THEN *Y*.

Where *X* and *Z* are fuzzy input variables and *Y* is the fuzzy output variable. Center of mass method implements the defuzzification operation [40], as shown in (1)

$$x_{\text{out}} = \frac{\sum_i w_i x_i}{\sum_i w_i} \tag{1}$$

where w_i represents the value of the weight of each rule *i* and x_i is the crisp value of each rule *i* condition, understanding weight as the degree in which the crisp current values of the inputs satisfy the set of rule condition.

A more extensive description of the ORBEX fuzzy coprocessor can be found in [41]

A PD fuzzy controller has been defined in order to achieve these related tasks. The first step of this development was the CC fuzzy system. After that, and based on the same controller, we developed the second step in our research, the addition of adaptive capabilities to the CC controller in order to achieve an ACC system.

a) The Fuzzy Cruise Control System: The equipped testbed vehicle provides the necessary data to obtain the input information for the control system: the instantaneous speed and the time interval between two speed measures. This input consists of two fuzzy variables:

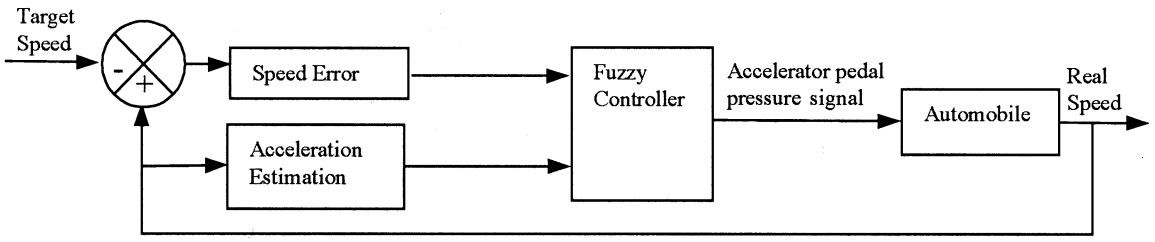


Fig. 2. CC system diagram.

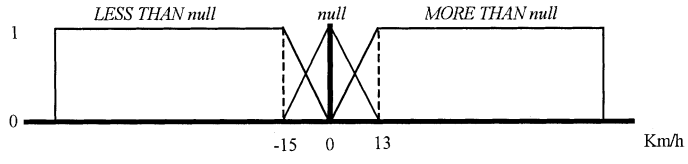


Fig. 3. Speed Error membership function after the applying MORE THAN and LESS THAN modifiers.

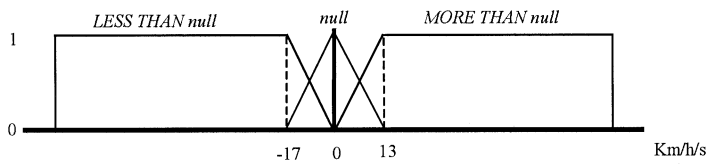


Fig. 4. Acceleration membership function after applying MORE THAN and LESS THAN modifiers.

Speed Error is the difference between the current speed and the user-preset speed. We can express it mathematically as follows in (2).

$$\text{Speed\_Error} = \text{Current\_Speed} - \text{Pre-set\_Speed}. \quad (2)$$

Acceleration is approximated by the derivative of the speed for the instant t, described in expression (3).

$$\text{Acceleration}_t = \frac{\text{Current\_Speed}_t - \text{Current\_Speed}_{t-1}}{\Delta t}. \quad (3)$$

For an optimal calculation of the acceleration it is essential to have a precise measurement of the time interval. We use the low-level clock of the PC for measuring time with an accuracy of 1/1 193 180 s. The measure even if somehow noisy, is good enough for our purposes. However a Fourier Filter [42] improves considerably this measure.

The output of the fuzzy controller will be the new accelerator-pedal-pressure-like signal (electrical voltage) that will be sent to the car internal on-board computer through the electronic layer. Schematically we can represent the fuzzy PD controller as shown in Fig. 2, where inputs to the fuzzy controller are normalized between 0 and 1. The output is also a value in the $(-1, 1)$ range, which stands for the increment (or decrement) that must be applied to the throttle to achieve the desired speed.

Let us define now the fuzzy membership functions of the system variables. There are three membership functions, two per each input variable and one for the output result.

Speed Error Membership Functions: This input variable is measured in kilometers per hour and defines the profile of the speed error with only one linguistic label named "null," scaled between 0 and 1 (Fig. 3). Only this label is required because ORBEX allows fuzzy modifiers like MORE THAN or LESS THAN to be used.

Because of the use of these kind of modifiers, the fuzzy labels MORE THAN null and LESS THAN null can be generated (Fig. 3).

When the speed error is more than null, it means that the real speed is higher than the user desired speed so the accelerator pedal must move up. If the speed error is less than null, it means that the car runs slower that it should so the system must accelerate. The Fig. 3 shows that the membership functions are not symmetrical. The dynamics of the vehicle and the mechanical actuation when the car brakes or accelerates are the reasons for this asymmetry.

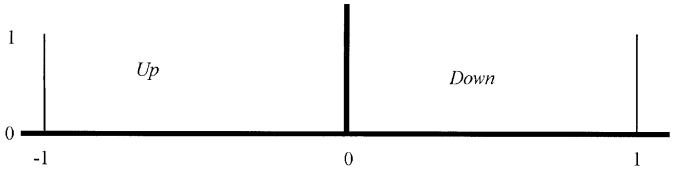


Fig. 5. CC fuzzy controller output.

Acceleration Membership Functions: In order to use intuitive units, this input variable is measured in kilometers/hour*Second. Usually, the car tachometer units are kilometers per hour, and the acceleration shows the speed increment per second, a very easy representation for user understanding. There is also only a linguistic label named "null" that is normalized between 0 and 1 (Fig. 4). As with the speed error variable, we use fuzzy modifiers.

With the application of the relational operators LESS THAN and MORE THAN, the membership function is transformed like shown in Fig. 4.

The acceleration is more than null means the car is speeding up; if it is less than null, the car is speeding down. The asymmetry of the functions is due to vehicle dynamics in braking and accelerating.

The controller output is the stepping on the accelerator pedal, performed through an electrical signal. The fuzzy controller adds a weight to this output between 0 to 1 in two linguistics labels; the "Up" label means the control must release the accelerator pedal and the "Down" label means the control must step on this pedal. This output is incremental because it is added to the accumulated value of the accelerator in each control loop iteration. Its membership function is shown in Fig. 5.

Finally, we define the fuzzy rules of the CC control system. Only four rules are necessary:

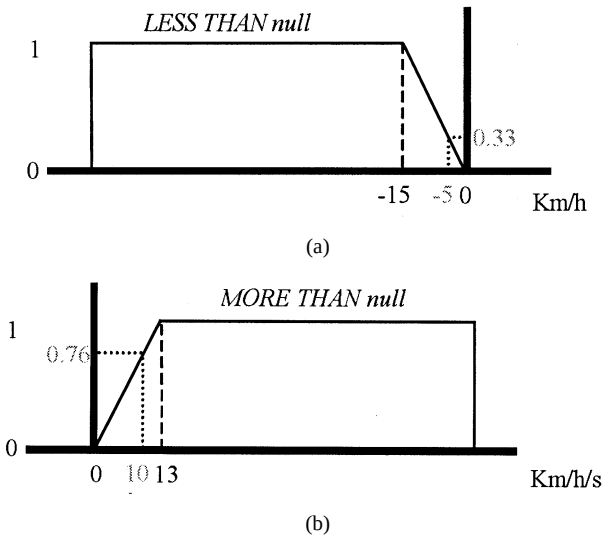


Fig. 6. (a) Assertion of "-5 km/h is LESS THAN null." (b) Assertion of "10 km/h/s is MORE THAN null."

IF *Speed_Error* **MORE THAN** null **THEN** *Accelerator* up
IF *Speed_Error* **LESS THAN** null **THEN** *Accelerator* down
IF *Acceleration* **MORE THAN** null **THEN** *Accelerator* up
IF *Acceleration* **LESS THAN** null **THEN** *Accelerator* down

In order to show the power of ORBEX, we can combine these rules into the following two rules:

IF *Speed_Error* **MORE**
THAN null **OR** *Acceleration* **MORE THAN** null **THEN** *Accelerator* up
IF *Speed_Error* **LESS**
THAN null **OR** *Acceleration* **LESS THAN** null **THEN** *Accelerator* down

The Speed_Error is the proportional component of the control and the Acceleration is the derivative component. This means that, when the speed of the car is not at the desired value, the Speed_Error adjusts the throttle pressure order, and the Acceleration smoothes out the actuation of this command.

Let us give a short overview of how a fuzzy control system can be implemented so that computationally cryptic statements (although easily understandable by a person) such us the preceding ones can yield a number to feed a digital to analog converter.

Speed_error and acceleration are fuzzy input variables and accelerator is an output variable represented by a singleton. Figs. 3 and 4 show the degree up to which the numerical values of Speed_error and acceleration satisfy the assertions "null," "LESS THAN null" or "MORE THAN null." Note, that in fuzzy control, input variables have a numeric value. Thus, in our example we will assume

$$\text{Speed\_error} = -5\ \text{km/hr}$$
$$\text{Acceleration} = 10\ \text{km/hr}\cdot\text{s}.$$

Fig. 3 shows the degree up to which an assertion such as "-5 km/h is LESS THAN null" is true. In our case in Fig. 6(a) we see this degree is 0.33. In a similar way we see in Fig. 6(b) that the degree up to which the assertion "10 km/h·s is MORE THAN null" is 0.76. From Figs. 3 and 4 we find that both assertions "-5 km/h is MORE THAN null" and "10 km/h·s is LESS THAN null" have a truth degree of 0.

The above paragraph can be rewritten saying that the fuzzified value of the assertion "the speed error is lower than null" is 0.33 and that of "the acceleration is more than null" is 0.76. The other two rules have a truth degree of 0. Then only the rules:

IF *Speed_error* **LESS THAN** null **THEN** *Accelerator* down
IF *Acceleration* **MORE THAN** null **THEN** *Accelerator* up

are to be applied, with a weight 0.33 for the first rule and 0.76 for the second.

Now, the controller applies the Center of Mass defuzzification method, and the output for the accelerator pedal will be (1). The output fuzzy variable accelerator has two linguistic values UP and DOWN, represented by singletons 1 and -1. Thus, the output is computed in (4).

$$x_{\text{out}} = \frac{0.33\cdot\text{up} + 0.76\cdot\text{down}}{0.33+0.76} = \frac{0.33\cdot 1 + 0.76\cdot(-1)}{0.33+0.76} = -0.39. \quad (4)$$

This means that the fuzzy controller will send a command to the accelerator low-level controller, normalized from -1 to 1: in this case -0.39. The low-level controller will increment or decrement its actual pressure over the throttle by a proportional value to this output.

b) The Adaptive Fuzzy Cruise Control System: This second control system is based on the above-mentioned fuzzy CC, with its objective being to maintain a safe gap with the vehicle ahead in the same lane of the road. This operation implies a speed adaptation to the speed of the preceding vehicle that must be done automatically by the controller, overriding the CC speed maintenance of the desired speed. The main application of this controller is in highway driving and platoons, in order to improve the safety and the comfort of driver and passengers in a high-speed driving environment. The extreme situation is when the preceding car stops; then the ACC equipped car must stop too. This is a classical event in traffic jam driving, and is generically named Stop&Go. Let it be clear that this is not a different controller; it is an upgrade of the previous controller. If no other car is detected or the time-gap with a detected car is long enough, this control works as the previous.

The keeping of a user defined safe distance from the next vehicle is a speed dependent function. This means that the inter-vehicle gap is to be larger as the speed is higher. This is the Time-Headway concept basis: a time dependent safe distance from the preceding car. For example, if we set a safe time-gap of 2 s when we drive at 40 km/h, the space-gap is 22.2 m but if our speed is 100 km/h, this gap is about 55.5 m. The setting of the time gap depends on the braking power of the car, the weather, the maximum driving speed, etc. For example, in Article 54 of the Spanish Highway Code the following is stated:

"The driver of a vehicle driven behind another shall leave a gap between them that permits the vehicle to stop in case of a sudden brake, without a collision, taking into account the speed, adherence and braking conditions."

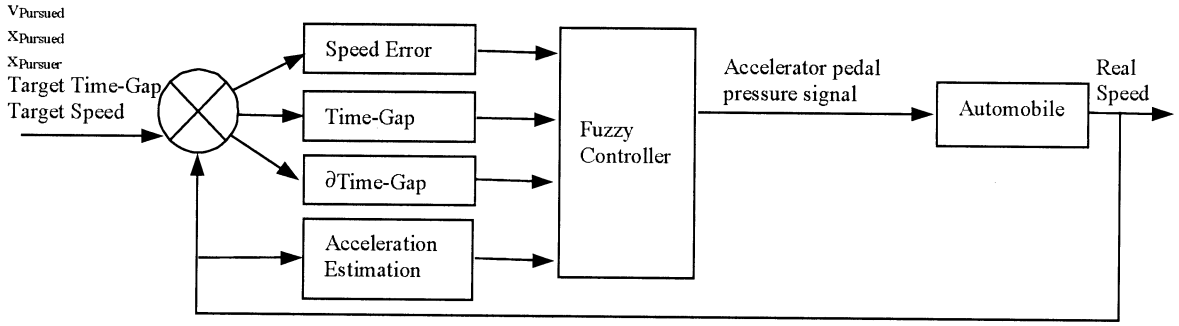


Fig. 7. ACC controller schematic.

Two new input fuzzy variables, a new rule and two rule modifications were added to the controller in order to perform the ACC. The output is the same as that in the CC controller: the actuation over the throttle pedal.

When we define this controller and its associated experiments, we will introduce a new terminology. We will name as pursued car the human driven vehicle that is driven in front of the ACC equipped car. The pursuer car will be the automatically driven vehicle that adapts its speed to the preceding one. At this point, we shall define the new input variables.

Time_Gap_Current: It is the time headway, the time it takes the pursuer vehicle to reach the point where the pursued vehicle is at present speed, in seconds. The mathematical representation is (5)

$$\text{Time\_Gap}_{\text{current}} = \frac{x_{\text{Pursued}} - x_{\text{Pursuer}}}{v_{\text{Pursuer}}} \tag{5}$$

where x_{Pursued} and x_{Pursuer} are the x coordinate of the pursued and the pursuer cars over the reference line, in meters, and v_{Pursuer} is the speed of the pursuer car in meters per second.

Time_Gap_Target: It is the time-headway set by the user to be maintained by the ACC system from the preceding vehicle. In commercial vehicles it should be between 1 and 2 s.

Derivative of Time_Gap: Is the variation of the Time-Gap_current with time. Its mission is to smooth the actuation in the same way that the Acceleration smoothes the Speed_Error. The equation used to calculate this variable for the control iteration *i* is (6)

$$d\_\text{Time\_Gap}_i = \frac{\text{Time\_Gap}_i - \text{Time\_Gap}_{i-4}}{4\Delta t}. \tag{6}$$

This variable is very unstable because it fluctuates wildly between positive and negative values easily in consecutive control iterations. An average filter is used for stabilizing the system and the fourth part of the Time-Gap increment in the last four iterations is used. This filter is very simple but it is enough in order to stabilize the variable for a good control, as shown in experiments.

A new sensor has to be added in order to obtain the information of the preceding car location. The control level is indifferent to the source of the data because the system will work correctly with any sensor that supplies the required position such as a laser, radar or a Radio-Ethernet link, which is what was used in our case [43]. All the cars involved in our experiments are equipped with an RTK-GPS and a Radio-Ethernet environment access; their positions are transmitted in a continuous loop and every car knows at every moment where the other is. The new controller architecture schematic looks as shown in Fig. 7, where the input values are normalized between 0 and 1 and the controller output is the increment (or decrement) that must be applied to the accelerator pedal to achieve the ACC in $(-1, 1)$ interval, as in CC.

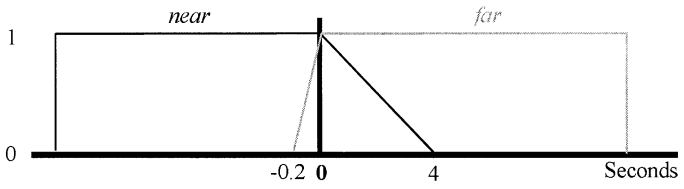


Fig. 8. Time-Gap_Error membership function.

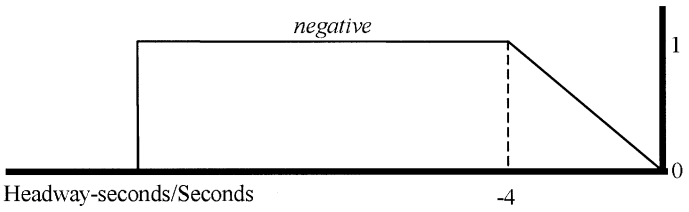


Fig. 9. Derivative of Time-Gap membership function.

In order to implement this in the ORBEX Fuzzy controller, two new input membership functions were added to the CC controller:

Time_Gap_Error Membership Function: It represents the time-gap error, the difference between the user-defined target time-gap and the current time-gap. Then, the input value for the gap fuzzy controller is represented in (7) and measured in seconds

$$\text{Time\_Gap\_Error} = \text{Time\_Gap}_{\text{current}} - \text{Time\_Gap}_{\text{target}} \tag{7}$$

Two linguistic labels have been defined: near and far. Near label shows the difference to the pre-determined reference safe gap and it activates when the two cars are near enough to start the ACC driving. The far label indicates when the cars are far enough to restart the classical CC driving. The graphical representation of the labels for this membership function is shown in Fig. 8.

The system will consider that when the gap error is less than 0 the car is fully near, when it is between 0 to 4 the pursuer is partially near to the pursued and if the error is bigger than 4, it is fully not near. The interpretation of the far label is analogous.

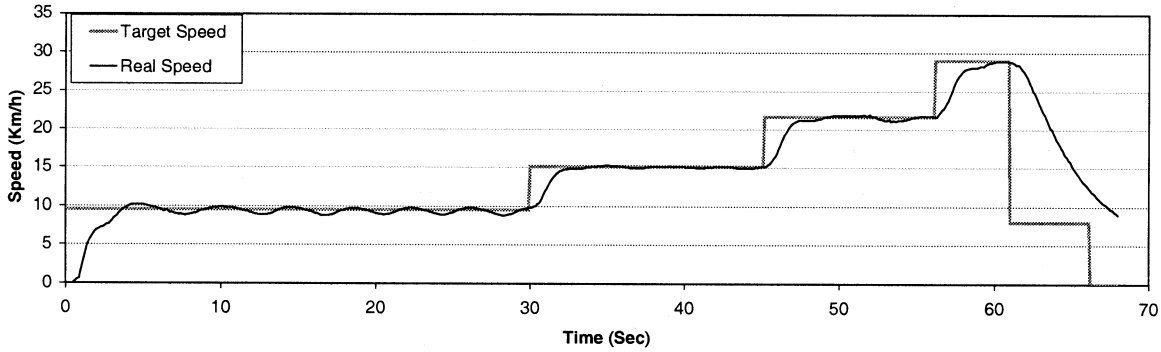


Fig. 10. Fuzzy cruise control performance at 9.6, 15, 21.6, and 29 km/h in the same round.

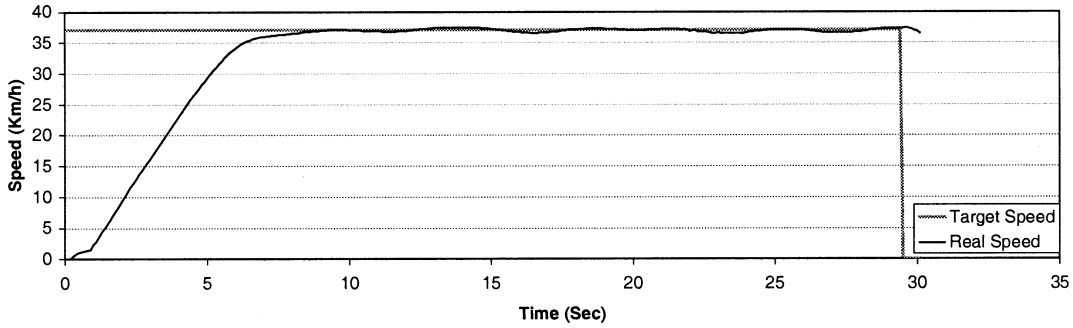


Fig. 11. Fuzzy cruise control performance at 37 km/h.

The aim of the controller is for the automated car to adjust exactly to the user-preset target time-headway when the target speed of the pursuer is bigger than the speed of the pursued. For example, if the driver has set the target time-headway at 2 s, when the distance between his vehicle and the next one is less than 6 s (that is 4 s of time gap error) the control action starts. This variable actuates with greater strength as the gap reduces. Should the gap get to be less than the preset value (2 s in this case), the label would be considered totally near. Similarly, when the gap is between 1.8 and 2 s the far linguistic label increases from 0 to 1 and if the distance is more than 2 it is considered fully far.

Derivative of the Time_Gap Membership Function: The units of this input variable are headway-seconds per second. This is the variation of the time-gap per time unit. A linguistic label named negative is defined. It means that the actual safe distance is less than the previous one thus the gap tends to reduce and the cars are nearer than in the last control iteration. Fig. 9 is the graphical representation of this membership function. When the value is lower than -4, the negative value is set to the maximum.

The last components of the ACC controller are the fuzzy rules. We added a new rule and we also modified two CC previous rules. The new set is as follows:

IF *Speed_Error* **MORE THAN** null **THEN** *Accelerator* up

IF *Speed_Error* **LESS THAN** null **AND** *Time_Gap_Error* **MORE THAN** near **THEN** *Accelerator* down

IF *Acceleration* **MORE THAN** null **THEN** *Accelerator* up

IF *Acceleration* **LESS THAN** null **AND** *Time_Gap_Error* far **THEN** *Accelerator* down

IF *Time_Gap_Error* near **AND** *d_Time_Gap* negative **THEN** *Accelerator* up

The aim of these rules is to maintain the CC and to keep a safe distance. To do this, the gap control only actuates when the pursuer car is near the pursued and it inhibits itself in other cases, the control thus becoming the classical CC. Another significant consideration is that there will only be speed adaption when the pursued car initially drives slower than the pursuer because it is necessary to inhibit the acceleration rules (Accelerator down) and boost the braking rules (Accelerator up). Let us see in detail the modified acceleration rules.

IF *Speed_Error* **LESS THAN** null **AND** *Time_Gap_Error* **MORE THAN** near **THEN** *Accelerator* down

The throttle signal decreases when the pursuer car is near the pursued so it will never accelerate enough to crash with the other car.

IF *Acceleration* **LESS THAN** null **AND** *Time_Gap_Error* far **THEN** *Accelerator* down

This rule allows stepping on the throttle only when the pursuer car is far from the pursued one.

IF *Time_Gap_Error* near **AND** *d_Time_Gap* negative **THEN** *Accelerator* up

With this rule, the control steps off the throttle when the safe distance is near. The stabilization of the system is the reason for the inclusion of the derivative in this rule.

The output variable of the controller is the same as in the CC.

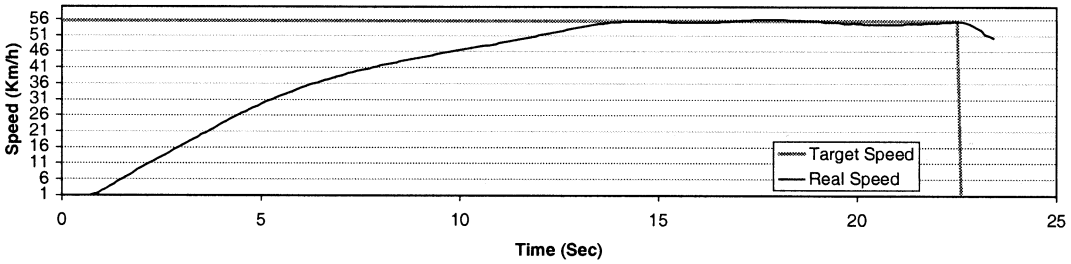


Fig. 12. Fuzzy cruise control performance at 55 km/h.

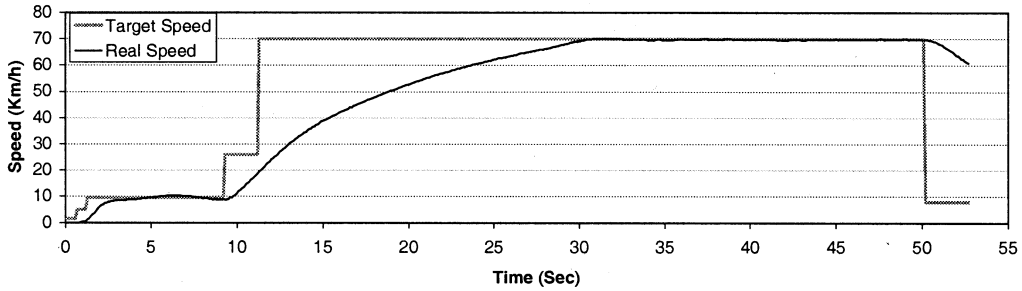


Fig. 13. Fuzzy cruise control performance at 70 km/h.

III. Related Experiments

After installing the above defined controller in the testbed cars, some experiments were made in order to demonstrate its performance. These tests were done in two different installations: The CSIC's Instituto de Automática Industrial in Arganda del Rey, Madrid, where we make experiments on urban automatic driving, and the probe circuit of GIAT Industries in Satory, Versailles, France, during the demos of the IEEE IV2002 Congress, sponsored by LCPC-INRETS's LIVIC (Laboratorie Sur Les Interactions Vehicule-Infraestructure-Conducteur), at which the high speed experiments could be done.

The first set of experiments test the longitudinal control to find how it adjusts to the user's desired speed; this is the fuzzy cruise control. In Fig. 10 we show the achieving and maintenance of a target speed which varies in time depending on the track events.

In this figure we can see that the user has selected four different speeds; 9.6, 15.1, 21.6, and 29 km/h. The transition of the change from one speed to another depends on the car kinematics and dynamics. As far as our goals are concerned these transitions are not important and it is sufficient to obtain the desired speed.

The short length of the IAI circuit, about 250 m, is the reason for making experiments only up to 55 km/h. Figs. 11 and 12 show the behavior of the control at two significant speeds: 37 km/h and 55 km/h (identical time scale in abscise).

Fig. 13 represents the fuzzy cruise control performance in the experiments realized at France in the LIVIC installations. In its 2 km circuit we have made tests up to 80 km/h.

In these experiments we can see that, in all cases, the performance of the fuzzy cruise control is very smooth, with a negligible error. Table I shows a statistical analysis of the CC controller performance. This represents the average error, the average deviation and the maximum error for each reference speed. The values are presented in kilometers per hour and the percent from the target speed. The combination of average error and its typical deviation shows that the running interval is always less than one kilometer per hour. At low speeds, this error is not a great percentage of the speed and at high speeds it is not relevant. The aim of the CC system is its application to high-speed roads in order to improve safety and increase the driver comfort; therefore, the low speed errors are not very important.

TABLE I
Statistical Analisys of the Speed Tracking (Excluding Transitory)

Target Speed (Km/h)	Average Error (Km/h)	Average Deviation (Km/h)	Maximum Error (Km/h)
9.6	0.23 (2.4%)	0.23 (2.4%)	0.80 (8.3%)
15	0.08 (0.5%)	0.10 (0.7%)	0.37 (2.5%)
21.6	0.16 (0.7%)	0.18 (0.8%)	0.60 (2.8%)
37	0.15 (0.4%)	0.18 (0.5%)	0.65 (1.7%)
55.5	0.35 (0.6%)	0.29 (0.5%)	1.05 (1.9%)
70	0.19 (0.3%)	0.12 (0.17%)	0.55 (0.8%)

Furthermore, another important cruise control system evaluation measurement is the comfort and the safety sensation for the passengers. This is critical for automatic systems applied to mass-produced cars because car buyers are usually not very fond of the automation of these kinds of tasks.

The second experiment set consists of the combination of the safe gap maintenance and stop-and-go capacity. This is the ACC system. Two testbed cars were used to make the controller tests. Both of them are equipped with a computer, an RTK-GPS receptor and Radio-Ethernet, but only the pursuer has activated the ACC. This means full automation of the throttle. The pursued is manually driven in order to simulate real conditions in which the reactions of the car ahead are unpredictable. The pursuer car must adapt its speed in all driving situations: when there is a car in its way and when it is alone on the road. In the first case, the rear car speed must be adapted in order to maintain a user defined safe gap, until it reaches the target speed also de-

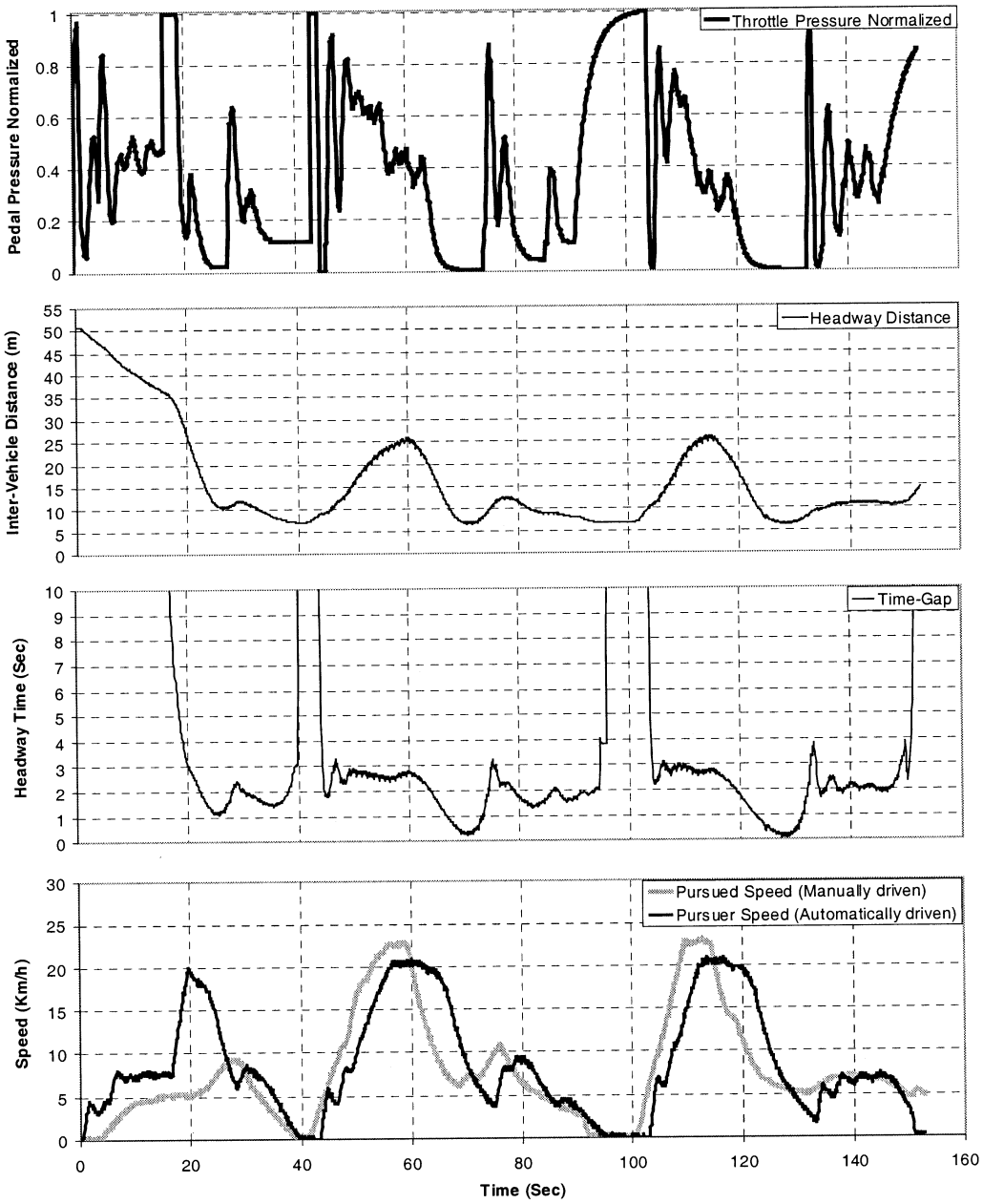


Fig. 14. Fuzzy ACC performance.

fined by the driver for the cruising. The previously described CC will be in effect when the car is alone on the road.

During the 156 s of this experiment, the pursued car runs at some variable speed between 0, when the car is stopped, and 30 km/h. The target speed of the pursuer car is always higher than the pursued one so, the cruise control will only maintain this speed when the pursuer is farther than the pre-selected safe gap. The experiment was made in a circuit with a straight lane with 2 km of length, oriented from West to East. We have also pre-set a minimum safe headway gap in the pursuer car of 2 s. This means that if the pursuer's target speed is higher than the pursued one, the pursuer's advanced cruise control will maintain a safe distance of 2 s from this preceding car. The reason for selecting this value is that 2 s are enough to fulfil the safety requirements in a limited environment such as our circuit.

Fig. 14 shows four different graphics. The bottom graphic shows the real speed of both cars for the duration of the experiment. The third graphic is the real-time headway time-gap between these testbed cars. The second represents the inter-vehicle distance, in meters, including the length of the pursued car (4 m). The top graphic shows the pressure on the throttle of the pursued car, meaning 0 foot quite off the pedal and 1 throttle fully pressed.

At the beginning of the experiment both cars are stopped and separated by about 50 m. The driver of the pursued car starts slowly while the automatic driver of the pursued car sets the target speed at 8 km/h. The time gap is initially very high, because the speeds are too low, so as the pursuer car speed increases, the gap reduces. After the first 16 s, the pursuer car gets to its target speed of 20 km/h. Then, the gap reduces drastically until it becomes about 2 s.

TABLE II
STATISTICAL ANALISYS OF THE ACC TRACKING

Minimum Gap	0.06 seconds
Minimum Distance	1.86 meters
Pursuer Speed	4.6 Km/h
Pursued Speed	5.11 Km/h
Pursuer Acceleration	-0.826 Km/h/sec
Pursued Acceleration	-0.108 Km/h/sec

At 40 s from the beginning, the pursued car stops. In this case, the pursuer car approaches the other car until the gap is about 2 m (6 m in the graphic), when it stops too (STOP). The reason for this change of units is that when the pursuer speed tends to zero, the time-gap tends to infinity and in this case it is not useful for control, because the cars would crash. It can be seen in the gap graphic around the 40th second. The choosing of 2 m is very reliable because when the pursuer reaches this distance the car is slowing down and stopping is imminent.

After this first stop, the pursued car accelerates strongly. The pursuer one accelerates too (GO). We can see in the figure that while the time gap is about 2 s, the distance in meters increases because the speed of the two cars increases rapidly. Generally, the average gap interval fluctuates between 3 to 1.5 s.

We can also observe in the gap graphic two reductions of the time gap between the 60–80 s and the 120–140 s. The reason of this fall is a hard braking of the pursued car and the pursuer delay for braking. In both cases the controller stops the second car without problems because the speed is very low. The distance between the cars is never less than 2 m. In order to improve the safety at these low speeds it is recommended to increase the minimum safe gap to 3 or 4 s.

The statistical results of the controller are shown in Table II, showing the worse results happen 128.4 s after starting. As it was already said in the introduction, [1], [44] and in the documentation of different commercial ACC's, they work only with speeds above 30 km/hr, which makes then incapable of working in slow traffic and traffic jams. The ACC described in this paper behaves in a similar way at these speeds, although it also works at very low speeds and performs stop and go, as it was clearly shown at the demonstrations performed at Versailles IV2002.

This means that at the instant where the gap and the distance are minimum, the pursuer has decelerated adequately so its speed becomes lower than the speed of the pursued. For this reason, this distance increases later.

IV. CONCLUSION

The alliance of fuzzy logic and Global Navigation Satellite Systems (GNSS) and its application to automatic driving systems can generate powerful controllers. In most of cases these kind of controllers go beyond the classical systems and offer a different point of view about the implementation of intelligent transportation systems.

The combination of ACC+Stop&Go is a good solution in order to achieve a safer driving, from high workload roads to traffic jams.

A lot of autonomous transport missions have already been carried out on private urban-like circuits using two Citroën Berlingo electric vans. Some of these missions are CC and Stop&Go tracking or platoon driving, assisted by a Fuzzy ACC system, which was the main feature that we presented.

In our opinion, the full automatic driving is a utopia and it will not be possible for twenty or thirty years. The developed experiences represent a starting point in order to reach this aim. The present real applications of these kind of controllers are grouped together in the field of safety elements of the driving as well as driver's aids. Similar systems are presently installed in mass produced cars or are in advanced development phase.

ACKNOWLEDGMENT

The authors express thanks to LIVIC because some of the tests could only be carried out in their magnificent facilities.

REFERENCES

[1] W. D. Jones, "Keeping cars from crashing," *IEEE Spectrum*, pp. 40–45, Sept. 2001.
[2] J. Crosse, "Tomorrow's world," *Auto. World*, pp. 46–48, Jan./Feb. 2000.
[3] *Scenarios and Evaluation Framework for City Case Studies*, 2002. European Comission Fifth Framework Programme Energy, Environment and Sustainable Development Programme Key Action 4: City of Tomorrow and Cultural Heritage, Deliverable 2, 3.
[4] C. Serafin, *Driver Preferences and Usability of Adjustable Distance Controls for an Adaptive Cruise Control (ACC) System*: Ford Motor Company Systems Technology Inc., Oct. 1996.
[5] "DIATS Final Report Evaluation of ATT System-Scenario Deployment Options,", Brussels, Belgium, RO-96-SC.301 CEC, DGVII, 1999.
[6] P. A. Ioannou and C. C. Chien, "Autonomous intelligent cruise control," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 657–672, Nov. 1993.
[7] R. Holve, P. Protzel, J. Bernasch, and K. Naab, "Adaptive fuzzy control for driver assistance in car-following," in *Proc. 3rd Eur. Congr. Intelligent Techniques and Soft Computing—EUFIT'95*, Aachen, Germany, Aug. 1995, pp. 1149–1153.
[8] J. I. Suárez, B. M. Vinagre, and Y. Q. Chen, "Spatial path tracking of an autonomous industrial vehicle using fractional order controllers," in *Proc. ICAR 2003, 11th Int. Conf. Advanced Robotics*, Coimbra, Portugal, June 30–July 3 2003, pp. 405–410.
[9] E. J. Rossetter and J. C. Gerdes, "Performance guarantees for hazard based lateral vehicle control," in *Proc. 2002 IMECE Conf.*, 2002.
[10] S. Bentalva *et al.*, "Fuzzy path tracking control of a vehicle," in *IEEE Int. Conf. Intelligent Vehicles*, 1998, pp. 195–200.
[11] S. Sheikholeslam and C. A. Desoer, "Design of Decentralized Adaptive Controllers for a Class of Interconnected Nonlinear Dynamical Systems: Part I," Department of Electrical Engineering and Computer Sciences, Institute of Transportation Studies University of California, Berkeley, CA, PATH Technical Memorandum 92-1, Feb. 3, 1992.
[12] R. Holve, P. Protzel, and K. Naab, "Generating fuzzy rules for the acceleration control of an adaptive cruise control system," in *Proc. Fuzzy Information Processing Soc. NAFIPS. Biennal Conf. North American*, 1996, pp. 451–455.
[13] *ALVINN: An Autonomous Land Vehicle in a Neural Network, Advances in Neural Information Processing Systems 1*, 1989.
[14] M. Sugeno, "An introductory survey to fuzzy control," *Info. Sci.*, vol. 36, 1985.
[15] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man Cybern.*, vol. SMC-15, pp. 116–132, Jan./Feb. 1985.
[16] L. A. Zadeh, "Fuzzy sets," *Info. Contr.*, vol. 8, pp. 338–353, 1965.
[17] E. H. Mamdani, "Application of fuzzy algorithms for control of a simple dynamic plant," *Proc. Inst. Elect. Eng.*, vol. 121, no. 12, pp. 1585–1588, 1974.
[18] M. Sugeno *et al.*, "Fuzzy algorithmic control of a model car by oral instructions," in *Proc. IFSA'87 Special Issue on Fuzzy Control*, K. Hirota and T. Yamakawa, Eds., Oct. 1987.
[19] M. Sugeno, I. Hirano, S. Nakamura, and S. Kotsu, "Developement of an intelligent unmanned helicopter," in *Proc. IEEE Int. Conf. Fuzzy Systems*, vol. 5, 1995, pp. 33–34.
[20] M. Sugeno, *Industrial Applications of Fuzzy Control*. New York: North Holland, 1985.

[21] M. Persson, F. Botling, E. Hesslow, and R. Johansson, "Stop&go controller for adaptive cruise control," in *Proc. IEEE Control Applications Conf.*, vol. 2, 1999, pp. 1692–1697.
[22] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujiri, "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications," *IEEE Trans. Intell. Transport. Syst.*, vol. 3, no. 3, pp. 155–161, Sept. 2002.
[23] M. Sugeno, "Theory of fuzzy integrals and its applications," Ph.D. dissertation, Tokyo Institute of Technology, Japan, 1974.
[24] M. Bursa, "Big names invest heavily in advanced ITS technology," in *ISATA Mag.*, Dec./Jan. 2000, pp. 24–30.
[25] J. M. Blosseville and M. Parent, "The french program: La route automatisee," *IEEE Intelligent Syst.*, pp. 10–13, May/June 2000.
[26] *Literature Review of ADAS/AVG (AVCSS) for Japan and United States*, June 27, 2001.
[27] R. Bishop, "Intelligent vehicle applications worldwide," *IEEE Intell. Syst.*, pp. 78–81, Jan./Feb. 2000.
[28] M. A. Sotelo, S. Alcalde, J. Reviejo, J. E. Naranjo, R. Garcia, T. DePedro, and C. Gonzalez, "Vehicle Fuzzy Driving Based on DGPS and Vision," in *Proc. 9th IFSA World Congr.*, Canada, July 2001.
[29] R. García and T. de Pedro, "Automatic car drivers," presented at the Proc. 31th Int. Symp. Automotive Technology and Automation, ISBN 0 9 532 576 0 6, June 1998.
[30] C. González, J. Reviejo, J. E. Naranjo, R. García, and T. de Pedro, "Sistema de ayuda a la plantación de viñas basado en navegación por satélite (GNSS)," presented at the Latin-American Conf. Automatic Control, CLCA 2002, Guadalajara, Mexico, Dec. 2002.
[31] S. Alcalde, "Instrumentación de un Vehículo Eléctrico Para Una Conducción Automática," Degree Thesis, Escuela Universitaria de Informática, Universidad Politécnica de Madrid, Madrid, Spain, Jan. 2000.
[32] L. A. Zadeh, "Fuzzy languages and their relation to human and machine intelligence," in *Proc. Man and Computer Processing Int. Conf.*, Bordeaux, France, 1970, pp. 130–165.
[33] ——, *Approximate Reasoning Based on Fuzzy Logic*. Berkeley, CA: Electronics Research Lab., College of Engineering, Univ. California, May 1979, Memorandum UCB/ERL M79/32.
[34] ——, *A Theory of Approxinate Reasoning, Machine Intelligence 9*, J. E. Hayes, D. Michie, and L. I. Mikulich, Eds. New York: Wiley, 1979.
[35] ——, "A new direction in AI, toward a computational theory of perceptions," in *AI Mag.*, Spring 2001, pp. 73–84.
[36] J. Reviejo, T. De Pedro, J. E. Naranjo, R. Garcia, C. Gonzalez, and S. Alcalde, "An open frame to test techniques and equipment for autonomous vehicles," in *Proc. IFAC-MIM 2000*, Patras, Greece, July 2000, pp. 350–354.
[37] R. Garcia and T. De Pedro, "Modeling a fuzzy coprocessor and its programming language," *Mathware and Soft Computing*, vol. V, no. 2–3, pp. 167–174, 1998.
[38] ——, "First application of the ORBEX coprocessor: Control of unmanned vehicles," in *Proc. EUSFLAT-ESTYLF Joint Conf.*, vol. 2–3, 2000, pp. 265–273.
[39] R. Garcia, T. De Pedro, J. E. Naranjo, J. Reviejo, and C. Gonzalez, "Frontal and lateral control for unmanned vehicles in urban tracks," presented at the IEEE Intelligent Vehicle Symp., June 2002.
[40] A. V. Patel and B. M. Mohan, "Some numerical aspect of center of area defuzzification method," *Fuzzy Sets and Syst.*, vol. 132, pp. 401–409, 2002.
[41] R. García and T. de Pedro, "A powerful model for a fuzzy coprocessor," presented at the Proc. Congreso Interamericano de Control Automático CLCA'98 & IFAC'98, Chile, 1998.
[42] J.-C. Radix, *Introduction au Filtrage Numèrique*, Paris: Eyrolles, 1970.
[43] J. E. Naranjo, "Sistemas de Comunicaciones Inalámbricos Aplicados a la Conducción Automática de Vehículos," Degree thesis, Universidad Politécnica de Madrid, Spain, Sept. 2001.
[44] *Evaluation of the Intelligent Cruise Control System. Volume I—Study Results*. Cambridge, MA: Volpe National Transportation Systems Center, Research and Special Programs Administration.

José E. Naranjo was born in Orense, Spain, in 1975. He received the B.E. and M.E. degrees in computer science engineering from the Polytechnic University of Madrid (UPM), Spain, in 1998 and 2001, respectively.

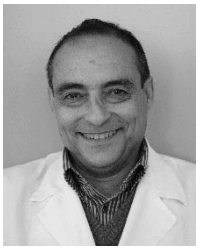
Since 1999, he has been with the the Industrial Computer Science Department, Instituto de Automtica Industrial (CSIC). His research interest includes fuzzy logic control and ITS.

Carlos González (M'86) was born in Torrelavega, (Cantabria), Spain, in 1947. He received the B.S. and the Ph.D. degrees in physics from Madrid University, Madrid, Spain, in 1969 and 1978, respectively, and the M.S. degree in computer science from the University of California at Los Angeles, in 1974.

Since 1971, he has been on staff of the Industrial Automation Institute of the Spanish Council for Scientific Research, where he is a software specialist for automation projects.

Jesús Reviejo received the B.E. and M.E. degrees in computer science from the Polytechnic University of Madrid, Madrid, Spain, in 1989 and 2003 respectively.

Since 1989, he has been working at the Industrial Automation Institute of the Spanish Council for Scientific Research, where from 1989 to 1998, he worked on robotics and walking robots. Since 1999, he has been working on ITS.



Ricardo García received the Ph.D. degree in physics from the Bilbao University, Spain.

He was a Founder of the Industrial Automation Institute of the Spanish Council for Scientific Research, where he works on intelligent robotics.

Dr. García was named in 2002 for the "Barreiros" Research on Automotive Field Prize for his AUTOPIA project on ITS.

Teresa de Pedro received the Ph.D. degree in physics from the Universidad Complutense of Madrid, Madrid, Spain, in 1976.

Since 1971, she has been working on artificial intelligence applied to automation at the Industrial Automation Institute, Spanish Council for Scientific Research. She is the Head of a Spanish team involved in the Integration of Sensors to Active Aided Conduction (ISAAC) project. Her research interests include fuzzy models for unmanned vehicles.