Proceedings of 2004 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 28 - October 2, 2004, Sendai, Japan

Vision-based Adaptive Cruise Control for Intelligent Road Vehicles

Miguel Angel Sotelo, IEEE member, David Fernández
Department of Electronics
University of Alcalá
Alcalá de Henares, Madrid, Spain
michael@depeca.uah.es

José E. Naranjo, Carlos González, IEEE member, Ricardo García, Teresa de Pedro, Jesús Reviejo
Department of Industrial Computer Science
Industrial Automation Institute. CSIC
Arganda del Rey, Madrid, Spain.
jnaranjo@iai.csic.es

***Abstract*— There is a broad range of Robotics Technologies that are currently being applied to the generic topic of Intelligent Transportation Systems (ITS). One of the most important research topics in this field is Adaptive Cruise Control (ACC), aiming at adapting the vehicle speed to a predefined value while keeping a safe gap with regard to potential obstacles. For this purpose, a monocular vision system provides the distance between the ego vehicle and the preceding vehicle on the road. The complete system can be understood as a Vision-based ACC controller, based on Fuzzy Logic, which assists the velocity vehicle control offering driving strategies and actuation over the throttle of a car. This controller is embedded in an automatic driving system installed in two testbed mass-produced cars operating in a real environment. The results obtained in these experiments show a very good performance of the Vision-based gap controller, which is adaptable to all speeds and safe gap selections.**

Keywords - Intelligent Road Vehicle; Vision-based Adaptive Cruise Control; Fuzzy Logic; Autonomous Vehicles

I. INTRODUCTION

Intelligent Transportation Systems apply Robotics 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 to provide assistance to the control of some 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 on top of the line-end vehicles. A second and more sophisticated step in the development of speed assistance 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 the speed in order to maintain a proper headway distance (gap) between vehicles in the same lane [3]. In the current work this is achieved through a visual headway sensor, a PC, and a fuzzy-logic speed controller. Previous research has shown that ACC systems can improve traffic conditions significantly [4, 5]. 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 [6], 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. In such a case the equipped car must stop at a safe headway if the preceding car stops. Likewise, Stop&Go is one of the most tedious and tiring operations human drivers have to carry out. Automatic 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 allows queues to discharge faster from bottlenecks [3, 7,8].

This paper addresses the integration of computer vision, mechatronics, and fuzzy control techniques [9] in order to get robotic aids to driving cars. The present application includes a car computer throttle control powered by a fuzzy logic controller, with the capability of performing a Vision-based Adaptive Cruise Control in an unmanned/manual driving. The experiments were made in a private test circuit using two automated mass-produced vehicles as testbed cars.

II. VISION-BASED VEHICLE DETECTION

Obstacles detection is a basic skill every autonomous vehicle must be endowed with in order to achieve reliable navigation. A monocular colour vision system has been deployed in this work in order to provide the vehicle with visual reactive capacity. Vision-based ACC is expected to provide more accurate range and range rate resolution than laser scanners [10] at a lesser expense.

A. Searching Area

The execution time is abruptly reduced by limiting the obstacles detection to some predefined area where the

0-7803-8463-6/04/$20.00 ©2004 IEEE

obstacles are more likely to appear. For this purpose, vehicles are searched for only within the limits of the lanes. Lane marking detection is then necessary. In this work lane marking detection and tracking is carried out using clothoid models [11] to describe and reconstruct the road geometry. Figure 1 shows a sequence of images depicting the lane edges, as well as the distance (in meters) from the left wheel to the left lane (left), the distance from the right wheel to the right lane (right), the radious of curvature of the road (R), and the maximum velocity recommended by the system (V) according to R.

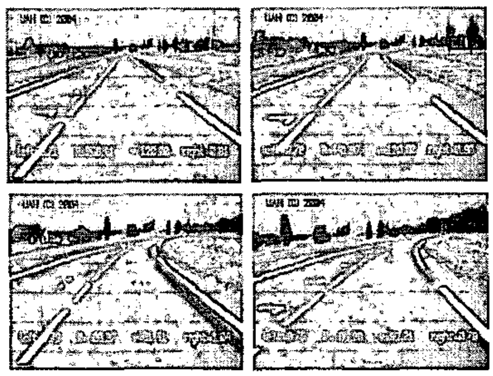


Fig. 1. Lane tracking in a sequence of images as a method to determine a region of interest for obstacles searching.

In order to robustly detect and track vehicles along the road (either in the ego-vehicle lane or in adjacent lanes), two consecutive processing stages are necessary. In the first step vehicles are localized based on edges and symmetry properties, while in the second one the already detected vehicle is tracked using a real time estimator.

B. Edges and symmetry discriminating analysis

A first discriminating analysis is carried out within the limits of the ROI determined by the lane edges. Edges and symmetry features are the key properties for this purpose. A set of horizontal edges along the lane has demonstrated to be a clear evidence indicating the possible presence of a vehicle, as depicted in figure 2, where a sequence of gradient images (Canny-based) is shown. Obviously, this evidence must be further validated using additional characteristics. Vertical edges and symmetry features are utilized then for validation. The joint use of these properties allows to obtain candidate edges representing the limits of the vehicles currently circulating along the road. Thus, a symmetry map [13] of the ROI (based on the lane edges) is computed so as to enhance those objects in the scene that present strong colour symmetry characteristics. After that, vertical edges are considered in pairs around those regions of the ROI where a sufficiently high symmetry measure (rejecting uniform areas) has been obtained, in order to account only for couples that represent possible vehicle contours, disregarding those combinations that lead to unrealistic vehicle shapes. Only those regions complying with realistic edges structures are validated as candidate vehicles.

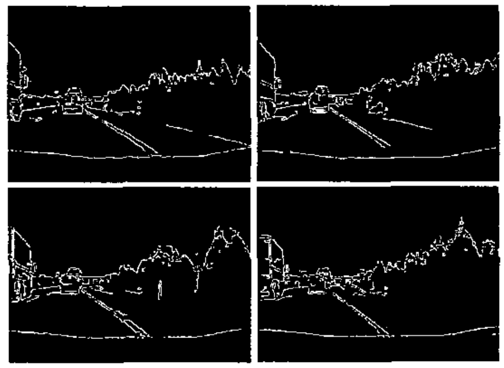
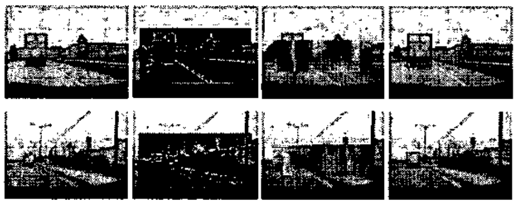


Fig. 2. Sequence of gradient images using Canny's filter. The presence of vehicles is clearly indicated by strong horizontal edges along the lane.

C. Temporal consistence

Using spatial features as the only criterion for detecting obstacles yields to sporadic incorrect vehicles detection in real situations due to noise. Hence, a temporal validation filter becomes necessary to remove non-consistent objects from the scene. This means that an object validated under the spatial features criterion described in the previous section must be detected several consecutive iterations in the image in order to be considered as a real vehicle. Otherwise it is discarded. A value t=0.5s has been used in practice to ensure that a vehicle appears in the image in a consistent time sequence. During the time-spatial validation stage a major problem is to identify the appearance of the same vehicle in two consecutive frames.

For this purpose, its (x,y) position in correlative frames is used. In other words, the position differences permit to describe the evolution of the vehicle in the image plane. At time instant t_0 the (x,y) position of each validated object under the spatial criterion is annotated in a dynamic list and a time count is started in order to keep track of temporal consistence of all candidate vehicles. At time t_0+1 the process is repeated using the same spatial validation criterion. The time count is increased only for those objects whose distance to some of the previous candidate vehicles is less than d_v. Otherwise the time count is reset. A candidate object is validated as a real vehicle when its time count reaches t=0.5s. Considering that the complete execution time of the vision algorithm is 40 ms, an empirical value d_v=1m has successfully proven to effectively detect real vehicles in the scene. Figure 3 depicts a couple of examples where the original and filtered images are illustrated together with the symmetry map of the ROI and the final position of the detected vehicle. The real distance to the preceding vehicle is measured using the reconstructed clothoidal geometry of the road.



a) b) c) d)

Fig. 3. Vehicle detection examples. a) Original image. b) ROI edge enhancement. c) Vertical symmetry map. d) Position of detected vehicle.

D. Vehicle Tracking

The position of the vehicle detected in the previous stage is tracked in two steps: position measurement and position estimation. The image contained inside the bounding box of the vehicle detected in the previous iteration is used as template in order to detect the updated position of the vehicle in the current image based on a best fit correlation. After that, data association for position validation is carried out using the (x,y) location of the newly detected vehicle. Basically it must be determined whether some of the objects in the current frame corresponds to the vehicle under tracking. For this purpose, a limited searching area is specified around the vehicle position yielding to efficient and fast detection. Likewise, a minimum correlation value and template size are established so as to determine the end of the tracking process, whenever poor correlations are attained or in case the vehicle gets too far away or out of the scene. The vehicle position measurements are then filtered using a Kalman filter. To avoid the problem of partial occlusions, the previously estimated vehicle position is kept during 5 consecutive iterations without obtaining any validated position before considering that the vehicle track has been lost. Should this happens, vehicle tracking is stopped and the vehicle detection stage is started again. To illustrate the vehicle tracking algorithm, figure 4 depicts a real traffic situation in which the preceding vehicle position is tracked in a sequence of images. The position of the vehicle is indicated by a blue bounding box, while the distance between the ego-vehicle and the preceding vehicle is shown below the bounding box. Additionally, the presence of other vehicles in the adjacent lane is indicated by a red horizontal line. The distance to the preceding vehicle as well as the relative velocity between the preceding vehicle and the ego-vehicle consitute the inputs to the Adaptive Cruise Control (ACC) System.

III. VISION-BASED ACC

The Vision-based Adaptive Cruise Control (ACC) is based on a fuzzy Cruise Control System. Although a detailed description of the fuzzy CC and ACC can be found in [12], a brief summary is provided next.

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:

Speed Error is the difference between the current speed and the user-preset speed. We can express it mathematically as follows in (1).

$$Speed\_Error = Current\_Speed - Pre\text{-}set\_Speed \qquad (1)$$

Acceleration is approximated by the derivative of the speed for the instant t, described in expression (2).

$$Acceleration_t = \frac{Current\_Speed_t - Current\_Speed_{t-1}}{\Delta t} \qquad (2)$$

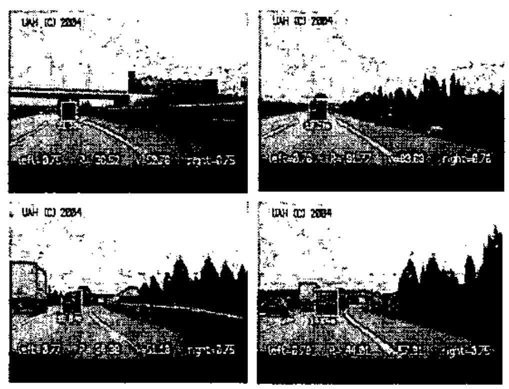


Fig. 4. Example of vehicle tracking in a real traffic situation.

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. Inputs to the PD 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. There are three membership functions, two per each input variable and one for the output result. 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 and 1 in two linguistic 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. By using the fuzzy modifiers MORE THAN and LESS THAN only two combined rules are necessary for the CC control system:

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

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 regard to the preceding vehicle 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 version 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 one. The keeping of a user defined safe distance from the preceding 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 two seconds when we drive at 40 Km/h, the space-gap is 22.2 meters but if our speed is 100 Km/h, this gap is about 55.5 meters. 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".

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 (3).

$$Time\_Gap_{current} = \frac{x_{Pursued} - x_{Pursuer}}{v_{Pursuer}} \tag{3}$$

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 seconds.

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 (4).

$$d\_Time\_Gap_i = \frac{Time\_Gap_i - Time\_Gap_{i-4}}{4\Delta t} \tag{4}$$

This variable is unstable because it fluctuates wildly between positive and negative values in consecutive control iterations. An average filter is used for stabilizing the system. This filter is very simple but it is enough in order to stabilize the variable for a good control, as shown in experiments. A vision-based sensor is incorporated 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. In order to implement this in the so-called 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 fuzzy gap controller is represented in (5) and measured in seconds.

$$Time\_Gap\_Error = Time\_Gap_{current} - Time\_Gap_{target} \tag{5}$$

Two linguistic labels have been defined: near and far. Near label shows the difference to the pre-determined reference safe gap and it gets activated when the two cars are near enough to start the ACC driving. The far label indicates that the cars are far enough to restart the classical CC driving. The graphical representation of the labels for this membership function is shown in Figure 5.

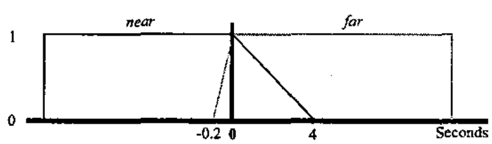


Fig. 5. Time-Gap_Error membership function.

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. 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 seconds, when the distance between his vehicle and the next one is less than 6 seconds (that is 4 seconds 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 seconds in this case), the label would be considered totally near. Similarly, when the gap is between 1.8 and 2 seconds 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 get reduced and the cars are nearer than in the last control iteration. 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 Cruise Control and to keep a safe distance. To do this, the gap control only actuates when the pursuer car is near the pursued one 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.

IV. Experimental Results

After installing the above defined system in the testbed cars, some experiments were made in order to demonstrate its performance. The vision-based vehicle detection system has been tested on recorded images of real highways in real conditions. Nonetheless, our test vehicles can not be taken out of our test tracks due to license problems. Accordingly, for the time being all vision-based ACC tests have been done at the CSIC's Instituto de Automática Industrial in Arganda del Rey, Madrid. The experiment set consists of the combination of vision-based safe gap maintenance and stop-and-go capacity. Two testbed cars were used to make the controller tests. Both of them are equipped with a computer, an RTK-GPS receiver and Radio-Ethernet, but only the pursuer has an onboard vision-based ACC. The GPS receiver allows to automatically drive any of the testbed vehicles using a previous map of the circuit. However, 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 defined by the driver for the cruising. During the 156 seconds of this experiment, the pursued car runs at some variable speed between 0, when the car is stopped, and 30 Km/h. 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 seconds. 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 seconds from this preceding car. Figure 6 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 meters. 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 seconds, the pursuer car gets to its target speed of 20 Km/h. Then, the gap reduces drastically until it becomes about 2 seconds. At 40 seconds from the beginning, the pursued car stops. In this case, the pursuer car approaches the other car until the gap is about 2 meters

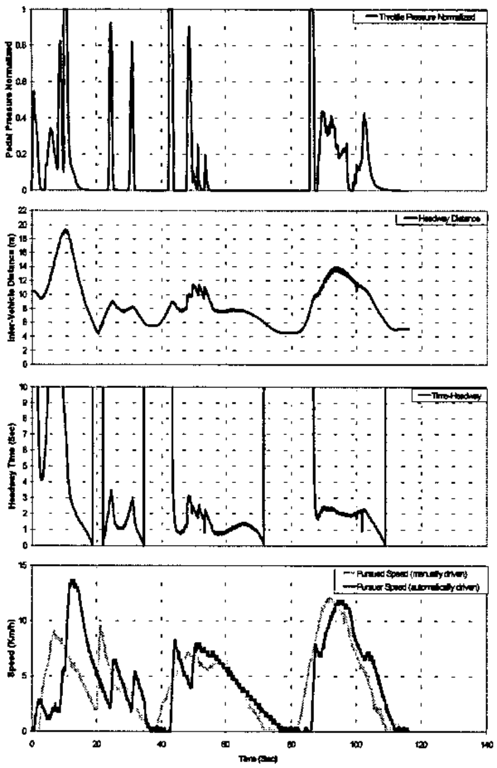


Fig. 6. Fuzzy ACC performance.

(6 in the figure), 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 purposes, because the cars would crash. It can be seen in the gap graphic around the 40th second. The distance between the cars is never less than 2 meters. In order to improve the safety at these low speeds it is recommended to increase the minimum safe gap to 3 or 4 seconds. A live demonstration using this system was carried out at the World Congress on Intelligent Transport Systems held in Madrid on november 2003.

V. CONCLUSIONS

The alliance of fuzzy logic and Computer Vision and its application to automatic driving systems can generate powerful controllers. In most 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 cruise control and stop and go tracking or platoon driving, assisted by a fuzzy adaptive cruise control system. 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.

ACKNOWLEDGEMENT

The components of this work have been granted by several Spanish Foundations, the last ones are: Ministery of Science CICYT DPI2002-04064-05-04 and Ministery of Fomento (Transports).

REFERENCES

[1] Willie D. Jones, "Keeping Cars from Crashing," *IEEE Spectrum*, September 2001, pp. 40 – 45.

[2] Jesse Crosse, "Tomorrow's World," Automotive World, January/February 2000, p. 46 – 48.

[3] STARDUST, "Scenarios and Evaluation Framework for City Case Studies", European Comission Fifth Framework Programme Energy, Environment and Sustainable Development Programme Key Action 4: City of Tomorrow and Cultural Heritage, Deliverable 2, 3, 2002.

[4] DIATS Final Report Evaluation of ATT system-scenario deployment options. RO-96-SC.301 CEC, DGVII. Brussels, Belgium. 1999

[5] P. A. Ioannou and C. C. Chien, "Autonomous Intelligent Cruise Control", IEEE Transactions on Vehicular Technology, pp. 657-672, volume 42, Nov 1993.

[6] R. Holve, P. Protzel, J. Bernasch, K. Naab, "Adaptive Fuzzy Control for Driver Assistance in Car-Following", Proceedings of the 3rd European Congress on Intelligent Techniques and Soft Computing – EUFIT'95, Aachen, Germany, pp. 1149-1153, Aug. 1995.

[7] M. Persson, F. Botling, E. Hesslow, R. Johansson, "Stop&Go Controller for Adaptive Cruise Control", Control Applications, 1999, Proceedings of the 1999 IEEE Conference on, Vol. 2, pp. 1692-1697, 1999.

[8] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, H. Fujiri, "Vehicle Control Algorithms for Cooperative Driving With Automated Vehicles and Intervehicle Communications", IEEE Transactions on Intelligent Transportation Systems, vol. 3, No. 3, September 2002, pp. 155 – 161.

[9] M. Sugeno, "Theory of Fuzzy Integrals and its Applications", Dissertation, Tokyo Institute of Technology, Japan, 1974.

[10] G. P. Gideon, O. Mano, A. Shashua. "Vision-based ACC with a Single Camera: Bounds on Rnage and Range Rate Accuracy. IEEE Intelligent Vehicles International Conference. Versailles, June 2002.

[11] E. D. Dickmanns and B. D. Mysliwetz. "Recursive 3-D Road and Relative Ego-State Recognition". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, February 1992.

[12] JE. Naranjo, C. González, J. Reviejo, R. García, T. De Pedro, "Adaptive Fuzzy Control for Inter.-Vehicle gap Keeping". IEEE Transactions on Intelligent Transportation Systems, vol. 4, No. 3, September 2003.

[13] A. Broggi, M. Bertozzi, A Fascioli, C. Guarino Lo Bianco, and A. Piazzi. "The Argo autonomous vehicle's vision and control systems." International Journal of Intelligent Control and Systems. Vol. 3, No. 4, 409-441, 2000.

[14] M. A. Sotelo, F. J. Rodríguez, L. Magdalena, L. M. Bergasa, L. Boquete. "A Color Vision-Based Lane Tracking System for Autonomous Driving on Unmarked Roads". Autonomous Robots 16, 95-116, 2004.