

Interoperable Control Architecture for Cybercars and Dual-Mode Cars

José E. Naranjo, Laurent Bouraoui, Ricardo García, Michel Parent, and Miguel Ángel Sotelo, *Member, IEEE*

Abstract—This work is driven by our vision that cybernetic transport systems (CTSs) based on fully automated urban vehicles (cybercars) will be seen on city roads and new dedicated infrastructures in the near future. These automated vehicles can be heterogeneous systems, such as human-driven traffic, each having different features and functionalities. In our case, we have developed a control architecture that can manage automatic driving of two cars: 1) CyCabs and 2) automated mass-produced cars. This architecture is interoperable and generates humanlike control of vehicles in any situation. Installation and communication with each vehicle are easy. The autonomous route-tracking behaviors are similar, even if the mechanical, electronic, software, and hardware configurations are different for both cars. The results of the developments shown in this paper are part of the European Union (EU) CyberCars-2 Project, which is currently under deployment.

Index Terms—Fuzzy control, hybrid control, intelligent control, proportional–integral–differential (PID) control, road vehicle control.

I. INTRODUCTION

IT IS WELL known that the field of autonomous vehicles was derived from autonomous robots. Consequently, the control schemas applied to these robots are applicable to cars. There are, of course, substantial differences between the subject of application and the operating environment, but general architectures are directly applicable. This way, the three classical mobile control architecture paradigms, i.e., the hierarchical/deliberative [1], [2], reactive [3], and hybrid paradigms [4], are applicable, with a few changes, to autonomous vehicles.

Some autonomous vehicle architectures based on these paradigms have been proposed.

The CyberCars European Union (EU) Project uses the Sharp architecture [5], which is a hybrid three-layer architecture

Manuscript received April 17, 2008; revised September 16, 2008. First published February 2, 2009; current version published February 27, 2009. This work was supported in part by European Commission Specific Targeted Research Project CyberCars-2. The Associate Editor for this paper was N. Zheng.

J. E. Naranjo was with the Instituto de Automática Industrial, Consejo Superior de Investigaciones Científicas, 28500 Madrid, Spain. He is now with the School of Computer Science, Polytechnics University of Madrid (UPM), 28031 Madrid, Spain (e-mail: joseeugenio.naranjo@upm.es).

L. Bouraoui and M. Parent are with the Institut National de Recherche en Informatique et Automatique, Informatique Mathématiques et Automatique pour la Route Automatisée (IMARA), 78153 Le Chesnay, France.

R. García is with the Instituto de Automática Industrial, Consejo Superior de Investigaciones Científicas, 28500 Madrid, Spain.

M. Á. Sotelo is with the Department of Electronics, University of Alcalá, 28871 Alcalá de Henares, Spain.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TITS.2008.2011716

(planner, mission scheduler, and motion controller) whose main contribution is the sensor-based maneuver concept. This concept consists of building into the system the possibility of combining planning with reactive action using some basic low-level maneuvers. Reactive actions are based on sensorial information only and have no planning. This architecture has also been extended to cooperative driving [6] among a set of cybercars.

Cybercars are road vehicles with fully automated driving capabilities. A fleet of such vehicles forms a managed transportation system, for passengers or goods, on a network of roads with on-demand and door-to-door capability. This concept emerged in Europe in the early 1990s and was introduced for the first time in The Netherlands in December 1997 for passenger transport at Schipol airport. Since then, it has been developed under a number of European projects, such as CyberCars, CyberMove, EDICT, Netmobil, and CyberC3. New projects, which are also supported by the European Commission, are now under way, i.e., CyberCars-2 and CityMobil.

A four-layer hierarchy architecture is proposed for the California Partners for Advanced Transit and Highways (PATH) Project in [7]. These layers are the network, link, planning, and regulation layers, which decompose complex maneuvers into a set of simpler and handier actions. A fault tolerance extension was later added to this architecture [8].

Another example of a vehicle control architecture is the hybrid Carnegie Mellon University (CMU) Navlab [9], which was directly adapted from the CMU's lengthy experience with mobile robots. This architecture has a planning stage, with strategic and tactical layers, and a low-level layer that includes specific behavioral skills for each driving situation.

Prof. A. Broggi's team at the University of Parma, Parma, Italy, developed an autonomous vehicle called ARGO. ARGO was able to semiautonomously travel a route, with an onboard computer managing the steering wheel of a mass-produced car and computer vision being used as the main sensorial input. The guidance system was based on a classical proportional (P) controller with a proportional gain defined by a supervision layer whose input signals were directly supplied by the lane-recognition vision system [10].

One of the most remarkable applications of autonomous vehicles is the DARPA Grand Challenge, which is a competition supported by the U.S. Department of Defense, where autonomous vehicles have to successfully pass a set of tests designed to demonstrate the feasibility of automatic driving for future transportation. These vehicles are fitted with sensors, actuators, and a control layer for tracking their routes. Thus,

the CIMAR NaviGATOR vehicle used a classical hierarchical high-level control architecture in the 2005 Grand Challenge [11]. This architecture is composed of four layers: 1) planning, 2) control, 3) perception, and 4) intelligence. The NaviGATOR autonomously traveled 14 mi before stopping and ranked eighteenth of the 23 finalists.

Similarly, Team Gray's KAT-5 vehicle uses an architecture that includes path planning and navigation stages, managing the vehicle actuators based on the information acquired by the sensors and intelligent systems for the low-level controllers [12]. KAT-5 completed the Grand Challenge after 7 h and 30 min of the competition.

The Autopia Program architecture for autonomous mass-produced-vehicle control [13] is based on the three-layer hierarchical paradigm, which includes a planning, a decision-making, and an actuation layer. This architecture should also be open and scalable, and even include different elements for each car. Depending on the user needs, this kind of vehicles can manually or automatically be managed, and are considered to be dual-mode cars.

This paper describes the common control architecture developed to compatibilize the same automatic driving control system (based on the Autopia architecture) across heterogeneous vehicles, such as CyCabs and Autopia-automated mass-produced cars.

In Japan, Prof. Tsugawa's team has been developing autonomous vehicle systems for more than 20 years. They apply multiple architectures, including a hierarchical system, to manage an automatic parking system installed in a mass-produced car [14].

II. EXPERIMENTAL VEHICLE

The CyberCars-2 Project involves two kinds of vehicles: 1) CyCabs and 2) dual-mode mass-produced cars (Fig. 1). CyCabs are robotized vehicles. They have all the features of a commercial vehicle but are equipped to be controlled from an onboard computer and have a wide range of navigation sensors installed [15]. Dual-mode vehicles offer an automatic mode in specific situations (such as platooning) and specific locations (such as automated parking lots), and a manual-assisted mode in regular situations [16].

In this case, the dual-mode vehicle is a Citroën C3 Pluriel, which is automated by the Spanish National Research Council (CSIC).

The electronic and computer equipment of both vehicles are very similar. They both have an onboard computer housing the control system. Sensors send information in several ways, depending on the car. For example, Global Positioning System (GPS) and laser scanners are connected via RS232 serial ports, whereas the speed signal is read through the controller-area-network bus interface. This bus is used to gather more information about the vehicle status. A wireless networking infrastructure is used to transmit the differential correction from a GPS base station to each vehicle and to broadcast position information between cars.

The C3 actuators are automated using an analog output attached to a servoamplifier that manages the original electric



Fig. 1. Testbed vehicles at INRIA Rocquencourt near Versailles, France, in September 2007.

power steering motor using several classical control and fuzzy controllers in a cascade control architecture. The throttle is managed by an analog signal that represents the pressure on the pedal, which is simulated by an analog card that receives the pressure value calculated by the pilot. Finally, the brake pedal is automated by a hydraulic pump with a variable-opening valve that regulates the pressure put on the brake clamp to attain the desired speed reduction [17].

On the other hand, the CyCab includes original electric, electronic, and computer systems to manage the speed and the steering. Each of the four wheels is independently powered and coupled with an additional dc motor that increases and reduces the vehicle's speed without additional braking systems. The steering has another dc motor engaged that is managed by a joystick in manual mode and electronic signals in automatic mode. An original internal computer receives the actuator movement commands from another high-level computer and sends the necessary electric signals to move the actuators.

III. ARCHITECTURE DESCRIPTION

To make two different vehicles, such as an automated mass-produced car and a CyCab, compatible, we decided to define a modular control architecture with abstract high-level modules that are valid for any mobile vehicle and low-level modules that are specific for each vehicle but can receive the same input signals. This interoperable architecture is based on the Autopia architecture [13], which was extended to be compatible with CyCabs. Fig. 2 shows a diagram of this architecture.

This architecture is based on a classical hierarchical system that is widely used in mobile robotics [1] but adapted to deal with automatic vehicle driving. Three layers comprise the system: 1) perception, 2) planning, and 3) actuation.

The perception layer includes all the sensorial information that will be necessary to successfully drive an automatic route. These data are used as input for the next layer. The planning layer calculates the necessary control actions that must be taken on the vehicle actuators to correctly track the route. Finally, the actuation layer receives the actuation orders from the other high-level layer and generates the related power signal for the actuators, so the vehicle can track the desired route.

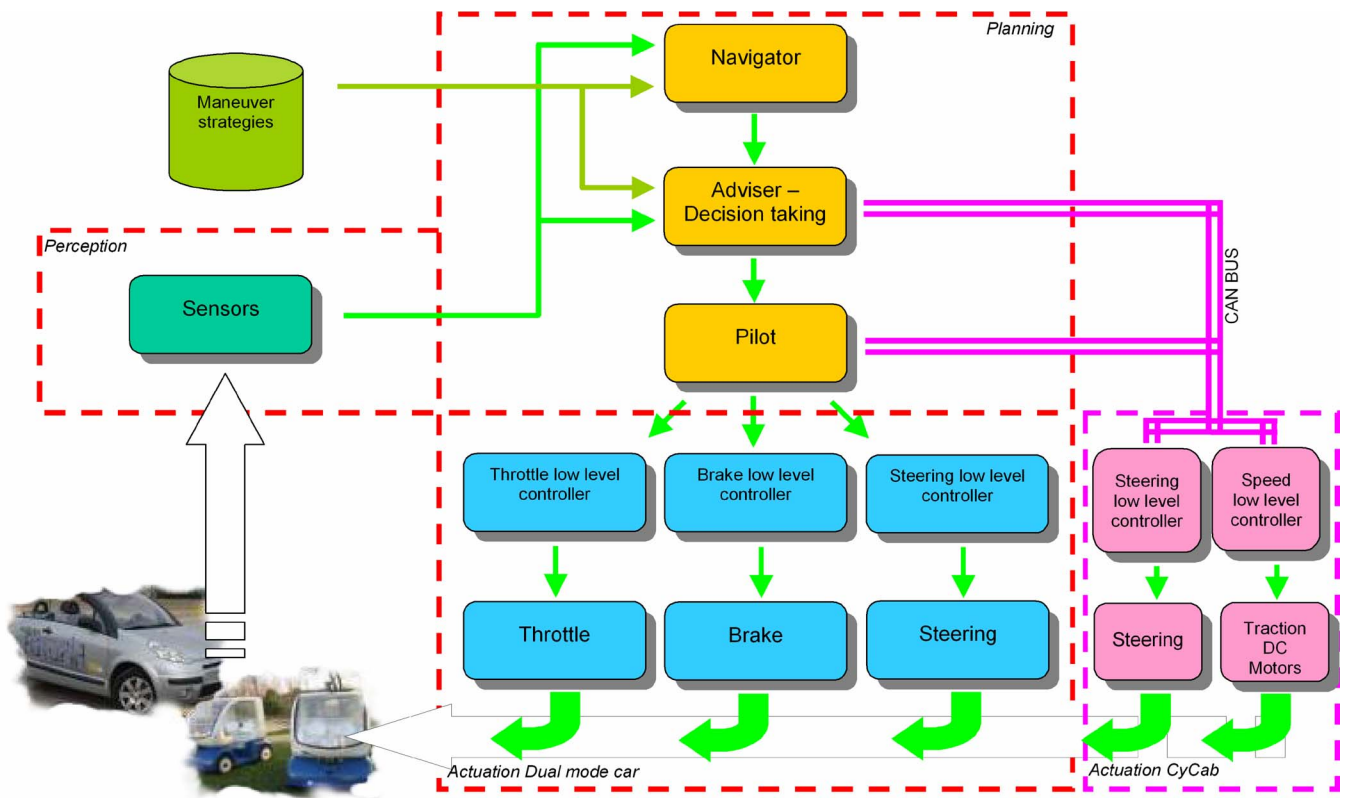


Fig. 2. Cybercar interoperable control architecture.

A. Perception Layer

In this paper, both kinds of cars use the same perception layer. This means that the information supplied by the onboard sensors is the same and that the outputs to be used for the next layer are the same.

At present, the main input of the sensorial layer is the data generated by a real-time kinematics (RTK) differential GPS receiver. The positioning accuracy of this kind of receiver is 1 cm (the precision of an autonomous GPS receiver is ~ 20 m), using the additional information wirelessly supplied by a georeferenced GPS base station and named differential correction.

To solve the need for data reception using wireless communications, we have built some communication systems into each car. There are three systems for transmitting and receiving information that are used depending on the location, environment, and signal availability. The default method is a 2-Mb/s wireless-fidelity (WiFi) network. This allows communication with the base station in direct vision zones. In situations where a WiFi link is not available, the networking is supplied by Universal Mobile Telecommunications System/General Packet Radio Service (GPRS) cellular telephony. Recently, we have been testing a new-generation WiFi link based on mesh networks. This system should upgrade classical links for an autoreconfigurable network and amplify the coverage by adding new nodes to the infrastructure. In addition, it should transform each vehicle into a mesh node, i.e., a router [15].

Other sensorial information variables are speed, acceleration, and steering-wheel turning angle. However, the perception layer is open and geared up to include new data from different

sensors. To achieve cooperation among vehicles, it is necessary to detect each circulating car. The method used to do this is to broadcast the GPS position of each vehicle across the wireless environment. With this information, each car is able to find out the position and direction of each car in real time and to take the appropriate actions to cooperatively circulate. This wireless interface is open and geared up to transmit any other necessary information, depending on how cooperative it is and the evolution of the research.

In the presented architecture, we have also added a knowledge base that includes the procedural information necessary to perform humanlike driving, e.g., the traffic code.

B. Planning/Control Layer

The second architecture layer includes the nucleus of the control system. In this layer, the sensorial inputs are managed, and some control actions are taken. These actions will be sent to pilot the actuators of the vehicle. This layer is divided into three sequential stages: 1) navigator, 2) adviser, and 3) pilot.

1) *Navigator*: This is the highest level in the control layer. In our paper, we have included this stage to add the possibility of intelligently planning the route for the vehicle to follow. In our case, however, this task is directly assigned to the users of the automated vehicles. It means that the commanded route for the autonomous vehicle is manually entered and defined as a set of GPS waypoints (digital cartography) that form polynomial lines, which are used as reference by the control system in tracking the route.

2) *Adviser*: It tells the next planning stage which maneuver is to be executed to successfully complete the route indicated by the navigator. Based on the information supplied by the sensorial inputs, it selects the driving mode that should run at any time. In this way, the copilot selects one of the sets of available maneuvers: straight-road tracking, bend tracking, overtaking, and adaptive cruise control or Stop&Go. These are complex maneuvers, and the copilot decomposes them into a sequence of simpler actions for execution in the next stage. For example, overtaking is divided into a lane change to the left lane of the road, a straight section in the left lane until the car passes the vehicle to be overtaken, and a second lane change to the right lane to return to normal [18].

The adviser processes and interprets the scene. Then, it decides which maneuver to make and which low-level controller or controllers from the next planning layer stage to activate. It decides what the reference speed should be at any time, depending on the road conditions and deviations. It also does the map matching between the reference digital cartography route supplied by the navigator and the real-time GPS coordinates, generating the respective deviation data.

3) *Pilot*: It is formed by a set of low-level controllers that define the basic human driver maneuvers. In our case, we use fuzzy controllers, but the pilot is open to any control method. It receives a set of input parameters and a low-level maneuver selection from the copilot and is able to generate an output signal that can be applied to the vehicle actuators.

The use of fuzzy logic for control systems has two main features: First, these kinds of controllers do not need an exact mathematical model of the system to be controlled. This characteristic is very important when dealing with systems that are hard to linearize, such as cars. Fuzzy logic avoids using very complex approximate models that are not very efficient if they are very realistic or not very realistic if they are efficient [19]–[22]. Second, fuzzy control does not aim to use the mathematical system representation but to emulate the behavior and experience of human drivers, mimicking their reactions. It can also add to the system subjective user knowledge, which is certainly a very useful characteristic for emulating human behavior [23].

C. Actuation Layer

The third architecture layer is actuation. In this layer, the control signals generated by the planning layer are adapted for transmission to the respective actuators. This layer is differently defined for each vehicle. This is a logical solution, because both kinds of the used vehicles incorporate different mechanical, electric, and electronic actuators. Even though the configuration of the physical modules is different, the received inputs are the same and are generated by the same planning architecture layer.

In this way, a dual-mode car has three actuators to be controlled, i.e., 1) throttle, 2) brake, and 3) steering, each with its own low-level controller that receives a target command from the planning layer and generates a signal that takes the same action as that of humans to effect this target action on these actuators.

On the other hand, the CyCab actuation layer is composed of two modules: 1) speed control and 2) steering control. The reason for this configuration is that this kind of vehicle does not have the usual throttle and brake pedals. Speed is managed from a joystick, whose proportional signal powers four electric motors (one per wheel) that electrically manage the speed decreases and increases.

IV. ARCHITECTURE IMPLEMENTATION

Having described the overall architecture, we proceed to explaining its actual implementation. All the vehicles are fitted with different elements that do, however, fulfill a similar function. From the architectural point of view, the functionalities of each element are the same.

The perception stage is the same for both vehicles, using GPS as the main sensorial input for navigation and internal vehicle speed signals for speed control. The core of the architecture is the planning stage, which houses the actual control system. The navigator module is exactly the same for both vehicles and uses the same digital cartography input. The adviser module is also the same, because, from a vehicle-management point of view, the selection of the next maneuver or driving mode to be performed is transparent. The selection of the different driving modes is the same when controlling a dual-mode car and a CyCab, or advising a human driver. The necessary conditions for changing from one driving mode to another are not dependent on the car type. The adviser also selects the reference speed for each road segment. This operation consists of sending the target speed to the low-level planning module and is the same for both vehicles.

The next module, i.e., the pilot, is different for each vehicle. From an operational point of view, managing the steering of dual-mode cars, with specific dynamics and wheel turning (e.g., tracking a straight road), is not the same as controlling a CyCab, which is smaller and has different dynamics, wheel size, and turning.

These differences are represented in the implementation of the specific fuzzy controllers that are in charge of controlling the steering and emulate the behavior of human drivers manually driving either vehicle.

This fuzzy controller performs the three basic tasks known as fuzzyfication, inference, and defuzzyfication. Fuzzyfication is the stage in which the crisp input values are transformed into fuzzy data. Inference is the procedure whereby the values of the fuzzy variables are inferred from a rule base, generating a fuzzy value for the output variable, i.e., the target steering-wheel angle. The final stage, i.e., defuzzyfication, transforms this output fuzzy value into crisp data that can be sent to an actuator. A more detailed fuzzy steering controller definition can be found in [24].

To parameterize the controller, it is necessary to define four parameters: 1) fuzzy variables (including the membership functions for these variables), 2) inference method, 3) defuzzyfication method, and 4) fuzzy rules.

Two fuzzy variables are necessary to manage the steering wheel of the two cars involved in the experiments: 1) *Lateral_Error*, which represents the distance between the vehicle and

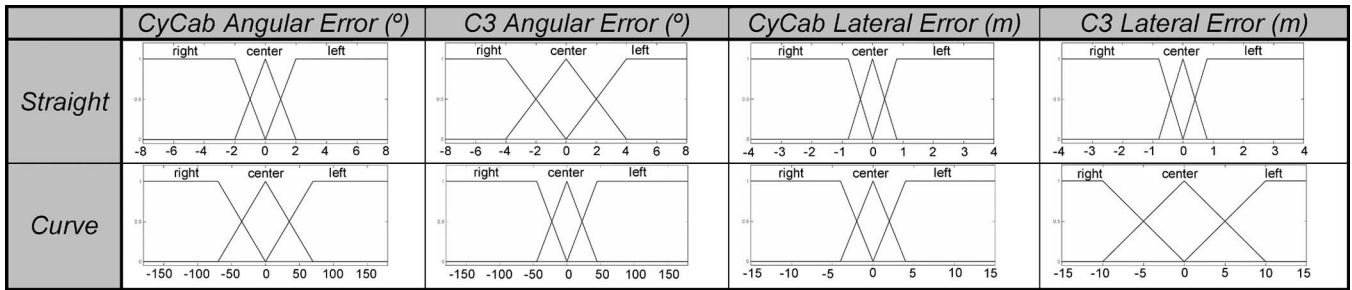


Fig. 3. Comparison of the fuzzy membership functions used in the CyCab and C3 controllers. With these functions, the controller can manage the steering wheel on bends and straight road sections.

TABLE I
FUZZY RULES FOR STEERING CONTROL

R1	IF Angular_Error <i>Left</i> THEN Steering <i>Right</i>
R2	IF Angular_Error <i>Right</i> THEN Steering <i>Left</i>
R3	IF Lateral_Error <i>Left</i> THEN Steering <i>Right</i>
R4	IF Lateral_Error <i>Right</i> THEN Steering <i>Left</i>

the reference route, and 2) *Angular_Error*, which represents the angle formed by the car's vector and the reference route. Their associated linguistic labels and membership functions are shown in Fig. 3.

In this case, our fuzzy system uses the center-of-mass

$$\frac{\sum w_i o_i}{\sum w_i} \quad (1)$$

defuzzification method, because it simplifies the controller's result calculation [25].

This formula is applied to each output variable for its linguistic labels i . The o_i elements are the crisp values of each linguistic label i of the named output variable, and w_i are the weights assigned to these linguistic values i , which are computed using Mamdani's inference method [26], where OR is represented by the maximum operator, and AND is the minimum operation. Mamdani's inference method was chosen, because its results have widely been tested in control systems.

Even though the two testbed vehicles are different, the driver's operational knowledge is the same for both cars. This is only logical because when we are driving, we move the steering wheel to the right to correct a trajectory deviation to the left and vice versa when the deviation is to the right. This procedural knowledge is represented using fuzzy rules. Only four rules model this human behavior, as shown in Table I.

In Table I, the fuzzy variables are shown in bold, and the linguistic labels are shown in italics.

The membership functions mark the difference between the fuzzy controller definition for the two vehicles. The membership function definition represents the degree of truth as an extension of valuation. This means that the membership functions define a fuzzy variable's degree of membership, which, e.g., has a linguistic value of left or right.

The definition differs for the two vehicles, because the steering actuation and the dimensions and dynamics of the vehicles are very different. For the same data input, then one of the systems could consider the vehicle to be a long way to the left, and the other could consider it to be only slightly off to

the left. The definition of these membership functions and the parameterizations of these subjective considerations are one of the key difficulties of fuzzy controller tuning.

Fig. 3 shows the shape of the membership functions for the dual-mode car and the CyCab.

V. RELATED EXPERIMENT

We have defined, implemented, and installed the aforementioned architecture in the testbed cars. Some experiments were conducted at the National Institute for Research in Computer Science and Control (INRIA) Rocquencourt during a CyberCars-2 meeting in September 2007. In these experiments, we used the same fuzzy controllers to manage the steering wheel of a Citroën C3 and a CyCab, modifying only the sets of membership functions for the input fuzzy variables.

The test was run on an automatic route around the Informatique Mathématiques et Automatique pour la Route Automatisée (IMARA) Group building. The route reference trajectory was previously defined as a sequence of GPS coordinates that the automatic driving systems can use as a reference. In the case of the CyCab, the reference route waypoints were manually taken, and in the case of C3, they were automatically taken. However, the control system is able to deal with any kind of reference files, independent of the number of points.

Both vehicles were also fitted with an RTK GPS receiver. This receiver obtains differential correction from a base station placed at INRIA via GPRS, achieving to-the-centimeter positioning accuracy.

Fig. 4 shows the automatic route around the IMARA building at INRIA, where the x -axis represents the Universal Transverse Mercator (UTM) East coordinates and the y -axis represents the UTM North coordinates in meters. Fig. 4 also shows an aerial image of the building (from Google Earth), the GPS reference route (black line), the tracked route (gray line), and the starting point represented as an arrow pointing to the starting direction.

Looking at Fig. 4, we find that the CyCab correctly follows the reference route back to the starting point. There are just a few oscillations at the top left of the figure, where the GPS precision decreased due to a weaker signal caused by tree canopy. However, the vehicle recovered the highest precision and continued the route. In this experiment, the inertial system on board the CyCab designed to correct GPS signal accuracy loss was not operational, and only the GPS was used for driving.

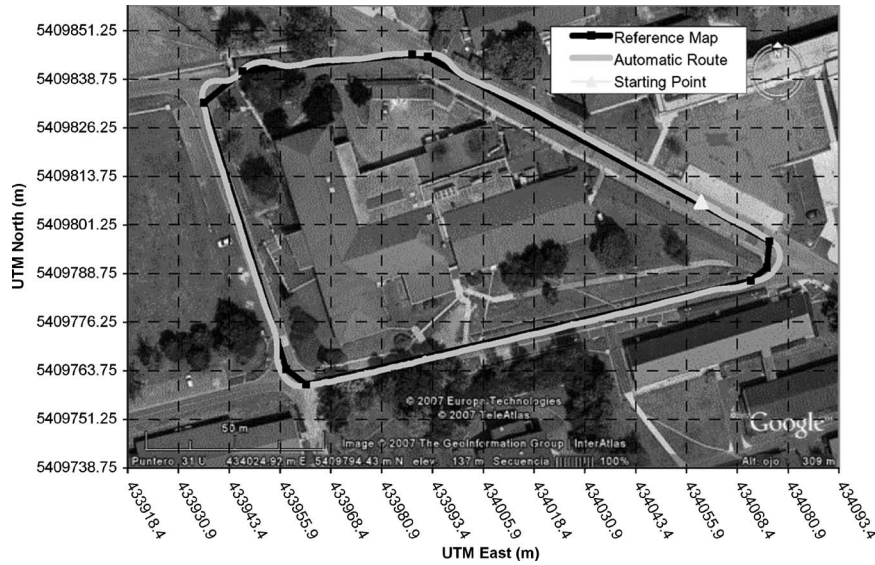


Fig. 4. CyCab automatic route around the IMARA building.

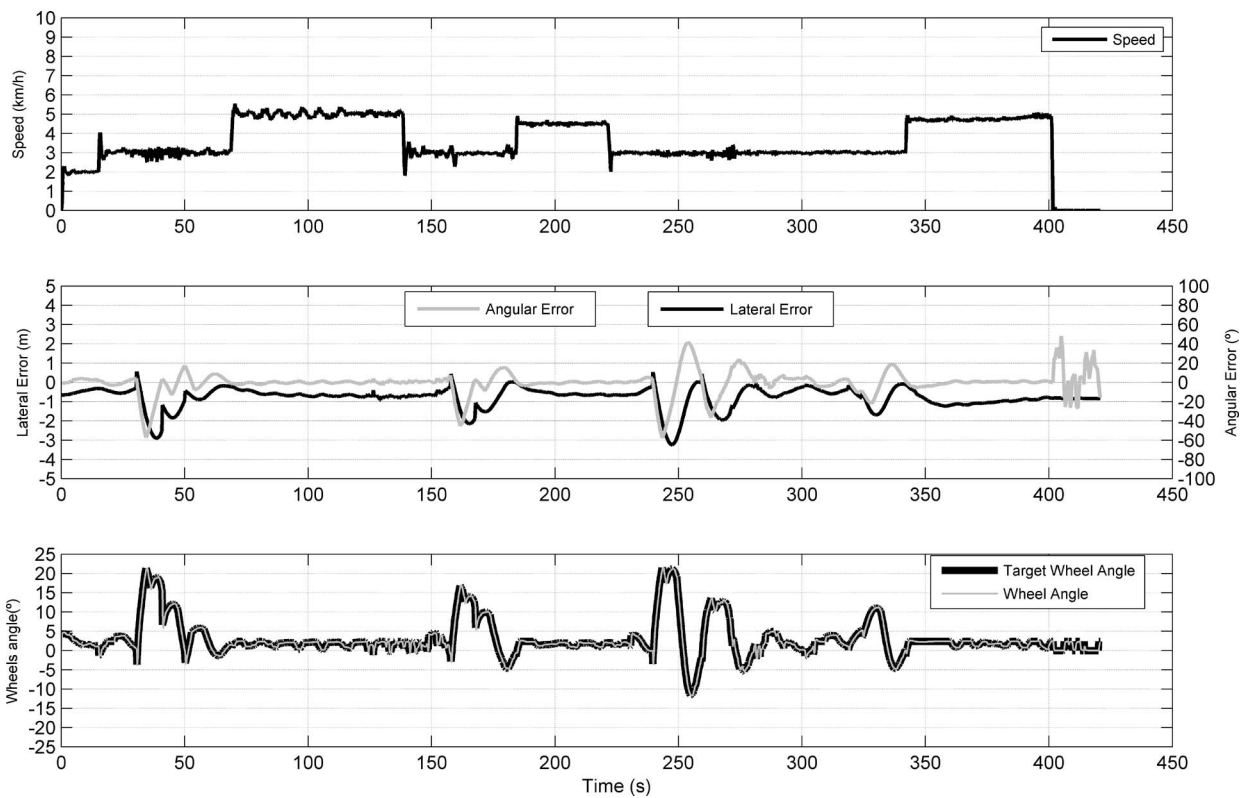


Fig. 5. CyCab control system variables for the experiment around the IMARA building.

Fig. 5, which comprises three graphs, shows the values of the internal control variables for this experiment. The top graph represents the car’s circulation speed during the experiment. The middle graph represents the value of the input variables’ lateral and angular errors. The values are clearly coherent with the consecutive bends taken by the car and shown in Fig. 4. Finally, the bottom graph represents the output of the fuzzy controller, the target wheel turning angle, and the value of the wheel turning angle. In this case, both values are almost the

same due to the fast response of the CyCab steering actuators. Fig. 5 shows that the controllers maintain the navigation errors close to zero, quickly correcting the vehicle trajectory to remain in the preset route around the IMARA building. The reference speed is kept at 6 km/h on straight roads and 3 km/h on bends. These speeds are very low for a car but are relatively high for a CyCab, which is designed to run at low speeds.

The graphic clearly shows the performance of the system in the first two turns to the right (positive wheel angle). In the third

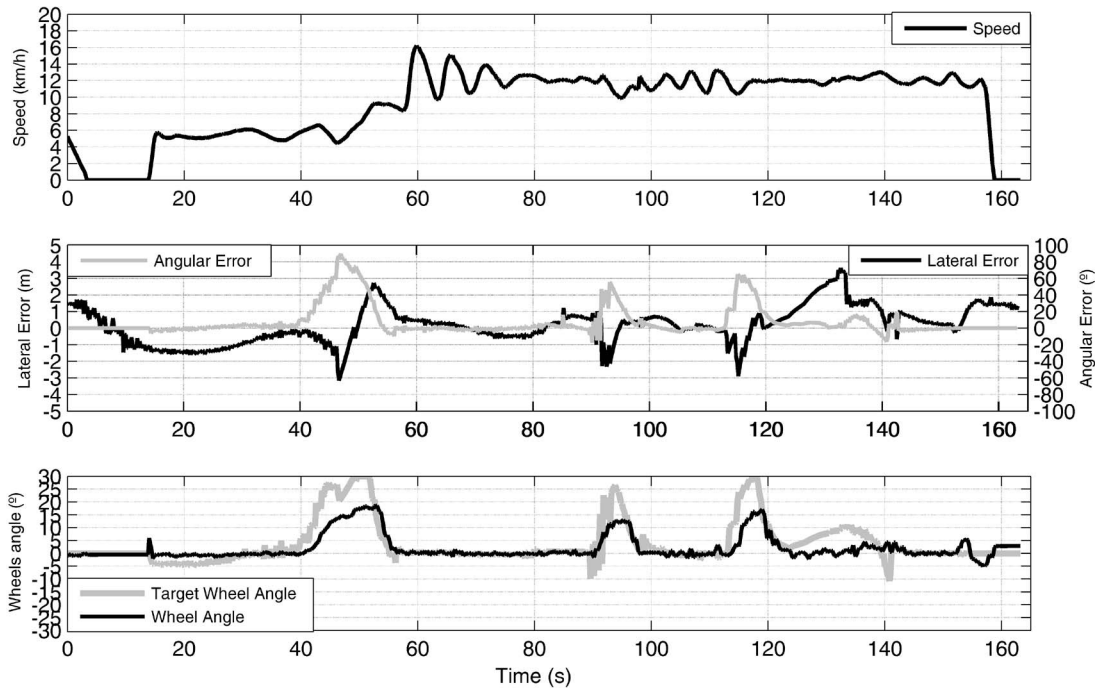


Fig. 7. Citroën C3 control system variables for the experiment around the IMARA building.

TABLE II
COMPARISON BETWEEN THE BEHAVIORS OF BOTH CARS IN THE SECOND STRAIGHT AND CURVE SEGMENTS

	Straight road segment				Curve road segment			
	Mean Error		Maximum Error		Mean Error		Maximum Error	
	Lateral (m)	Angular (°)	Lateral (m)	Angular (°)	Lateral (m)	Angular (°)	Lateral (m)	Angular (°)
CyCab	0.65	0.89	0.91	2.71	1.29	14.4	2.15	45.1
C3	0.36	1.9	0.59	6.3	0.87	24.2	2.31	55.9

VI. CONCLUSION

We have presented an open architecture for autonomous vehicles to support the automatic driving control of heterogeneous vehicles. This architecture has been developed and installed in two different vehicles as part of the EU CyberCars-2 Project: 1) a robotized CyCab vehicle and 2) an automated mass-produced Citroën C3. These vehicles were thus capable of individual autonomous behavior. Some autonomous driving experiments were run on these two vehicles during a CyberCars-2 meeting that was held at INRIA, Versailles, France. The main sensorial input for the vehicles in this experiment was the information supplied by a high-precision RTK GPS and the GPS digital cartography reference. The control layer of the architecture is composed of fuzzy controllers whose procedural rules are the same for both cars. The only difference is the tuning of the input variable membership functions. The overall architecture is the same for both cars. Only the low-level controller layer has particularly been designed for each vehicle and is specific for each kind of car. This architecture has experimentally been validated, and the tests performed show that the behavior of the testbed vehicles in autonomous driving is very similar, mimicking the human-driving behavior in the same situations and maintaining enough precision to consider this control architecture to be safe enough.

ACKNOWLEDGMENT

The authors would like to thank the European Commission, the Ministerio de Fomento, the Ministerio de Ciencia e Innovación, and Citroën España SA without whom this paper would not have been finished.

REFERENCES

- [1] R. R. Murphy, *Introduction to AI Robotics*. Cambridge, MA: MIT Press, 2000.
- [2] W. L. Nilsson, "Shakey the Robot," AI Center, SRI Int., Menlo Park, CA, Tech. Note 323, 1984.
- [3] R. A. Brooks, "A robust layered control system for a mobile robot," in *Readings in Uncertain Reasoning*. San Mateo, CA: Morgan Kaufmann, 1990, pp. 204–213.
- [4] R. C. Arkin, "Integrating behavioral, perceptual, and world knowledge in reactive navigation," *Robot. Auton. Syst.*, vol. 6, pp. 105–122, 1990.
- [5] C. Laugier and T. Fraichard, "Decisional architectures for motion autonomy," in *Intelligent Vehicle Technologies*. Warrendale, PA: SAE Int., 2001, pp. 333–391.
- [6] J. Kolodko and L. Vlacic, "Cooperative autonomous driving at the intelligent control systems laboratory," *IEEE Intell. Syst.*, vol. 18, no. 4, pp. 8–11, Jul./Aug. 2003.
- [7] P. Varaiya, "Smart cars on smart roads: Problems of control," *IEEE Trans. Autom. Control*, vol. 38, no. 2, pp. 195–207, Feb. 1993.
- [8] J. Lygeros, D. N. Godbole, and M. Broucke, "A fault tolerant control architecture for automated highway systems," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 2, pp. 205–219, Mar. 2000.
- [9] M. Bayouth, I. Nourbakhsh, and C. Thorpe, "A hybrid human-computer autonomous vehicle architecture," in *Proc. 3rd ECPD Int. Conf. Adv. Robot., Intell. Autom. Control*, 1997.
- [10] A. Broggi, M. Bertozzi, A. Fascioli, C. G. Lo Bianco, and A. Piazzi, "The ARGO autonomous vehicle's vision and control systems," *Int. J. Intell. Control Syst.*, vol. 3, no. 4, pp. 409–441, 1999.
- [11] C. D. Crane, D. G. Armstrong, R. Touchon, T. Galluzzo *et al.*, "Team CIMAR's NaviGATOR: An unmanned ground vehicle for the 2005 DARPA grand challenge," *J. Field Robot.*, vol. 23, no. 8, pp. 599–623, 2006.
- [12] P. G. Trepagnier, J. Nagel, P. M. Kinney, C. Koutsougeras, and M. Dooner, "KAT-5: Robust systems for autonomous vehicle navigation in challenging and unknown terrain," *J. Field Robot.*, vol. 23, no. 8, pp. 509–526, 2006.
- [13] J. E. Naranjo, C. González, T. de Pedro, R. García, J. Alonso, M. A. Sotelo, and D. Fernández, "AUTOPIA architecture for automatic

- driving and maneuvering,” in *Proc. IEEE ITSC*, Toronto, ON, Canada, Sep. 17–20, 2006, pp. 1220–1225.
- [14] N. Hashimoto, S. Kato, N. Minobe, and S. Tsugawa, “Automated vehicle mobile guidance system for parking assistance,” in *Proc. IEEE Intell. Veh. Symp.*, Istanbul, Turkey, Jun. 13–15, 2007, pp. 630–635.
- [15] L. Bouraoui, S. Petti, A. Laouiti, T. Fraichard, and M. Parent, “Cybercar cooperation for safe intersections,” in *Proc. IEEE ITSC*, Toronto, ON, Canada, Sep. 17–20, 2006, pp. 456–461.
- [16] J. E. Naranjo, C. González, R. García, and T. de Pedro, “ACC+Stop&Go maneuvers with throttle and brake fuzzy control,” *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 2, pp. 213–225, Jun. 2006.
- [17] V. Milanés, J. E. Naranjo, C. González, J. Alonso, and T. de Pedro, “Autonomous vehicle based in cooperative GPS and inertial systems,” *Robotica*, vol. 26, no. 5, pp. 627–633, Sep. 2008.
- [18] J. E. Naranjo, C. González, R. García, and T. de Pedro, “Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver,” *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 438–450, Sep. 2008.
- [19] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modelling and control,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.
- [20] B. K. Bose, *Fuzzy Control of Industrial Systems—Theory and Applications*. Dordrecht, Germany: Kluwer, 1998.
- [21] M. Jamshidi, “Fuzzy control of complex systems,” in *Soft Computing—A Fusion of Foundations, Methodologies and Applications*. Berlin, Germany: Springer-Verlag, 1997.
- [22] G. K. Viswanadha Raju and P. R. Bijwe, “Reactive power/voltage control in distribution systems under uncertain environment,” *IET Generation, Transmission Distrib.*, vol. 2, no. 5, pp. 752–763, 2008.
- [23] J. M. Mendel, “Fuzzy logic systems for engineering: A tutorial,” *Proc. IEEE*, vol. 83, no. 3, pp. 345–377, Mar. 1995.
- [24] J. E. Naranjo, C. González, R. García, T. de Pedro, and R. E. Haber, “Power-steering control architecture for automatic driving,” *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 4, pp. 406–415, Dec. 2005.
- [25] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.
- [26] 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, Dec. 1974.



José E. Naranjo was born in Orense, Spain, in 1975. He received the B.E. and M.E. degrees in computer science engineering and the Ph.D. degree in computer science from the Polytechnic University of Madrid (UPM), Madrid, Spain, in 1998, 2001, and 2005, respectively.

From 1999 to 2007, he was with the Instituto de Automática Industrial, Consejo Superior de Investigaciones Científicas, Madrid. In 2007, he joined the School of Computer Science, UPM, as an Associate Professor and then joined the Region of Madrid

Excellence Group Safety in Automobile Vehicles with special interest in Reduced Mobility Persons (SEGVAUTO). He is currently the Vice-Dean of the School of Computer Science, UPM. His research interests include intelligent systems, fuzzy logic control, intelligent transportation systems, usability, and accessibility.



Laurent Bouraoui received the B.E. degree from the French University “Paul Sabatier,” Toulouse, France, which is linked to the LASS/CNRS Robotic Department.

Since 2003, he has been the Chief Technical Engineer of the R&D Team on Advanced Road Transport (Informatique Mathématiques et Automatique pour la Route Automatisée (IMARA) Research Group) with the Institut National de Recherche en Informatique et Automatique, IMARA, Le Chesnay, France. He coordinates software development and integra-

tion for different European projects and manages public technical demonstrations that involve IMARA autonomous vehicles in different fields, such as communication, navigation, and computer vision. Prior to this, he was with the THALES Group, where he was involved in the development of small robots for military purposes. His current research interests include industrial mobile robotics for education and surveillance while helping a small company called WIFIBOT to grow.



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

He was a Founder of the Instituto de Automática Industrial, Spanish National Research Council (CSIC), Madrid, Spain, where he is currently working on intelligent robotics.

Dr. García was the recipient of the “Barreiros” Research on Automotive Field Prize for his AUTOPIA project on intelligent transport systems in 2002.



Michel Parent received the B.E. degree from the French Aeronautics School, École Nationale de la Statistique et de l’Administration Économique (ENSAE), Malakoff, France, and the M.E. degree in operation research and the Ph.D. degree in computer science from Case Western Reserve University, Cleveland, OH.

Since 1991, he has been a Program Manager with the R&D Team on Advanced Road Transport (Informatique Mathématiques et Automatique pour la Route Automatisée (IMARA) Research Group),

Institut National de Recherche en Informatique et Automatique (INRIA), Le Chesnay, France. This group focuses on the research and development of new forms of road transport and, in particular, on fully automated vehicles (cybercars). He has been or is the coordinator of several European projects on this topic. From 2001 to 2004, he was the Coordinator of the European Project CyberCars, which is continued now in CyberCars-2 (2006–2009). Before joining INRIA, he spent half of his time in research and academia at Stanford University, Stanford, CA; the Massachusetts Institute of Technology, Cambridge; and INRIA, and the other half in the robotics industry. He is the author of several books on robotics, vision, and intelligent vehicles, and numerous publications. He is also the holder of several patents.



Miguel Ángel Sotelo (M’02) received the Dr. Ing. degree in electrical engineering from the Technical University of Madrid, Madrid, Spain, in 1996 and the Ph.D. degree in electrical engineering from the University of Alcalá, Alcalá de Henares, Madrid, in 2001.

From 1993 to 1994, he was a Researcher with the Department of Electronics, University of Alcalá, where he is currently an Associate Professor. He has been serving as an Auditor and Expert for the FITSA Foundation, working on R&D projects on automotive applications since September 2004. He is the author of more than 100 refereed publications in international journals, book chapters, and conference proceedings. His research interests include real-time computer vision and control systems for autonomous and assisted intelligent road vehicles.

Dr. Sotelo is a member of the IEEE Intelligent Transportation Systems (ITS) Society and the ITS-Spain Committee. He was the recipient of the Best Research Award in the domain of Automotive and Vehicle Applications in Spain in 2002 and 2009, the 3M Foundation Awards under the category of eSafety in 2003 and 2004, and the Best Young Researcher Award from the University of Alcalá in 2004. He is currently an Associate Editor for the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.