Cooperative Maneuvering in Close Environments Among Cybercars and Dual-Mode Cars

Vicente Milanés, Javier Alonso, Laurent Bouraoui, and Jeroen Ploeg

***Abstract*—This paper describes the results of vehicle-to-vehicle (V2V) and infrastructure-to-vehicle (I2V) experiments implementing cooperative maneuvering for three different vehicles driving automatically. The cars used were cybercars from the Institut National de Recherche en Informatique et Automatique (INRIA), (France), which are fully automated road vehicles, and two mass-produced cars—one a Smart Fortwo car from TNO (Netherlands) equipped with additional actuators and sensors and the other a convertible Citroën C3 from IAI (Spain) that uses sensorial information to manage the actuators. The cars communicate by a wireless mesh network over Wi-Fi using the optimized link state routing (OLSR) ad-hoc protocol. The entire communication task is embedded in a small MIPS Linux Box (4G System Cube) that is transparent for the cars. A standard framework was defined with the parameters needed to perform adaptive cruise control (ACC) and intersection maneuvers among the cars, as well as emergency stops via a signal sent by the infrastructure. The experiments were carried out in La Rochelle (France) during the final demonstration of the European Union (EU) Cybercars-2 Project.**

***Index Terms*—Cooperative systems, intelligent control, road vehicle control, vehicle safety communications.**

I. Introduction

COOPERATION by communication among vehicles [vehicle to vehicle (V2V)] and between vehicles and infrastructure [vehicle to infrastructure (V2I)] is one of the most important research and development topics today. The European Commission includes projects relating to communications under a joint initiative called e-safety [1]. There are over 60 FP6 and FP7 research projects in this initiative, most of them dealing with communications.

Toulminet *et al.* made a comparative study [2] of the three projects dealing with cooperative systems that they consider particularly worthy of note [Cooperative Vehicle-Infrastructure Systems (CVIS), Cooperative systems for road safety “Smart Vehicles on Smart Roads” (SAFESPOT), and CO-Operative Systems for Intelligent Road Safety (COOPERS)]. The main goal of CVIS [3] is to develop technologies to permit vehicles to communicate with each other and with the infrastructure using the proposed communications, air interface, long and medium range (CALM) standard. CVIS would also be the testbed for CALM. SAFESPOT’s [4] aim is to improve the safety of roads by detecting critical situations in advance. In COOPERS [5], a set of vehicles is connected via wireless communication with the infrastructure to exchange information related to the road segment along which the vehicle is traveling, the goal being to increase road safety.

In a parallel line of work, the preventive and active safety applications contribute to the road safety goals on European roads project has as its main aim to develop safety systems at a competitive cost. Its approach is to integrate in-vehicle systems and combine them through communications into a network of integrated safety systems [6].

California’s Partners for Advanced Transit and Highways (PATH) is a multidisciplinary program established in 1986, in collaboration with the State Department of Transportation. One of the main contributions of the PATH program is in platooning experiments in which communication is used to improve highway throughput by demonstrating how a platoon of cars can act in unison [7]. Another U.S. contribution to intelligent transportation systems is organized by the Defense Advanced Research Projects Agency (DARPA). Three competitions with unmanned vehicles have taken place so far. DARPA challenges are not research programs but contests in which the winner receives a prize. The most recent, named the DARPA Urban Challenge, was held in November 2007, in which 11 teams qualified for the final. During the final event, 30 manned vehicles and over 50 vehicles in total were simultaneously navigating the city streets. Boss [8], of the Tartan Racing team, won this edition by covering the circuit in a little over 4 h. Six robots ended the race, which was an outstanding development given that none finished the first DARPA Grand Challenge in 2004.

In Japan, the advanced safety vehicle (ASV) program [9]–[11] started in 1991. Its fourth phase, i.e., Nissan ASV-4, was presented in October 2007. It is an attempt to reduce the number of traffic accidents by using V2V communications that warn the driver of the presence of other vehicles in bad weather conditions or areas of poor visibility.

The main objective of the Cybercars-2 project [12] is to enhance and enrich the cybernetic transport system concept already dealt with in FP5 project Cybercars [13] and CyberMove [14] by means of V2V and V2I communications used for

Manuscript received February 6, 2009; revised August 31, 2009, January 20, 2010, and April 27, 2010; accepted May 3, 2010. Date of current version March 3, 2011. This work was supported in part by the European Commission Specific Targeted Research Project Cybercars-2. The Associate Editor for this paper was S. C. Wong.

V. Milanés and J. Alonso are with the Centre of Automation and Robotics, Centro de Automática y Robótica, 28500 Madrid, Spain (e-mail: vmilanes@iai.csic.es; jalonso@iai.csic.es).

L. Bouraoui is with the Informatique Mathématiques et Automatique pour la Route Automatisée, Institut National de Recherche en Informatique et Automatique, 78153 Le Chesnay, France (e-mail: laurent.bouraoui@inria.fr).

J. Ploeg is with The Netherlands Organisation for Applied Scientific Research TNO, Science and Industry, Business Unit Automotive, 5700 AT Helmond, The Netherlands (e-mail: jeroen.ploeg@tno.nl).

Digital Object Identifier 10.1109/TITS.2010.2050060

vehicles to perform cooperative maneuvering at close range and cooperative traffic management from remote control centers. The final demonstration took place in La Rochelle, France, in September 2008, with cars from different manufacturers and different groups cooperating. This paper presents the integration of the architectures of the different participating vehicles and describes the development of the communications network and the cooperative maneuvering control algorithm. The trials were conducted in an ad-hoc figure-eight closed circuit in a public square.

This paper is organized as follows: Section II details the different architectures of the participating vehicles. The communications architecture is explained in Section III. Section IV presents the data structure of the messages exchanged by the vehicles and the infrastructure. Section V describes the control algorithms used, and Section VI describes the experiments that were performed. Finally, Section VII gives some concluding remarks.

II. Participating Vehicle Architectures

In this section, we present the three different architectures used by the participants. Since the critical layer is that of communications, all the architectures were prepared to support the 4G System Cube.

A. IAI Dual-Mode Vehicle: C3 Pluriel Clavileño

The architecture model for autonomous vehicle control developed by IAI-Centro de Automática y Robótica (CSIC) is capable of dealing with different vehicle models, actuators, and control methods.

Functionally, this is a dual architecture. On one hand, it defines the internal control structure for each vehicle to provide it with individual autonomous driving capabilities. On the other hand, it must support cooperation among a set of vehicles that have been individually equipped with such autonomous driving capabilities. It needs to be open and distributed, allowing scalability without substantial changes to its configuration, even with the inclusion of different elements in each car and irrespective of the vehicle model [15], [16].

The general architectural scheme is shown in Fig. 1. A set of independent autonomous vehicles are linked with each other and with a central monitoring station, sharing all the information necessary to cooperate and perform human-like maneuvers. The main novelty of this architecture is its transparency, independently of the kind of car that is added to the automatic driving environment. Thus, while each vehicle has its own driving system, uses its own sensorial inputs, and acts on its own actuators, they are all equipped with a similar configuration.

Clavileño (see Fig. 3), i.e., the IAI-CSIC's C3 convertible used in these experiments, is equipped with a double-frequency real-time kinematics differential Global Positioning System (RTK-DGPS) receiver and an inertial measurement unit, which are used as the main sensorial inputs and provide the vehicle's absolute coordinates with centimeter accuracy [17]. Speed and acceleration data are also taken from the vehicle via a controller-area network (CAN) bus, together with the steering angle, the current gear, and the pressure exerted on the pedals.

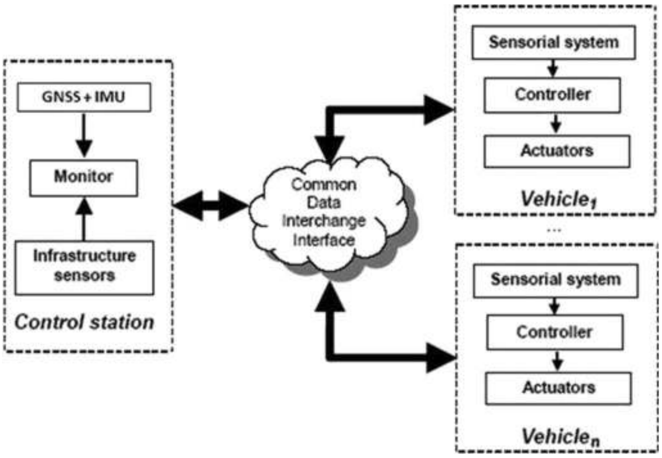

Fig. 1. IAI-CSIC car system architecture.

With respect to the sensors, an analog output attached to a servo-amplifier manages the motor using several classical and fuzzy controllers in a cascade architecture. The throttle is controlled with an analog signal that represents the pressure on the pedal, simulated with an analog card that receives the pressure value calculated by the pilot. Finally, the brake pedal is automated by means of an electrohydraulic braking system in parallel with the original braking circuit.

B. INRIA Cybercar Vehicle

A cybercar is a four-wheeled electric vehicle that has robotic abilities, i.e., it can drive fully autonomously (see Fig. 3). It has been designed to transport up to two persons at a maximum speed of 30 km/h [18].

The mechanics is borrowed from a small electrical golf car frame. Consequently, the architecture is modular, and the vehicle is easier to drive. The steering is performed by means of an electrical jack mechanically linked to all the wheels.

Each wheel motor block has its own power amplifier, driven by a microcontroller. This intelligent node consists of three linked layers. The lowest provides power to two motors. The middle layer permits sensor data acquisition and communication with the other nodes. The top layer, consisting of a Motorola MC68332 microcontroller well suited for motor control, drives the two others. Each wheel node controls the drive engine and a brake motor, with all their associated sensors (optical encoder, brake torque measurement, temperature, etc.). A fifth node is attached to the steering jack and the joystick. Communications between the nodes are established through a CAN serial bus (see Fig. 2).

This architecture can be split into two components: 1) the partial motion planner (PMP) block and its visualization module and 2) the Taxi application block, which is in charge of the vehicle's low-level control and coordinates the different information flows.

The GPS information received in each vehicle is processed by a central application named Taxi and sent to the PMP as if it corresponded to an obstacle. The PMP calculates a new

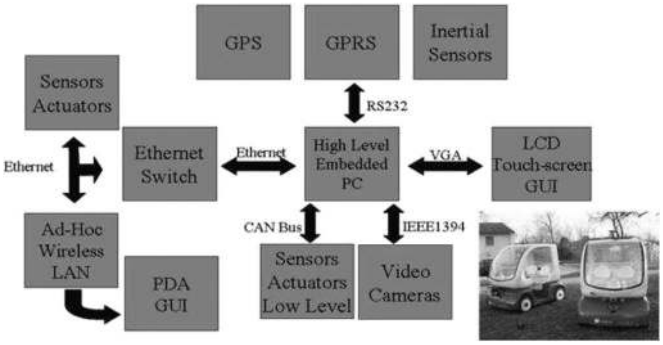

Fig. 2. INRIA Cybercar architecture.

Fig. 3. Cars participating in the experiment in La Place de Verdun, La Rochelle, France.

trajectory on the basis of this new obstacle information. When the vehicles are close to each other, the results generated by the PMP make the cybercar slow down as being the only possible way to remain safe in this situation, letting the second car continue its trajectory.

With respect to the bandwidth available for V2V communication, around 500 kb/s can be guaranteed when the vehicles are linked through the infrastructure node (static mesh cube), whereas around 2 Mb/s have been achieved through a direct connection between the vehicles.

C. TNO Dual-Mode Vehicle: AGS

To serve as a platform for automatic vehicle guidance (AVG) research [19], optionally combined with advanced driver assistance (ADA), TNO equipped a Smart Fortwo (see Fig. 3) with additional actuators and sensors. It is a standard Smart three-cylinder 600-cc production car with a weight of 795 kg (without additional equipment). This Smart has been used in several European Union (EU) projects, e.g., CarTALK2000 and Will-Warn, being gradually improved and extended over the years.

To describe the main system architecture of the Automatic Guided Smart (AGS) (see Fig. 4), the hardware will be considered first, followed by a short explanation on the human–machine interfaces, the description of the system architecture itself together with the safety system, and, finally, a short overview of the functionality of the AGS.

To operate the AGS as an automatic guided vehicle, several actuators were added or changed, and a number of additional sensors were installed. To carry out lateral control, an electric motor is mounted parallel to the steering shaft, allowing for automatic control of the steering wheel angle. Longitudinal control is managed by two actuators. First, an electronically controlled throttle has been implemented to allow automatic throttle actuation. Second, an entirely new braking system is installed (the DaimlerChrysler EHB system) for automatic control of the brake pressure. A second braking system (LeGrand) is installed as a backup system for reasons of safety. This latter system mechanically applies pressure to the brake pedal. Finally, the electronic gear shifting system has been modified to allow for gear shifts controlled by an external computer.

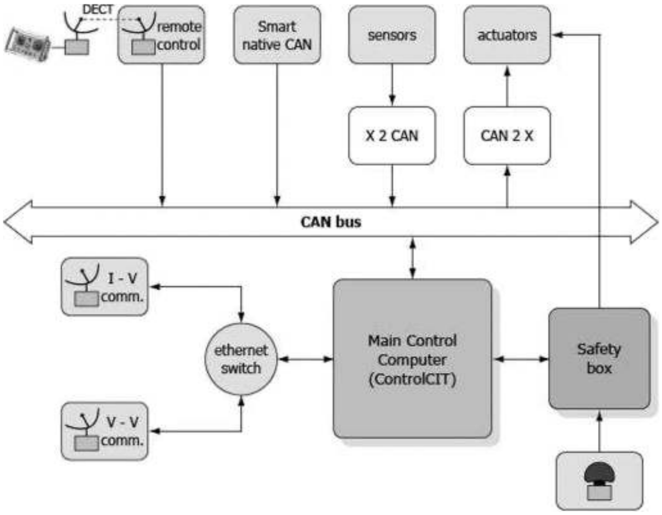

Fig. 4. TNO AGS architecture.

A Trimble RTK-DGPS is installed, which allows AVG due to its high precision. The AGS's position estimation and control algorithms are executed by the control computer of the intelligent transportation system. This is a low-cost modular rapid prototyping computer platform.

The aforementioned components interact with each other through a CAN bus system, as shown in Fig. 4. Emphasis is put on safety, which is why a separate "Safety box" is present that continuously checks the vehicle's operation and performs autonomous emergency actions if necessary via a direct link with the actuators.

All the sensors are connected via the CAN bus (actually, there are three CAN buses for reasons of system reliability). Some sensors, however, have no CAN interface, and therefore, vehicle interface components (CAN 2 X, X 2 CAN) are used to establish the CAN connection. Ideally, all actuators should also be CAN based, but for practical reasons, this is not fully implemented.

Using the components previously described, the AGS is capable of autonomously driving a route predefined in terms of position (and inherently velocity) as a function of time. One way of setting a desired trajectory and velocity profile is to manually drive the trajectory while recording the positions and the velocity profile. Then, using a playback function, the vehicle can repeat the trajectory.

III. COMMUNICATION ARCHITECTURE

An approach to an architecture model that enables cooperative communication between cybercars and dual-mode cars is presented in this section. This model provides interoperability and also permits cooperative maneuvers to safely be performed between these vehicles. The main consideration is of V2V communication, but some specific V2I communications between the vehicles and the complementary infrastructure elements are also taken into account.

The proposed Cybercars-2 cooperative architecture is based on the SAFESPOT scheme, which is characterized by a 3-D approach as follows:

1) Functional architecture: This comprises the functional modules and data structures interconnected by standard interfaces.
2) Physical architectures: These are related to the way in which the functionality can be implemented as applications to meet user needs. It involves translation from the functional modules into subsystems with the corresponding HW/SW components, operating systems, and implementation languages involved.
3) Communications architecture: This defines the links between the physical architecture components and the communication protocols.

To establish communication between the vehicles, a dynamic self-organized wireless mesh network has been set up as a vehicular ad-hoc network. All the vehicles involved have been equipped with specialized embedded hardware. Both V2V and V2I communications are based on Wi-Fi technologies.

The communications capability is integrated into the vehicles by means of a 4G access cube, i.e., a small reprogrammable Linux MIPS Box from 4G Systems. Each box has two wireless local area network cards (currently 802.11b), one of which is used for the mesh network, whereas the other acts as a base station. The base station function is important to monitor traffic performance from an infrastructure control and visualization unit without disturbing the mesh network and to achieve successful connections with common devices, such as personal digital assistants, Smart Phones, or any kind of wireless sensor.

Dynamic routing over the V2V physical link is achieved by using the optimized link state routing (OLSR) protocol contributed by INRIA [20]. The proactive behavior of the OLSR is appropriate for vehicle communications. Indeed, vehicle ad-hoc networks can quickly be created, and the impact of car association or disassociation to and from the network is minimized. Moreover, the OLSR is designed for multihop ad-hoc networks with a strong and efficient mechanism for data flow delivery to the entire network. Since handling large networks may be inappropriate for V2V communications, a series of modifications to OLSR has been implemented to limit the relaying of topology information and rapidly build small ad-hoc networks that are well suited to the dynamic context of vehicle communications at a crossing. Control information sent by a node (vehicle) is relayed within a restricted area to get the necessary routing and topology information of the nearby network. This information is sufficient to enable data relaying and broadcasting to an area that is larger than the nearby known topology.

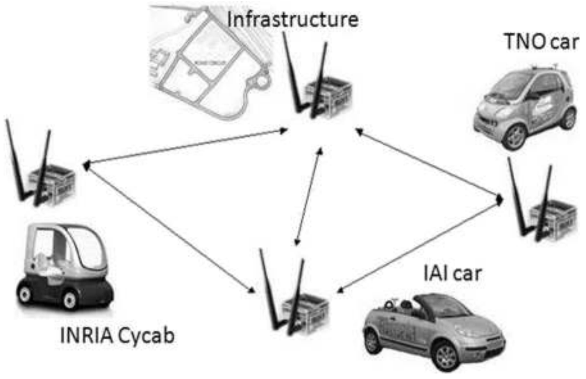

Fig. 5. V2V/V2I communication schema with 4G Access Cube.

The advantage of using mesh networks (that are based on Wi-Fi) over other wireless communication system is its capability to automatically rewrite the routing table of each node to guarantee connectivity between all the nodes to provide meshing.

The vehicles were linked via a mesh network over Wi-Fi using the OLSR ad-hoc routing protocol. The whole communication task was embedded in the 4G System Cube. During testing, GPS information from each car was periodically delivered to the mesh network. A static wireless mesh cube was added as an infrastructure relay node to enable communication between vehicles by relaying the forwarded messages when out of each other's range in terms of radio coverage (see Fig. 5).

The OLSR uses user datagram protocol (UDP) packages to test the link for discovering nodes, and it uses transmission control protocol (TCP) packages for user communications (service discovery, data exchange, etc.). The UDP messages manage route loss and route discovering. The OLSR is completely transparent for the user, and the user only deals with the TCP stack. In operation, an Ethernet cable is used to connect each vehicle with its cube. The main advantage in the use of the OLSR is that it immediately knows the status of the link. Moreover, it is possible to extend quality of service information to this protocol so that the hosts know in advance the quality of the route.

IV. Communication Data Structures

Information must be shared in order for vehicles to cooperate. Data have to continuously be exchanged to permit the cars to know what is in their environment. A data package was established that will be the same for both V2V and infrastructure-to-vehicle (I2V) communications. The structure of the package is presented in Table I.

The messages are sent either by the vehicles or by the infrastructure and are broadcast over the network. When the message originates from a vehicle, the Car Identifier is the name of the originator. When the message originates from the infrastructure, this field contains the name of the intended recipient. In the latter case, the Northing and Easting coordinates (in Universal Transverse Mercator (UTM) coordinates) are set to zero to permit messages originating from the infrastructure to be distinguished from those originating from cars.

TABLE I
COMMUNICATIONS DATA PACKAGE

Field	Size	Data Type
Car Identifier	10 B	Char
Timestamp	4 B	Float
Northing UTM Coordinate	8 B	Double
Easting UTM Coordinate	8 B	Double
Speed (cm/s)	8 B	Double
Heading (rad)	4 B	Float
Emergency Flag	1 B	Char

The infrastructure receives all the messages and analyzes them. Its main task is to monitor the route of each car and detect that its behavior is correct. If a V2V fault occurs or a possible mishap is detected, then the emergency flag is activated. In this case, the car or cars in danger are stopped. The cars will continue their route once the fault situation has been corrected.

In V2V communications, the Timestamp is the key to ensuring correct functioning of the system. The messages are sent at a 2-Hz rate. While this is a low frequency, the physical limitation of the circuit limited speeds to very low values, and tests showed that 500-ms sample time was sufficient. If the cars detect a correct Timestamp, then the control actions are permitted in accordance with the other data received. The maneuvers to perform are the following:

1) Adaptive Cruise Control (ACC): The equipped vehicle uses the GPS coordinates of the preceding car to maintain a preset safety headway. The control system automatically manages the car's throttle and brake. The minimum data update rate is 5 Hz, which is also the rate for the internal car control loop. The data needed to perform this maneuver are Northing, Easting, and Speed. The validity range for this data exchange is about 6 m since this is the maximum distance at which ACC is useful in the context of the circuit of La Place de Verdun (La Rochelle).
2) Stop & Go: This maneuver is considered under Cybercars-2 to be an extension of ACC, where a distance domain is used instead of the time domain to maintain a safety gap with the preceding vehicle, and whose extreme action is stopping, in a traffic jam, for example. All the constraints are the same as those in ACC.
3) Intersection: In this case, the data needed to perform the maneuver are Northing, Easting, Speed, and Heading. A hot area of ±6 m is defined around the intersection center. When a vehicle is occupying this hot area and another vehicle is at a distance of less than 6 m from the intersection approaching with lower priority, it will stop 1 m before the intersection, i.e., 2.5 m from the center of the intersection.
4) Emergency Stop: This infrastructure action forces any car to stop when it is considered necessary. The action will occur when a message is received with the emergency flag raised. The stopped car or cars will resume their route when a message is received with the emergency flag lowered.

Table II presents the requirements needed in each case.

TABLE II
COMMUNICATION REQUIREMENTS FOR EACH MANEUVER

	ACC	Stop&Go	Intersection	Emergency Stop
Type of Comm.	V2V	V2V	V2V	V2I
Transmission Mode	Periodic	Periodic	Periodic	Occasional
Frequency (Hz)	2	2	2	-
Required Fields	Northing, Easting, Speed	Northing, Easting, Speed	Northing, Easting, Speed, Heading	Emergency Flag

V. CONTROL ALGORITHMS

While each research group had been testing various complex algorithms for the intersections, to use a common procedure, those algorithms were simplified. For a one-way crossroad, the traffic code is as follows: The vehicle approaching the crossroad from the right has right of way. The vehicle approaching from the left is therefore stopped. Hence, two areas were defined. The first is the union of the crossroad area and the entrance to the crossroad from the right. If there is a car in this section of the loop, then an occupation signal will light up in the other cars of the loop. The second area is the entrance to the crossroad from the left. If there is a car in this area, then it will check whether the crossroad is occupied and, if so, will stop 1 m before the intersection.

This simple algorithm, in conjunction with ACC + Stop & Go procedures, guarantees safe circulation around the loop. In the case that the ACC + Stop & Go controller and crossroad controller are simultaneously activated, the latter has preference. The vehicle is stopped to guarantee safety. The particular form in which each car includes these directives in its own vehicle software does not reduce the system's reliability.

A. IAI Control Algorithm

Clavileño has been tested in different driving circuits, showing its ability to adapt itself to any environment [21], [22]. However, the experiments that were carried out in the present project required the use of different sets of controllers to confront different choices, in particular, ACC + Stop & Go, intersection maneuvers, and emergency stops.

Two control algorithms were developed. One was used to perform ACC + Stop & Go and is based on fuzzy logic. The other is an event control algorithm, since traversing a crossroad is a high-risk maneuver. Conditions for safety were defined such that, when they are satisfied, the car is stopped before entering the crossroad. Finally, the emergency stop was implemented as a constant high pressure on the brake pedal that permitted the car to be halted quickly.

In the case that the ACC rules are activated, the IAI-CSIC group uses fuzzy logic because it is a well-tested method for dealing with this kind of system, provides good results, and can incorporate human procedural knowledge into control algorithms. In addition, fuzzy logic lets us partially mimic human driving behavior. Min–Max have been selected to implement T-norm and T-conorm because of their computing ease. The

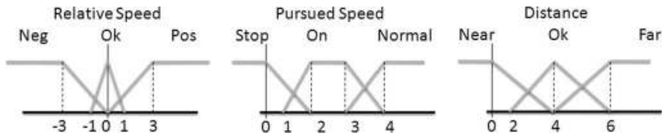

Fig. 6. IAI-CSIC's ACC fuzzy controller membership functions.

TABLE III
FUZZY CONTROLLER OUTPUT

Distance Equal to Near			
Relative/Pursued Speed	Neg	Ok	Pos
Stop	Brake b02	Brake b01	Throttle t00
On	Brake b01	Brake b01	Throttle t00
Normal	Brake b01	Brake b01	Throttle t00

Distance Equal to Ok			
Relative/Pursued Speed	Neg	Ok	Pos
Stop	Brake b01	Brake b00	Throttle t00
On	Brake b00	Brake b00	Throttle t01
Normal	Brake b00	Brake b00	Throttle t01

Distance Equal to Far			
Relative/Pursued Speed	Neg	Ok	Pos
Stop	Throttle t00	Throttle t01	Throttle t01
On	Throttle t00	Throttle t01	Throttle t01
Normal	Throttle t00	Throttle t01	Throttle t01

inference engine used is Mamdani type based on three input variables: 1) the relative speed determined as the speed difference between the pursued and the pursuer; 2) the pursued speed (both in kilometers per hour); and 3) the distance between the two cars obtained from the UTM coordinates (in meters)—Northing and Easting. Once having defined the inputs of the fuzzy controller, one must establish the linguistic values that each can take [23]. To this end, three membership functions—based on human drivers' experiences—were defined, as shown in Fig. 6. The output variables of this controller are the actuation on the throttle and the brake pedals that are defined as Sugeno's singletons [24], [25]. Thus, the singletons used for these two variables were restricted to t00, t01, and t02 for the throttle, and b00, b01, and b02 for the brake, representing normalized pressures on the pedal of 0 (no pressure), 0.1 (10% of pressure on the pedal), and 0.2 (20% of pressure on the pedal), respectively. The performance criteria are evaluated on the basis of the maximum error committed in a loop. The rule base is presented in Table III.

Finally, following the common algorithm for intersection directives, if the car is approaching the intersection in the left-hand side entrance, and there is another car approaching the intersection to the right, then the car is stopped 1 m before the intersection.

B. INRIA Control Algorithm

The control algorithm developed by the INRIA group for the guidance is explained in this section. It is based on PMP and applied for ACC + Stop & Go and crossroad safety control using communication (see Fig. 7). Once a coarse plan has been established, a specific motion has to be defined for the vehicle. This motion needs to define the state of the vehicle for the future time instants (Fig. 7, step 1). The sequence of desired states in time is called a trajectory. The vehicle has to consider its own limitations, the future movement of the other vehicles, and the obstacles detected by sensors or communication to guarantee the safe motion of the vehicle when computing the trajectory (Fig. 7, step 2). Partial planning takes place when the vehicle has limited visibility, and its plans can only reach a limited horizon. Since a wall (traffic jam, road blockage) can exist on the frontier of the unobserved area, all trajectories are required to stop before reaching the end of the visibility region. When the observed region is updated, the trajectory is also updated. Because of the partial nature of the provided trajectory, we call this approach partial motion planning [26]. Regarding safety, to ensure that the trajectory is feasible, the trajectory-generation strategy is based on a search in the commands space. Given an initial vehicle state (position, speed, steering), the control algorithm searches the set of commands that will allow it to best reach the goal. The model used to integrate the effect of a sequence of commands takes into account the saturation of the vehicle in steering and acceleration [26].

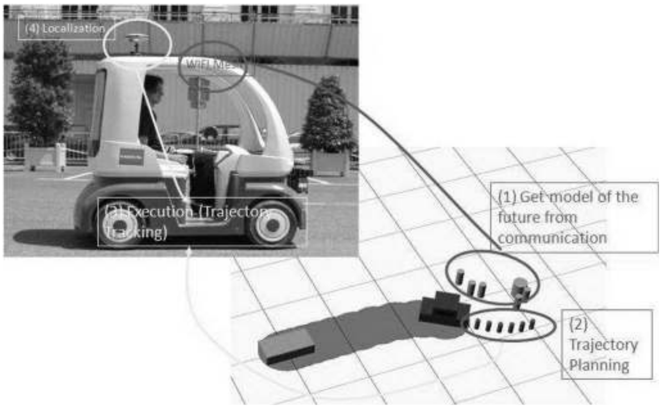
(4) Localization
WiFi Mes
(3) Execution (Trajectory Tracking)
(1) Get model of the future from communication
(2) Trajectory Planning

Fig. 7. INRIA PMP control step scheme.

In addition, for any given state of the partial trajectory, it is verified that the vehicle is capable of stopping without colliding. By doing so, we ensure that at any time the solution available will not actively cause the vehicle to collide. To provide this guarantee, it is necessary to use a conservative prediction of the vehicle's surroundings and have a good perception of the world. Here, it was done using only communication data as perception sensors.

C. TNO Control Algorithm

TNO's control software is developed in Matlab/Simulink. The Simulink block charts are converted into executable code and run in real time. Two blocks are in charge of the intersection.

The intersection decision block (see Fig. 8) determines whether the car is in the left intersection entrance area, whether there is another car in the intersection area, and the distance to the intersection.

The intersection decision flow chart block (see Fig. 9) was simplified, reducing it to two straightforward operations that can trigger a stop signal and determine the distance to the desired stop point. With this distance known, the speed can smoothly be varied to bring the car to a halt at the stop point 1 m before the intersection.

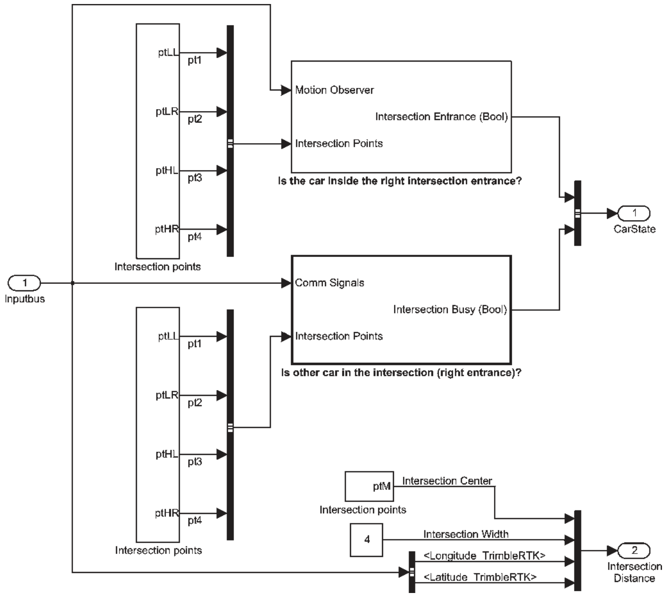

Fig. 8. TNO intersection decision subsystem.

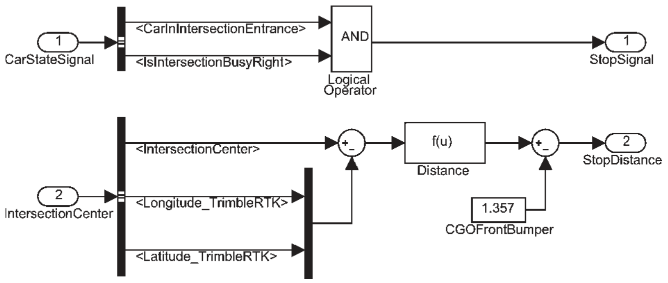

Fig. 9. TNO intersection decision flow chart.

As mentioned in Section II, the Smart is position controlled, much like a wheeled mobile robot. To this end, a so-called "virtual control point" V is defined ahead of the vehicle. The control objective is to let V follow a predefined position trajectory that is stored *a priori* in a table consisting of desired positions and velocities as a function of time. This was implemented using input–output linearization by time-invariant state feedback, which renders the system (partially) linear [19], after which simple PD controllers can be applied to achieve the desired position in the 2-D plane.

This approach, however, does not permit the type of vehicle following needed for ACC + Stop & Go since the desired trajectory has been defined beforehand. This is resolved by real-time scaling of the reference trajectory: Given the measured distance to the preceding vehicle and the desired user-defined time headway, a reference speed is calculated as the quotient of the two variables. This reference velocity is compared with the predefined reference velocity, leading to a scale factor. Finally, the time basis for the reference positions in the table is then scaled with the same factor.

VI. Experiments

For the test site, an open area was lent to the project by the La Rochelle (France) Town Council at La Place de Verdun. The circuit is a figure eight (see Fig. 10). The Place de Verdun's characteristics (such as the carousel, parking access, and trees) clearly determine the proportions of the track, with the upper loop having a 12-m external radius and the lower loop a 9-m external radius.

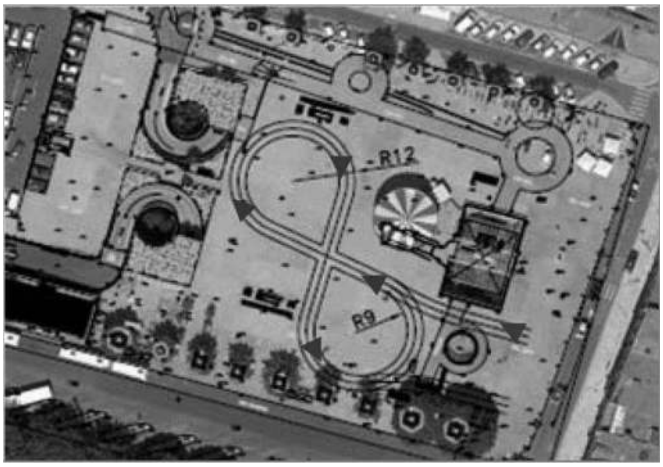

Fig. 10. Final demonstration driving circuit in La Rochelle (France).

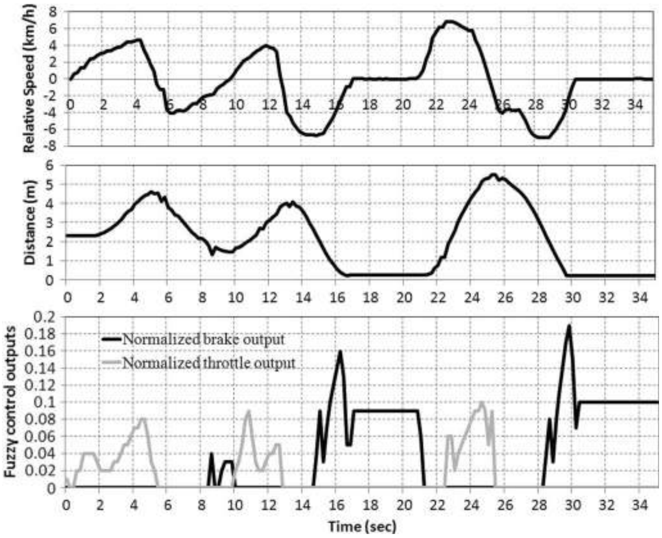

Fig. 11. ACC and Stop & Go experiment with the IAI-CSIC vehicle as trailing car and INRIA Cybercar as leading car.

The experiments were initiated with the cars stationary at different points on the circuit. A signal from the infrastructure (by simply dropping the emergency flag) started the cars moving. The experiments were carried out once all the cars were in motion.

Five experiments are shown. The first and the second show the behavior of the IAI-CSIC and TNO ACC + Stop & Go systems. The third and the fourth present the behavior in an intersection situation with the IAI-CSIC and INRIA systems. The fifth presents an emergency-stop experiment. Experiments involving three cars were not properly recorded, and the data cannot be shown in this paper.

Fig. 11 shows the record of an ACC and Stop & Go experiment done at La Rochelle on the circuit of Fig. 9. The test consisted of Clavileño following the INRIA Cybercar. The top plot shows the relative speed between the two cars. At a given moment, the leading vehicle stops, forcing the trailing car to

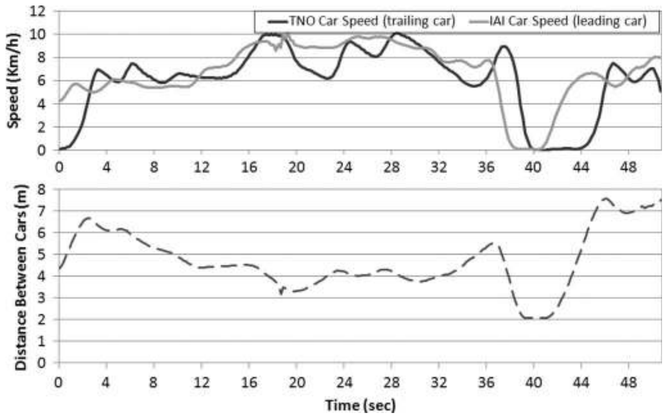

Fig. 12. ACC and Stop & Go experiment with the TNO Smart as trailing car and IAI-CSIC vehicle as leading car.

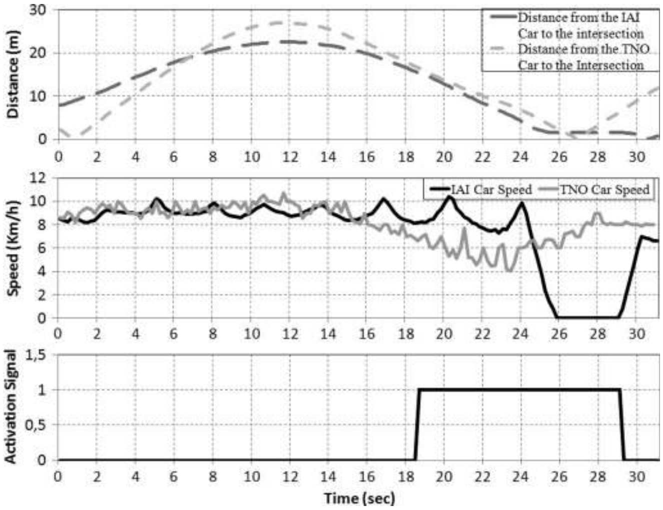

Fig. 13. Intersection experiment with TNO having the right of way and the IAI-CSIC vehicle arriving at the crossroad.

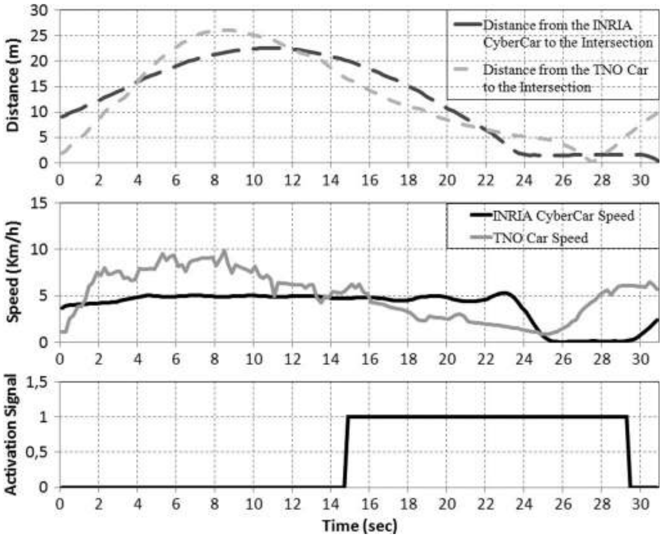

Fig. 14. Intersection experiment with TNO having right of way and the INRIA Cybercar arriving at the crossroad.

stop. The middle plot shows the distance between the cars. The periods during which this distance is perfectly constant are when they are stopped. Finally, the bottom plot shows the normalized output of the fuzzy controllers. The gray line shows the accelerator output, and the black line shows the brake output. Although, for ACC systems, one typically uses the time gap—defined as the distance between vehicles divided by the speed of the trailing car—at low speeds, this is not a reliable variable since small changes in the trailing car's speed cause large changes in the value of the time gap. This experiment showed that the variables defined—relative speed, pursued speed, and distance between vehicles—were capable of leading to acceptable performance.

Fig. 12 shows another ACC + Stop & Go experiment. In this case, the IAI-CSIC car was the leading vehicle, followed by the TNO Smart. The data record starts when TNO Smart began its movement. It speeds up to reduce the separation between the two vehicles, keeping the distance to around 4 m. The fluctuation in the speed of the gasoline TNO vehicle is due to the low speed. At second 38, the IAI-CSIC car stops, and 2 s later, the TNO car does the same to maintain a safe distance between the two cars. After 2 or 3 s, the IAI car resumes its trajectory, and the TNO car restarts to follow the leading vehicle.

Fig. 13 shows the record of an intersection experiment. The vehicles travel along the same circuit (see Fig. 9) at a very low speed under ACC (the black line shows the speed of the trailing car). The INRIA Cybercar is the leading vehicle, and the IAI-CSIC is the trailing vehicle. At the start of the plot, the trailing car has just passed the crossing point. It goes around the circumference and approaches the crossing point again. The cars' controllers are constantly monitoring the distance to the nearest crossing point, as reflected in the sinusoidal dashed lines: the darker gray line showing the IAI-CSIC vehicle's distance to the crossroad and the lighter gray line the distance of the vehicle, which has the right-of-way—the TNO Smart. At a certain moment timed to permit the INRIA Cybercar to pass through the intersection, the TNO vehicle heads toward the intersection point. The continuous gray line shows its speed. As can be seen, it reduces its speed so that it can occupy the hot area, allowing time for the trailing car to stop. The flat dotted gray line at the bottom of the figure represents the detection of a vehicle in the hot area of the intersection point. When this danger signal is activated, the car stops at the preset distance from the intersection and restarts after the danger has disappeared. The binary value "danger in crossing" is computed from the data received via the communications line.

Fig. 14 shows another intersection experiment. This time, an INRIA Cybercar and the TNO car were in the loop. They began their movement so that, at around second 4, they were at the same distance from the intersection, with the TNO car being manually driven with the immediate objective of reaching the intersection at the same time as the cybercar. The INRIA Cybercar was driven in autonomous mode. Arriving at the intersection at the same time as the TNO car, it stopped to allow the latter to pass, obeying the precedence rules (the TNO car having right of way). Once the TNO car left the intersection, the cybercar resumed its course.

In the figure, the dark gray dashed line represents the distance to the intersection measured by the INRIA Cybercar. The light

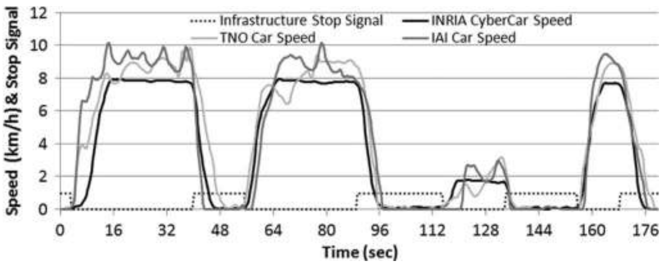

Fig. 15. I2V experiment with emergency stop signals sent from the central station.

gray dashed line represents the distance to the intersection of the TNO car. The dotted line represents the (binary) obstacle detection signal, its value being unity when the intersection is detected as being occupied by a vehicle reaching it with priority, so that other vehicles have to stop just in front of the intersection. The obstacle detection signal is reset to zero once the priority vehicle has left the intersection. When this signal was set to unity, the INRIA Cybercar had to stop upon reaching the entrance to the intersection. It resumes its coursed once the signal had returned to zero. The black and gray solid lines represent the speeds of the two cars involved in the maneuver.

Fig. 15 shows three cars in the loop being stopped by the infrastructure. Using special messages, the infrastructure had the capacity of ordering all the cars to stop or to resume their course. In these experiments, the three cars started at different positions in the loop. Once the central station detected a risk situation, the emergency stop signal was activated, and the three cars were stopped. With this experiment, the objective was to test the system's behavior with a sequence of emergency stop activations and deactivations. The experiment shown consisted of four examples of how the cars reacted when the infrastructure broadcast a stop signal.

VII. Conclusion and Future Works

We have presented the results of a cooperative study involving three European institutions working on automatic driving. As part of the Cybercar-2 project, the Informatique Mathématiques et Automatique pour la Route Automatisée group of INRIA (France), the TNO Automotive Division of TNO (The Netherlands), and IAI of the CSIC (Spain) cooperated in a project whose goal was to execute different cooperative maneuvers by means of shared communications. The results presented here were those of the final demonstration in La Rochelle (France).

A 4G Cube communications architecture was developed to manage V2V and V2I communications to permit cooperation among the vehicles involved. The results of the experiments showed that, given such a communications standard, three different vehicles with different architectures and different control systems can cooperate using the data they exchange and a common decision algorithm to perform complex cooperative maneuvers.

Five of the experiments were presented in this paper. The first and the second were of ACC with the Stop & Go extension based on V2V communication. They were carried out with two gasoline vehicles—TNO and IAI-CSIC. Not only were the vehicles able to drive at low speed in a closed environment, but they were also capable of following a leading vehicle with no problem. The third and the fourth involved the management of crossing an intersection and were also based on V2V communications. They showed that the INRIA Cybercar and the IAI-CSIC vehicle were able to halt in front of the intersection when another vehicle was approaching the crossroad with right of way. The fifth experiment tested the behavior of the V2I communication. Sequences of emergency stop signals were sent to the three vehicles, which demonstrated their ability to appropriately stop and resume their course.

Notwithstanding the small size of the demonstration circuit, all the architectures and communications demonstrated good behavior in all the tests performed, indicating that it should be feasible to straightforwardly increase the number of vehicles in the system, upgrading the communication architecture.

Although the results shown here are good, there is still a long way to go until they can be applied to commercial vehicles. To the best of our knowledge, the trend toward a safer transport system is to be based on communications—V2V and V2I—to prevent accidents or minimize their effects if they cannot be prevented. As future works, tests about incorporating or eliminating vehicles from the traffic flow as well as using circuits closer to the real-word traffic conditions will have to be conducted.

Acknowledgment

The authors would like to thank the EU Cybercars-2 project for its support in the development of this paper and the La Rochelle Town Council for its inestimable assistance in the execution of the experiments.

References

[1] [Online]. Available: http://www.esafetysupport.org

[2] G. Toulminet, J. Boussuge, and C. Laurgeau, "Comparative synthesis of the 3 main European projects dealing with Cooperative Systems (CVIS, SAFESPOT and COOPERS) and description of COOPERS Demonstration Site 4," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, Beijing, China, Oct. 2008, pp. 809–814.

[3] V. Singh, T. Singh, D. Langan, and P. Kumar, "A framework for Internet GIS based computerized visitor information system for theme parks," in *Proc. IEEE 11th Conf. Intell. Transp. Syst.*, Washington, DC, Oct. 2004, pp. 679–683.

[4] G. Vivo, P. Dalmasso, and F. Vernacchia, "The European integrated project 'SAFESPOT'—How ADAS applications co-operate for the driving safety," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Sep./Oct. 2007, pp. 624–629.

[5] J. Piao and M. McDonald, "Safety impacts of variable speed limits—A simulation study," in *Proc. IEEE 11th Conf. Intell. Transp. Syst.*, Beijing, China, Oct. 2008, pp. 833–837.

[6] J. P. Loewenau, W. Richter, C. Urbanczik, L. Beuk, T. Hendriks, R. Pichler, and K. Artmann, "MAPS & ADAS: Real time optimization of active cruise control with map data using," in *Proc. 12th ITS World Congr.*, San Francisco, CA, Nov. 2005.

[7] C. Toy, K. Leung, L. Álvarez, and R. Horowitz, "Emergency vehicle maneuvers and control laws for automated highway systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 2, pp. 109–119, Jun. 2002.

[8] C. Urmson, J. Anhalt, H. Bae, J. A. Bagnell, C. Baker, R. E. Bittner, T. Brown, M. N. Clark, M. Darms, D. Demitrish, J. Dolan, D. Duggins, D. Ferguson, T. Galatali, C. M. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. Howard, S. Kolski, M. Likhachev, B. Litkouhi, A. Kelly, M. McNaughton, N. Miller, J. Nickolaou, K. Peterson, B. Pilnick,

R. Rajkumar, P. Rybski, V. Sadekar, B. Salesky, Y. Seo, S. Singh, J. M. Snider, J. C. Struble, A. Stentz, M. Taylor, W. Whittaker, Z. Wolkowicki, W. Zhang, and J. Ziglar, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.—Special Issue on the 2007 DARPA Urban Challenge*, vol. 25, pt. I, no. 8, pp. 425–466, Jun. 2008.

[9] H. Kamiya, Y. Fujita, T. Tsuruga, Y. Nakamura, S. Matsuda, and K. Enomoto, "Intelligent technologies of Honda ASV," in *Proc. IEEE Intell. Veh. Symp.*, Tokyo, Japan, Sep. 1996, pp. 236–241.

[10] H. Watanabe, S. Kondo, and K. Hirano, "Introduction to Suzuki ASV technologies," in *Proc. IEEE Intell. Veh. Symp.*, Tokyo, Japan, Sep. 1996, pp. 219–223.

[11] F. Sugasawa, H. Ueno, M. Kaneda, J. Koreishi, R. Shirato, and N. Fukuhara, "Development of Nissan's ASV," in *Proc. IEEE Intell. Veh. Symp.*, Tokyo, Japan, Sep. 1996, pp. 254–259.

[12] [Online]. Available: http://www-c.inria.fr/cybercars2

[13] L. Bouraoui, S. Petti, A. Laouiti, T. Fraichard, and M. Parent, "Cybercar cooperation for safe intersections," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Toronto, ON, Canada, Sep. 2006, pp. 456–461.

[14] M. Parent, "Automated urban vehicles: State of the art and future directions," in *Proc. Int. Conf. Control Autom. Robot. Vis.*, Kunming, China, Dec. 2004, pp. 138–142.

[15] 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 Conf. Intell. Transp. Syst.*, Toronto, ON, Canada, Sep. 2006, pp. 1220–1225.

[16] J. E. Naranjo, L. Bouraoui, R. García, M. Parent, and M. A. Sotelo, "Interoperable control architecture for cybercars and dual-mode cars," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 146–154, Mar. 2009.

[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] N. Morette, C. Novales, and P. Vieyres, "Inverse versus direct kinematics model based on flatness and escape lanes to control Cycab mobile robot," in *Proc. IEEE Int. Robot. Autom. Conf.*, Pasadena, CA, May 2008, pp. 2240–2245.

[19] J. Ploeg, A. C. M. Van der Knaap, and D. J. Verburg, "ATS/AGV-design, implementation and evaluation of high performance AGV," in *Proc. IEEE Int. Veh. Symp.*, Versailles, France, Jun. 2002, pp. 127–134.

[20] L. Bouraoui, A. Fortelle, and A. Laouiti, "OLSR improvement for distributed traffic applications," in *Proc. IFIP Int. Federation Inf. Process.*, Île de Porquerolles, France, Jun. 2005, vol. 197/2006, pp. 73–77.

[21] J. E. Naranjo, C. González, R. Garcia, 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.

[22] V. Milanés, J. Pérez, E. Onieva, and C. González, "Controller for urban intersections based on wireless communications and fuzzy logic," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 243–248, Mar. 2010.

[23] L. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Inf. Sci.*, vol. 8, no. 3, pp. 199–249, 1975.

[24] 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, Feb. 1985.

[25] M. Sugeno, "On stability of fuzzy systems expressed by fuzzy rules," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 2, pp. 201–224, Apr. 1999.

[26] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *Proc. Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 3726–3731.

Vicente Milanés was born in Badajoz, Spain, in 1980. He received the B.E. and M.E. degrees in electronic engineering from Extremadura University, Badajoz, in 2002 and 2006, respectively, and the Ph.D. degree in electronic engineering from Alcala University, Madrid, Spain, in 2010.

Since 2006, he has been with the Industrial Computer Science Department (presently Centre of Automation and Robotics), Instituto de Automática Industrial, Madrid. His research interests include autonomous vehicles, fuzzy-logic control, and intelligent transportation systems.

Javier Alonso was born in Zaragoza, Spain, in 1975. He received the European Doctor degree in computer science from the Polytechnic University of Madrid, Madrid, Spain, in 2009.

Since 2001, he has been with the Consejo Superior de Investigaciones Científicas, presently the Centro de Automática y Robótica, Madrid. He is also a Software Specialist for automation projects. His fields of expertise are artificial intelligence, fuzzy logic, automatic guidance of vehicles, global navigation satellite system navigation, wireless communications, and artificial vision.

Laurent Bouraoui received the engineering degree from the French University "Paul Sabatier," Toulouse, France.

He was previously involved in developing small robots for military purposes with the THALES group. He is currently the Chief Technical Engineer of the of the R&D team on advanced road transport with the Informatique Mathématiques et Automatique pour la Route Automatisée (IMARA) Research Group, Institut National de Recherche en Informatique et Automatique, Le Chesnay, France. He coordinates the software development and integration for different European project like Cybercars2 and Citymobil. He also manages public technical demonstrations that involve IMARA autonomous vehicles in different topics like communication, navigation, and computer vision.

Jeroen Ploeg was born in 1964. He received the M.Sc. degree in mechanical engineering from Delft University of Technology, Delft, The Netherlands, in 1988. He is currently working toward the Ph.D. degree on the synchronization of road vehicles for cooperative driving with the Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands.

From 1989 to 1999, he was a Researcher with Koninklijke Hoogovens (currently Corus), Ijmuiden, The Netherlands, where his main interest was the dynamic process control of large-scale industrial plants. Since 1999, he has been a Senior Development Engineer with the Netherlands Organisation for Applied Scientific Research TNO, Helmond, The Netherlands. His current interests are focusing on control system design for advanced driver-assitance systems and for path tracking of automatic guided vehicles.