Using Fuzzy Logic in Automated Vehicle Control

José E. Naranjo, Carlos González, Ricardo García, and Teresa de Pedro, *Instituto de Automática Industrial*

Miguel A. Sotelo, *Universidad de Alcalá de Henares*

Automated versions of a mass-produced vehicle use fuzzy logic techniques to both address common challenges and incorporate human procedural knowledge into the vehicle control algorithms.

Until recently, in-vehicle computing has been largely relegated to auxiliary tasks such as regulating cabin temperature, opening doors, and monitoring fuel, oil, and battery-charge levels. Now, however, computers are increasingly assuming driving-related tasks in some commercial models. Among those tasks are

- maintaining a reference velocity or keeping a safe distance from other vehicles,
- improving night vision with infrared cameras, and
- building maps and providing alternative routes.

Still, many traffic situations remain complex and difficult to manage, particularly in urban settings. The driving task belongs to a class of problems that depend on underlying systems for logical reasoning and dealing with uncertainty. So, to move vehicle computers beyond monitoring and into tasks related to environment perception or driving, we must integrate aspects of human intelligence and behaviors so that vehicles can manage driving actuators in a way similar to humans.

This is the motivation behind the AUTOPIA program, a set of national research projects in Spain. AUTOPIA has two primary objectives. First, we want to implement automatic driving using real, mass-produced vehicles tested on real roads. Although this objective might be called "utopian" at the moment, it's a great starting point for exploring the future. Our second aim is to develop our automated system using modular components that can be immediately applied in the automotive industry.

AUTOPIA builds on the Instituto de Automática Industrial's extensive experience developing autonomous robots and fuzzy control systems and the Universidad de Alcalá de Henares's knowledge of artificial vision. (The "Intelligent-Vehicle Systems" sidebar discusses other such projects.) We're developing a testbed infrastructure for vehicle driving that includes control-system experimentation, strategies, and sensors. All of our facilities and instruments are available for collaboration with other research groups in this field. So far, we've automated and instrumented two Citroën Berlingo mass-produced vans to carry out our objectives.

Automated-vehicle equipment

Figure 1 shows two mass-produced electric Citroën Berlingo vans, which we've automated using an embedded fuzzy-logic-based control system to control their speed and steering. The system's main sensor inputs are a CCD (charge-coupled device) color camera and a high-precision global positioning system. Through these, the system controls the vehicle-driving actuators—that is, the steering, throttle, and brake pedals. Both vehicles include an onboard PC-based computer; a centimetric, real-time kinematic differential GPS (RTK DGPS); Wireless LAN support; two servomotors; and an analog/digital I/O card. We added a vision system in another computer connected to the control computer. Figure

1541-1672/07/$25.00 © 2007 IEEE
Published by the IEEE Computer Society

According to Ernst Dickmanns, it's likely to be at least 20 years before we'll apply intelligent transportation systems to road transportation and achieve the ultimate goal of full autonomous vehicle driving.[1] The Stardust final report assumes that ITS progress toward this goal will be rooted in adaptive cruise control.[2]

In 1977, Sadayuki Tsugawa's team in Japan presented the first intelligent vehicle with fully automatic driving.[3] Although that system was limited, it demonstrated autonomous vehicles' technical feasibility, and Tsugawa's group continues its autonomous vehicle work today, and many other international projects have also been launched, including the following efforts in Europe and the US.

European projects

In Europe, researchers completed autonomous-vehicle developments such as VaMoRs (advanced platform for visual autonomous road vehicle guidance) and Argo as part of the Prometheus program (the Program for European Traffic with Highest and Unprecedented Safety).

As part of the VaMoRs project, Ernst Dickmanns' team at the Universität der Bundeswehr developed an automatic-driving system using artificial vision and transputers in the late 1980s. The team installed the system in a van with automated actuators and ran a range of automatic-driving experiments, some of which were on conventional highways at speeds up to 130 kmh. Projects such as VAMP (the advanced platform for visual autonomous road vehicle guidance) are continuing this work today.[4]

Alberto Broggi's team developed the ARGO vehicle at Parma University.[5] In 1998, the team drove ARGO 2,000 km along Italian highways in automatic mode using artificial-vision-based steering.

In France, projects such as Praxitele and "La route automatisée" focus on driving in urban environments, as do the European Union's Cybercars and CyberCars-2 projects. Another European project, Chauffeur, focuses on truck platoon driving.

US projects

California's PATH (Partners for Advanced Transit and Highways)[6] is a multidisciplinary program launched in 1986. Its ultimate goal is to solve the state's traffic problems by totally or partially automating vehicles. PATH uses special lanes for automatic vehicles, which will circulate autonomously (guided by magnetic marks) and form platoons.

Carnegie Mellon University's Navlab has automated 11 vehicles since 1984 to study and develop autonomous-driving techniques. In 1995, Navlab #5 carried out the No Hands across America experiment, a trip from Washington D.C. to California along public highways in which artificial vision and a neural network control system managed the vehicle's steering wheel.

The University of Arizona's Vista (Vehicles with Intelligent Systems for Transport Automation) project started in 1998 to conduct intelligent-vehicle research and develop technology for vehicle control.[7] Since 2000, the project has been cooperating with the Chinese Academy of Sciences, whose intelligent-transportation-system activities include research[8] and managing the National Field Testing Complex.[9] In 2004, Darpa decided to test automatic-vehicle technology by organizing the Darpa Grand Challenge. The experiment consisted of a set of activities for autonomous vehicles, with 25 out of 106 groups selected to participate.[10] Only 14 groups qualified for the final, which was a race of approximately 320 km; the winner made it only 12 km.[11]

A second edition was held on 8 October 2005 in the US. The Stanford Racing Team won, with a winning time of 6 hours, 53 minutes. In all, five teams completed the Grand Challenge course, which was 132 miles over desert terrain. Darpa has announced a third Grand Challenge, set for 3 November 2007. In it, participants will attempt an autonomous, 60-mile route in an urban area, obeying traffic laws and merging into moving traffic.

References

1. E. Dickmanns, "The Development of Machine Vision for Road Vehicles in the Last Decade," *Proc. IEEE Intelligent Vehicle Symp.*, vol. 1, IEEE Press, 2002, pp. 268–281.
2. *Sustainable Town Development: A Research on Deployment of Sustainable Urban Transport (Stardust), Critical Analysis of ADAS/AVG Options to 210, Selection of Options to Be Investigated*, European Commission Fifth Framework Programme on Energy, Environment, and Sustainable Development Programme, 2001.
3. S. Kato et al., "Vehicle Control Algorithms for Cooperative Driving with Automated Vehicles and Intervehicle Communications," *IEEE Trans. Intelligent Transportation Systems*, vol. 3, no. 3, 2002, pp. 155–161.
4. U. Franke et al., "Autonomous Driving Goes Downtown," *IEEE Intelligent Systems*, vol. 13, no. 6, 1998, pp. 40–48.
5. A. Broggi et al., *Automatic Vehicle Guidance: The Experience of the ARGO Autonomous Vehicle*, World Scientific, 1999.
6. R. Rajamani et al., "Demonstration of Integrated Longitudinal and Lateral Control for the Operation of Automated Vehicles in Platoons," *IEEE Trans. Control Systems Technology*, vol. 8, no. 4, 2000, pp. 695–708.
7. F.Y. Wang, P.B. Mirchandani, and Z. Wang, "The Vista Project and Its Applications," *IEEE Intelligent Systems*, vol. 17, no. 6, 2002, pp. 72–75.
8. N.N. Zheng et al., "Toward Intelligent Driver-Assistance and Safety Warning System," *IEEE Intelligent Systems*, vol. 19, no. 2, 2004, pp. 8–11.
9. F.Y. Wang et al., "Creating a Digital-Vehicle Proving Ground," *IEEE Intelligent Systems,* vol. 18, no. 2, 2003, pp. 12–15.
10. Q. Chen, U. Ozguner, and K. Redmill, "Ohio State University at the Darpa Grand Challenge: Developing a Completely Autonomous Vehicle," *IEEE Intelligent Systems*, vol. 19, no. 5, 2004, pp. 8–11.
11. J. Kumagai, "Sand Trap," *IEEE Spectrum*, June 2004, pp. 34–40.

2 shows the control system that we developed to handle all these devices.

The computer drives the vans using two fuzzy-logic-based controllers: the steering (lateral) control and the speed (longitudinal) control. To automate the steering, we installed a DC servomotor in the steering wheel column. The Berlingo has an electronic throttle control, so we shortened the electronic circuit to actuate the throttle using an analog output card. The brake pedal is fully mechanical; we automated it using a pulley and a DC servomotor. We equipped the transmission with an electronic gearbox with forward and reverse selection. We automated this using a digital

Figure 1. The AUTOPIA testbed vehicles. An embedded fuzzy-logic-based control system controls both speed and steering in each Citroën Berlingo.

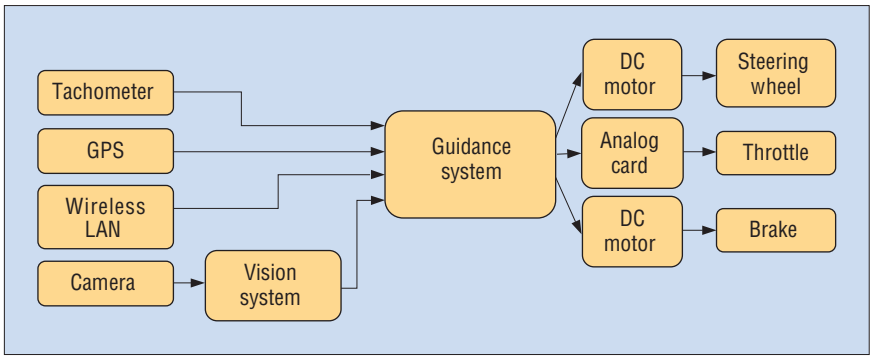


Figure 2. The AUTOPIA system control structure. The sensorial equipment supplies the necessary data to the fuzzy-logic-based guidance system, which decides the optimal control signals to manage the vehicle actuators (steering wheel, throttle, and brake).

I/O card that sends the correct gear to the internal vehicle computer. We designed our driving area to emulate an urban environment because automatic urban driving is one of ITS's less researched topics.

Guidance system

We modeled the guidance system using fuzzy variables and rules. In addition to the steering wheel and vehicle velocity functionalities, we also consider variables that the system can use in adaptive cruise control (ACC) and overtaking capabilities. Among these variables are the distance to the next bend and the distance to the lead vehicle (that is, any vehicle driving directly in front of the automated vehicle).

Car driving is a special control problem because mathematical models are highly complex and can't be accurately linearized. We use fuzzy logic because it's a well-tested method for dealing with this kind of system, provides good results, and can incorporate human procedural knowledge into control algorithms. Also, fuzzy logic lets us mimic human driving behavior to some extent.

Steering control

The steering control system's objective is to track a trajectory. To model lateral and angular tracking deviations perceived by a human driver, we use two fuzzy variables: Lateral_Error and Angular_Error. These variables represent the difference between the vehicle's current and correct position and its orientation to a reference trajectory.

Both variables can take left or right linguistic values. Angular_Error represents the angle between the orientation and vehicle velocity vectors. If this angle is counterclockwise, the Angular_Error value is left. If the angle is clockwise, the Angular_Error value is right. Lateral_Error represents the distance from the vehicle to the reference trajectory. If the vehicle is positioned on the trajectory's left, the Lateral_Error value is left; it's right if the vehicle is on the right.

We compute the variables' instantaneous value using the DGPS data and a digital environment map. The fuzzy output variable is Steering_Wheel and indicates which direction the system must turn the steering wheel to correct the input errors. Again, the variable also has left and right linguistic values. The value is left if the steering wheel must turn counterclockwise, and right if it must turn clockwise. We define the fuzzy sets that define the left and right values in an interval of –540 degrees and 540 degrees.

As with human behavior, our guidance system works differently for tracking lanes or turning on sharp bends. When traveling along a straight road, people drive at relatively high speeds while gently turning the steering wheel. In contrast, on sharp bends, they rapidly reduce speed and quickly turn the steering wheel. We emulate such behavior by changing the membership function parameters of the Lateral_Deviation, Angular_Deviation, and Steering_Wheel linguistic variables. To represent the human procedural knowledge in the driving task, we need only two fuzzy rules. These rules tell the fuzzy inference motor how to relate the fuzzy input and output variables:

IF Angular_Error left OR Lateral_Error left THEN Steering_Wheel right
IF Angular_Error right OR Lateral_Error right THEN Steering_Wheel left

Although these rules are simple, they generate results that are close to human driving. The rules are the same for all situations, but the definition of the fuzzy variables' linguistic values change. Figure 3 shows this feature in the membership function definition for Lateral_Error and Angular_Error. Figures 3a and 3b show the degree of truth for the input error values in straight-path tracking situations. This definition lets the system act quickly when trajectory deviations occur—again in keeping with human behavior.

To prevent accidents, we must limit the maximum turning angle for straight-lane driving. This limitation is also similar to human behavior; we achieve it by defining the output variable membership function as a singleton, confining this turning to 2.5 percent of the total. Figure 3c and 3d show similar function definitions, but their shape's gradient is lower. This makes the driving system less reactive when tracking a straight trajectory and assures that they'll adapt to the route smoothly. We can also represent the output

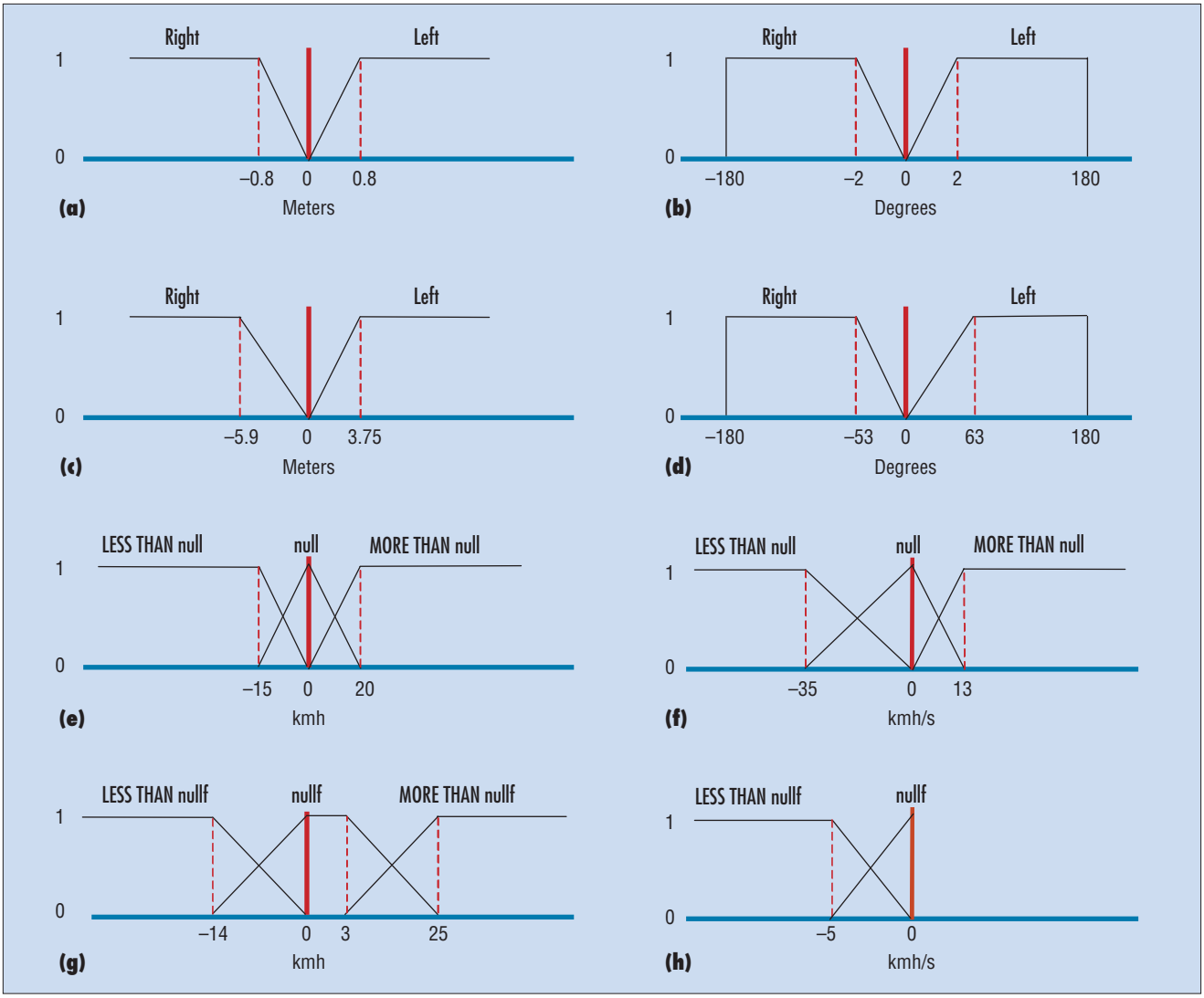


Figure 3. The membership function definition for fuzzy variables: (a) Lateral_Error straight, (b) Angular_Error straight, (c) Lateral_Error curves, (d) Angular_Error curves, (e) Speed_Error throttle, (f) Acceleration throttle, (g) Speed_Error brake, and (h) Acceleration brake.

using a singleton without turning limitations. We fine-tuned the membership functions experimentally, comparing their behavior with human operations and correcting it accordingly until the system performed acceptably. So, the driving system selects a fuzzy membership function set depending on the situation, which leads to different reactions for each route segment.

Speed control

To control speed, we use two fuzzy input variables: Speed_Error and Acceleration. To control the accelerator and the brake, we use two fuzzy output variables: Throttle and Brake. The Speed_Error crisp value is the difference between the vehicle's real speed and the user-defined target speed, and the Acceleration crisp value is the speed's variation during a time interval. The throttle pressure range is 2–4 volts, and the brake pedal range is 0–240 degrees of the actuation motor.

The fuzzy rules containing procedural knowledge for throttle control are

IF Speed_Error MORE THAN null THEN Throttle up
IF Speed_Error LESS THAN null THEN Throttle down
IF Acceleration MORE THAN null THEN Throttle up
IF Acceleration LESS THAN null THEN Throttle down

The rules for brake control are

IF Speed_Error MORE THAN nullf THEN Brake down
IF Speed_Error LESS THAN nullf THEN Brake up
IF Acceleration LESS THAN nullf THEN Brake up

where brake/throttle down means depress the brake and throttle, and brake/throttle up means release the brake and throttle. The associated membership functions of the fuzzy linguistic labels null and nullf define the degree of nearness to 0 of Acceleration and Speed_Error, respectively.

Figures 3e through 3h show the membership functions of null (for the throttle controller) and nullf (for the brake controller) for Speed_Error and Acceleration, respectively. An asymmetry exists in the two variable definitions for two reasons:

- to account for the difference in how accelerating and braking vehicles behave, and

Figure 4. The adaptive cruise control + Stop&Go controller's performance in an automated vehicle. Keeping a safe distance (a) at 30 kmh, (b) during speed reduction, and (c) in stop-and-go situations.

- to coordinate both pedals' actuation to emulate human driving.

Throttle and brake controllers are independent, but they must work cooperatively. Activating the two pedals produces similar outcomes and can

- increase the target speed (stepping on the throttle or stepping off the brake on downhill roads),
- maintain speed (stepping on or off either pedal when necessary), and
- reduce the vehicle's speed (downshifting the throttle or stepping on the brake.

ACC+Stop&Go

With ACC, the system can change the vehicle's speed to keep a safe distance from the lead vehicle. As an extreme example, the lead vehicle might come to a complete stop owing to a traffic jam. In this case, the ACC must stop the vehicle using a stop-and-go maneuver; when the road is clear, the ACC reaccelerates the vehicle until it reaches the target speed. Combining ACC with stop-and-go maneuvers increases driving comfort, regulates traffic speed, and breaks up bottlenecks more quickly. Many rear-end collisions happen in stop-and-go situations because of driver distractions.

ACC systems have been on the market since 1995, when Mitsubishi offered the Preview Distance Control system in its Diamante model. Several sensors can provide the vehicle with ACC capability: radar, laser vision, or a combination thereof. Almost all car manufacturers now offer ACC systems for their vehicles, but they all have two clear drawbacks. First, the ACC systems don't work at speeds lower than 40 kmh, so they can't offer stop-and-go maneuvers. Second, the systems manage only the throttle automatically; consequently, the speed adaptation range is limited.

Our system overcomes these limitations by automating the throttle and brake, which lets the system act across the vehicle's entire speed range. In our case, we selected GPS as the safety-distance sensor. We installed GPS in both vehicles, and they communicate their position to one another via WLAN.

Keeping a user-defined safety distance from the next vehicle is a speed-dependent function: the higher the speed, the larger the required intervehicle gap. This is the *time-headway concept*—a time-dependent safety distance maintained between two vehicles. If we set a safety time gap of two seconds, for example, the space gap is 22.2 meters for a vehicle moving at 40 kmh but approximately 55.5 meters for 100 kmh. The time gap setting depends on the vehicle's braking power, the weather, the maximum speed, and so on. For example, Article 54 of the Spanish Highway Code states,

> The driver of a vehicle trailing another shall keep a distance such that he or she can stop his or her vehicle without colliding with the leading vehicle should this brake suddenly, taking into account speed, adherence and braking conditions.

Figure 4 shows our ACC+Stop&Go controller's performance in one of our automated vehicles. At the experiment's beginning, the trailing vehicle starts moving, speeds up, and eventually stops because the lead vehicle is blocking the way. The lead vehicle then starts moving, gains speed, and brakes again, emulating a congested traffic situation. A few seconds later, the trailing vehicle starts up again, eventually stopping behind the lead vehicle. Figure 5a shows the system using the throttle and brake to continuously control the distance and time headway. Figure 5b shows that the trailing vehicle maintains its distance even if the speed is low and the time headway is no longer significant. When the lead vehicle starts up, the trailing vehicle follows, respecting the time headway (see figure 5c). Finally, figure 5d shows each vehicle's speed profile, which indicates the cars' behavior in relation to pedal pressure.

Overtaking

The system can also manage obstacles or other vehicles in the vehicle's path by calculating when the vehicle should change lanes to overtake the (mobile or immobile) obstacle. First,

- the vehicle must be in the straight-lane driving mode,
- the left lane must be free, and
- there must be room for the overtaking.[2]

Given this, overtaking occurs as follows:

1. Initially, the vehicle is in straight-lane mode.
2. The driving mode changes to lane-change mode, and the vehicle moves into the left lane.
3. The driving mode changes to straight-lane mode until the vehicle has passed the obstacle or vehicle.
4. The driving mode again changes to lane-change mode, and the vehicle returns to the right lane.

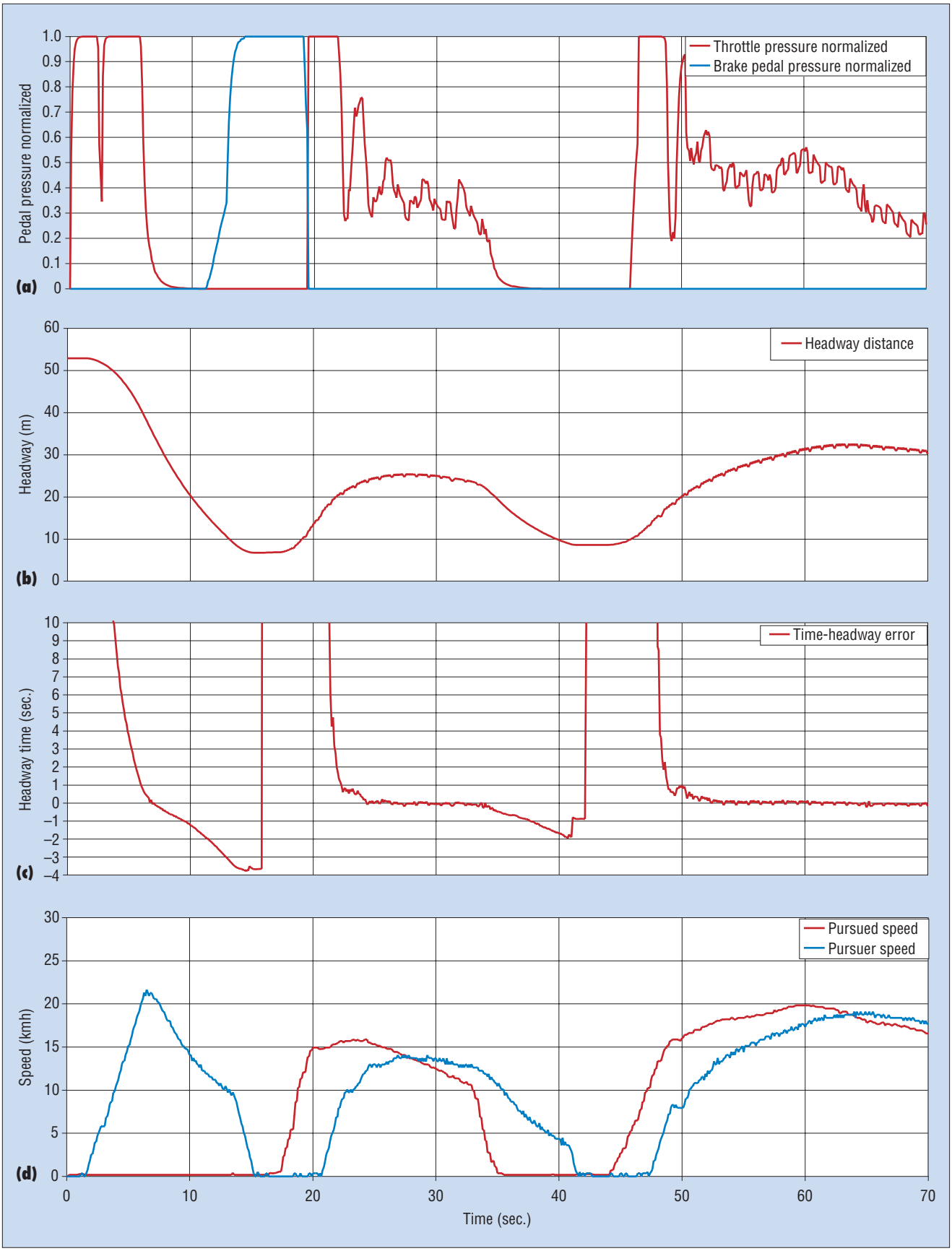


Figure 5. Performance measures: (a) normalization of throttle and brake pressure, (b) headway distance, (c) time-headway error, and (d) each vehicle's speed.

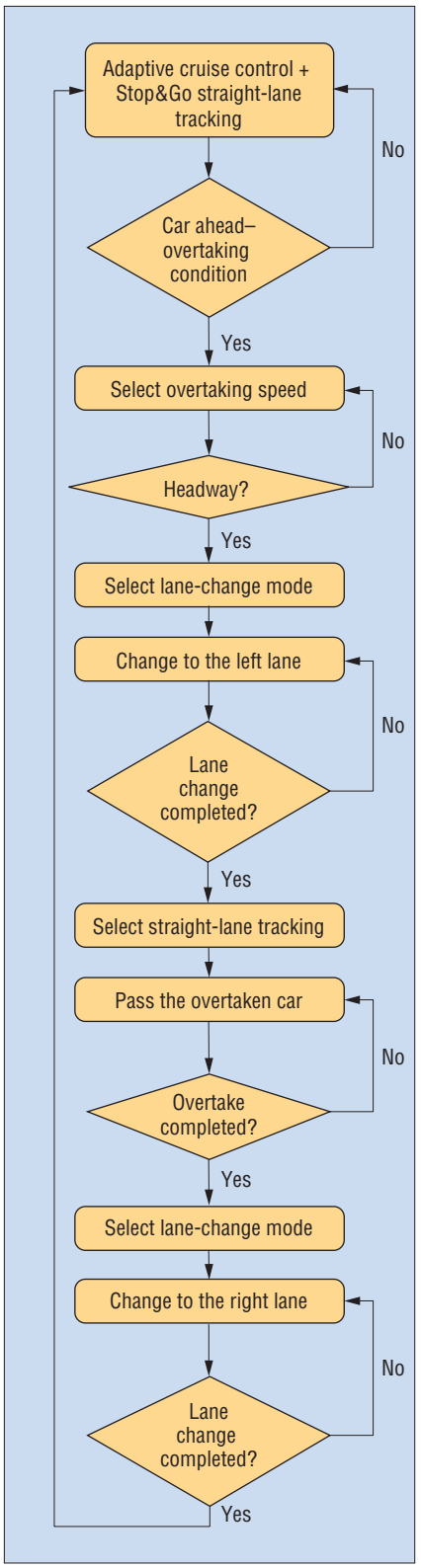


Figure 6. A flow chart of the overtaking algorithm.

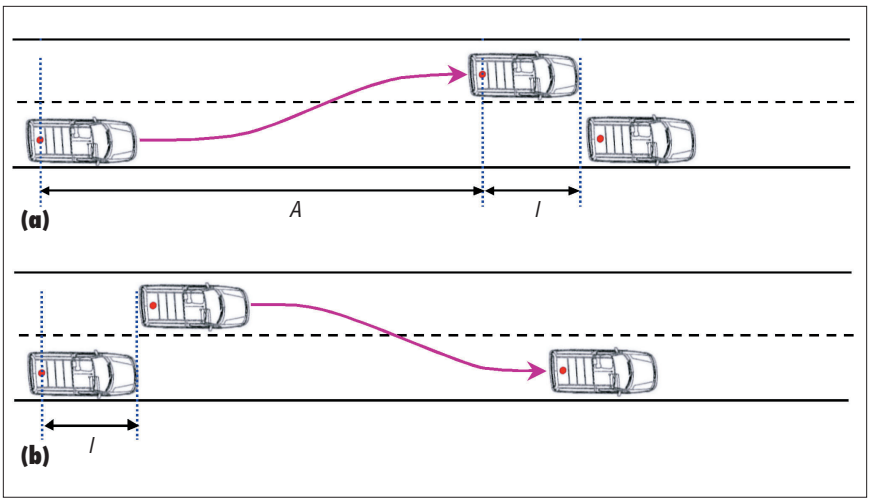


Figure 7. The overtaking maneuver (a) starts with a change from the right to the left lane and (b) ends with a change from the left to the right lane. *A* is the distance at which the vehicle changes lanes, and *l* is the vehicle's length.

5. When the vehicle is centered in the lane, the driving mode changes back to straight-lane mode, and driving continues as usual.

Figure 6 shows a detailed flowchart of this algorithm. We calculate the time for starting the transition from the first to the second step as a function of the vehicles' relative speed and the overtaking vehicle's length. Figure 7 illustrates the overtaking maneuver. The overtaking vehicle must change lanes at point $A + l$, where A is the distance at which the vehicle changes lanes and l is the vehicle's length. The dot on the back of each vehicle represents a GPS antenna, located over the rear axle. Vehicles use the GPS receptor and the WLAN link to continuously track their own position and that of other vehicles. The lane change proceeds only if the front of the overtaking vehicle is completely in the left lane upon reaching the rear of the overtaken vehicle in the right lane.

A is speed dependent—$A = F(v)$, where v is the relative speed between the overtaking and overtaken vehicles because the higher the velocity, the larger the lane-change distance. A is a function of the relative speed between both vehicles because overtaking depends on the two mobile objects' speed. In this case, l is 4 meters, a Citroën Berlingo's length.

The system transitions from step 2 to step 3 when the overtaking vehicle's angular and lateral errors are both low. Specifically, Angular_Error must be less than 2 degrees and Lateral_Error less than 0.8 meter. The system transitions to step 4 when the overtaking vehicle's rear end passes the overtaken vehicle's front end and the separation is l (see figure 7b). Finally, the transition to step 5 is the same as from steps 2 to 3.

Vision-based vehicle detection

To achieve reliable navigation, all autonomous vehicles must master the basic skill of obstacle detection. This vision-based task is complex. Consider, for example, common situations in urban environments, such as missing lane markers, vehicles parked on both sides of the street, or crosswalks. All such situations make it difficult for a system to reliably detect other vehicles, creating hazards for the host vehicle. To address this, we use a monocular color-vision system to give our GPS-based navigator visual reactive capacity.

Search and vehicle detection

We sharply reduce execution time by limiting obstacle detection to a predefined area in which obstacles are more likely to appear. This rectangular area—or region of interest (ROI)—covers the image's central section.

To robustly detect and track vehicles along the road, we need two consecutive processing stages. First, the system locates vehicles on the basis of their color and shape properties, using vertical edge and color symmetry characteristics. It combines this analysis with temporal constraints for

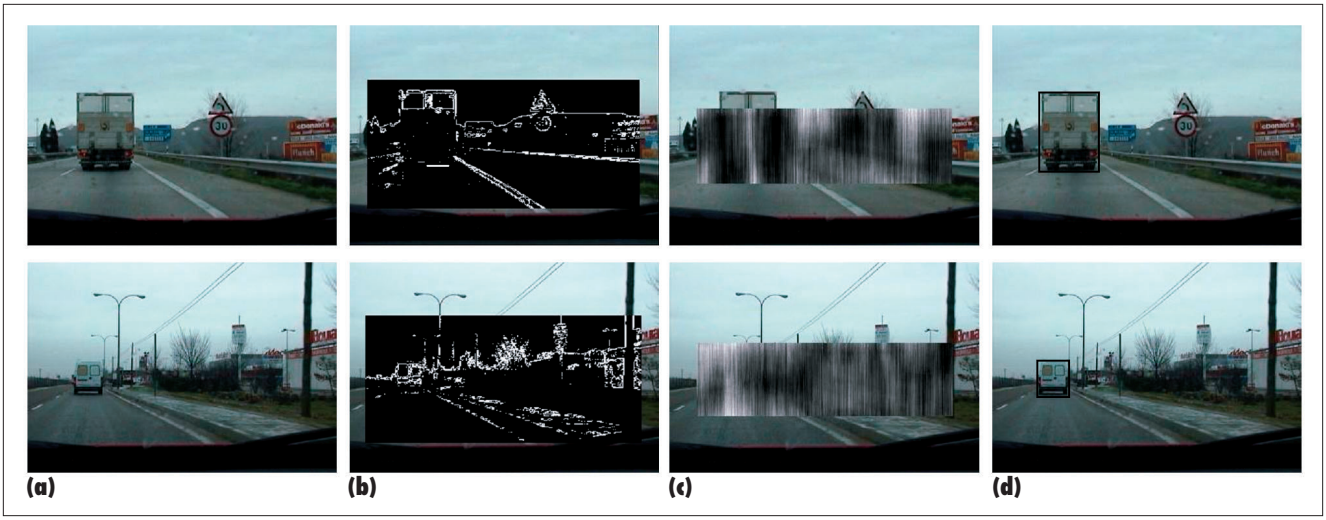


Figure 8. Two vehicle detection examples: (a) the original image, (b) the image with region-of-interest edge enhancement, (c) a vertical symmetry map, and (d) the detected vehicle's position.

consistency, assuming that vehicles generally have artificial rectangular and symmetrical shapes that make their vertical edges easily distinguishable. Second, the system tracks the detected vehicle using a real-time estimator.

Vertical-edge and symmetry discriminating analysis

After identifying candidate edges representing the target vehicle's limits, the system computes a symmetry map[3] of the ROI to enhance the objects with strong color symmetry characteristics. It computes these characteristics using pixel intensity to measure the match between two halves of an image region around a vertical axis. It then considers the vertical edges of paired ROIs with high symmetry measures (rejecting uniform areas). It does this only for pairs representing possible vehicle contours, disregarding any combinations that lead to unrealistic vehicle shapes.

Temporal consistency

In the real world, using only spatial features to detect obstacles leads to sporadic, incorrect detection due to noise. We therefore use a temporal-validation filter to remove inconsistent objects from the scene.[4] That is, the system must detect any spatially interesting object in several consecutive image iterations to consider that object a real vehicle; it discards all other objects.

We use the value $t = 0.5s$ to ensure that a vehicle appears in a consistent time sequence. A major challenge of temporal-spatial validation is for the system to identify the same vehicle's appearance in two consecutive frames. To this end, our system uses the object's (x, y) position in correlative frames. That is, it can use the position differences to describe the vehicle's evolution in the image plane. At time instant t_0, the system annotates each target object's (x, y) position in a dynamic list, and starts a time count to track all candidate vehicles' temporal consistency. At time $t_0 + 1$, it repeats the process using the same spatial-validation criterion. We increase the time count only for those objects whose distance from some previous candidate vehicles is less than d_v. Otherwise, we reset the time count. A candidate object is validated as a real vehicle when its time count reaches $t = 0.5s$.

Given that the vision algorithm's complete execution time is 100 ms, an empirical value $d_v = 1m$ has proven successful in effectively detecting real vehicles in the scene. Figure 8 shows examples of the original and filtered images along with the ROI symmetry map and the detected vehicle's final position.

Vehicle tracking

We track the detected vehicle's position using position measurement and estimation. We use the detected vehicle's ROI image as a template to detect position updates in the next image using a best-fit correlation. We then use the vehicle's (x, y) location in data association for position validation. Basically, we want to determine whether any object in the current frame matches the vehicle being tracked. To do this, we specify a limited search area around the vehicle position, leading to fast, efficient detection. We also establish a minimum correlation value and template size to end the tracking process if the system obtains poor correlations or if the vehicle moves too far away or leaves the scene.

Next, we filter the vehicle position measurements using a recursive least-squares estimator with exponential decay.[5] To avoid partial occlusions, the system keeps the previously estimated vehicle position for five consecutive iterations—without calculating any validated position—before considering the vehicle track as lost. Given a loss, the system stops vehicle tracking and restarts the vehicle detection stage. Figure 9 illustrates our algorithm, showing how the system tracked the lead vehicle in real traffic situations.

Adaptive navigation

After detecting the lead vehicle's position, we must ensure safe navigation in ACC mode if the lead vehicle suddenly brakes within the safety gap limits. This event could easily lead to a crash unless the host vehicle rapidly detects the braking situation and brakes hard. To ensure this, the system must detect the lead vehicle's brake light activation, which clearly indicates braking.

A vehicle's brake light position varies depending on its model and manufacturer. So,

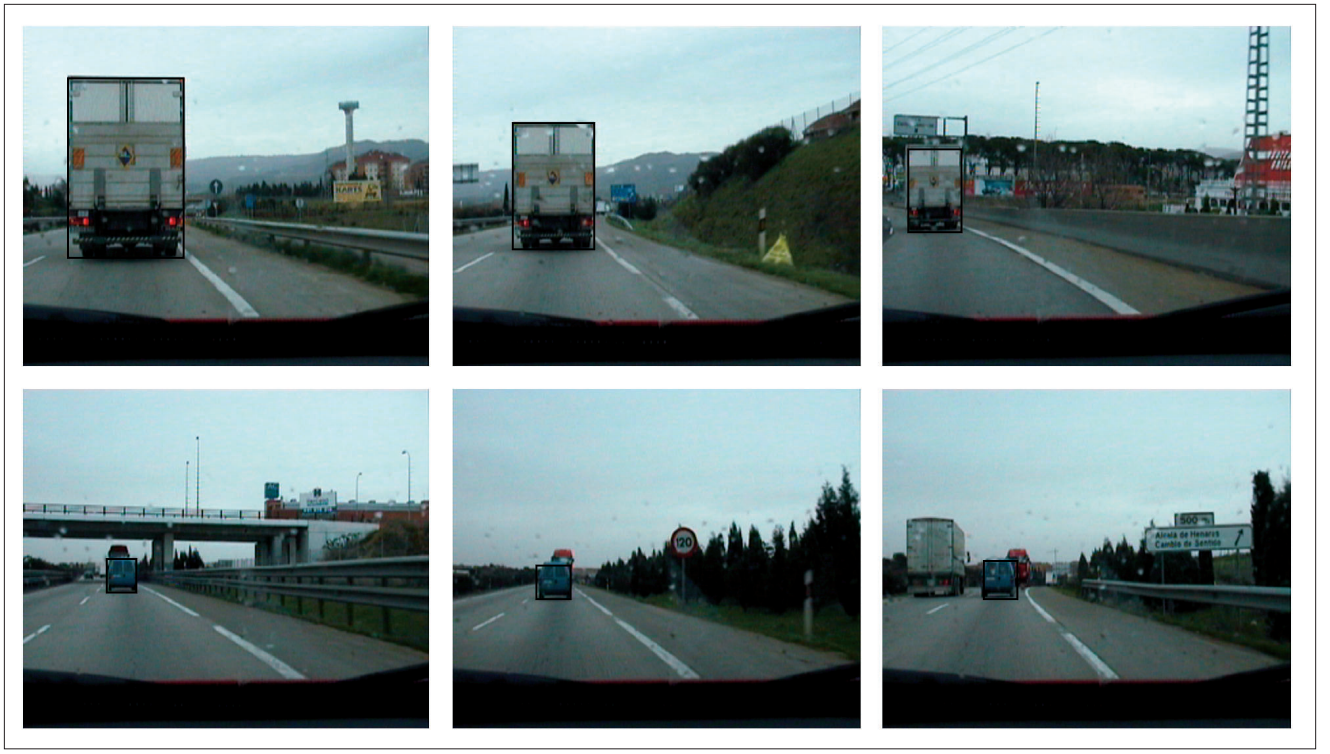


Figure 9. Vehicle tracking in two real-world traffic situations.

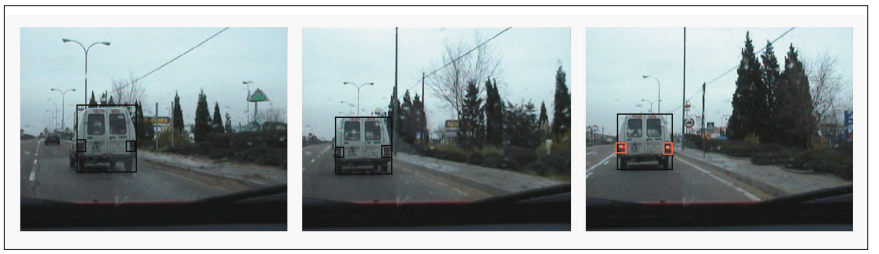


Figure 10. Sudden-braking detection. Once the system locates the brake lights, it continuously monitors their luminance to detect brake-light activation.

the system must carry out a detailed search to accurately locate these lights inside the vehicle's ROI. We do have some a priori information to ease the search: brake indicators are typically two red lights symmetrically located near the vehicle's rear left and right sides. Once the system locates these lights, it must detect sudden brake light activation; it does this by continuously monitoring the lights' luminance. In case of sudden activation, the system raises an alarm to the vehicle navigator to provoke emergency braking.

Figure 10 shows an example of sudden-braking detection. Brake lights are a redundant safety feature: if they're activated, a braking procedure has already started. Fortunately, our system continuously computes the distance to the lead vehicle. If this distance is too short, it automatically stops the vehicle.

We carried out all of our experiments with real vehicles on real roads, albeit within a private circuit. The results show that the fuzzy controllers perfectly mimic human driving behavior in driving and route tracking, as well as in more complex, multiple-vehicle maneuvers, such ACC or overtaking. In the near future, we're planning to run new experiments involving three automatic driving cars in more complex situations, such as intersections or roundabouts.

Fuzzy control's flexibility let us integrate a host of sensorial information to achieve our results. Also, using vision for vehicle and obstacle detection lets the host vehicle react to real traffic conditions, and has proven a crucial complement to the GPS-based navigation system. To improve and further this work, we're collaborating with other European institutions specializing in autonomous vehicle development under the UE Contract CyberCars-2. Through this collaboration, we plan to perform a cooperative driving involving more than six vehicles, adding new sensors for pedestrian detection, traffic-sign detection, and infrastructure monitoring. We'll also integrate new wireless communication systems that include vehicle-to-vehicle, vehicle-to-infrastructure, and in-vehicle information transmission. Finally, we're planning to use Galileo and GPS-2, next-generation GPS systems that address some existing GPS positioning problems and improve location accuracy. ■

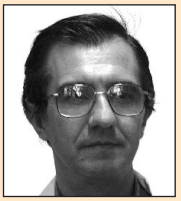
References

1. M.A. Sotelo et al., "Vehicle Fuzzy Driving Based on DGPS and Vision," *Proc. 9th Int'l Fuzzy Systems Assoc.*, Springer, 2001, pp. 1472–1477.

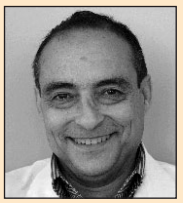
The Authors

José E. Naranjo is a researcher in the Industrial Computer Science Department at the Instituto de Automática Industrial in Madrid. His research interests include fuzzy logic control and intelligent transportation systems. He received his PhD in computer science from the Polytechnic University of Madrid. Contact him at Instituto de Automática Industrial (CSIC), Ctra. Campo Real Km. 0,200 La Poveda, Arganda del Rey, Madrid, Spain; jnaranjo@iai.csic.es.

Miguel A. Sotelo is an associate professor in the University of Alcalá's Department of Electronics. His research interests include real-time computer vision and control systems for autonomous and assisted intelligent road vehicles. He received Spain's Best Research Award in Automotive and Vehicle Applications in 2002, the 3M Foundation awards in eSafety in 2003 and 2004, and the University of Alcalá's Best Young Researcher Award in 2004. He received his PhD in electrical engineering from the University of Alcalá. He's a member of the IEEE and the IEEE ITS Society. Contact him at Universidad de Alcalá de Henares, Departamento de Electrónica, Madrid, Spain; michael@depeca.uah.es.



Carlos González is a software specialist in automation projects at the Instituto de Automática Industrial. He received his PhD in physics from Madrid University and is an IEEE member. Contact him at Instituto de Automática Industrial (CSIC), Ctra. Campo Real Km. 0,200 La Poveda, Arganda del Rey, Madrid, Spain; gonzalez@iai.csic.es.



Ricardo García is research professor and founder of the Consejo Superior de Investigaciones Científicas' Instituto de Automatica Industrial, where he works in intelligent robotics. His AUTOPIA project earned the 2002 Barreiros Research on Automotive Field Prize. He received his PhD in physics from Bilbao University. Contact him at Instituto de Automática Industrial (CSIC), Ctra. Campo Real Km. 0,200 La Poveda, Arganda del Rey, Madrid, Spain; ricardo@iai.csic.es.

Teresa de Pedro is a researcher at the Consejo Superior de Investigaciones Científicas' Instituto de Automatica Industrial, where she works on AI as applied to automation and leads the ISAAC (Integration of Sensors to Active Aided Conduction) project. Her research interests include fuzzy models for unmanned vehicles. She received her PhD in physics from the Universidad Complutense of Madrid. Contact her at Instituto de Automática Industrial (CSIC), Ctra. Campo Real Km. 0,200 La Poveda, Arganda del Rey, Madrid, Spain; tere@iai.csic.es.

2. R. Garcia et al., "Frontal and Lateral Control for Unmanned Vehicles in Urban Tracks," *IEEE Intelligent Vehicle Symp.* (IV2002), vol. 2, IEEE Press, 2002, pp. 583–588.
3. A. Broggi et al., "The Argo Autonomous Vehicle's Vision and Control Systems," *Int'l J. Intelligent Control and Systems*, vol. 3, no. 4, 2000, pp. 409–441.
4. M.A. Sotelo, *Global Navigation System Applied to the Guidance of an Terrestrial Autonomous Vehicle in Partially Known Outdoor Environments*, doctoral dissertation, Univ. of Alcalá, 2001.
5. H. Scheneiderman and M. Nashman, "A Discriminating Feature Tracker for Vision-Based Autonomous Driving," *IEEE Trans. Robotics and Automation*, vol. 10, no. 6, 1994, pp. 769–775.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.