Proceedings of the IEEE ITSC 2006
2006 IEEE Intelligent Transportation Systems Conference
Toronto, Canada, September 17-20, 2006

AUTOPIA Architecture for Automatic Driving and Maneuvering

José E. Naranjo, Carlos González, *Member, IEEE*, Teresa de Pedro, Ricardo García, Javier Alonso, Miguel A. Sotelo, *Member, IEEE*, and David Fernández

Abstract— **Cybercars and dual mode vehicles are presently the most innovative testbeds for vehicular automation applications. The definition of standards and control architectures of the different automatic vehicle onboard systems is a necessary task to build a final prototype to be produced. Several classical architecture definitions have been made in the field of mobile robotics. These architectures are capable of dealing with sensorial inputs and environment and procedural knowledge to manage the different actuators of mobile robots in order to accomplish their missions. Autonomous vehicles are conceived as a link between mobile robotics and the field of vehicular technology, obtaining cars that may be as autonomous as a mobile robot but circulating in high demand environments and in different conditions, as compared to robots. In this paper we present the control architecture used in AUTOPIA program, used for automating mass produced cars. This architecture is to deal with sensorial information and wireless communication as main sensorial input and manages the three fundamental actuators in a car: throttle, brake and steering wheel. The final aim of this architecture is to cover an automatic driving system that can manage a set of maneuvers of a car in the same way human drivers do. At this moment, straight circulation, curve circulation, adaptive cruise control, stop and go and overtaking maneuvers are available and research continues in order to increment its number.**

I. INTRODUCTION

As it is well known, the field of autonomous vehicles could be derived from autonomous robots and, consequently, the control schemas applied to these robots are applicable to cars. Logically, the subject of application and circulation environment contains substantial differences but general architectures are directly applicable. This way, the three historical mobile control architecture paradigms [1], hierarchical/deliberative [2], reactive [3] and hybrid [4] are applicable to autonomous vehicles, with some modifications.

From these paradigms, some autonomous vehicle architectures have been proposed.

In the Cybercars EU project, the Sharp architecture [5] is used. This is a hybrid three layer one (planner, mission scheduler and motion controller). The main contribution of this architecture is the sensor based maneuver (SBM) concept. It consists on adding to the system the possibility of combining planning and acting reactively in some low basic maneuvers, based only on the sensorial information and without planning. This architecture has also been extended to cooperative driving [6] among a set of cybercars.

In the California PATH project, a four layer hierarchy architecture is proposed in [7]. These layers are network, link, planning and regulation and they decompose complex maneuvers in a set of more simple and handy ones. A fault tolerance extension has been included later in this architecture [8].

Another example of vehicle control architecture is the CMU Navlab [9] hybrid one, adapted directly from the wide experience of this university on mobile robots. This architecture has a planning stage, with strategic and tactical layers and a low level layer that includes some behavior skills, specific for each circulation situation.

In this paper we present the control architecture developed in the Autopia Program for autonomous vehicle control. The function of this architecture is dual. On one hand it defines the internal control structure for each vehicle in order to provide individual autonomous driving capabilities. On the other hand the architecture must support the cooperation among a set of vehicles, equipped with the aforementioned individual architecture. This architecture also has to be open and scalable, even with the inclusion of different elements in each car.

Manuscript received February 24, 2006. This work was supported in part by the Spanish Ministry of Education under Grant ISAAC2 CICYT DPI200507980-C03. J. E. Naranjo, C. González, R. García, T. de Pedro and J. Alonso are with the Instituto de Automática Industrial (CSIC), Ctra. Campo Real Km. 0,200, 2850, Madrid, Spain (phone: +34 91 8711900; fax: +34 91 8717050; e-mail: jnaranjo@ iai.csic.es).

M. A. Sotelo and D. Fernández are with the Department of Electronics. Escuela Politécnica Superior, University of Alcalá, Alcalá de Henares, Madrid, Spain (e-mail: michael@depeca.uah.es).

II. AUTOMATIC DRIVING CONTROL ARCHITECTURE

The goal of AUTOPIA is to develop of a set of automated vehicles that can be automatically driven in a closed circuit. In order to do this, it is necessary to define a general architecture, common to all vehicles, capable of dealing with different vehicle models, actuators and control methods. This general architecture should be distributed and it has to allow scalability without substantial changes in its configuration. In our case, we only deal with three autonomous vehicles, two Citroën Berlingo named Babieca and Rocinante, and a Citroén C3 named Clavilenyo (fig. 1), but the architecture design may manage an undetermined number of cars.

Fig. 1. Image of Clavilenyo automated vehicle in an autonomous trajectory. This is a Citroën C3 Pluriel whose actuators has been automated and equips an onboard computer and some sensors.

This general architecture uses the schema defined in fig. 2, where a set of independent autonomous vehicles are linked among them and with a central monitoring station, sharing the necessary information to cooperate and perform human-like maneuvers.

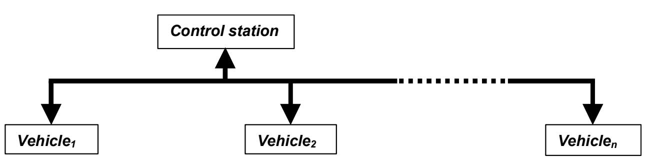

Fig. 2. Schematic of the AUTOPIA general architecture.

The main novelty of the developed architecture is the transparency about the kind of car that is added to the automatic driving environment. This way, each vehicle incorporates its own driving system, uses its own sensorial inputs and acts over its own actuators. However, all the vehicles are equipped with a similar configuration. Each car has a centimetric GPS which is used as the main sensorial input. They also read speed and acceleration information from the vehicle, the steering angle, the actual gear and the pressure performed over the pedals. The available information is the same for the three cars but the sensorial source is different depending on each model. For example, in the Berlingos, the speed signal is acquired reading an analog signal from the vehicle's tachometer but, in the C3, this information is read directly from the vehicle CAN bus.

There are also some specific sensors installed in the vehicles. This way the Rocinante vehicle uses a laser scanner for detecting obstacles in front of it [10], Babieca includes an artificial vision camera to detect the road borders [11] and Clavilenyo equips a stereo vision system, used to detect pedestrians and road obstacles.

The second component of the figure 2 schema is a central station. This station deals with the information supplied from a GPS base receiver and the available infrastructure sensors. Its main mission is to broadcast differential correction information from the GPS base and the important information from the sensors (fig. 3) as well as the relevant data supplied by the infrastructure, emergency or whatever other signal.

All these components are linked through a wireless LAN (WLAN) to share information among them almost in real time. In order to do this, it is necessary to define a common data interchange interface that will be the same for all the components of the architecture. This way, all the vehicles may be individually different but all of them "speak" the same language so they can share the required information to circulate cooperatively.

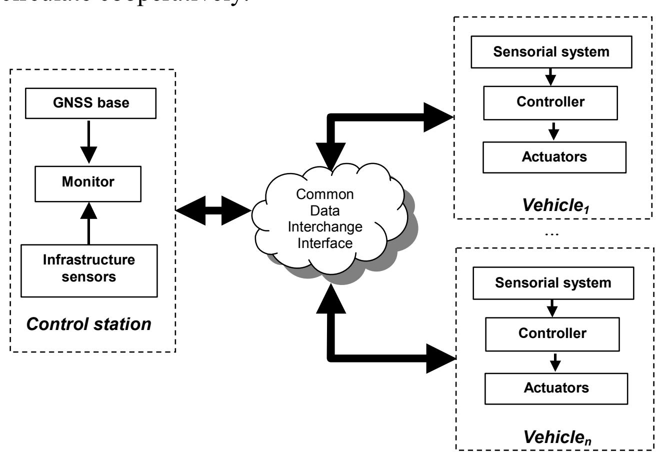

Fig. 3. AUTOPIA general architecture.

III. AUTOMATIC DRIVING SYSTEM ARCHITECTURE

Once the general architecture that covers the cooperative driving of a set of autonomous vehicles has been defined, we will present the individual architecture installed in each vehicle. In fig 3, this architecture is represented as a three layer hierarchy classical one: a sensorial layer where the input signals are acquired from the sensors, a control layer that decides the actions to be taken based in this input data and an actuation layer that acts upon the vehicle control elements. Fig 4 represents the schema of this architecture for the autonomous vehicles.

At present moment, the sensorial layer uses as main input the data generated by its own centimetric GPS, that receives via WLAN with differential correction information and is used to locate the car within the driving zone. Speed, acceleration, steering wheel turning angle and other variables are also available sensorial information. However, this layer is open and ready to include new sensorial information from different sensors. In order 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 through the WLAN environment. With this information, each car is able to know the position and direction of each car in real time and to take the appropriate actions to circulate cooperatively. This WLAN interface is open and available to transmit any other information that would be necessary, depending on the degree of cooperation desired 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 a human-like driving, for example the traffic code.

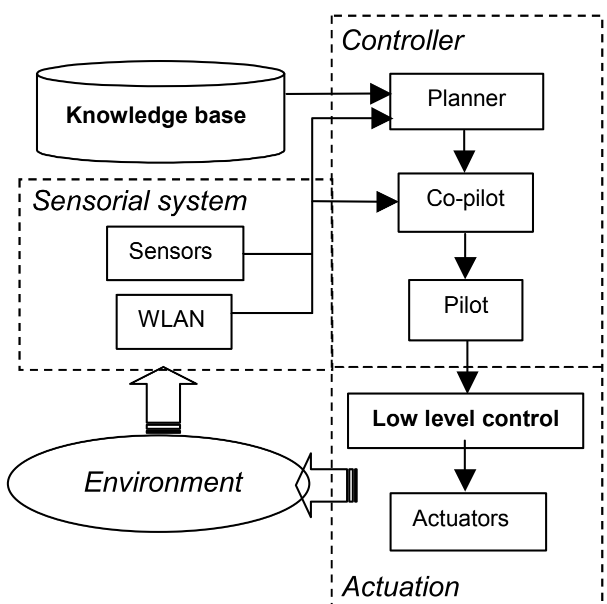

Fig. 4. Individual boarded control architecture for each AUTOPIA autonomous vehicle.

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, that will be sent to the actuators. This layer is divided into three sequential stages: planner, the co-pilot and the pilot. From a general point of view, the co-pilot aim is to follow a route and the pilot mission is to always maintain the car into the reference lane.

A. The planner

It is the highest level in the control layer. In our work we have included this stage in order to add the possibility of planning the route to follow by the vehicle in an intelligent way. 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 that form polynomial lines, used as reference by the control system to track the route.

B. The Co-pilot.

The name of this stage has been chosen because its function is very similar to the mission of a rally co-pilot. It indicates to the low level layer the maneuver to execute next, selecting the driving mode that must run in each moment, based on the route information from the planner and the sensorial inputs. This way, the copilot chooses among the set of available maneuvers, to follow a straight road, to follow a curve road, to overtake, and to do ACC, or stop&go. These maneuvers are complex and they are decomposed in a sequence of more simple ones by the co-pilot, and will be executed by the next stage. For example, the overtaking is divided in a lane change to the left lane of the road, a straight circulation through the left lane until surpassing the overtaken vehicle and a second lane change to the right lane to return to the normal circulation [12].

The copilot processes, interprets the scene, decides which maneuver to do and which low-level controller/s has to be activated. It decides which speed has to be taken as reference at each moment depending the route circumstances and deviations and also does the map matching between the reference route supplied by the planner and the real time GPS coordinates, generating the corresponding deviation data.

C. The 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 co-pilot and it is able to generate an output signal that can be applied to the vehicle actuators.

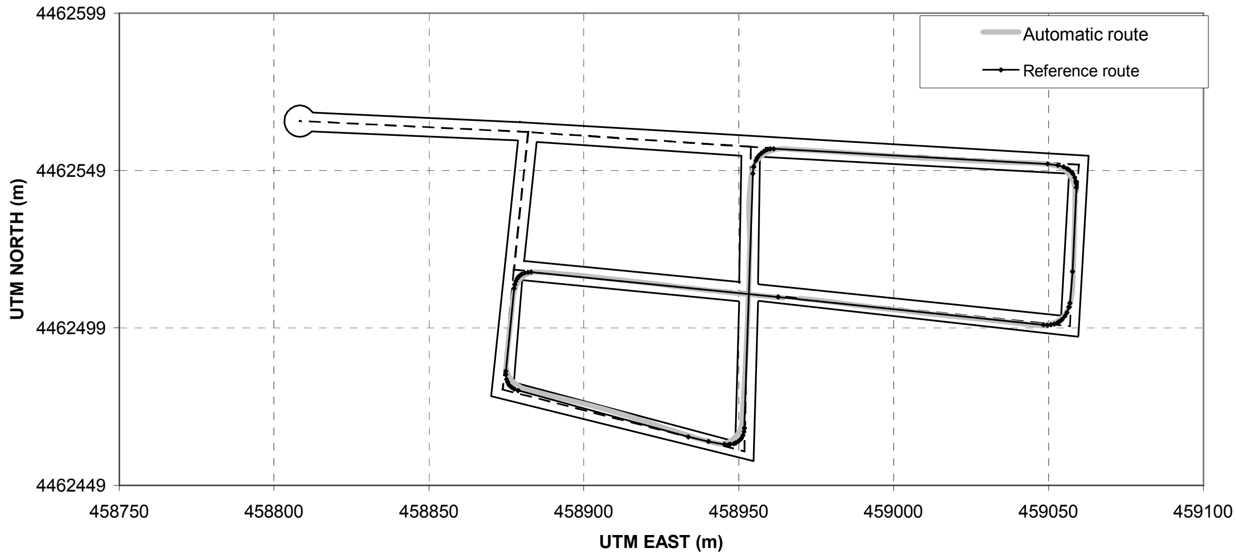

Fig. 5. Automatic route representation.

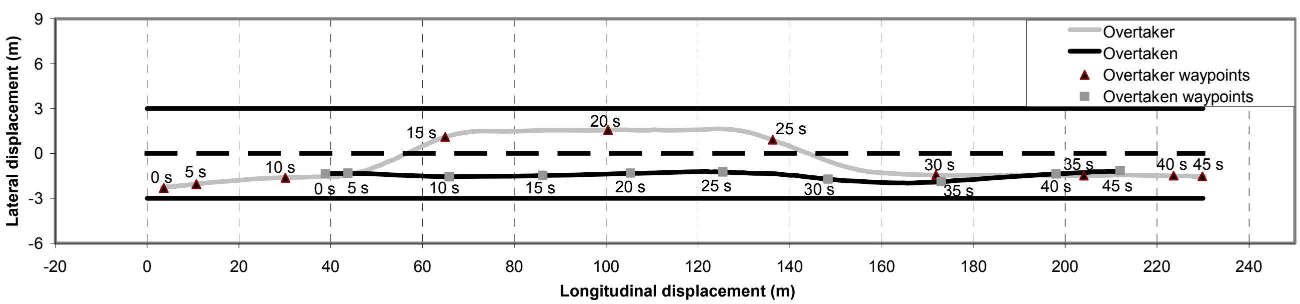

Fig. 6. Overtaking experiment .

Basically, there are two fuzzy controllers that manage both vehicle control signals: speed and steering, also known as longitudinal and lateral control. In other words, the longitudinal control manages the throttle and brake, and the lateral control, the steering wheel movement. Each one of this two fuzzy controllers is different depending the situation the car is. This way, for the steering management there are three fuzzy controllers, depending the car is in straight, curve or overtaking situations. There are also two controllers for the speed; one manages the throttle and the other one manages the brake. In this case, this controllers are capable of managing the speed in cruise control situations as well as in adaptive cruise control, with the information supplied by the sensors. Steering fuzzy control [13] is in charge of minimizing the copilot's calculated trajectory deviations looking for the car to adapt correctly to a curve trajectory, straight trajectory, or reference trajectory change. The output of this controller is the angle that the steering wheel must be turned to correct the trajectory deviation. Throttle and brake fuzzy control [14] permit the vehicle to adapt its speed to a reference in each part of the road, reducing or increasing this speed when necessary in order to maintain its route or to maintain a safety distance from the precedent vehicle.

The outputs of these controllers are the incremental pressure that must be effected over both pedals in order to minimize the speed errors.

The third architecture layer is the actuation one. In this part, the control signals generated by the pilot are adapted so as to be sent to the corresponding actuators.

IV. Implementation

Once described the overall architecture, we proceed to explain its actual implementation stage. All the vehicles equip different elements but with a similar function. However, from the point of view of the architecture, the functionality of each one is the same. This way, all of them equip an onboard computer in which resides the control system. Sensors send information in several ways, depending the car; GPS's and laser scanner are connected via RS232 serial ports; the speed signal is read through an analog input card in the Berlingos and with the CAN bus interface in the C3. This bus is use to obtain much more information about the car status. Each computer vision system is placed in individual onboarded PCs since the vision task requires higher processor consumption. Each computer (Babieca and Clavilenyo) is connected to the onboard control one via wired LAN. Presently, WLAN networking infrastructure is used for transmitting the differential correction from the central station to each vehicle and to broadcast the position information between cars.

About the sensors, the steering wheel is managed through a DC servo motor engaged to the steering bar. The motor is controlled by a control/power card with a built-in PID which receives the target angle that the steering wheel must be turned in the Berlingos. In the C3, an analog output attached to a servoamplifier manages the motor, using several classical control and fuzzy controllers in a cascade architecture. The throttle is managed with an analog signal that represents the pressure over the pedal, simulated with an analog card which receives the pressure value calculated by the pilot. Finally, the brake pedal is automated through a DC servo motor with a pulley that receives an angle command from the same card than the steering wheel.

Some examples of the performance of the automatic vehicle architecture can be found at http://www.iai.csic.es/autopia.

V. Experiments

In this section we present some examples that show the performance of the defined architecture when it is installed in our testbed vehicles. First experiment represents the automatic route performed by the Citroën C3 following reference trajectory. It is shown in fig. 5.

The X axis represents the UTM East coordinate and the Y axis represents the UTM North, in meters. In this case, the car is performing an standalone route with no other vehicle circulating in the same driving zone and the steering wheel and speed of the car are automatically managed. In this graphic, the gray and dotted black lines correspond to the vehicle trajectory and the reference route respectively. In this case, the circulation speed oscillates between 10 – 20 Km/h, adapting the system it when is necessary.

The second experiment corresponds to an overtaking maneuver. In this case, two cars are involved in the operation, being driving cooperatively and interacting between them. The overtaker vehicle is automatically driven and the overtaken one is controller by a human. This situation increases the difficulty of the control because a human driver is much more unpredictable than an automatic one, as is represented in the erratic route depicted by the black line in fig. 6.

In this figure, the routes during a overtaking operation are shown. In these trajectories some temporal waypoints are marked to indicate the position where each car is at a time instant from the beginning of the experiment. With these points, we can see that the overtaking maneuver is correctly achieved, and the cooperation and information interchange specified in the architecture works correctly. In this experiment, the circulation speed of the overtaker is about 30 Km/h and the overtaken one is 15 Km/h.

VI. Conclusion

We have developed an open architecture for autonomous vehicles to support the interaction and cooperation among a set of autonomous vehicles circulating in the same driving zone. Similarly, a second architecture has been developed and installed in each one of these vehicles, which contains the capacity of individual autonomous behavior. Both architectures are similar because the second one allows each car a standalone automatic driving and the first one provides all the vehicles with the capacity of a cooperative driving. A common communication interface has been defined and the low level driving computation in which resides the human knowledge and experience has been modeled using fuzzy logic.

Acknowledgment

We want to thank Ministerio de Fomento and Ministerio de Educación. We thank specially to Citroën España SA, without its collaboration, this work wouldn't have been achieved.

References

[1] R.R. Mutphy, "Introduction to AI Robotics", MIT Press, 2000.
[2] W.L. Nilsson, "”Shakey the robot", Technical note 323 AI Center, SRI International, 1984.
[3] R.A. Brooks, "A Robust Layered Control System for a Mobile Robot", Readings in Uncertain reasoning , Morgan-Kaufmann, pp. 204-213, 1990.
[4] R.C. Arkin, "Integrating Behavioral, , Perceptual and world Knowledge in Reactive Navigation", Robotics and Autonomous Systems, 6, 105-122, 1990.
[5] C. Laugier and T. Fraichard, "Decisional architectures for motion autonomy", Intelligent Vehicle Technologies, Ed. SAE International, pp. 333-391, 2001

[6] J. Kolodko and L Vlacic," Griffith University Cooperative Autonomous Driving at the Intelligent Control Systems Laboratory", IEEE Intelligent Systems July/August, pp. 8-11. 2003.

[7] P. Varaiya, "Smart Cars on Smart Roads: Problems of Control", IEEE Transactions on Automatic Control, Vol. 38, No. 2, pp. 195–207, February 1993

[8] J. Lygeros, D. N. Godbole, and M. Broucke. "A Fault Tolerant Control Architecture for Automated Highway Systems", IEEE Transactions on Control Systems Technology, Vol. 8, No. 2, pp. 205-219, March 2000

[9] M. Bayouth, I. Nourbakhsh and C. Thorpe, "A Hybrid Human-Computer Autonomous Vehicle Architecture,". In Proceedings, Third ECPD International Conference on Advanced Robotics, Intelligent Automation and Control, 1997

[10] M.A. Sotelo, et al., "Laser-Based Adaptive Cruise Control for Intelligent Vehicles", 1st International Conference on Informatics in Control, Automation and Robotics, ICINCO 2004. August 2004, Setúbal, Portugal. pp. 388-401.

[11] M.A. Sotelo, F.J. Rodriguez; L. Magdalena, "VIRTUOUS: vision-based road transportation for unmanned operation on urban-like scenarios" IEEE Transactions on Intelligent Transportation Systems, Volume 5, Issue 2, pp. 69-83, June 2004.

[12] J.E. Naranjo, et al., "Overtaking Maneuver Experiments with Autonomous Vehicles", Proceedings of the 11th International Conference on Advanced Robotics 2003, ICAR 2003. Coimbra, Portugal, pp. 1699-1703, June 2003.

[13] J. E. Naranjo, C. González, R. García, T. de Pedro, and R.E. Haber "Power Steering Control Architecture for Automatic Driving", IEEE Transactions on Intelligent Transportation Systems, Volume 6, Number 4, pp. 406-415, December 2005.

[14] J.E. Naranjo, C. González, J. Reviejo, R. García, T. de Pedro, "Adaptive Fuzzy Control for Inter-Vehicle Gap Keeping" IEEE Transactions on Intelligent Transportation Systems, Volume 4: No. 3, pp. 132-142, September 2003