RESEARCH ON THE ACC AUTONOMOUS CAR

Received 24th February 2007; accepted 1st March 2007.

Claudiu Pozna, Fritz Troester

Abstract:

The aim of this paper is to present a general view about the autonomous driving researches made in the University of Applied Science Heilbronn, Germany. More exactly we will present aspects about our autonomous car design and construction. We will point out the original elements of our achievements: the driving robot construction and the control system structure design. This will be also the occasion to focus on the tactical level of the mentioned control system in order to present results on the trajectory tracking strategies.

Keywords: *autonomous car; driving robot; control system architecture, trajectory tracking*

1. Introduction

Nowadays, autonomous car designs are taken into consideration more and more. The mentioned concept can be defined like a € car, which is able to drive on its own. So this means that we deal with a car, which copies a human driver's performances. Is this a mobile robot? The answer is yes, if we consider only the navigation performance, but no, if we have in mind there is necessary to add specific interfaces that are not needed in mobile robots (we mention here the interface between the human passenger and the autonomous car, interface, which allows the possibility to interrupt the control system and drive the car in traditional way). The present work has ignored this difference and has focused on the driving (navigation) performance that means that our autonomous car is in fact a mobile robot.

Scientific literature proves that many research teams study the autonomous car subject. Spectacular results have been obtained by Volkswagen and by Stanford University. It is important to match these two success stories , because it highlights the convergence of two directions of development: the automotive industry, which increases permanently the automation in the cars and the universities, which try to increase the navigation performances of the autonomous cars.

In order to exemplify the automotive industry projects, we will mention the “Autonomous Driving” project, which was managed by Volkswagen. The purpose of the project was to develop an autonomous vehicle with the options of accidents avoidance and automatic driving. The project partners were the Brunswick Technical University, Robert Bosch GmbH, Kasprich-IBEO GmbH and Sondermachinen GmbH. Accordingly, up to ten vehicles were simultaneously driven automatically by robot-drivers. A driving robot was implemented in the car to transform a VW in a mobile robot. This driving robot has three “legs” (which allows it to manipulate the gas, clutch and brake pedals) and two “arms” (which manipulates the steer and the gearbox lever). The environment recognition was possible by: radar sensors, laser scanner, and two video cameras. All these systems provide the vehicle guidance, high precision, computation of the desired trajectory; vehicle regulation, sensors functions etc. Another project managed by automotive industry (in the frame of European Prometheus project), was VITA II. Daimler-Benz presents a vehicle, which is able to drive autonomously on highways and perform overtaking maneuvers without any interaction. One of the research developments is the Intelligent Stop&Go.

At the universities, several projects on autonomous cars have been made as well. The Stanford University's Stanley mobile robot won the 2005 DARPA Grand Challenge. From [1,2] we know that Stanley was developed for high-speed desert driving and was controlled through artificial intelligence methods. Another example is the Safemove FranceKorea project that developed the CyCab robot - designed at INRIA - and the pi-Car prototype of IEF [3]. We will mention also the NAVLAB robot developed by Carnegie Mellon University Robotics Institute [4] and the examples can continue. Each of the projects allowed researches in several directions: car tracking [5,6]; vision-based navigation systems that tracks car taillights at night [7]; safe navigation which allows to reduce the use of the private automobile in downtown areas [8]; timing failure detection service which reacts to unexpected perception delays [9]; sensor fusion and parking [10,11], autonomous car programming and navigation by using the Bayesian theory [12, 13]; car control using fuzzy logic or neural network which intends to increase the system robustness [14-20] etc.

Based on these remarks, we briefly sum up our conclusions concerning the developments of autonomous cars:

- There are several projects made by powerful car companies. The results of these projects are summarized by the sentence: “It can be done, but it is too expensive”. Moreover, much knowledge from these projects have been used in the development of new automated systems: “stop& go” maneuver; automotive cruise control; automatic parking systems etc;
- The autonomous car construction implies a mechanical and electrical design. There are two solutions: transform a real car into an autonomous car [1] or design a new vehicle [3].
- The mobile robots' navigation has been defined from mathematical point of view [13], this means that we

have the axiomatic background for mathematical solutions;

- Each mentioned work try to solve the navigation problem or a part of this problem in a particular way. There are used classical robust control techniques [14], fuzzy logic or neural network strategies [4,15-18] or solution based on Bayesian theory of probability [3, 8, 10, 12, 13];
- One drawback of the mentioned solutions is the time consuming computation, needed in solving complex situations [13].

This paper presents aspects about the ACC autonomous car that we have constructed in the University of Applied Science in Heilbronn, Germany. The presentation will include aspects from project management, mechanical and electrical design of the autonomous car, and will focus on some original elements, which refer to the control architecture of the mobile robot. More precisely, we will present our three level control system, which operates with a collection of programs named behaviors, and we will detail some of them.

It is important from the beginning, to mention what benefits we expected from our project:

- Construction of an autonomous car. This means to transform an ordinary car into a mobile robot, by building a driving robot in the car. The driving robot is a mechatronical construction which replaces the human driver in the car;
- Designing an intelligent control system for a mobile robot. This means to understand the environment, to identify driving circumstance, to find an appropriate behavior; to compute the control signal for the driving robot and to send this signal;
- Solving the mentioned time-consuming computation drawback by developing control architecture, which manage a collection of (off line made) solution for several driving situation.

We have organized our presentation in the following sections: description of the mobile robot construction, where we will show the electrical and mechanical design of the driving robot, description of the conceptual construction for the control system, followed by the presentation of control program.

2. The driving robot construction

The mobile robot is a system composed of following subsystems: the car, the driving robot, the control system and the extra sensory system. In order to accomplish our goals, we have organized our project using the well-know functional design concept presented in figure 1. There are three major levels in this procedure: construct, implement and test the driving robot in the car (an AKlasse Mercedes), construct the control system, integrate sensors needed for the environment recognition and finally test the autonomous car.

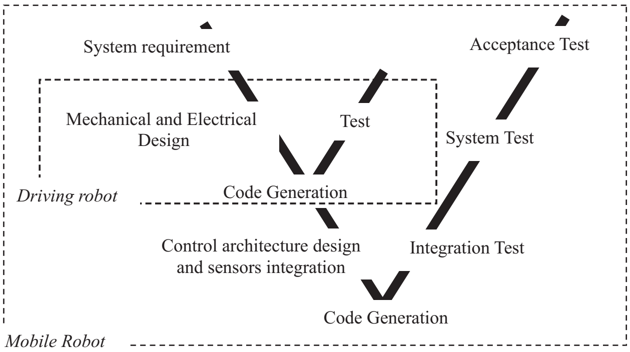


Figure 1. The functional design concept used in project management

In order to present the construction of our driving robot, several preliminary pieces of information about the driving robots are needed. These robots, which replace the human driver in the car, are usually made for car testing. Because in the car industry there are many tests which imply a huge number of cyclic operations (exhaust emission test, fuel economy test, running losses test, acoustic test etc.) it is necessary to use driving robots. Because of this demand, several firms produce the mentioned robots. We will mention here the Stahle and Antony Best driving robots [19]. Unfortunately these systems have not been suitable for our project, and we have constructed ours on driving robot. This effort is motivated by following reasons: driving robots are made for indoor testing; control programs, which run on these robots, are designed to perform cyclical operations and data acquisition and it is difficult to develop them in order to obtain artificial intelligent systems.

The electrical and data transfer block diagram of the constructed driving robot is presented in figure 2. It can be seen that the driving robot is composed of five subsystems, each one needed to copy a certain action of the human driver: steering, acceleration, turning the ignition key, turning the gearbox lever and braking the car.

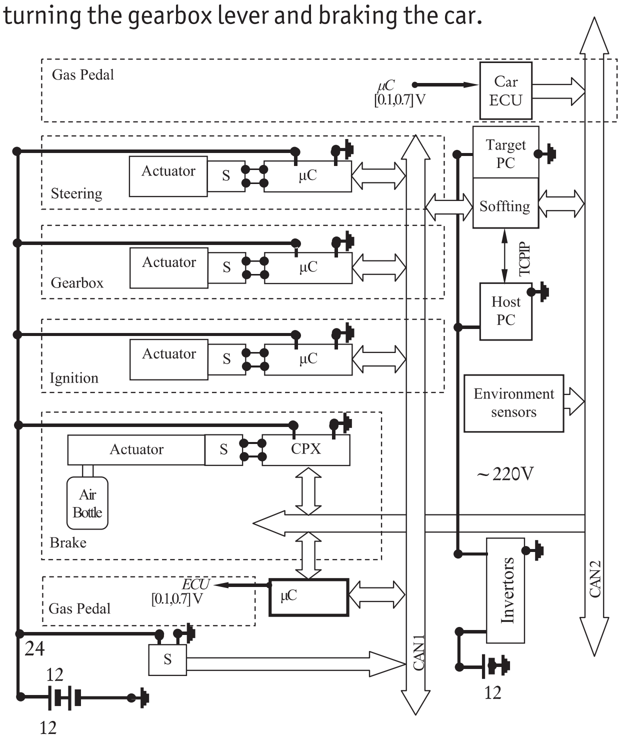
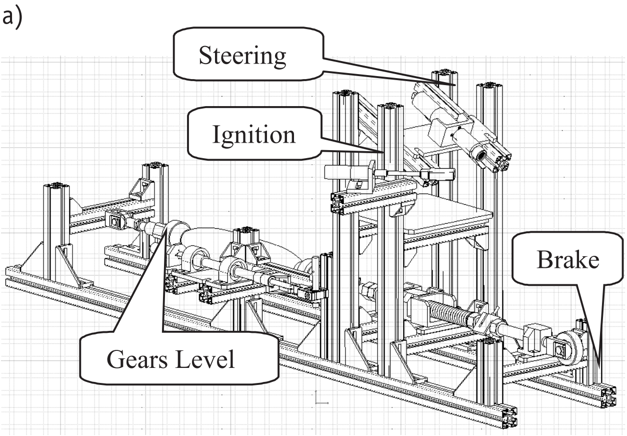


Figure 2. The Driving Robot electrical block diagram

Each subsystem consists of actuators (we have chosen MAXON DC motors [20] and FESTO Muscle) sensors and micro-controllers, which solve the local control problems. Over these five local feedback loops, a main feedback loop is design. The hard of this loop consists of the environment sensors, the target PC (real time machine) and the CAN data transfer network. The program, which runs on the target computer, is the control system that we will be presented in the next section.

In figure 3.a we have illustrated the CAD drawing of the robot, where for a better understanding we have identified the mentioned subsystems.

a)

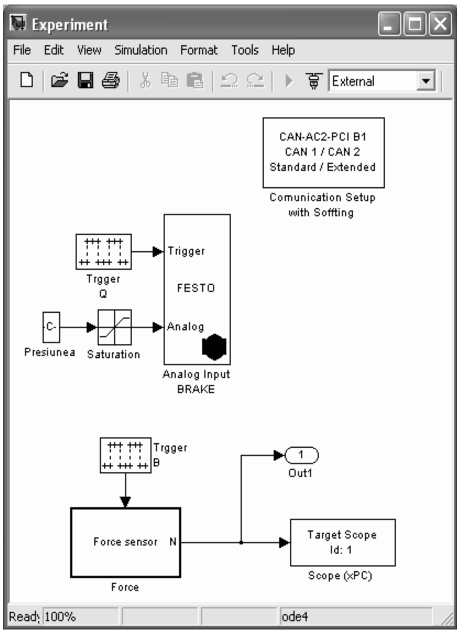


b)

Figure 3 a) The Mechanical design of the driving robot, b) Integration of the robot in the car

In order to illustrate the integration of the driving robot in the car, in figure 3.b we have presented a picture of the autonomous car. According to the project management (see figure 1), after the driving robot integration, indoor tests must be made. For this purpose, test programs have been made in Matlab (using xPC toolbox) on the host computer (see figure 2) and have been down-loaded on the target computer. To exemplify these tests, we have chosen the braking subsystem. More precisely, one of the braking tests programs is presented in figure 4.a, and the result of this test is presented in figure 4.b.

a)



b)

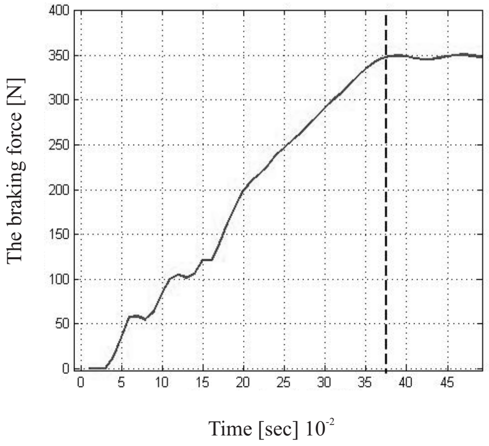


Figure 4, a) The test program for braking, b) Test result

We can conclude that the driving robot is an operational model of the human driver. All the human driver actions (steering, braking, etc.) can be approximated by using the driving robot. Successful test results have allowed us to step to the next level of the project (see figure 1): design the control system. Results of this level will be presented in the next section.

3. The control system structure

Our control system architecture is based on the human driver behavior model concept. So, in order to present our ideas, some preliminary discussions are necessary. From [21-28] we know that the "Driver Behaviors" model is used in the simulation [23], [24] and also in autonomous car design field [22]. The first researches on the subject started in 1950 [23] and began with the "Skill-based driving model", continued with the "Motivational model" which considered the drivers' emotional state (from this class we can enumerate the "Risk compensation"; "Risk avoidance" and "Risk threshold models"). Recently, the model turned to a "Hierarchical control structure" (by Milchon). The "Hierarchical control structure" divides driving into three levels of control: a strategic level, which establishes the goal of the driving, a tactical level, which finds the solution to accomplish the goal and an operational level, which implements this solution on low level control of the vehicle. Behind this "Hierarchical control structure" many scientific papers consider and develop problems like: "Longitudinal behaviors models" [22]; "Lateral behaviors model" [25], [21]; "Brake behavior" [22] etc. The solutions of these problems are varied: "Linear optimal Control", "Heuristic human driver models", "Adaptive control strategy", "Neuronal Network and fuzzy logic", "Mental models", etc.

Because we have intended to make a heuristic approach, we have been interested in finding control programs architectures, which model the human behavior.

Such architectures are presented in [23] and [24].

Some conclusions about these briefly overviews:

- In the scientific literature referring to "Driver Behavior Model" we have found several results which can be adapted and used in our mobile robot control;
- Recent works accept the Milchon three levels architecture;
- Many papers focus on the tactical level where the program must find the solutions in condition of changeable driving circumstance.

Our idea starts from this point: we consider that it is more suitable to model and implement the "human driver decisions act", than the "human driver actions". This idea transfers the approximation of the human driver behaviors from a mechanical to an artificial intelligence problem. This kind of problem involves preliminary analysis, which must answer to the following questions: "What are driving behaviors?" "Can we obtain some fundamental true about these behaviors and use them in our construction?"

We have made a phenomenological research [29], which starts with the semantic characteristics of "Driving behavior". First, it is important to establish the category tree of this word: from [30] we have {act › activity › (behavior, practice,)}. According to this, the behavior is: "an action or a set of actions performed by a person under specified circumstances that reveal some skill, knowledge or attitude". From the scientific literature which concerns the driving behavior [21-28] and from our experience, the driving behavior has a special feature. To describe it, we focus on the word "custom" which belongs to the same category tree {act › activity › practice › habit,}and which is defined as: "accepted or habitual practice". In many situations, these habits have a special nature: automatism - any reaction that occurs automatically without conscious thought or reflection. Using the previous definition, "Driving Behavior" is an action or a set of actions performed by a person under driving circumstances, actions that tend to be transformed in habits and even in automatisms. In fact the "Driving Behavior" is composed of a series of behaviors (the driver's behavior when he makes the ignition, the driver's behavior when he stops the car, etc.). From the mentioned theoretical and practical research, we established the following "fundamental truth" for the "Driving Behavior":

1. A priori, the driver establishes the current driving goal;
2. A behavior is a set of actions;
3. These behaviors are linked together, creating a system which allows the driver to obtain solutions in the driving circumstance;
4. The translation from one behavior to another is triggered by the occurrence of an event;
5. This system is developed by learning - experience;
6. Behaviors presume decisions with an incomplete set of information;
7. In time, these sets of actions tend to be transformed in habits and automatisms.

These propositions agree with the well-known three level architecture of Milchon: the strategic level, where the driver establishes his goal, the tactical level, where the driver finds the solution to accomplish the goal and the operational level, where the driver implements these solutions. Using these propositions, we can focus on the tactical level and model (approximate) the "Human Driving Behaviors" by a collection of high linked programs (behaviors), which are stored in a memory. The decision to run a certain program is made by a manager program. This decision is based on the driving goal and on the environment understanding (driving circumstance). Each program (behavior) is a set of instructions (actions), which impose parameters and trigger actuators. Using these seven propositions, we can imagine the utility of state machines, for handling the behaviors, fuzzy logic to enable decisions or to describe the environment and neuronal network to implement the learning processes.

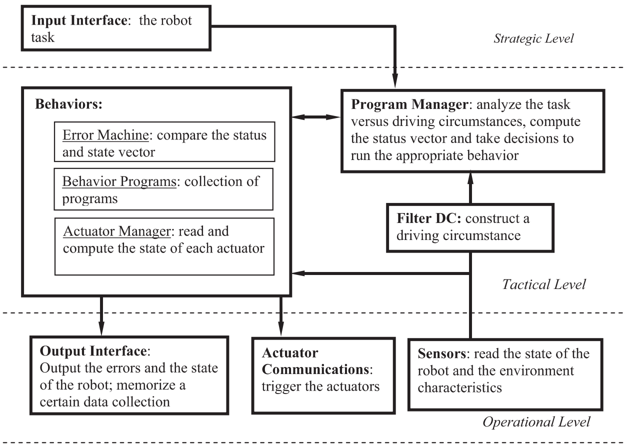


Figure 5. The Control system architecture

The aim of figure 5 is to make our concept more understandable and to allow the necessary explanations:

- The strategic level, where the robot receives its task (goal) is an interface which helps the human operator to impose the goal
- The "Program Manager" analyzes the goal versus the driving circumstances which are obtained from the sensors; the result of this process is the status vector of the robot (the desired position, velocity, etc.) and also the decision to run a certain program from the "Behaviors" subsystem;
- The "Behaviors" contains three parts:

- The "Error Machine" which compares the status vector with state vector (the positions, velocity, etc. obtained from the sensors);
- The "Behavior Programs": is a collection of programs (behaviors); each program is able to solve a special environment situation (ignition, emergency stop, zero position, errors....);
- The "Actuators Manager" which manage the actuators of the robot;
- The "Output Interface" allows reading the states and errors and also the robot state history memorization;
- The "Actuators Communications" outputs data to the micro- controllers of each actuator;

In order to build the "Behaviours" subsystem, it is important to imagine the structure of the programs, which are included in the "Behaviours Programs". In figure 6.a we present three different structures, named: "basic behaviours", "error behaviours" and "simple behaviours". The main differences between these programs are the connection type (P previous, N next, E error, QI quick in, QO quick out) and also the direction of information flow.

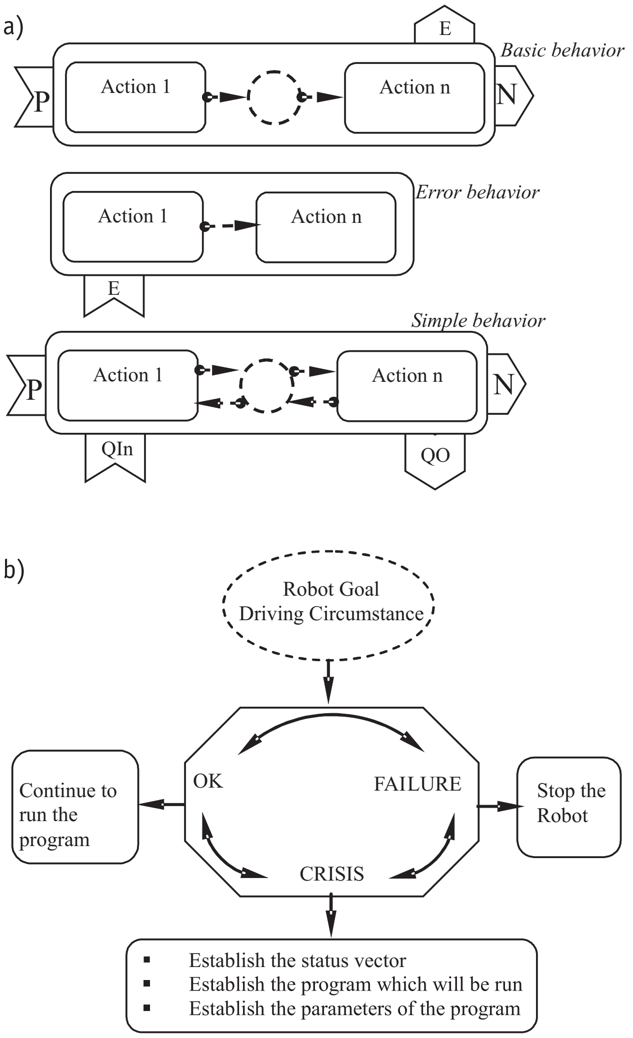


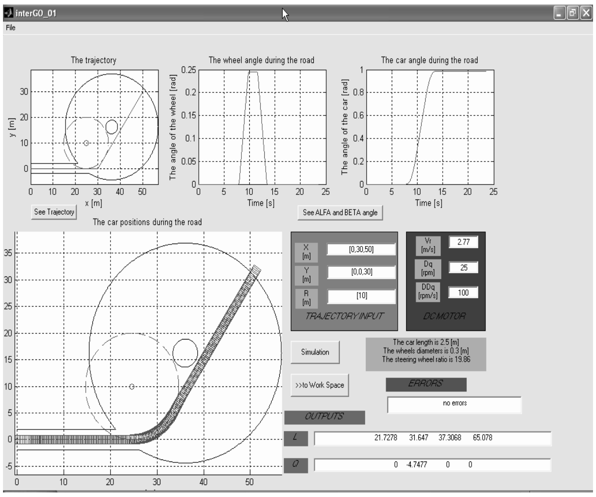
Figure 6 a) The structure of the programs included in "Behavior Programs" b) The "Program Manager" structure

In the "Behaviors" subsystem, there is an "Error Machine" program running. The aim of this program is to compare de "status vector" (the desired variable of the robot: car speed, steering angle etc.) with the "state vector" (the variable read from the sensors: car speed, steering angle etc.). The "Program Manager" made decisions about the program, which will be run (see figure 6.b). This program compares the goal of the robot with the driving circumstance; establishes the status vector and enables the program to run. After these decisions, the program continues to compare the robot goal with driving circumstance. If the result is acceptable, nothing is changed (the same program is run), on the contrary, a "Crisis" or a "Failure" event is signaled. "Crisis" means that a new behavior is needed, so the status vector as well as the program will be changed. "Failure" means that we don't have solutions (behaviors) to solve the problem and we must stop the robot safely.

4. The control program

In order to build the control program, we have used Matlab (xPC toolbox). Once again, the program is constructed in the Host computer and downloaded on the Target computer (which is a real time machine) see figure 2. According to previous section, the program is composed of three levels: the input interface (see figure 7.a), the tactical level (see figure 7.b) and the output interface which runs on the target computer.

a)



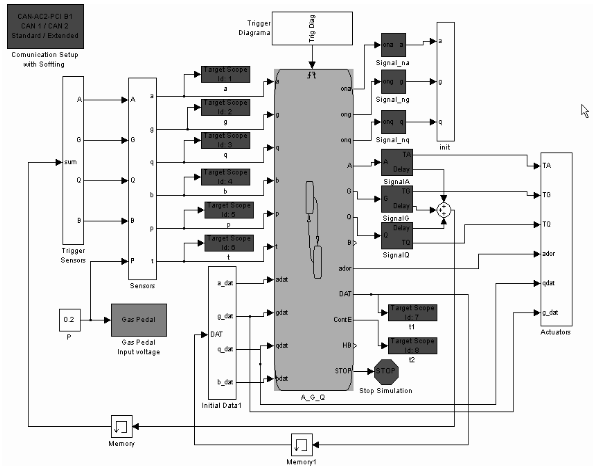
X [m] [0,30,50]
Y [m] [0,0,30]
R [m] [10]
Vr [m/s] 2.77
Dq [rpm] 25
DDq [rpm/s] 100

The car length is 2.5 [m]
The wheels diameters is 0.3 [m]
The steering wheel ratio is 19.86

no errors

L 21.7278 31.647 37.3068 65.078
Q 0 -4.7477 0 0

b)



c)

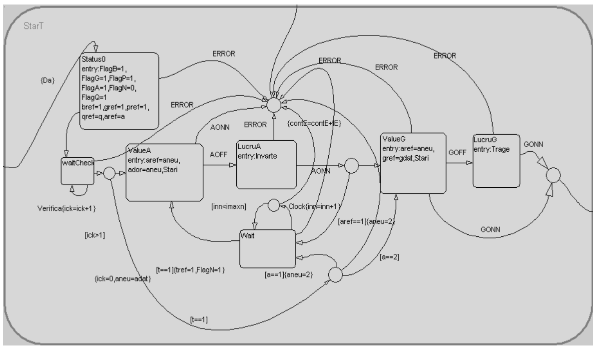


Figure 7 a) The input interface, b) The tactical level, c) The "Start the Car" behavior

The navigation problem that we intend to solve can be defined in the following way: the task of the mobile robot is to follow a certain trajectory between two points; the trajectory is mathematical defined on a map, but (initial) unexpected obstacles must be avoided during the navigation. In order to be able to accomplish this goal, the tactical level of the control program, more exactly the "Behaviors Program" (see figure 5) must contain following programs (behaviors): starting the car; following an a priori defined trajectory; returning to the a priori defined trajectory; avoiding the obstacle and return to the a priori defined trajectory; stopping de car. The interface presented in figure 7.a transforms the robot desired trajectory which is defined in the Cartesian coordinates in desired trajectory which refer to the steering actuator and the car speed. After this transformation, the interface can simulate the working volume of the mobile robot in a desired map. The interface gives also possibilities to verify if the dynamical characteristics of the driving robot (the maximum steering torque, the braking force, etc.) admit the kinematics of the mobile robot (the possibility to follow a certain trajectory in a specific amount of time).

From the mentioned behaviors in figure 7.c we illustrate the "Start the Car" program. The "Start the Car" program is a state machine which manages the following actions: check if the initial state of the autonomous car is appropriate; command the ignition key turning; check the car ignition; command the ignition key returning; command the gearbox lever on D; wait until a new behavior is triggered by the "Manager program" (see figure 6.b). If the ignition has failed, the program gives the possibility to a second ignition maneuver. In case of errors, the program is linked to error behavior (see figure 6.a).

5. Following the trajectory

Following a trajectory is a more complicated behavior and we have split it in two solutions:

- following a trajectory which was computed off line;
- following a trajectory which was computed on line;

In the first case, the control system compute, off line, a trajectory and a tolerance band right around the de mentioned trajectory. Using these data, the "trajectory follower behavior" commands the autonomous car and checks (with a GPS) if the car is inside of the tolerance band. If the car is out of the tolerance band, the "Manager program" chooses a new behavior program, which makes specific corrections and drives the car back into the mentioned tolerance band.

In order to run the "*trajectory follower behavior*", it is necessary to compute (off line) the command vector. Because the desired trajectory consists of linear trajectories, which are connected, by circular trajectories (see figure 8) and because the car is a non-holonomic vehicle, imposing an appropriate trajectory means to find a good compromise, which minimizes the tracking errors in the rectilinear zone.

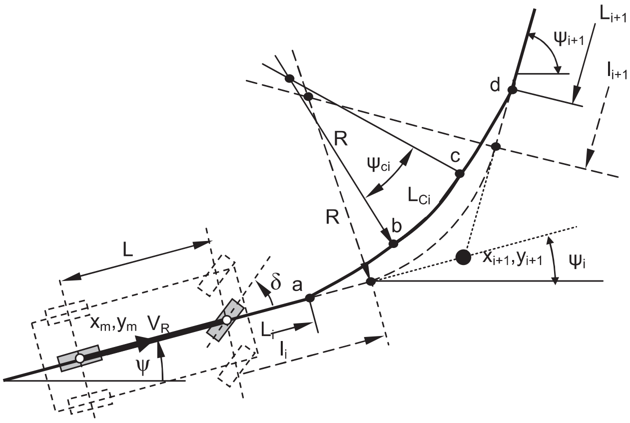


Figure 8. Transformation of the desired trajectory

In order to compute the mentioned command vector (1) we have used the kinematical models of the car (2), of the steering DC motor (which follows a Bang-Bang trajectory) (3) and of the mechanical transmission between the steering wheel and the car wheel (4).

$$\begin{Bmatrix} L \\ \delta \end{Bmatrix} = \begin{Bmatrix} L_1 & L_2 & \dots & L_n \\ \delta_1(t) & \delta_2(t) & \dots & \delta_n(t) \end{Bmatrix} \tag{1}$$

$$\begin{cases} \dot{x}_m = V_R \cos(\delta)\cos(\psi) \\ \dot{y}_m = V_R \cos(\delta)\sin(\psi) \\ \dot{\psi} = \dfrac{V_R}{L}\sin(\delta) \end{cases} \tag{2}$$

$$i_R = \frac{q}{\delta} \tag{3}$$

$$q = q(q_0, q_d, \dot{q}_d, \ddot{q}_d, t) \tag{4}$$

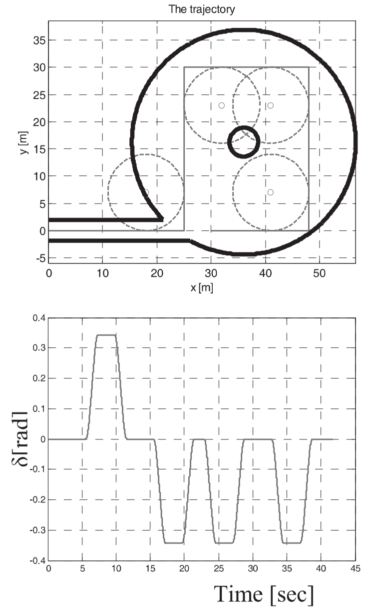
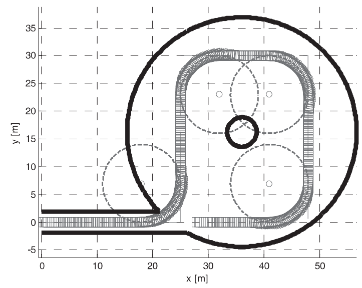
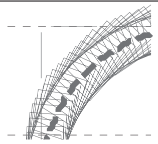
where: L_i is the length of the trajectory that must be covered with the steering angle $\delta_i(t)$; ψ is the car direction angle; V_R is the car speed; q_0 is the initial position; $\dot{q}_d$ is the desired position; $\ddot{q}_d$ is the maximum angular velocity; is the acceleration and deceleration.

The approximation that we have proposed replaces the circular trajectories by clotoidal trajectories. More precisely, (see figure 9) the trajectory will be made of three regions: $\overline{ab}$ steering from δ=0 to δ= δ_d; $\overline{bc}$ steering with

$\delta = \delta_d$; $\overline{cd}$ steering from $\delta = \delta_d$ to $\delta = 0$. The value of δ_d is computed (off line) by numerical integration of equation (2). We have solved this problem by defining a Matlab function. An example of these computations is presented in Table 1.

We intend to use the second strategy (following a trajectory which was computed on line), in the avoiding obstacles maneuvers. More precisely, if the autonomous car recognizes an obstacle, on line procedures compute the avoiding trajectories and control the car on these trajectories.

Table 1

Results for $\dot{q}_d = 25rpm$; $\ddot{q}_d = 25rpm/s$; $V_r = 2.77m/s$

The first step is to impose a trajectory;

The steering function δ(t)
 Detail of the working volume
{13.85, 3.64, 7.61, 3.64, 1.61, 3.64, 7.61, 3.64, 7.75} { 0, 0.34, 0, -0.3430, 0, -0.3430, 0, -0.3430, 0} The command vectors

In order to design the control law, the dynamic model of the car is divided in longitudinal and lateral dynamic. We have chosen the dynamic model, presented in [31]. The equation of this model is presented in (5) (see also figure 9.a), the main hypotheses are that the car speed is constant during the locomotion and the angles are small.

$$\begin{bmatrix}\dot{\beta}(t)\\ \ddot{\psi}(t)\end{bmatrix}=\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\cdot\begin{bmatrix}\beta(t)\\ \dot{\psi}(t)\end{bmatrix}+\begin{bmatrix}b_1\\ b_2\end{bmatrix}\cdot\delta \qquad (5)$$

$$a_{11}=-\frac{2c}{mV};a_{12}=\frac{cl_s-cl_f}{mV^2};a_{21}=\frac{c(l_s-l_f)}{J_z};a_{22}=$$

$$=-\frac{c(l_s^2+l_f^2)}{J_zV};b_1=\frac{c}{mV};b_2=\frac{cl_f}{J_z}$$

where: β is the angle between the velocity V and the car direction; m, J_z are the mass and the momentum of the car; V is the car velocity which is considered constant; c is the rotational stiffness of the wheels; l_f, l_s are the lengths from the mass center to the front and back wheels.

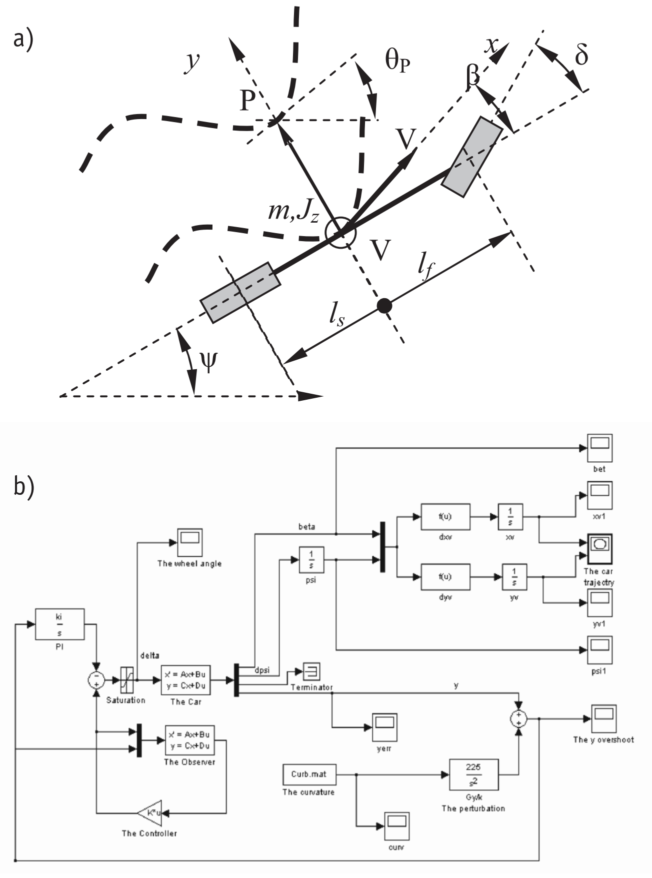


Figure 9.a) The car and the trajectory model, b) The block diagram of the controlled system

The differential equations of the trajectory tracking are (6) [32] - see also figure 9.a. Here the main hypothesis is:

$$dx(r,s)=0;x(r,s)=0 \qquad (6)$$

$$y=V(\theta_\Delta+\beta)$$

$$\dot{\theta}_\Delta=V\kappa_P-\dot{\psi}$$

where: $\theta_\Delta=\theta_P(r)-\psi(s)$; κ_P is the trajectory curvature

$$\begin{bmatrix} \dot{\beta} \\ \ddot{\psi} \\ \dot{\theta}_{\Delta} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & -1 & 0 & 0 \\ V & 0 & V & 0 \end{bmatrix} \cdot \begin{bmatrix} \beta \\ \dot{\psi} \\ \theta_{\Delta} \\ z \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ 0 \\ 0 \end{bmatrix} \cdot \delta + \begin{bmatrix} 0 \\ 0 \\ V \\ 0 \end{bmatrix} \cdot \kappa_P \tag{7}$$

$$y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \beta & \dot{\psi} & \theta_{\Delta} & y \end{bmatrix}^T + [0] \cdot \delta$$

If we add the car model to the trajectory model, we will obtain equations (7) where we can also obtain the transfer function representation:

$$Y(s) = G_{y/\delta}\Delta(s) + G_{y/\kappa}\mathrm{K}_P(s) \tag{8}$$

where: $Y(s) = G_{y/\delta}\Delta(s) + G_{y/\kappa}\mathrm{K}_P(s)$;Δ(s) is the Laplace transform of δ(t); $G_{y/\kappa}(s) = \frac{V^2}{s^2}$; KP(s) is the Laplace transform of κ_P(t)

For the controller design, we have chosen the state-space design methods, which are contained in Matlab Control toolbox. Using equations (8), the simulation diagram (see figure 9.b) is constructed in SimuLink. The mentioned diagram is composed of following blocks: the car model (7); the controller; the observer; the curvature source and the perturbation transfer function. Experiments have been made to track linear or circular trajectories. Results of these experiments are presented in figure 10.a. In figure 10.b a picture of the robot in test field is presented.

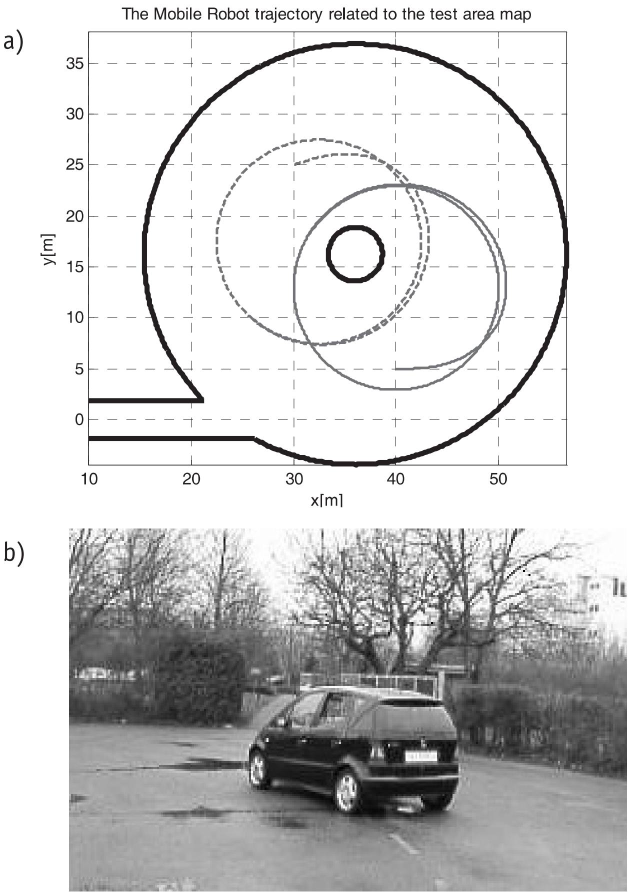


Figure 10. a) Two trajectories of the Mobile Robot related to the test area map; b) The Mobile Robot in the test area

6. Conclusions

The intention of this paper was to give a general image about scientific works made on autonomous locomotion inside the ACC (Automotiv Competence Center) project of Applied Science University from Heilbronn, Germany. For this reason, we have presented aspects about our project management, about the driving robot construction and about the control system design. The original achievements of these works have been highlighted. More precisely, a low price-driving robot was designed and integrated in the car; original control system architecture, based on human behavior model was proposed. We must mention that nowadays, only a part of the control architecture is made, so future work will develop this architecture. We intend also to develop our environment sensory system for circumstance recognition, and "Program manager" development. Other direction in which we intend to develop our driving robot [33] is the obstacle-avoiding maneuver.

AUTHORS

Claudiu Pozna* - Transilvania University of Brasov, Romania, e-mail: cp@unitbv.ro.

Fritz Troester - University of Applied Science Heilbronn, Germany, e-mail: troester@fh-heilbronn.de.

* corresponding author

References

[1] Thrun S., Montemerlo M., Dahlkamp H., et al., "Stanley, the robot that won the DARPA Grand Challenge", *Journal of Field Robotics*, forthcoming. Available online at: http://robots.stanford.edu/papers/thurn.

[2] Gerkey B.P., Thrun S., and Gordon G. , "Visibility-based pursuit-evasion with limited field of view", *International Journal on Robotics Research*, 2006, no. 25(4) pp. 299-316.

[3] Coué C., Pradalier C., Laugier T., Bessiere P., "Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application", *The International Journal of Robotics Research*, vol. 25, January 2006, no. 1, pp. 19-30.

[4] Pomerleau D., "Neural Network Based Autonomous Navigation", in: *Vision and Navigation: The CMU Navlab*, Ed. by C. Thorpe, Boston: Kluwer Academic Publishers, 1990.

[5] Kehtarnavaz N., Griswold N. and Lee S, "Visual Control of an Autonomous Vehicle (BART) | The Vehicle-Following Problem", *IEEE Transactions on Vehicular Technology*, vol. 40, 1991, no. 3, pp 255-272.

[6] Kuehnle A., "Symmetry-based Recognition of Vehicle Rears", *Pattern Recognition Letters*, vol. 12, 1991, pp. 249-258.

[7] Rahul Sukthankar RACCOON, "A Real-time Autonomous Car Chaser Operating Optimally at Night", in: *Proceedings of IEEE Intelligent Vehicles*, 1993, pp. 245-256.

[8] Pradalier C., Hermosillo J., Koike C., Braillon C., Bessihre P., Laugier C., "An Autonomous Car-Like Robot Navigating Safely Among Pedestrians", *Preceding of the 2004 IEEE International Conference on Robotics & Automation*, New Orleans - April 2004.

[9] Sousa P., Verýssimo P., "Towards a cooperating autonomous car", in: *Proceedings of the 7th CaberNet Radicals Workshop*, Bertinoro, Italy, Oct. 2002, pp 133-141.

[10] Braillon C., Kane Usher K., Pradalier C., Crowley J., Laugier C., "Fusion of stereo and optical flow data using occupancy grids", *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, Toronto (CA), September 2006.

[11] Shih-Jie Chang, Tzuu-Hseng, "Design and Implementation of Fuzzy Parallel-Parking Control for a Car-Type Mobile Robot", *Journal of Intelligent and Robotic Systems*, 2002, no. 34, pp. 175194.

[12] Chagas C. M., Koike C., *Bayesian Approach to Action Selection and Attention Focusing An Application in Autonomous Robot Programming*, PhD Thesis, l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Institut National Polytechnique de Grenoble, 2005.

[13] Pradalier C., *Navigation intentionnelle d'un robot mobile*, PhD Thesis, l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Institut National Polytechnique de Grenoble, 2004.

[14] Ackerman J., "Robust control for automatic steering", *Proc. Amer. Conf.*, San Diego, 1990, pp.795-800.

[15] Antsalkis P., *An introduction to Intelligent and Auto-nomous Control*, Norwell: Kluver Academic Publisher, 1993.

[16] Layne J.K.,"Fuzzy Learning Control for Antiskid Braking System", *IEEE Trans. on Control Systems Techn.*, vol.1, June 1993, no. 2, pp. 122-129.

[17] Jang J.-S.R., C.-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft *Computing*, NJ:Prentice Hall 1997.

[18] Kosko B., Neural Network and Fuzzy Systems, Englewood Cliffs, NJ:Prentice Hall 1992.

[19] http://www.stahle.com

[20] http://www. maxonmotor.com

[21] Mizukami R., *Development of Autonomous Ground Vehicle Design of Control System*. Available at: http://www.ee.ualberta.ca.

[22] Bengtsson J., *Adaptive Cruise Control and Driver Modeling*, Department of Automatic Control Lund Institute of Technology Lund, November 2001.

[23] Al-Shihabi T., Mourant R., "A Framework for Modeling Human-like Driving Behaviors for Autonomous Vehicle in Driving Simulators", *Proceedings 5th International Conference on Autonomous Agents*, June 2001, pp. 286-291.

[24] Quispel L., *Automan, a psychologically based model of human driver. Experimental and Work Psychology*, Department of Psychology, University of Groningen. Available at: http://www.tcw2.ppsw.rng.nl.

[25] Ungoren A., *An Adaptive Lateral Preview Driver Model. Vehicle System Dynamics*. vol. 43, April 2005, pp. 225-259.

[26] Liu A., "Modeling and Prediction of Human Driver Behavior", *Proc. of the 9th International Conference on Human Computer Interaction*, August 2001, pp. 235-244.

[27] Huang S., "Design and Performance Evaluation of Mixed Manual and Automated Control Traffic", *IEEE Transaction on Systems Man and Cybernetics*, Nov. 2000, pp. 127-136.

[28] Salvucci D., "Modeling Driver Behavior in a Cognitive Architecture", in: *Human Factors* (press), 16th February 2005.

[29] Pozna C., "A phenomenological analyze trial of AI syntagma", in: *Proc. of the 7th International Symposium of Hungarian researches*, 2006, pp. 159-165.

[30] http://www.wordreference.com/definition/behavior

[31] Isermann R., "Diagnosis Methods for Electronic Controlled Vehicles", *Vehicle System Dynamics*, vol. 36, 2001, no.2-3, pp. 77-117.

[32] Hrich K., "Optimization of Emergency Trajectories of Autonomous Vehicles with Respect to Linear Vehicles Dynamic", in: *IEEE/ASME 2005 International Conference on Advance Intelligent Mechatronic*, California SUA 2005, pp 311-320.

[33] Pozna C., Troester F., "Collision Avoidance Trajectory Design", at *15th International Workshop on Robotics in Alpe-Adria-Danube Region RAAD*, 2006.