Proceedings of ICAR 2003
The 11th International Conference on Advanced Robotics
Coimbra, Portugal, June 30 - July 3, 2003

Path-Tracking Controller of a bi-steerable Cybernetic Car using Fuzzy Logic

Luis Conde Bento†, Urbano Nunes†, Abel Mendes† and Michel Parent‡

†Institute for Systems and Robotics
University of Coimbra-Polo II 3030-290 Coimbra, Portugal
{conde, urbano}@isr.uc.pt

‡INRIA Rocquencourt
B.P. 105 78153, Le Chesnay Cedex, France
michel.parent@inria.fr

Abstract

This paper presents a path-tracking controller of a bi-steerable cybernetic car. The velocity planner is a fundamental block in the architecture. Its implementation uses fuzzy logic, and dynamic issues like vehicle´s dynamics and road-vehicle interaction are taken into account. To make the travel as comfortable as possible for the passengers was one of the main objectives in the controller design. Simulation and experimental results are presented showing the effectiveness of the overall navigation control system.

1 Introduction

In the last decades, we have seen a significant growing world scale level of pollution and specially in big cities. The needfulness of people to move quickly from one place to another lead to the increase of the automobile park which drives the increase congestion, the decrease in safety and the growth in pollution [1].

To cope with the problems mention above small automated and non-pollutant vehicles are now being developed in many countries to form a new public urban transport with the same flexibility as the private automobile, i.e. anywhere, anytime.

These vehicles use advanced control techniques for navigation in an autonomous way, i.e. without a driver.

Path-tracking has been widely investigated, Dongbing Gu and Huosheng Hu developed a path-tracking scheme for a car-like mobile robot based on neural predictive control [2]; Jacky Baltes and Robin Otte, developed a Fuzzy Logic Path Controller for path-tracking [3], Jiri Sika and Joop Pauwelussen using lookahead point; Virtual Point (VP), developed a lateral controller to follow the pre-described path, the steering is continuously controlled by a closed loop system observing the difference between the actual and the desired pose of the vehicle [4].

The primary aim of this research work was to develop a control architecture for path tracking of a bi-steerable cybernetic car. It must provide the path tracking ability taking into account the vehicle velocity, atmospherical conditions and type and slope of the road. The smoothness of the acceleration profile was one of the main issues considered in the controller design.

The navigation system is mainly decomposed in two modules: Velocity Planner (VP) and Path-Tracking Controller (PTC). The Velocity Planner computes the linear passenger comfortable reference velocity and the linear maximum reference velocity in which the vehicle can travel. The control system drives the robot along the path while handling disturbances (Path Tracking), this module is Fuzzy Logic based and makes use of a lookahead point (Virtual Point) providing a prediction behaviour. Such modules operating together solve the global problem of moving to a goal with time constraints. To build a complete system a combination of these two modules was made resulting in our present control architecture.

The paper structure is the following. Section 2 presents the control system architecture and section 3 presents the vehicle parameters used in the research. The Velocity Planner Module and the PT controller are described in sections 4 and 5, respectively. The results either with or without the Virtual Point, are addressed in section 6. Finally, in section 7 some concluding remarks are given.

2 Control Architecture

The control architecture is made up of two main modules: the Velocity Planner and the Path-Tracking Controller, as shown in Figure 1. The Velocity Planner provides the desired trajectory and computes the maximum velocity and the comfortable velocity based on the *Curvature*, external factors, and the passengers comfortable acceleration. External factors are tyre characteristics, road slope and passenger comfortable lateral and longitudinal accelerations. In order to follow the trajectory the real position is feeded as input. The next desired position and the reference velocity are feeded to PTC. In the PTC the errors between the desired pose and actual pose are used as inputs to the Fuzzy Logic module that converts them to desired commands. A position error and a heading error are measured for every new iteration.

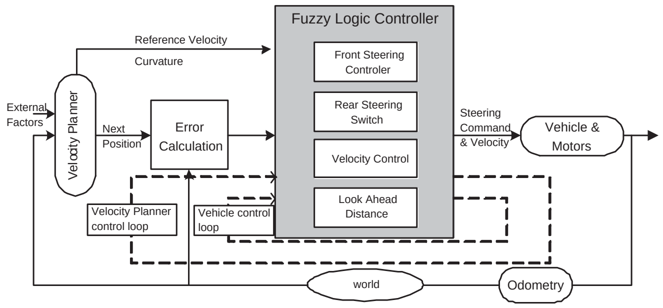

Figure 1: *Control Architecture.*

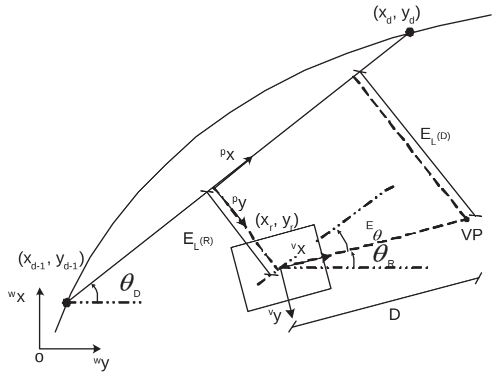

Figure 2: *Position and heading errors.*

These errors are calculated in respect to a coordinate system $\{^P x, ^P y\}$ that is defined to have its origin located at the perpendicular projection of the current vehicle location onto the planned path. This coordinate system, as well as the position and heading errors, are shown graphically in Figure 2. The position error, E_L, is defined to be the lateral error: $E_L = ^P y_V$, where $\{^P x_V, ^P y_V\}$ are the coordinates of the VP in the coordinate system defined above. The heading error, E_θ, is defined by $E_\theta = \theta_D - \theta_R$, where θ_D and θ_R are shown graphically in Figure 2.

The differential errors ΔE_L and ΔE_θ given by $\Delta E_L = E_L(k) - E_L(k-1)$ and $\Delta E_\theta = E_\theta(k) - E_\theta(k-1)$, and the *Curvature* radius are additional inputs to the Fuzzy Logic module.

The Vehicle Kinematics, Motors and Odometry models were considered in the simulations. They are vehicle dependent and their relevant parameters are described in the next section.

3 Kinematic Model of the Car

The Car has the ability to steer both the rear and the front pair of wheels [7]. The classical model, which considers an imaginary wheel at the mid point of the wheels axles so that it is oriented in the direction of the steering command is used. Wheels rolling without slipping assumption was made in simulations.

The kinematic model of the car, for a reference frame located at mid point of the rear axle, is given by the following equations:

$$\dot{x} = v\cos(\theta + \phi) \tag{1}$$

$$\dot{y} = v\sin(\theta + \phi) \tag{2}$$

$$\dot{\theta} = v\frac{\sin(\varphi \times \phi)}{L\cos(\varphi)} \tag{3}$$

where L is the length between axles, and the variables θ, ϕ and φ are defined as illustrated in Figure 3

The speeds for each wheel, are calculated as follows:

$$V_{lf} = V_{lr} = \frac{2|\tan(\alpha)|\sqrt{(\frac{R}{2} - \frac{e}{2})^2 + \frac{l^2}{4}}}{l} V_D \tag{4}$$

$$V_{rf} = V_{rr} = \frac{2|\tan(\alpha)|\sqrt{(\frac{R}{2} + \frac{e}{2})^2 + \frac{l^2}{4}}}{l} V_D \tag{5}$$

where e is the distance between wheels and l denotes the distance between the front and rear axles. The same absolute steering for both front and rear axles given by α was assumed. The desired speed V_D is expressed at the point G (Figure 3) in order to be tangent to the car trajectory.

4 Velocity Planner Module

The VP module estimates the linear reference velocity in which the vehicle should travel (which can be the maximum value allowed or not), as well as determines the next reference trajectory point. One main objective taken into account was to make the travel as comfortable as possible, i.e. to give the system the capability of fully control the smoothness of the acceleration profile either lateral, or longitudinal.

A Canadian study used a testing ground and an highway to test speed and lateral acceleration on both wet and dry pavement on horizontal curves. They found that “comfortable lateral acceleration” and “speed environment” limited a driver’s speed, while pavement surface conditions (dry or wet) and the driver’s gender did not. Drivers adjusted their comfortable speed according to their comfortable lateral acceleration tolerance, approximately between 0.35 and 0.40g [5]. Another study revealed the comfortable Longitudinal acceleration [6] i.e. Steady Deceleration Under expected-stop conditions. Research shows that drivers generally exert an average steady braking force of -0.35 g. This amount of braking force seems comfortable for drivers.

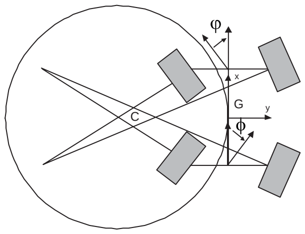

Figure 3: *Vehicle Kinematic Model.*

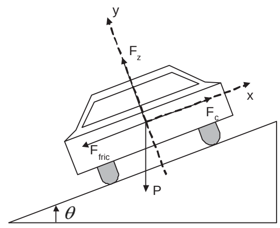

Figure 4: *Acting forces on the vehicle.*

The previous accelerations were used to set up the reference velocity, see in Figure 5 the **"Max Acc Comfort Constrained"** and the **"Max Vel Comfort Constrained"**, although a maximum velocity still had to be computed to cope with unexpected situations.

To estimate the maximum velocity in which the vehicle can navigate without slipping, it is necessary to know the forces that actuate on the vehicle, see Figure 4. In a general way, the forces that actuate on the vehicle are the horizontal forces, the wheel ground contact forces, the force that the vehicle performs on the ground and the wind force over the vehicle (air resistance). In this document, the wind force will not be taken into account in the calculation of the velocity. The friction force is proportional to the normal reaction, being the proportionality factor the friction coefficient (static or dynamic):

$$F_{fric} = \mu F_z. \tag{6}$$

The force that the vehicle carries out in the ground (weight) is proportional to its mass,

$$P = mg. \tag{7}$$

Furthermore there is the centrifugal force that actuates on the vehicle when it describes a curvilinear trajectory,

$$F_c = m\frac{v^2}{r} \tag{8}$$

where m is the vehicle mass, v its velocity and r the $Curvature$ radius.

The vehicle, when describing a curvilinear trajectory, must not slide (either the inside or outside of the $Curvature$). So, in order to avoid vehicle sliding, the sum of the forces, along x axis must be null:

$$F_c - F_{fric} - P\sin(\theta) = 0 \tag{9}$$

From equations (6 - 9), the maximum velocity, is expressed as follows:

$$v_{max} = \sqrt{rg(\mu\cos(\theta) + \sin(\theta))} \tag{10}$$

In Figure 5 v_max is denoted by **"Max Vel Slip Constrained"**. The main difficulty in the calculation of the reference velocity is the friction coefficient. This parameter must be estimated with a significant precision because all the process depends on it. We used the following equation to estimate it:

$$\mu(S_{Res}) = c_1(1 - e^{-c_2 S_{Res}}) - c_3 S_{Res}) \tag{11}$$

$$(e^{-c_4 S_{Res} v_{CoG}}(1 - c_5 F_z^2)$$

where S_{Res} is the resultant slip and the constants c_i ($i = 1..5$) are the characteristic parameters of various types of road [9].

The reference velocity (**"Max Vel Comfort Constrained"**) determines the intended vehicle behaviour applied in the Vehicle Motion and it has smooth variations. The maximum velocity (**"Max Vel Slip Constrained"**) as much more ruggedness profile, see Figure 5. In order to fulfill the maximum acceleration without slipping, or the maximum comfort acceleration constraints, the vehicle should start braking in advance being more restrictive for the comfort profile. The profit of being more restrictive is a smoother variation for the Comfort profile.

5 Path Tracking Module

The PTC is divided in four independent modules: Steering , Rear Steering Switch, Velocity Control and Look Ahead Distance (see Figure 6).

- The Steering module computes the steering command. The purpose is to minimize both the orientation error E_θ, and the lateral error E_L. Additionally a Steering Increment ($SteerringInc$) fuzzy variable is computed in order to achieve a faster recovery from an undesired pose. The variable $SteerringInc$ is the output of a fuzzy module which as has inputs the $Curvature$ and the ΔE_θ. This module ensures a geometrical convergence towards the path to be followed.

- The Rear Steering module decides whether the rear axle should steer to opposite direction of the front axle (Dual Mode) or in the same direction

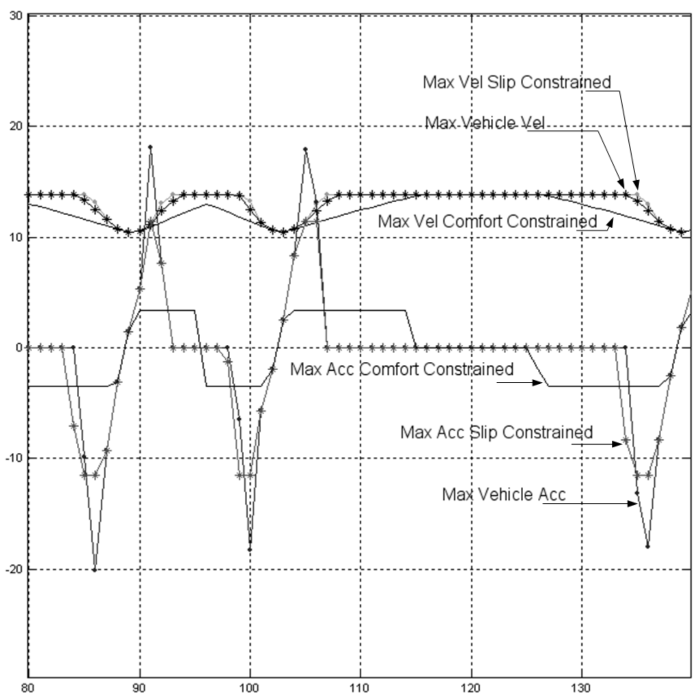

Figure 5: *Acceleration Profiles.*

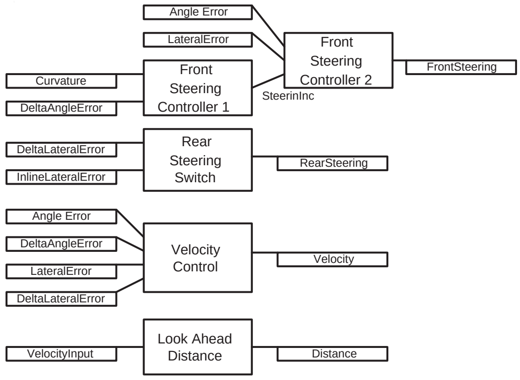

Figure 6: *Path Tracking Structure.*

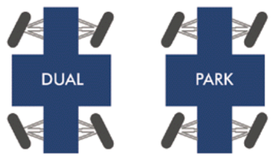

Figure 7: *Driving Modes.*

(Park Mode), (Figure 7). The inputs for this module are the ΔE_L and the Inline Lateral Error (ILE), given by $ILE = |\frac{E_L(D)}{E_L(R)}| + |E_\theta|$, where the $E_L(D)$ is the lateral error at the VP position and $E_L(R)$ is the lateral error at the vehicle position. This module steers the rear wheels in the same direction as the front wheels; the result is a decreasing of the vehicle yaw motion. The yaw motion is necessary for executing a manoeuvre but is undesired for the stability of the vehicle [8].

- The inputs of Velocity Control module are ΔE_L, E_L, ΔE_θ and E_θ. This module computes a weight factor assigning a level of significance to the reference velocity, i.e. if the errors have an high magnitude then the velocity must be decreased, otherwise the reference velocity is applied.

- The Look Ahead module computes a distance D in front of the vehicle called the VP. This distance is function of the vehicle velocity, therefore the input for this module is the vehicle velocity.

5.1 Fuzzy Input Sets

The Lateral Error (LE) membership function is a fifth-pronged triangular membership functions and two trapezoidal membership functions, representing lateral distances conditions: High Negative LE (HNLE), Negative LE (NLE), Low Negative LE (LNLE), Zero LE (ZLE), Low Positive LE (LPLE), Positive LE (PLE) and High Positive LE (HPLE). The Angle Error (AE) membership function is similar to the previous one, representing angle conditions: High Negative AE (HNAE), Negative AE (NAE), Low Negative AE (LNAE), Zero AE (ZAE), Low Positive AE (LPAE), Positive AE (PAE) and High Positive AE (HPAE).

The Differential LE (DLE) membership function is a one-pronged triangular membership functions and two trapezoidal membership functions, representing the dynamics of the LE: LE Decreasing (DDLE), LE Constant (CDLE) and LE Increasing (IDLE). The Differential AE (DAE) membership function is similar to the previous one, representing the dynamics of the AE: AE Decreasing (DDAE), AE Constant (CDAE) and AE Increasing (IDAE).

The *Curvature* (C) membership function is a one-pronged triangular membership function and two trapezoidal membership functions, representing the curvature magnitude: Zero (ZC), Medium (MC) and High (HC). The *InlineLateralError* (ILE) membership function is similar to the previous one, representing the ILE magnitude: Zero (ZILE), Medium (MILE) and High (HILE). The *VehicleVelocity* (VV) membership function is similar to the previ-

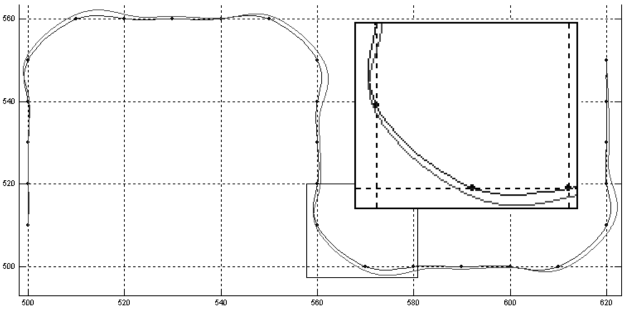

Figure 8: *Test Track Result of the Lateral Controller without Virtual Point (x-axis and y-axis in meters).*

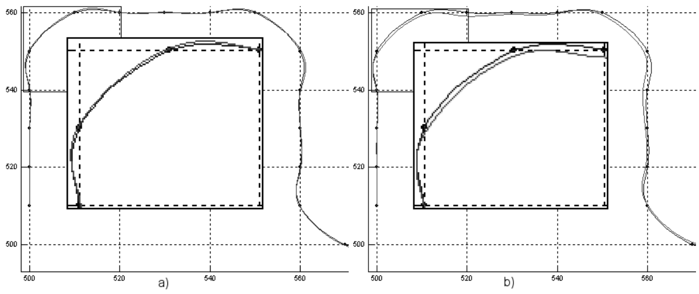

Figure 9: *Test Track Result of the Lateral Controller with Virtual Point(x-axis and y-axis in meters).*

ous one, representing the Vehicle Velocity magnitude: Zero (ZVV), Medium (MVV) and High (HVV). The *SteeringIncrement* (STI) membership function is similar to the previous one, representing the Increment magnitude of the Steering : Zero (ZSTI), Medium (MSTI) and High (HSTI).

5.2 Fuzzy Output Sets

The *FrontSteering* (FS) membership function is a fifth-pronged triangular membership function and two trapezoidal membership functions, representing the Front Steering commands: High Negative FS (HNFS), Negative FS (NFS), Low Negative FS (LNFS), Zero FS (ZFS), Low Positive FS (LPFS), Positive FS (PFS) and High Positive FS (HPFS).

The *RearSteering* (RS) membership function is a two non-overlapping trapezoidal membership functions, representing which driving mode is triggered: Dual Driving Mode (DDMRS) and Park Driving Mode (PDMRS).

The *VelocityWeight* (V) membership function is a one-pronged triangular membership function and two trapezoidal membership functions, representing the Velocity Weight magnitude: Zero (ZV), Medium (MV) and High (HV). The *LookAheadDistance* (LAD) membership function is similar to the previous one, representing the LAD magnitude: Zero (ZLAD), Medium (MLAD) and High (HLAD).

5.3 Fuzzy Knowledge Base

The following rules constitute part of the knowledge base of the PTC and express how the system should react:

- Front Steering Module- *1-If LE is HNLE and AE is HNAE and STI is HSTI then FS is HPFS; 2-If LE is LNLE and AE is LPAE and STI is ZSTI then FS is LNAE; 3-If LE is LNLE and AE is HNAE and STI is HSTI then FS is PFS; 4-If LE is MNLE and AE is ZAE and STI is ZSTI then FS is MPFS.*
- Rear Steering Switch Module- *1-If DLE is IDLE and ILE is ZILE then RS is PDMRS; 2-If DLE is IDLE and ILE is MILE then RS is DDMRS; 3-If DLE is IDLE and ILE is HILE then RS is DDMRS; 4-If DLE is DDLE and ILE is HILE then RS is DDMRS.*
- Velocity Control Module- *1-If LE is HNLE and AE is ZAE and DLE is IDLE and DAE is CDAE then V is MV; 2-If LE is ZLE and AE is ZAE and DLE is IDLE and DAE is IDAE then V is MV; 3-If LE is ZLE and AE is ZAE and DLE is CDLE and DAE is CDAE then V is HV; 4-If LE is HNLE and AE is HNAE and DLE is CDLE and DAE is CDLE then V is ZV;*
- Look Ahead Module *1-If VV is ZVV then V is ZV; 2-If VV is MVV then V is MV; 3-If VV is HVV then V is HV.*

6 Results

The control point can be defined at the center of gravity of the vehicle, as shown in section 6.1, but it is also possible to define it in front of the vehicle, with help of a virtual point (section 6.2).

6.1 Lateral Controller without Virtual Point

Figure 8 shows the simulation without Virtual Point, where it can be observed that the response has a damping behaviour. Although it has a good performance, still has some drawbacks. When the car approaches the curve it has a low lateral error E_L and a high orientation error E_θ and when it leaves the curve it has a high lateral error E_L and a low orientation error E_θ, which means a slower recovery from narrow curves.

6.2 Lateral Controller with Virtual Point

The Virtual Point it is chosen at a so called **"lookahead "** distance D in front of the vehicle. D is a function of vehicle velocity, which must be chosen in order to guarantee that:

- D is not too small: otherwise, the vehicle might reach the target point between two computations, or oscillations might appear.

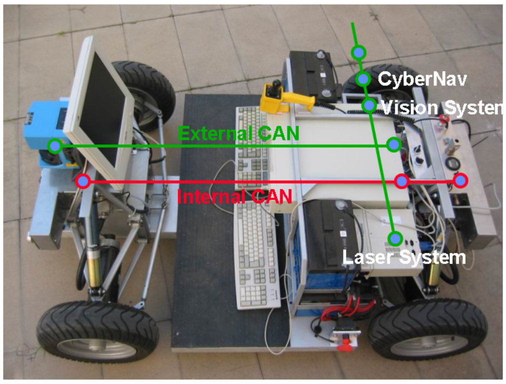

Figure 10: *CAN based Modular Robucar Architecture.*

- D is not too big: if this condition is not fulfilled, the vehicle might cut corners.

The look-ahead distance is variable and depends on the vehicle speed. Figure 9 shows the behaviour for a constant distance 9-b) and for velocity dependent distance 9-a). For a too big distance D, the vehicle cuts the curves.

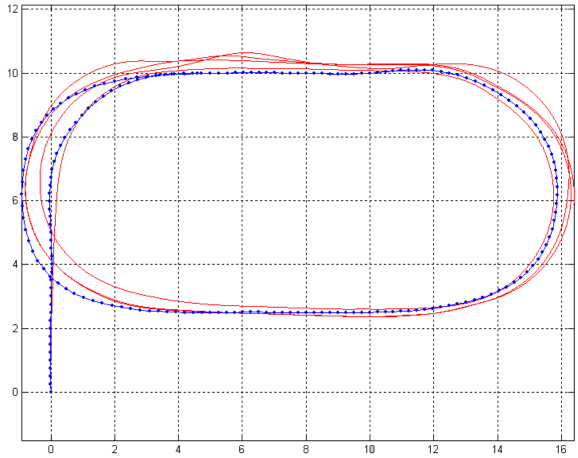

Figure 11: *Experiment relying only on odometry data (x-axis and y-axis in meters).*

7 Conclusions

In this paper we present a strategy for path tracking. The PTC showed good performance and smooth path tracking in simulation, this is due to the smooth transitions between rules and the smooth interpolation between different actions. It has been demonstrated that, as the velocity increases, the damping factor of the closed loop system gets worse and can be improved, under certain limits, by increasing the look-ahead distance, which improves the performance in sharp curves.

Experimantal results using a Robucar (manufactured by Robosoft see Figure 10) is illustrated in Figure 11. The dots represent the path to be followed and the lines represent the real path followed by the Robucar.

Acknowledgments

This work was partially supported by EU (European Commission) project CyberCars (IST-2000-28487) and FCT (Portuguese Science and Technology Foundation) project POSI/41618/SRI/2001.

References

[1] C. Laugier and M. Parent, “Automated Vehicles for Future Urban Transport Systems”, *Int. Conference on Field and Service Robotics*, 29-31, Pennsylvania, 1999.

[2] Dongbing Gu and Huosheng Hu, “Neural Predictive Control for a Car-like Mobile Robot”, *Int. Journal of Robotics and Autonomous Systems*,Vol. 39, 2-3, 2002.

[3] Jacky Baltes and Robin Otte, “A Fuzzy Logic Controller for Car-like Mobile Robots”, *Int. Symposium on Computation Intelligence in Robotics and Automation*, Monterrey, 1999.

[4] Jiri Sika and Joop Pauwelussen, “Entering of Automated Platoon”, *Int. Symposium on Advanced Vehicle Control*, Japan, 2002.

[5] Emmanuel Felipe and Francis Navin, “Canadian Researchers Test Driver Response to Horizontal Curves”, *Road Management & Engineering Journal TranSafety, Inc.*, 1, 1998.

[6] “Simulated On-the-Road Emergencies Used to Test Stopping Sight Distance Assumptions”, *Road Management & Engineering Journal TranSafety, Inc.*, 18, 1997.

[7] S. Sekhavat and J. Hermosillo, “The Cycab Robot: a Differentially Flat System”, *In Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, 312-318, Japan, 2000.

[8] J. Sika, J. Hilgert, T. Bertram, J. P. Pauwelussen and M. Hiller, “Test Facility for Lateral Control of Scaled Vehicle in an Automated Highway System”, *8th Mechatronics Forum Int. Conference - Mechatronics*, 24-26, Netherlands, 2002.

[9] U. Kiencke, L. Nielsen, “Automotive Control Systems”, *SAE-Society of automotive Engeneering*, ISBN 3-540-66922-1.