

Path-Tracking Controller with an Anti-collision Behaviour of a bi-steerable Cybernetic Car

Abel Mendes, Luis Conde Bento and Urbano Nunes

Institute for Systems and Robotics

University of Coimbra-Polo II 3030-290 Coimbra, Portugal

{abfm, conde, urbano }@isr.uc.pt

Abstract—This paper presents a path-tracking controller of a bi-steerable cybernetic car with an anti-collision behaviour. The velocity planner and the anti-collision system are fundamental modules in the architecture. The path tracking implementation uses fuzzy logic. The smoothness of the acceleration profile was one of requirements taken into account in the controller design. The anti-collision system based on Laser Range Data consists of estimating the trajectories and behaviour of surrounding objects. Simulation and experimental results are presented showing the effectiveness of the overall navigation control system.

I. INTRODUCTION

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 [3] using as inputs only the orientation and lateral errors; Jiri Sika and Joop Pauwelussen, using a lookahead point developed a lateral controller to follow the pre-described path [4].

In [11] object tracking with a static sensor is performed. A new approach for object tracking with a moving laser sensor, was developed in [10]. A cooperative approach using stereovision and laser scanner in order to perform robust, and in real-time, detection of multi-obstacles in the automotive context was proposed in [16].

In this paper, a path-tracking controller with anti-collision behaviour of a bi-steerable cybernetic car (Robucar [7], manufactured by Robosoft, shown in Figure 9) is described with some detail. To simplify the presentation, the overall system is divided in two systems, which are designated by navigation system and anti-collision system. The navigation system provides the path-tracking ability, taking into account the vehicle velocity, atmospheric conditions, tyre and road characteristics/conditions, and information regarding the dynamics of the surrounding obstacles provided by the anti-collision system. The navigation system is mainly composed by two modules: Velocity Planner Module (VPM) and Path-Tracking Controller (PTC), as shown in Figure 1. The VPM computes

the passenger comfortable velocity and the maximum linear velocity in which the vehicle can travel. The smoothness of the acceleration profile was considered in the VPM design. The PTC drives the robot along the path while handling disturbances, 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 navigation control architecture.

The anti-collision system is constituted by the following modules: segmentation, multi-object tracking and impact-time computation. This system estimates the position and velocity of existing objects in the environment, thanks to a number of scans provided by a laser measurement system (LMS), based on which the impact-time is computed.

The paper structure is the following. Section 2 presents the navigation system, and section 3 presents the anti-collision system. The hardware architecture used to integrate the modules is summarized in section 4. The results either with or without moving obstacles are discussed in section 5. Finally, in section 6 some concluding remarks are given.

II. NAVIGATION SYSTEM

A. Control Architecture

The control architecture is made up of two main modules: the VPM and the PTC. The VPM 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 weather conditions. In order to follow the trajectory, the actual position computed by odometry (with some re-calibration procedure using landmarks) 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. Additionally the impact-time computed in the anti-collision system (section III) is provided to the PTC. The collision avoidance is made by controlling the vehicle reference velocity, this means reducing or even stopping the vehicle in situations of eminent danger. A position error and a heading error are measured for every new iteration. These errors are calculated in respect to the

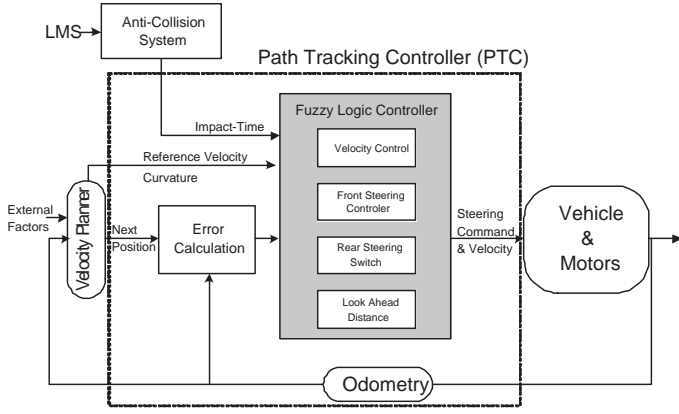


Fig. 1. Navigation system architecture, with an anti-collision behaviour.

coordinates 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 coordinates 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 coordinates 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

$$\Delta E_L = E_L(k) - E_L(k-1), \quad (1)$$

and

$$\Delta E_\theta = E_\theta(k) - E_\theta(k-1), \quad (2)$$

and the *curvature* radius are additional inputs to the Fuzzy Logic controller.

B. 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 vehicle kinematics, motors and odometry models were considered in the simulations. They are vehicle dependent and their relevant parameters are described in this section. The kinematic model of the car, for a reference frame located at mid point of the rear axle, is given by the following equations:

$$\dot{x} = v \cos(\theta + \phi) \quad (3)$$

$$\dot{y} = v \sin(\theta + \phi) \quad (4)$$

$$\dot{\theta} = v \frac{\sin(\varphi \times \phi)}{L \cos(\varphi)} \quad (5)$$

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

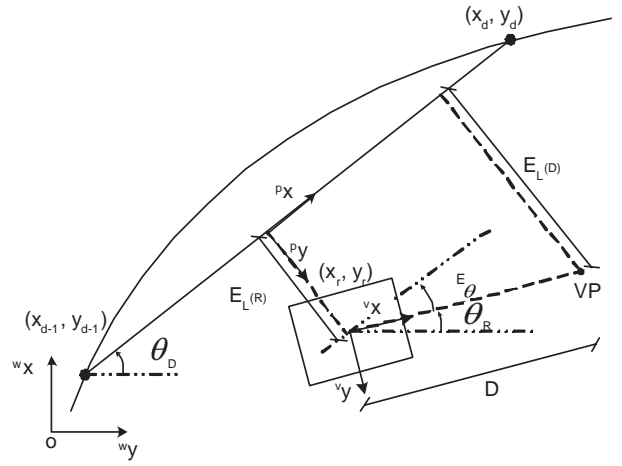


Fig. 2. E_L and E_θ - Position and heading errors.

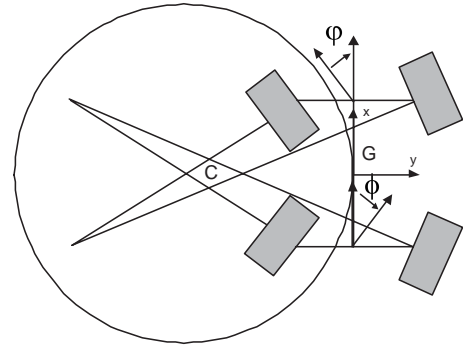


Fig. 3. Vehicle Kinematic Model.

The speeds for each wheel, are calculated as follows:

$$V_{lf} = V_{lr} = \frac{2|\tan(\alpha)|\sqrt{(\frac{R}{2} - \frac{e}{2})^2 + \frac{l^2}{4}}}{l} V_D \quad (6)$$

$$V_{rf} = V_{rr} = \frac{2|\tan(\alpha)|\sqrt{(\frac{R}{2} + \frac{e}{2})^2 + \frac{l^2}{4}}}{l} V_D \quad (7)$$

where e is the distance between wheels and for example V_{lf} denotes the velocity of the left front wheel. 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.

C. Velocity Planner Module

The VPM 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.

The estimation of the maximum velocity in which the vehicle can navigate without slipping is expressed as follows

[13]:

$$v_{max} = \sqrt{rg(\mu \cos(\theta) + \sin(\theta))} \quad (8)$$

where r , g , θ and μ are the curvature radius, gravity force, vehicle roll angle and friction coefficient, respectively.

The main difficulty in the calculation of the reference velocity is the friction coefficient, which must be estimated with a significant precision. We used the following equation to estimate it:

$$\mu(S_{Res}) = c_1(1 - e^{-c_2 S_{Res}}) - c_3 S_{Res}(e^{-c_4 S_{Res} v_{COG}}(1 - c_5 F_z^2)) \quad (9)$$

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].

D. Path-Tracking Controller

The PTC is divided in four independent modules: Steering, Rear Steering Switch, Velocity Control and Lookahead Distance (see Figure 4). The anti-collision system provides the impact-time as input to the PTC. In order to properly avoid collision with obstacles the impact-time is integrated in the velocity control module.

- The Front Steering module (composed by controller 1 and 2) 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 (*SteeringInc*) fuzzy variable is computed in order to achieve a faster recovery from an undesired pose. The variable *SteeringInc* is the output of a fuzzy module which has as inputs the *curvature* and the ΔE_θ . This module ensures a geometrical convergence towards the path to be followed.
- The Rear Steering Switch module decides whether the rear axle should steer to opposite direction of the front axle (Dual Mode) or in the same direction (Park Mode). The inputs for this module are the ΔE_L and the Inline Lateral Error (ILE), given by $ILE = \left| \frac{E_L(D)}{E_L(R)} \right| + |E_\theta|$, 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 the Velocity Control module are ΔE_L , E_L , ΔE_θ , E_θ and the *time of impact*. This module computes a weight factor assigning a level of significance to the reference velocity, i.e. if the errors have a high magnitude or the impact-time as a low magnitude then the velocity must be decreased, otherwise the reference velocity is applied. This module is of extreme importance since the collision avoidance is decided here, i.e. if the impact-time is small, then the vehicle velocity is reduced or null, otherwise if the impact-time is high the vehicle velocity (weight factor) is not affected by this variable.

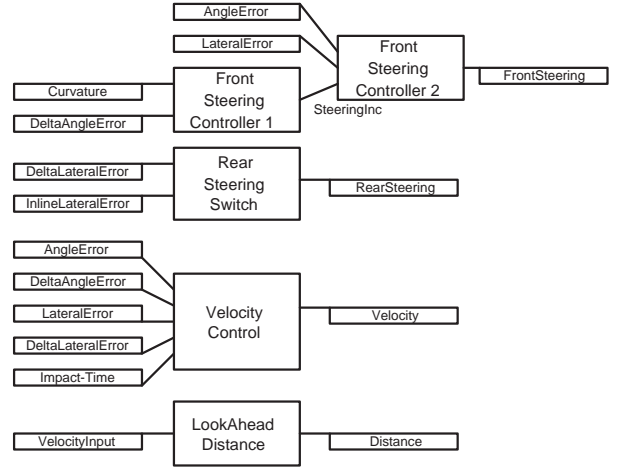


Fig. 4. PTC Structure.

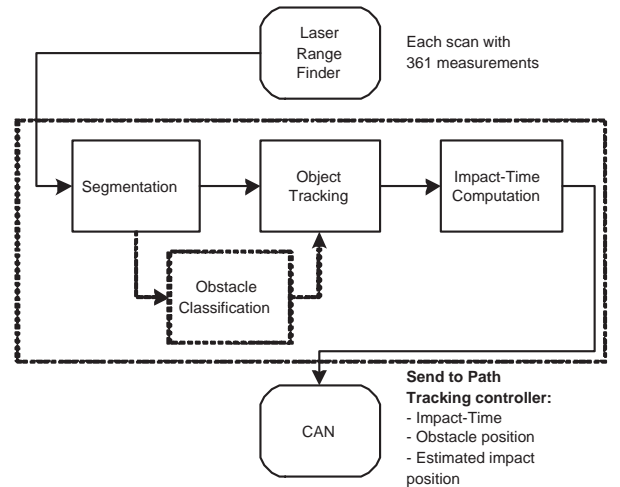


Fig. 5. Anti-collision system architecture.

- The Lookahead module computes a distance D to a point in front of the vehicle called the VP. This distance is function of the vehicle velocity.

III. ANTI-COLLISION SYSTEM

The block diagram in Figure 5 describes the dataflow between the modules, that constitute the anti-collision system: Segmentation, Object Tracking, Obstacle Classification and Impact-time Computation.

A. Segmentation

The goal here is to identify the limits of possible existing objects detected by the LMS and if so, to filter and provide additional information about the object in analysis. The readings are subdivided into small sets of neighbour points (segments), taking into account the proximity between two consecutive points of the scan. A segment is, hence, a set of measurement values (points of the scan) close enough to each other, which due to their proximity, probably belong to the same object. The segmentation criterion is based on [10]: two

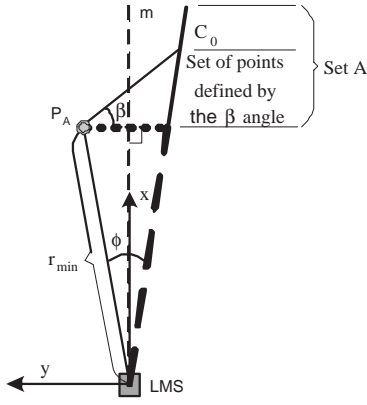


Fig. 6. Illustration of the segmentation algorithm. Set A - Set of values that the farthest point can take to belong to the same segment as the nearest point PA.

consecutive points, distant from the LMS, r_a and r_b , belong to the same segment as long as the distance between them fulfils the following expression:

$$|r_a - r_b| \leq C_0 + r_{min} \cdot \frac{\sqrt{2} \cdot (1 - \cos \phi)}{\cot \beta \cdot \cos \left(\frac{\phi}{2} \right) - \sin \left(\frac{\phi}{2} \right)} \quad (10)$$

where $r_{min} = \min\{r_a, r_b\}$, and ϕ is the angular resolution of the LMS. Angle β and constant C_0 are parameters of segmentation sensibility refinement. β was introduced to reduce the dependence of the segmentation with respect to the distance between the LMS and the object. If $C_0 = 0$, then β gives the maximum absolute inclination with respect to the perpendicular to the dashed line m (see Figure 6), that an object's face can belong to a segment. Line m is the median line between the two consecutive LMS beams. If r_{min} is short, the set of points defined by the β angle is small and if it is high, the set of points defined by the β angle is high. If β is too short, some objects will not be detected (Figure 7-a)). On the other hand, if β is too high, then a segment can represent more than one object (Figure 7-b)).

After the subdivision of each scan in segments, a selection of several points of the segment and the computation of the visible dimensions of the object take place.

B. Object Tracking

It is necessary to identify the segment-object pair in an unquestionable fashion. For that purpose, the reference point of the detected segments is compared with the reference point estimated with a Kalman filter for each tracked object. For each detected segment, an interest region is defined, in which the search of existing objects is performed [15]. Since the LMS is carried by vehicles it is necessary to consider the movement of the LMS in the tracking of the objects. When performing the Kalman filtering update cycle all the variables involved (including actual and previous state) are expressed in the same reference frame [15].

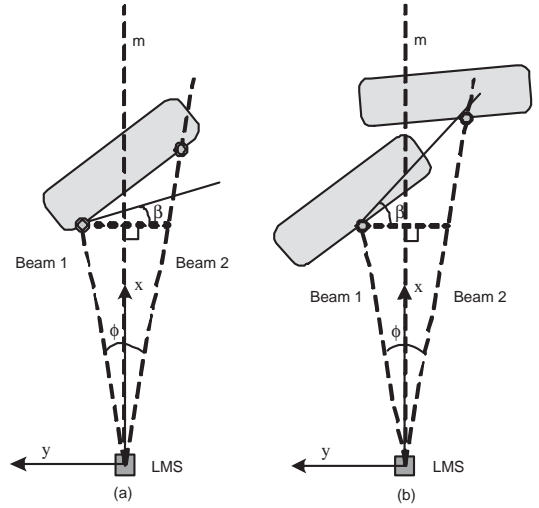


Fig. 7. Different angles (β). (a) β too short - points of the object face don't belong to the same segment. (b) β too high - two points detected from different objects belong to the same segment.

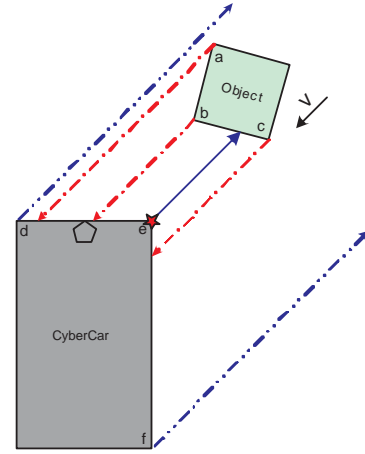


Fig. 8. Geometrical method of impact computation. (vector V represents the obstacle velocity relatively to the CyberCar; edge e will be the impact point)

C. Impact-Time Computation

The Impact-Time Computation module uses the results of all previous ones to estimate the impact-time and position, for each one of all detected objects. The method used is based on the projection of all possible points of impact in the direction of the object's velocity, and for each instant it is assumed a constant object velocity relative to the vehicle. As we can see in Figure 8, these points are the edges of the car (d, e, f) and of the object (a, b, c). So, from the projection lines starting on the object, defined by the starting points and velocity vector (expressed mathematically by Eq. 11), we select the shortest line that intercepts a line segment of the boundary of the vehicle (Eq. 12). Applying the same method for the projection lines that start on the car, we finish the process and achieve the colliding shortest distance.

The interception point between these two lines is achieved equalizing their expressions. Afterwards, we have the starting

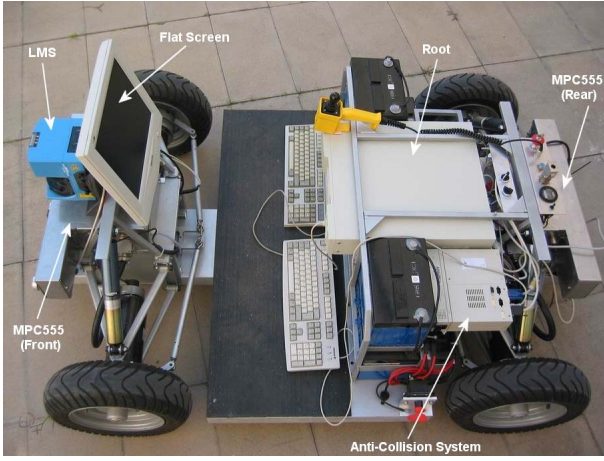


Fig. 9. Robucar prototype

point $P_0 = (x_0, y_0)$ and the interception point $P_i = (x_i, y_i)$, so the Euclidean distance is given by Eq. 13.

$$Y = \frac{V_y}{V_x} \cdot (X - x_0) + y_0 \quad (11)$$

$$Y = \frac{y_2 - y_1}{x_2 - x_1} \cdot (X - x_1) + y_2 \quad (12)$$

$$\Delta s = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2} \quad (13)$$

The object velocity is assumed to be constant (V), therefore the impact-time is obtained from Δs as

$$t = \frac{\Delta s}{V}. \quad (14)$$

IV. HARDWARE CONTROL ARCHITECTURE

The interconnections among all subsystems of our prototype, is made by two CAN buses, “Internal CAN” and “External CAN”. The first one is used in the control motion of the car. More specifically, it makes the connection between the motor controllers (Motorola micro-controller MPC555), the motor encoders and the central computer. On the other hand, the “External CAN” is mainly used to interconnect different sensors and the central computer. In the path tracking experiments, this bus only connects the “anti-collision system” and the central PC named “root”, which runs the “Navigation System” software.

This modular architecture enables an easy integration of new sensors and modules (see Figure 10). Other systems such as a global positioning system and stereo vision obstacle detection [14] are being integrated in the navigation system.

V. RESULTS

The results presented in this section were obtained in simulation and in real experiments, and they show the effectiveness of both anti-collision and navigation systems. The vehicle control point can be defined at the center of gravity of the vehicle, as shown in section V-A, but it is also possible to

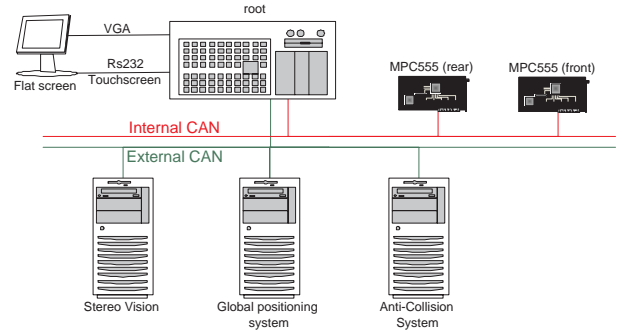


Fig. 10. CAN based Modular Robucar Architecture.

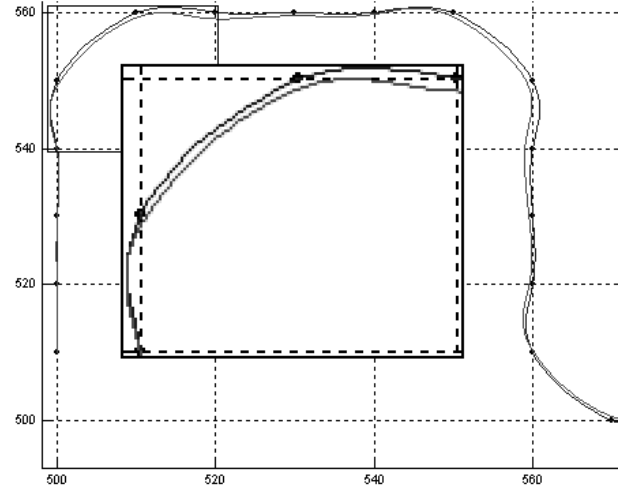


Fig. 11. Test track result of the lateral controller with a constant Virtual Point distance (x-axis and y-axis in meters).

define it in front of the vehicle, with help of a virtual point (section V-B).

A. PTC lateral performance without Virtual Point

The simulation without Virtual Point revealed a response with 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.

B. PTC lateral performance with Virtual Point

The Virtual Point is chosen at a so called “lookahead” distance D in front of the vehicle, which in these experiments was made dependent of the vehicle’s velocity. The variable D should take values in a given interval taken into account the following premises:

- D should be not too small, otherwise oscillations on the vehicle motion might appear.
- D should be not too big. If this condition is not fulfilled, the vehicle might cut curves.

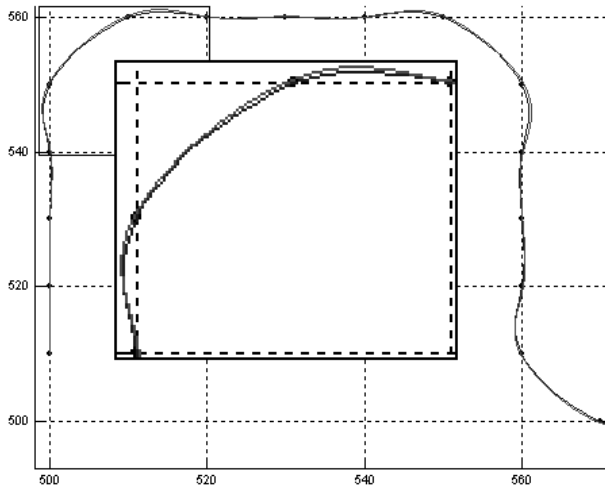


Fig. 12. Test track result of the lateral controller with velocity dependent Virtual Point distance (x-axis and y-axis in meters).



Fig. 13. Experimental environment with planned trajectory (in the real trajectory, the corners are rounded).

The Figure 11 shows the behaviour for a constant lookahead distance, and the vehicle behaviour for a lookahead distance function of the vehicle's velocity can be analyzed in Figure 12.

C. PTC and Anti-Collision System real experiment

The Figure 13 shows a picture of the experimental environment with the planned trajectory drawn on it. Experimental results using a Robucar are illustrated in Figures 14 and 16.

In the Figure 14 the dots represent the path to be followed and the lines represent the real path followed by the Robucar relying only on the odometric values. The first rounds contain low error, but the error is integrated along the run. To cope with this problem, a re-localization system using landmarks is being integrated in the navigation system.

The Figure 16 presents a screen snapshot took in the middle of the experimentation. The small squares represents the reference point of each tracked object and the associated vector represents its velocity. In the situation shown by Figure 16,

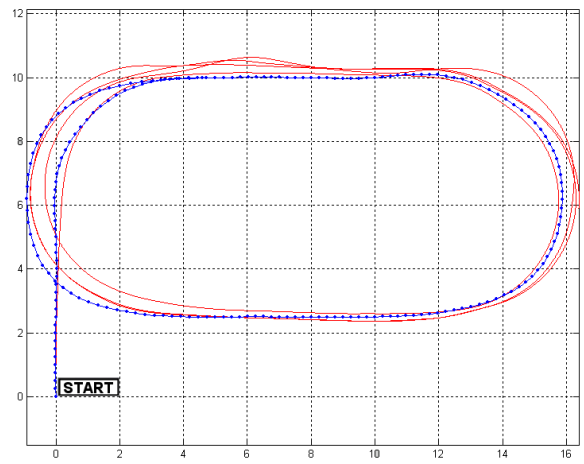


Fig. 14. Experiment relying only on odometry data (x-axis and y-axis in meters).

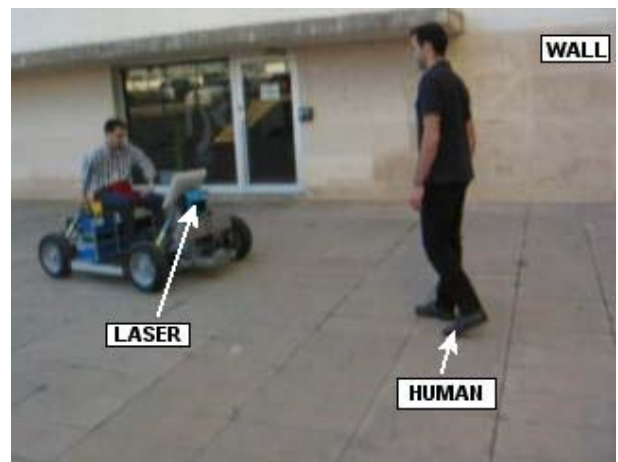


Fig. 15. Real experimentation photo.

two moving obstacles were detected, being one (the human) in the direction of the vehicle. An impact-time of 5 seconds was estimated for the human. Since the impact-time was not critical, the taken action was to decrease the vehicle velocity.

VI. CONCLUSIONS

This paper describes a strategy for path tracking with an anti-collision behaviour. The PTC showed good performance and smooth path tracking in simulation, which 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 lookahead distance, which improves the performance in sharp curves. The impact-time and position estimation, for each one of all detected objects was achieved on both simulations and experiments.

The overall system reacts properly in the presence of moving and static obstacles, the collision avoidance algorithm revealed to be successful, i.e. the vehicle's velocity was gradually reduced while the impact-time was decreasing.

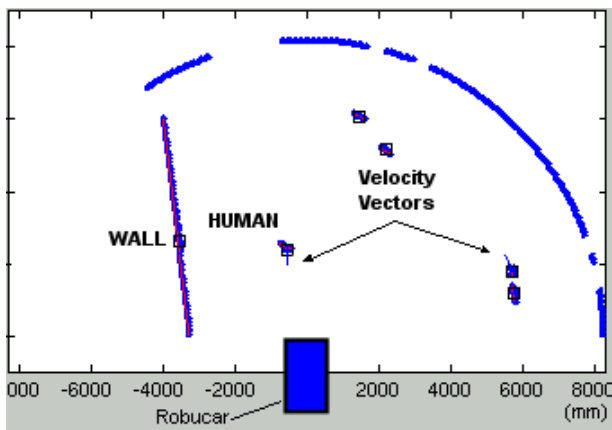


Fig. 16. Snapshot of real experimentation.

ACKNOWLEDGMENT

This work was partially supported by EU (European Commission) project CyberCars (IST-2000-28487) (www.cybercars.org) 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.
- [10] Dietmayer, K. C. J., J. Sparbert and D. Streller, "Model Based Object Classification and Object Tracking in Traffic Scenes", *IEEE Intelligent Vehicle Symposium*, Tokyo, Japan, 25-30, 2002.
- [11] Fod, A., A. Howard and M. J. Mataric, "A Laser-Based People Tracker", *IEEE Int. Conference on Robotics and Automation*, 3024-3029, Washington, 2002.
- [12] J. Patric and H.I. Christensen, "Pose tracking using laser scanning and minimalistic Environment Models", *IEEE Transactions on robotics and Automation*, Vol. 17, n 2, 138-147, 2001.
- [13] Luis Conde Bento, Urbano Nunes, Abel Mendes and Michel Parent, "Path-Tracking of a bi-steerable Cybernetic Car using Fuzzy Logic", *Int. Conf. on Advanced Robotics*, Vol. 3, 1556-1561, Coimbra, 2003.
- [14] Qian Yu, Helder Araújo, Hong Wang, "Stereo-Vision Based Real time Obstacle Detection for Urban Environments", *Int. Conf. on Advanced Robotics*, Vol. 3, 1671-1676, Coimbra, 2003.
- [15] Sérgio Santos, José Eduardo Faria, Fernando Soares, Rui Araújo and Urbano Nunes, "Tracking of multi-obstacles with laser range data for autonomous vehicles", *3º Festival Nacional de Robótica*, Vol. 3, 59-65, Lisboa, 2003.
- [16] R. Labayrade, C. Royere, D. Gruyer and D. Aubert, "Cooperative fusion for multi-obstacles detection with the use of stereovision and laser scanner", *Int. Conf. on Advanced Robotics*, Vol. 3, 1538-1543, Coimbra, 2003.