

Data fusion and path-following controllers comparison for autonomous vehicles

Urbano Nunes · L. Conde Bento

Received: 15 November 2005 / Accepted: 12 March 2006 / Published online: 18 January 2007
© Springer Science + Business Media B.V. 2007

Abstract This paper presents a comparative study of two path-following controllers developed for guiding autonomous vehicles in semi-structured outdoor environments. Part of this paper is focused on the performance of two path-following controllers, which are implemented using two different approaches, the first using fuzzy logic and the second using chained systems theory. The control effort and the errors magnitude along the path are evaluated in a comparative way. A magnetic guidance system for autonomous vehicles navigation in semi-structured outdoor environments is also described, integrating redundant encoders data and absolute positioning data provided by on-board magnetic sensors and magnetic markers buried in the road. Simulation and experimental results are presented showing the effectiveness of the overall control system.

Keywords Data fusion · Path-following · Intelligent vehicles · Fuzzy logic controller · Chained form systems · Magnetic guidance

1 Introduction

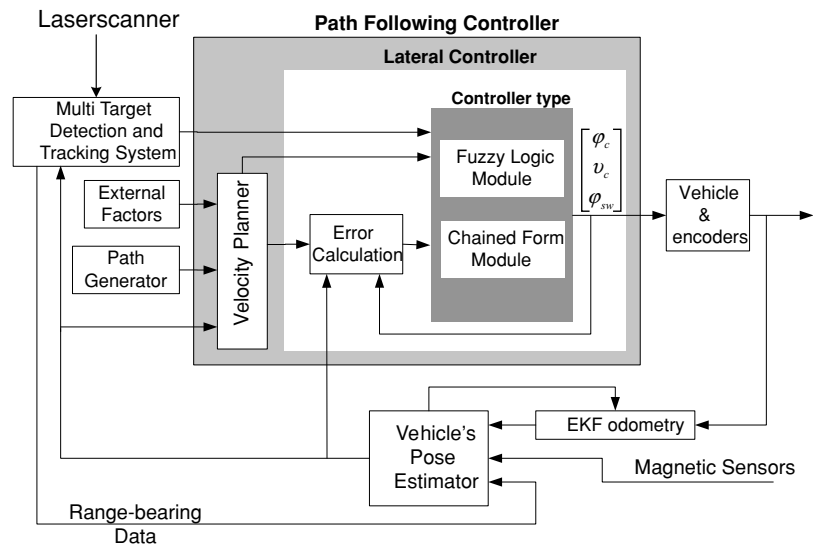
A new approach for mobility providing an alternative to the private passenger car, by offering the same flexibility but with much less nuisances, is emerging, based on fully automated electric vehicles, named cybercars [6, 18]. A fleet of such vehicles might be an important element in a novel individual, door-to-door, transportation system to the city of tomorrow. These vehicles must be user-friendly, easy to handle and functioning with total safety, not only for passengers but also for other road users. These vehicles are already in operation in specific environments featuring short trips at low speed [3, 6].

For fully automated operation, path-following and lateral controllers have been widely investigated, using different control strategies, such as fuzzy-logic, sliding mode and chained form based controllers. In [22], a simplified nonlinear kinematics model is proposed, intended to ease the design and implementation of a stable lateral controller. Fuzzy-logic controllers (FLC) are described in [10, 11]. The design and simulation evaluation of trajectory-tracking and path-following controllers based on sliding mode control is described in [21].

U. Nunes (✉)
Instituto de Sistemas e Robotica & Dep. Eng.
Electrotecnica e de Computadores, Universidade de
Coimbra, 3030-290 Coimbra, Portugal
e-mail: urbano@isr.uc.pt

L. C. Bento
Instituto de Sistemas e Robotica & Dep. Eng.
Electrotecnica, Instituto Politecnico de Leiria 2411-901
Leiria, Portugal
e-mail: conde@isr.uc.pt

Fig. 1 Navigation system architecture. Two lateral controllers were developed. Although only one runs for each set of experiments they are both represented in the figure as fuzzy logic module and chained form module



This paper describes developments of an autonomous navigation system applied on guiding a four-wheel actuated electrical vehicle moving in semi-structured outdoor environments. Its purpose is to provide guidance control with anti-collision behaviour for low-speed vehicles moving in cybercars scenarios. Cybercars [6] have to satisfy challenging requirements like following a path, with high accuracy, in narrow spaces shared with other vehicles and in some areas with pedestrians, providing ride comfort, with low level of jerk and assuring complete safety with human driverless control. A suitable controller has to be chosen fulfilling the previous requirements, which motivated the comparative study of the two lateral control strategies presented in this paper.

Odometry being essential for autonomous navigation is not enough due to its relative and integrative nature. So, it is required to complement odometric data with absolute positioning. Data fusion of ABS sensors and GPS for outdoor localization, based on an Extended Kalman Filter (EKF) had been presented in [4]. Self-localization, given a map of the environment, and the more challenging problem of simultaneous localization and mapping are two examples of key mobile robot problems requiring positioning data. The most commonly used localization probabilistic approaches employ Kalman filtering (e.g. [13]), grid-based Markov localization [9] and Monte Carlo methods [8]. On the other hand, the California Partners for Advanced Transit and Highways (PATH) Program has been given important contributions in the development of a

reference system based on magnets for vehicle lateral guidance/control [24, 27].

1.1 Navigation architecture

The overall navigation system (see Fig. 1) is composed of three main subsystems, which are designated by path-following controller (PFC), vehicle's pose estimator (VPE) and multi-target detection and tracking system (MTDTS). The MTDTS is described in [16], while the PFC and VPE modules are addressed in this paper. The PFC is made up of two main modules: the velocity planner (VP) and the lateral controller (LC). The VP provides local target points of the reference trajectory and computes the maximum and the comfortable velocities, taking into account external factors. The considered external factors are tyre characteristics and passenger comfortable lateral and longitudinal accelerations.

Two lateral controllers, one fuzzy-logic-based and another using chained form theory, were developed which are described and compared in this paper.

2 Kinematics and odometry model

2.1 Kinematics model

A Robucar (manufactured by Robosoft) is used in the autonomous navigation experiments. It is equipped with four wheels, each one driven by an independent

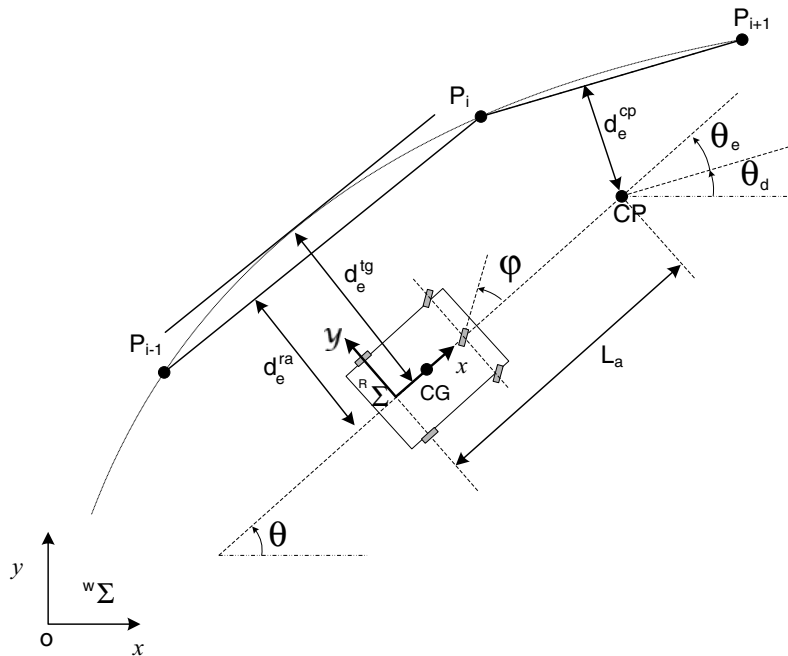


Fig. 2 Illustrative construction of lateral and heading errors (in real situation the discrete path points P_i are much closer between each other, when compared with the vehicle dimensions). P_i denote the points that define the reference trajectory; d_e^{cp} and d_e^{ra} are the lateral errors at CP and rear axle, respectively; d_e^{tg} is the perpendicular distance between the rear axle midpoint and the

current tangent to the path; θ , θ_e^{cp} and θ_d are the orientation of the vehicle, heading error and desired heading, respectively. $^W\Sigma$ and $^R\Sigma$ represent respectively the world coordinate system and the vehicle local coordinate system with its origin at the midpoint of the rear axle and its x -axis aligned with the longitudinal axis of the vehicle

motor equipped with its own encoder. The vehicle has the ability to steer both the rear and the front pair of wheels [19], but in our models and experiments only front steering has been used due to uncertainties on the odometry model of a double steered vehicle. The classical model considers an imaginary wheel at the midpoint of the wheels axles, so that it is oriented in the direction of the steering command.

The configuration of the vehicle can be described without ambiguity by (x, y, θ) (see Fig. 2):

- x and y are the coordinates of the rear axle centre with respect to the $^W\Sigma$ coordinates;
- θ is the vehicle heading with respect to the $^W\Sigma$ coordinates.

The vehicle kinematic equations are derived according to pure rolling, non-slipping and rigid body assumptions. Therefore, a linear velocity vector and instantaneous rotation centre exists at the reference frame located at the midpoint of the rear axle $^R\Sigma$ and the velocity is directed along the vehicle axle. Kinematics models have the property of keeping the steering and

velocity of the vehicle completely decoupled, therefore turning easy the kinematics-based control design. The kinematic model of the vehicle, for a reference frame located at the midpoint of the rear axle $^R\Sigma$, is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ \frac{\tan \varphi}{L} \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2 \tag{1}$$

where v_1 represents the linear velocity of the vehicle, v_2 is the steering angular velocity, L is the distance between the rear and front axles, φ is the front steering angle and θ is the vehicle orientation in the world coordinate system, as depicted in Fig. 2.

A different point of view and more useful in terms of path-following is the one that describes the vehicle behaviour in terms of the path coordinates [14]. Assuming that the vehicle has to follow a path defined by its arc length, one can define the following error variables:

- d_e^{tg} the perpendicular distance between the rear axle midpoint and the current tangent to the path.
- θ_e^{tg} the angle between the current tangent to the path and the x -axis of the vehicle.

Under these assumptions, one can derive the kinematic model in terms of the path coordinates,

$$\begin{bmatrix} \dot{s} \\ d_e^{tg} \\ \theta_e^{tg} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{\cos(\theta_e^{tg})}{1-d_e^{tg}c(s)} \\ \sin(\theta_e^{tg}) \\ \frac{\tan \varphi}{L} - \frac{c(s)\cos(\theta_e^{tg})}{1-d_e^{tg}c(s)} \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2 \quad (2)$$

where $c(s)$ is the curvature of the path.

2.2 Odometry model

Let the vehicle position be represented by the middle point rear axle M with Cartesian coordinates (x_k, y_k) at time t_k , as shown in Fig. 3. The vehicle local coordinate system is defined as having origin M and its x -axis aligned with the longitudinal axis of the car. θ_k is the vehicle heading angle at time t_k . Assuming that the vehicle’s motion is locally circular, its position and

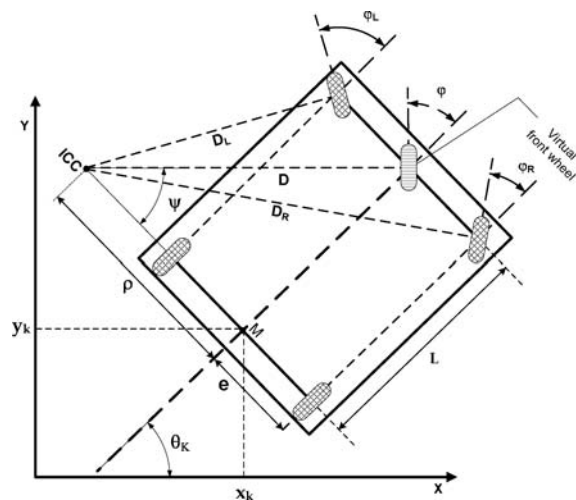


Fig. 3 Vehicle geometrical configuration

Table 1 Vehicle geometrical configuration parameters

L	Car length (distance between rear and front axle)
e	Half-track (half car width)
φ_R	Steering angle from right wheel
φ_L	Steering angle from left wheel
φ	Steering angle of the virtual front wheel
D	Curvature radius of virtual front wheel
ρ	Curvature radius of the rear axle center

orientation at time t_k is given by

$$\begin{cases} x_{k+1} = x_k + \Delta \cos(\theta_k + \omega/2) \\ y_{k+1} = y_k + \Delta \sin(\theta_k + \omega/2) \\ \theta_{k+1} = \theta_k + \omega \end{cases} \quad (3)$$

where Δ is the arc length and ω the elementary rotation. Assuming that there is no wheel slippage and using only data from the rear wheels encoders, then

$$\Delta = \frac{\Delta_{RR} + \Delta_{RL}}{2}, \quad \omega = \frac{\Delta_{RR} - \Delta_{RL}}{2e} \quad (4)$$

where e is the half distance between wheels and Δ_{RR} and Δ_{RL} are calculated using the right and left wheel encoders measurements, respectively.

The vehicle geometrical configuration parameters are illustrated in Fig. 3 and are summarized in Table 1.

3 Path-following controllers

The vehicle can execute a point-to-point stabilization, path-following and trajectory tracking. Point-to-point stabilization requires that the vehicle moves from point A to point B with no restrictions on its movement between these two points. While in path-following, the vehicle must move along a geometric path, in trajectory tracking, the vehicle must move along a geometric path at a given speed.

This paper addresses the path-following problem. The PFC is made up of two main modules: VP and LC. Two LCs are described in this section with the following set of inputs:

- for the chained-form-based controller:

$$u_{CF} = [\theta_e^{tg}, d_e^{tg}, c(s)] \quad (5)$$

– for the fuzzy controller:

$$u_{FL} = [\theta_e^{cp}, d_e^{cp}, \Delta\theta_e^{cp}, \Delta d_e^{cp}, t_{imp}, d_{ile}, c(s), v] \tag{6}$$

where v denotes the linear reference velocity, t_{imp} is the time-to-collision computed in the MTDTS, d_{ile} is the inline lateral error (see Section 3.2.2) and the differential errors Δd_e^{cp} and $\Delta\theta_e^{cp}$, at the control point (CP) are given by

$$\Delta d_e^{cp} = d_e^{cp}(k) - d_e^{cp}(k - 1) \tag{7}$$

and

$$\Delta\theta_e^{cp} = \theta_e^{cp}(k) - \theta_e^{cp}(k - 1) \tag{8}$$

Collision avoidance is achieved by controlling the vehicle’s reference velocity, reducing or even stopping the vehicle in situations of eminent danger. The main goal of the path-following controller is to ensure that the vehicle follows the predefined reference path with appropriate orientation. For the fuzzy logic controller this can be understood as a task of minimizing the vehicle lateral and heading errors (d_e^{cp} , θ_e^{cp}) with respect to the reference path, at a given control point (CP) located at a distance L_a denoted by lookahead distance, as illustrated in Fig. 2.

For both controllers the curvature along the path $c(s)$ is estimated as described in [15]. From the third row of (2) one can obtain a linearly parameterizable system in $c(s)$ written by

$$y = wa \tag{9}$$

where

$$y = \frac{v_1 d_e^{tg} \tan \varphi}{L} - \dot{\theta}_e^{tg} \tag{10}$$

$$w = v_1 \cos(\theta_e^{tg}) + \frac{v_1 d_e^{tg} \tan \varphi}{L} - \dot{\theta}_e^{tg} d_e^{tg} \tag{11}$$

$$a = c(s) \tag{12}$$

Knowing w and y , an estimate of a , i.e. \hat{a} , is obtained using the least squares estimator:

$$J = \int_0^t (y - w\hat{a})^2 d\tau \tag{13}$$

Solving for \hat{a} so as to minimize J , the following update equation for \hat{a} is obtained:

$$\dot{\hat{a}} = P(wy - w^2\hat{a}) \tag{14}$$

where P and its update equation \dot{P} are given by:

$$P = \frac{1}{\int_0^t w^2 d\tau} \tag{15}$$

$$\dot{P} = -P^2 w^2 \tag{16}$$

Each controller has to provide the same control vector ($[\varphi^c, v_c, \varphi_{sw}]$) to the traction control level, where φ^c (in degrees) is the steering angle, v_c ($m\ s^{-1}$) is the velocity command and φ_{sw} is the rear steering switch that controls the two possible driving modes: dual and park modes. In dual mode the rear axle steers in opposite direction of the front axle, while in park mode the rear and front axle steers in the same direction. The chained form controller provides a steering angle velocity command ($\dot{\varphi}^c$) which has to be integrated before being issued to the traction control level (see Fig. 4).

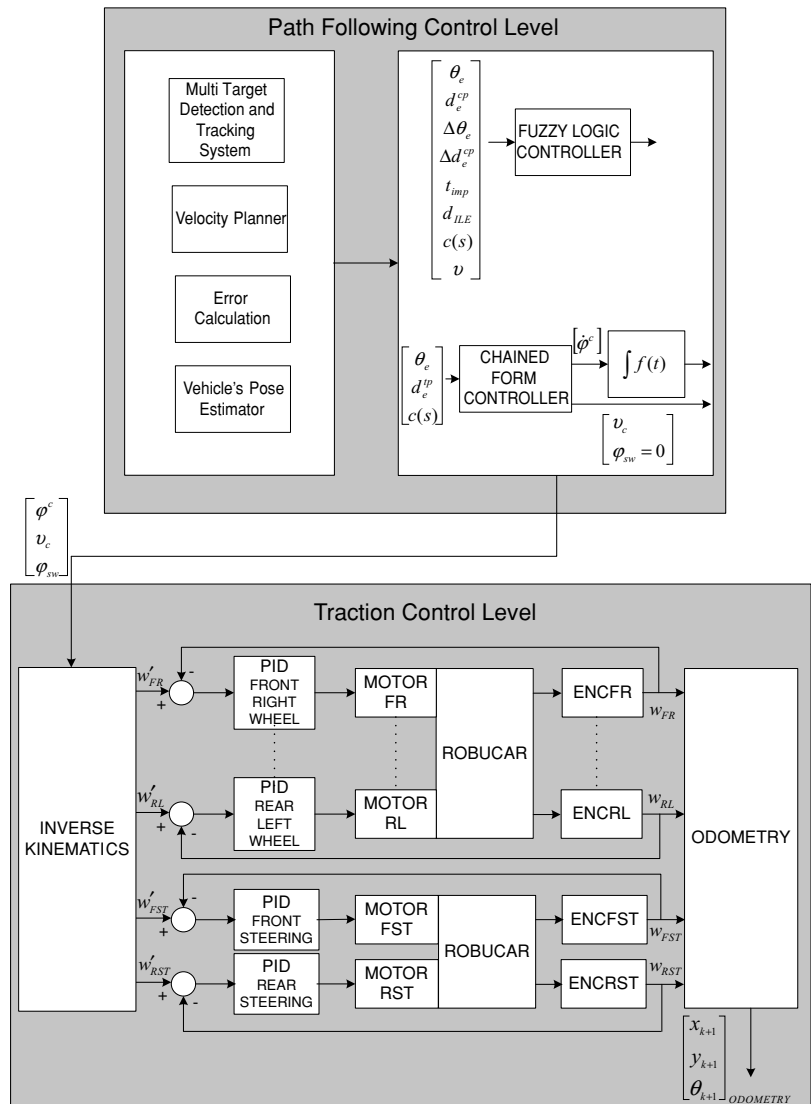
The traction controller is common for any controller type, thus providing modularity to the system architecture. For each wheel and steering axle there is an independent PID controller as shown in Fig. 4, enabling control of wheel slippage and the two possible driving modes: front steered and double steered. Although it is possible to use the double steered option, only front steering was used in the implemented controllers.

3.1 Velocity planner

The VP module calculates the linear reference velocity, as well as determines the local reference trajectory points. One main objective taken into account was to make the trip as comfortable as possible, i.e. to give the system the capability of fully controlling the smoothness of the acceleration profile either lateral or longitudinal.

A Canadian study [7] used a highway testing ground 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 the driver’s speed, while pavement surface conditions (dry or wet) and the driver’s gender did

Fig. 4 Lateral and traction control structures



not. Drivers adjusted their comfortable speed according to their comfortable lateral acceleration tolerance, approximately between 3.43 and 3.92 m s^{-2} . Another study [26] revealed the comfortable longitudinal acceleration, i.e. steady deceleration under expected-stop conditions; drivers generally exert an average steady braking force of -3.43 m s^{-2} . This amount of braking force seems comfortable for most drivers.

The previous acceleration limits were used to set up the maximum comfort acceleration a_{mc} and maximum comfort velocity v_{mc} . The maximum acceleration without slipping a_{mws} and maximum velocity without slipping v_{mws} still had to be computed to cope with unexpected situations. To estimate v_{mws} , it is necessary

to know the forces that actuate on the vehicle, which are basically the horizontal forces, the wheel ground contact forces, the force that the vehicle exerts on the ground and the wind force over the vehicle (air resistance). In this study we consider a plane road and no wind force effects are taken into account.

The friction force is proportional to the normal reaction, where the proportionality factor is the friction coefficient μ (static or dynamic).

Taking into account the previous assumptions one can derive the maximum velocity without slipping

$$v_{mws} = \sqrt{rg(\mu \cos(\psi) + \sin(\psi))} \tag{17}$$

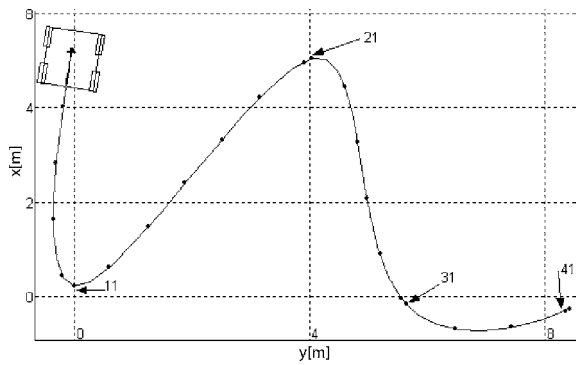


Fig. 5 Vehicle following a given path. In the figure some sampling points are marked (11, 21, 31 and 41) (x -axis and y -axis are in meters)

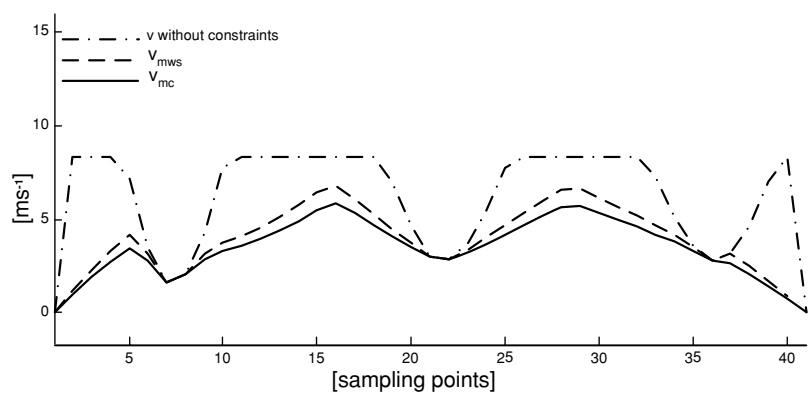
where g is the gravity acceleration, r the curvature radius and ψ denotes the *roll* angle of the vehicle.

An estimate of μ is obtained by the following equation:

$$\mu(S) = (c_1(1 - e^{-c_2S}) - c_3S)e^{-c_4Sv}(1 - c_5F_z^2) \quad (18)$$

where S is the resultant slip and the constants c_i ($i = 1 \dots 5$) are characteristic parameters of various types of road [12]. The velocity v_{mc} determines the intended vehicle velocity used in the vehicle motion. The v_{mws} has a more ruggedness profile, as can be observed in Fig. 6, which shows the velocity profiles corresponding to the example of a vehicle following the path depicted in Fig. 5. In order to fulfil the a_{mws} , or the a_{mc} constraints, the vehicle should start braking in advance being more restrictive for the a_{mc} profile (see Fig. 7). The profit of being more restrictive is a smoother variation on the a_{mc} profile.

Fig. 6 Velocity profiles for the example of Fig. 5 (x -axis denotes the sampling points and y -axis is in m s^{-1})



3.2 Fuzzy logic lateral controller

The fuzzy LC is composed of four independent modules: front steering controller, rear steering switch, velocity command generator and lookahead distance computation (see Fig. 8). In order to properly avoid collisions with obstacles the time-to-impact t_{imp} (also referred here as time-to-collision), provided by the MTDTS, is integrated in the velocity command generator. All modules are fuzzy logic based. Figure 8 shows the LC identifying the fuzzy logic inference flow from the input variables to the output variables.

The fuzzy controller is characterized in Table 2. The fuzzification transforms numerical variables into fuzzy sets, which can be manipulated by the controller. The controller uses fuzzy triangular membership functions and trapezoidal membership functions to encode inputs and outputs. The controller uses min and max connectives and a singleton sum-product inference mechanism. The *center of gravity* defuzzification method was used. Because more than one output term can be evaluated as valid, the defuzzification method must be a compromise between different results. The center of gravity method was chosen because it takes into account, better than any other method, the distribution of the resultant fuzzy set. In this method, the defuzzified value u is a weighted sum of the term membership:

$$u = \frac{\sum_i \mu(x_i)x_i}{\sum_i \mu(x_i)} \quad (19)$$

where x_i is the degree of activation of the i th rule and $\mu(x_i)$ is the output membership function.

The input sets, the output sets, part of the fuzzy knowledge base and some of the membership functions

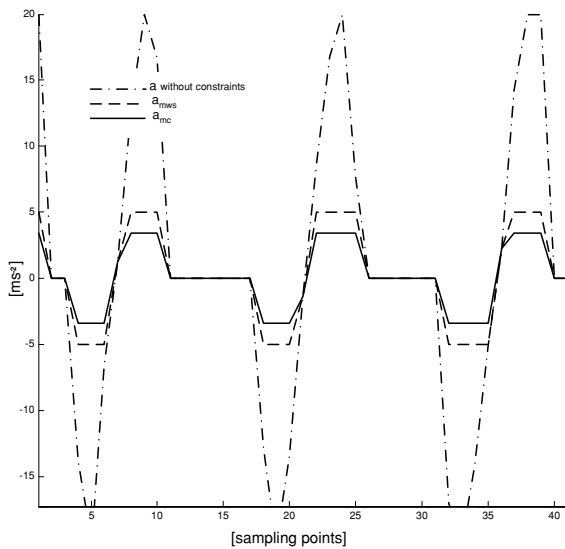


Fig. 7 Acceleration profiles for the example of Fig. 5 (x-axis denotes the sampling points and y-axis is in $m\ s^{-2}$)

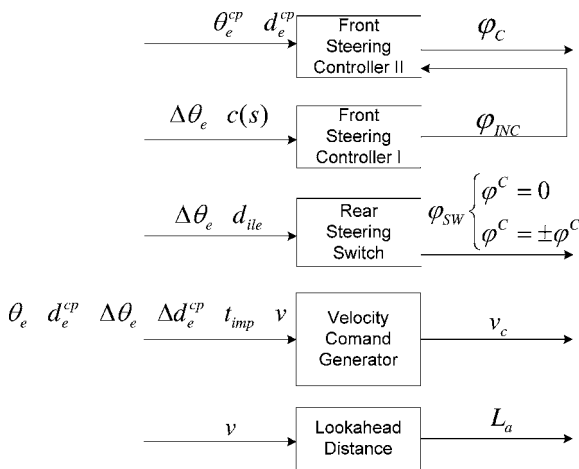


Fig. 8 Fuzzy lateral controller modules

are presented in [2]. The knowledge base of the LC expresses how the system should react, a complete description can be found in [1].

3.2.1 Front steering module

The front steering module computes the steering command φ_c . The purpose is to minimize both the orientation error θ_e^{cp} and the lateral error d_e^{cp} . A steering increment fuzzy variable (φ_{inc}) is computed in order to achieve a faster recovery from an undesirable pose. φ_{inc} is the output of a fuzzy module which has as inputs $c(s)$ and $\Delta\theta_e^{cp}$.

Table 2 Fuzzy controller structure

Fuzzy system	Structure
Input variables	8
Output variables	4
Intermediate variables	1
Rule blocks	5
Rules	615
Membership functions	48

3.2.2 Rear steering switch module

The rear steering switch module enables dual mode or park mode. The inputs of this module are Δd_e^{cp} and the inline lateral error d_{ile} given by

$$d_{ile} = \left| \frac{d_e^{cp}}{d_e^{cg}} \right| + |\theta_e^{cp}| \tag{20}$$

If Δd_e^{cp} is decreasing and d_{ile} is small, this module steers the rear wheels in the same direction as the front wheels; the result is a decrease in the vehicle’s yaw motion. The yaw motion is necessary for executing a manoeuvre but is not desired from the point of view of the vehicle’s stability control [20]. This module was only implemented and tested in simulations.

3.2.3 Velocity command generator module

The inputs of this module are Δd_e^{cp} , d_e^{cp} , $\Delta\theta_e^{cp}$, θ_e^{cp} and the t_{imp} . 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 time-to-collision has a low magnitude then the velocity must be decreased, otherwise the reference velocity is applied. This module is of extreme importance since collision avoidance is decided here, i.e. if the t_{imp} is small, then the vehicle velocity is reduced or the vehicle is even stopped.

3.2.4 Lookahead distance computation module

This module computes the lookahead distance, L_a , which is a function of the vehicle velocity, v . If the velocity increases, the damping factor of the closed loop system gets worse and is improved by increasing the lookahead distance. The lookahead distance provides a prediction behaviour to the controller, since it

enables the control point to be far ahead of the CG of the vehicle, see Fig. 2.

3.3 Chained-form-based lateral controller

The control law designed here, based upon the kinematics model, uses the chained systems theory [5]. Although mobile robot models cannot be linearized, it has been proven that one can convert the nonlinear system in an almost linear system, termed as chained form. A 2-input and n -state chain form system $(2, n)$ has the following structure:

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_2 u_1 \\ &\vdots \\ \dot{x}_n &= x_{n-1} u_1 \end{aligned} \tag{21}$$

The vehicle model from Equation (2) can be converted into chained form using the following change of coordinates and input transformations [14], respectively:

$$\begin{aligned} x_1 &= s \\ x_2 &= -c'(s)d_e^{lg} \tan(\theta_e^{lg}) \\ &\quad - c(s)(1 - d_e^{lg}c(s)) \frac{1 + \sin^2(\theta_e^{lg})}{\cos^2(\theta_e^{lg})} \\ &\quad + \frac{(1 - d_e^{lg}c(s))^2 \tan(\varphi)}{L \cos^3(\theta_e^{lg})} \\ x_3 &= (1 - d_e^{lg}c(s)) \tan(\theta_e^{lg}) \\ x_4 &= d_e^{lg} \end{aligned} \tag{22}$$

$$\begin{aligned} v_1 &= \frac{1 - d_e^{lg}c(s)}{\cos(\theta_e^{lg})} u_1 \\ v_2 &= \alpha_2(u_2 - \alpha_1 u_1) \end{aligned} \tag{23}$$

where $c'(s)$ denotes the derivative of c with respect to s , and

$$\alpha_1 = \frac{\partial x_2}{\partial s} + \frac{\partial x_2}{\partial d_e^{lg}} (1 - d_e^{lg}c(s)) \tan(\theta_e^{lg})$$

$$\begin{aligned} &+ \frac{\partial x_2}{\partial \theta_e^{lg}} \left(\frac{\tan(\varphi)(1 - d_e^{lg}c(s))}{L \cos(\theta_e^{lg})} - c(s) \right) \\ \alpha_2 &= \frac{l \cos^3(\theta_e^{lg}) \cos^2(\varphi)}{(1 - d_e^{lg}c(s))^2} \end{aligned} \tag{24}$$

and the other variables are defined in Fig. 2.

Although the system has two inputs, u_1 and u_2 , this model can be considered single input if u_1 is known *a priori*. Then the objective of the control law is to achieve path-following under the assumption that the vehicle linear velocity u_1 is constant.

The controller was made using the smooth time-varying feedback stabilization method described in [14], where control is either smooth or at least continuous with respect to the robot state.

As a first step, the variables of the chained form are redefined

$$\chi = (\chi_1, \chi_2, \chi_3, \chi_4) = (x_1, x_4, x_3, x_2) \tag{25}$$

resulting in the chained form system

$$\begin{aligned} \dot{\chi}_1 &= u_1 \\ \dot{\chi}_2 &= \chi_3 u_1 \\ \dot{\chi}_3 &= \chi_4 u_1 \\ \dot{\chi}_4 &= u_2 \end{aligned} \tag{26}$$

The above reordering is simply an exchange between the second and fourth coordinates. Path-following is achieved via input scaling, which requires zeroing the χ_2, χ_3 and χ_4 variables, independently from χ_1 . The system (26) is controllable if u_1 is a piecewise continuous, bounded and strictly positive (or negative) function, as stated in [14]. Therefore, u_2 is the only input to the system as long as u_1 is known *a priori*:

$$\begin{aligned} u_2(\chi_2, \chi_3, \chi_4, t) &= -k_1|u_1(t)|\chi_2 \\ &\quad - k_2u_1(t)\chi_3 - k_3|u_1(t)|\chi_4 \end{aligned} \tag{27}$$

The complete deducing of the controller and its background theory are described in [14].

4 Vehicle pose estimator

4.1 EKF odometry

The odometry model in (3) and (4) is based only on the rear wheels encoder. Using also the front wheels encoders, redundant data become available, which can be used to produce better estimates of Δ and ω . The steering angle of left and right wheels can be expressed by

$$\varphi_L = \arctan\left(\frac{\tan(\varphi)L}{L - e \tan(\varphi)}\right) \tag{28}$$

$$\varphi_R = \arctan\left(\frac{\tan(\varphi)L}{L + e \tan(\varphi)}\right) \tag{29}$$

where φ is the steering angle of the virtual front wheel.

From (28) and (29) and knowing that $\Delta = \rho\omega$, a set of equations can be established which relates the encoders measurements (from each of the four wheels and steering) with the parameters Δ and ω [4]:

$$\begin{cases} \tan(\varphi) = L \frac{\omega}{\Delta} \\ \Delta_{RL} = \Delta - e\omega \\ \Delta_{RR} = \Delta + e\omega \\ \Delta_{FL} \cos(\varphi_L) = \Delta - e\omega \\ \Delta_{FR} \cos(\varphi_R) = \Delta + e\omega \end{cases} \tag{30}$$

Therefore, considering $\mathbf{x} = [\Delta, \omega]^T$ and

$$\mathbf{y} = [\tan(\varphi), \Delta_{RL}, \Delta_{RR}, \Delta_{FL} \cos(\varphi_L), \Delta_{FR} \cos(\varphi_R)]^T$$

the state and measurement vectors, respectively, an EKF can be applied as in [4] to estimate the state vector, from the redundant data.

4.2 Odometry confidence tests and simulation

In this section, the EKF odometry described in Section 4.1 is applied. Real data measurements, gathered from Robucar encoders moving along a closed path as depicted in Fig. 9, were used in the reported simulations. The qualitative behaviour of the EKF odometric model is very satisfactory in normal road conditions of adherence as illustrated in Fig. 9(a) and (b). White Gaussian noise was added to the measures with

a signal-to-noise ratio of 10 dB. The EKF filters efficiently coped with the added noise. This result is well illustrated in Fig. 9(c), where it is shown that the EKF odometry approaches very closely the real trajectory (computed with the encoders data without noise), while the trajectory computed from the noisy measurements diverges a lot, as expected.

However, this odometric model does not solve the problem inherent to slippages. If a big slippage occurs, the Kalman filter will not eliminate its effects. This problem can be attenuated by pre-processing the redundant data before providing it to the EKF odometry algorithm. We can compute an approximate motion of the rear wheels based on the motion performed by the front wheels and vice versa, applying the following equations:

$$\begin{aligned} \Delta_R &= \frac{\Delta_{RR} + \Delta_{RL}}{2} & \omega_R &= \frac{\Delta_{RR} - \Delta_{RL}}{2e} \\ \Delta_F &= \frac{\Delta_{FR} + \Delta_{FL}}{2} & \omega_F &= \frac{\Delta_{FR} - \Delta_{FL}}{2e} \end{aligned} \tag{31}$$

$$\Delta_{R/F} = \frac{\Delta_R}{\cos(\varphi)} \quad \Delta_{R/F} = \cos(\varphi)\Delta_F \tag{32}$$

$$\begin{aligned} \Delta_{VRL} &= \Delta_{R/F} - e\omega_F & \Delta_{VRR} &= \Delta_{R/F} + e\omega_F \\ \Delta_{VFL} &= \Delta_{F/R} - e\omega_R & \Delta_{VFR} &= \Delta_{F/R} + e\omega_R \end{aligned} \tag{33}$$

In (31), (Δ_R, ω_R) and (Δ_F, ω_F) are the parameters with respect to the midpoint of the rear axle and front axle, respectively. In (32) and (33), $\Delta_{i/j}$ means Δ_i computed based on measurements from j , with $i, j = \{F, R\}$. Equations (33) express the designated virtual displacements for each wheel. Based on (33) we define the following confidence coefficients:

$$\begin{aligned} CC_R &= 1 - \frac{|\Delta_{VRL} - \Delta_{RL}| + |\Delta_{VRR} - \Delta_{RR}|}{|\Delta_{VRL} + \Delta_{RL} + \Delta_{VRR} + \Delta_{RR}|} \\ CC_F &= 1 - \frac{|\Delta_{VFL} - \Delta_{FL}| + |\Delta_{VFR} - \Delta_{FR}|}{|\Delta_{VFL} + \Delta_{FL} + \Delta_{VFR} + \Delta_{FR}|} \end{aligned} \tag{34}$$

The confidence coefficients are used to decide if a virtual measure ($\Delta_{VRR}, \Delta_{VRL}, \Delta_{VFR}$ and Δ_{VFL}) is used instead of the real measure. Figure 9(d)–(f) shows results of using the algorithm in a simulation of slippages injected on the rear right wheel at $t = 10$ s and $t = 50$ s. As we can see from Fig. 9(d), the EKF with this data

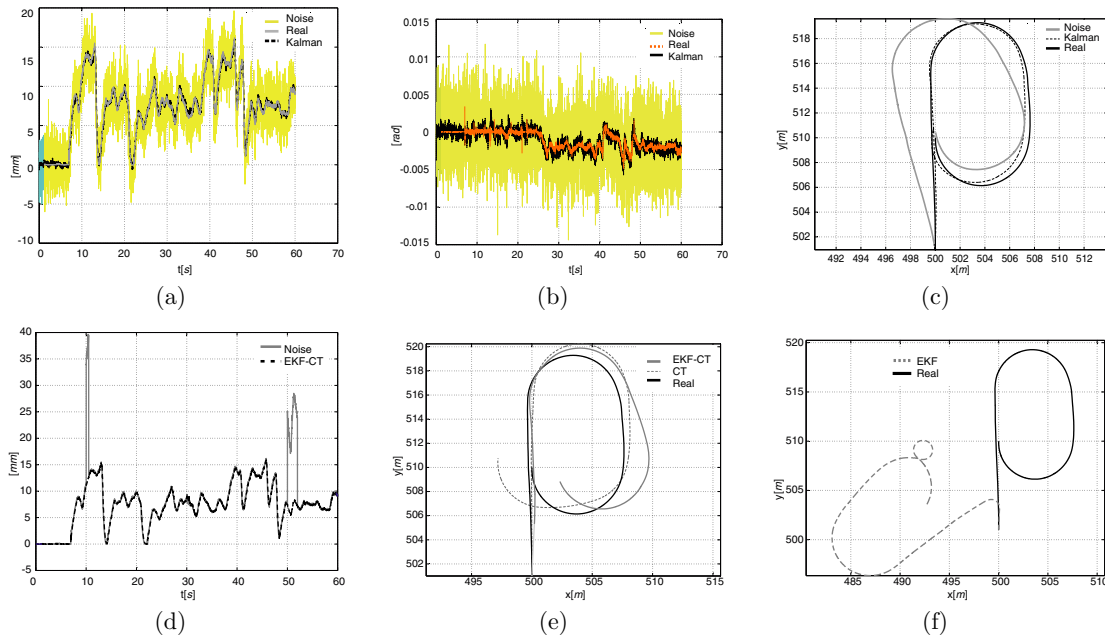


Fig. 9 (a) EKF estimation for Δ (real – encoders measures; noise – measures with white noise; Kalman – obtained estimation); (b) EKF estimation for ω (real – encoders measures; noise – measures with white noise; Kalman – obtained estimation); (c) Odometry results with and without the EKF (real – encoders measures; noise – measures with white noise; Kalman – obtained estimation); (d) Δ estimation with a simulated slippage (noise – encoders measures with a simulated slippage at $t = 10$

and $t = 50$ s; EKF-CT – estimation with EKF-CT); (e) Odometry obtained with CT (CT – path estimated applying the CT and odometry model (3); real – path without the simulated slippage; EKF-CT – path estimated applying the EKF-CT); (f) Odometry obtained with EKF without using CT (real – path without the simulated slippage; EKF – path estimated applying the EKF odometry)

pre-processing, henceforth known as EKF-CT odometry (EKF odometry with confidence tests), will not follow the slippage, hence the algorithm detected a wrong measure and replaced it by the virtual measure, computed based on the other measures. Estimated paths using the EKF-CT and the EKF algorithms are illustrated in Fig. 9(e) and (f). In both, the solid line represents the real path. In Fig. 9(e), the dotted line represents the estimated path using odometry model (3) and confidence tests.

4.3 Fusion of odometry and positioning absolute data

The vehicle is equipped with sensors which provide absolute positioning data: (1) a SICK laser which provides range-bearing data (d, ϕ) associated to visible landmarks; (2) two magnetic sensing rulers, one placed on the front and the other on the rear of the vehicle. The magnetic sensing rulers developed at ISR [17], based on an array of adjacent Hall effect sensors, de-

tect robustly magnetic markers which are placed on the ground defining centerpoints of the path to be followed by the vehicle.

The fusion of odometry data with absolute positioning data is made by means of EKFs. The vehicle’s pose is defined by the Cartesian coordinates (x, y) and heading (θ) , which are the state variables of the EKF. The state variables of the EKF odometry (Section 4.1) are here treated as inputs to the EKF data fusion, i.e $\mathbf{u}_k = (\Delta, \omega)$ with an associated noise covariance matrix Γ_k . The range-bearing measurements associated to each landmark are treated as measurements in the fusion process.

- (1) *System model:* The system model is defined by the kinematic nonlinear equations (3), with state vector $\mathbf{x}_k = [x_k \ y_k \ \theta_k]^T$, and input $\mathbf{u}_k = [\Delta_k \ \omega_k]^T$, which can be written in the compact form (including noises):

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \gamma_{k-1}, \sigma_{k-1}) \tag{35}$$

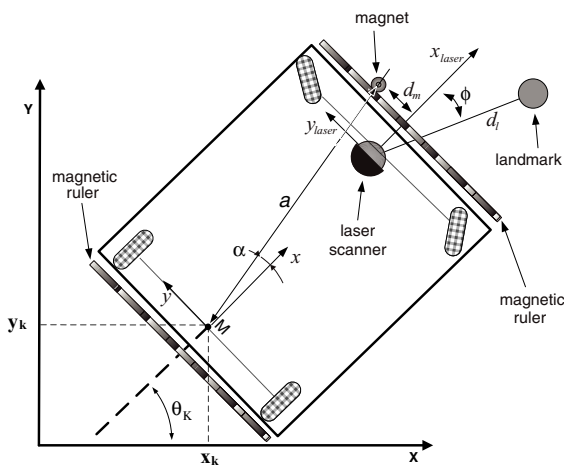


Fig. 10 Measurement model variables. In figure, (a, α) denotes range-bearing data associated to a magnet detection

where γ_k and σ_k denote the system and input noises, with associated matrices \mathbf{Q} and Γ_k .

(2) *Measurement model (example for the front magnetic ruler and laser-based detected landmarks):* let (a, α) be the range-bearing pair, associated to a detected landmark, defined in the local robot coordinate system (see Fig. 10). Thus, the following equations yield

$$a = \sqrt{(y_f - y_k)^2 + (x_f - x_k)^2}$$

$$\alpha = \arctan \frac{y_f - y_k}{x_f - x_k} - \theta_k \tag{36}$$

where (x_f, y_f) represents the Cartesian position of the landmark. From (36) we can define the nonlinear measurement model

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{37}$$

where $\mathbf{h}(\mathbf{x}_k)$ is the nonlinear vector function

$$\mathbf{h}(\mathbf{x}_k) = \begin{bmatrix} \sqrt{(y_f - y_k)^2 + (x_f - x_k)^2} \\ \arctan \frac{y_f - y_k}{x_f - x_k} - \theta_k \end{bmatrix} \tag{38}$$

and \mathbf{v}_k is the Gaussian sensor noise vector with covariance matrix \mathbf{R}_k . The range-bearing data (a, α) are the observation values entering the EKF, $\mathbf{z} = [a \ \alpha]^T$, which are calculated from sensor measures as follows:

(a) *for magnetic marker*

$$a = \sqrt{d_m^2 + L_1^2}$$

$$\alpha = \arctan \frac{d_m}{L_1} \tag{39}$$

where d_m is the magnetic ruler measure which corresponds to the distance between the marker with known position (x_m, y_m) and the magnetic-sensing ruler central point, and L_1 is the distance between the front magnetic ruler and the vehicle rear axis (we are assuming that the ruler is perfectly parallel with the y -axis of robot coordinate system).

(b) *for laser-based detected landmark*

$$a = \sqrt{d_l^2 + L_1^2 + 2d_l L_1 \cos(\phi)}$$

$$\alpha = \arctan \frac{d_l \sin(\phi)}{L_1 + d_l \cos(\phi)} \tag{40}$$

where (d_l, ϕ) are the range-bearing data described in the laser coordinate system. It is assumed that the laser coordinate system is aligned with the robot coordinate system, with a distance L_l , defined in the xy -plane, between them.

Another (non-standard) measurement model has been investigated and applied as described in [23], which consists on considering in model (37):

$$\mathbf{z} = [x_f \ y_f]^T \tag{41}$$

$$\mathbf{h}(\mathbf{x}_k) = \begin{bmatrix} x_k + a \cos(\theta_k + \alpha) \\ y_k + a \sin(\theta_k + \alpha) \end{bmatrix}$$

(3) *EKF algorithm:* It is composed of the following prediction and correction stages:

Prediction stage

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0, 0) \tag{42}$$

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{B}_k \Gamma_{k-1} \mathbf{B}_k^T + \mathbf{Q}$$

where the system (\mathbf{A}) and input (\mathbf{B}) matrices are calculated as the following Jacobian of the

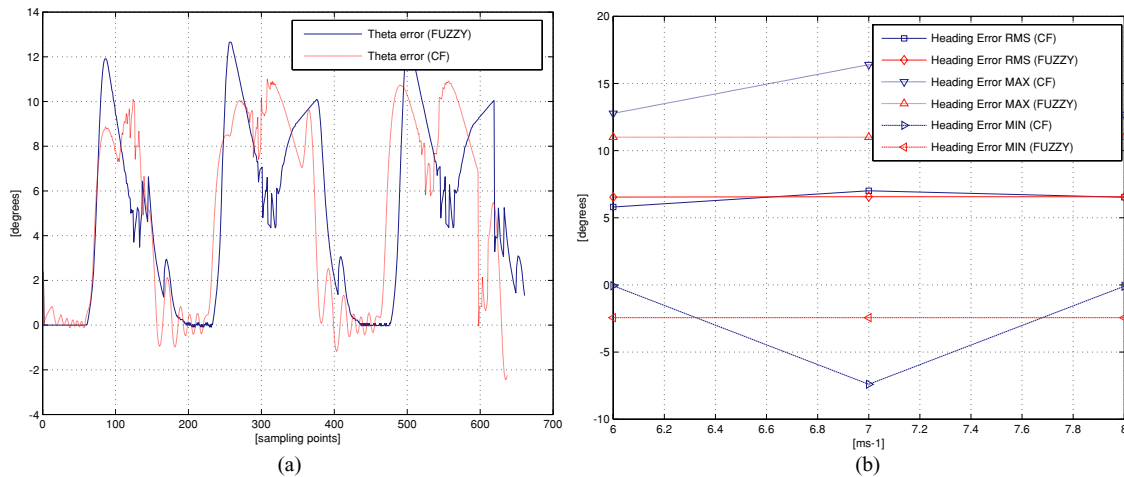


Fig. 11 (a) θ_e^{ig} : Heading error using the chained form controller (dashed line) and heading error using the fuzzy logic controller (solid line) in degrees; (b) root mean square (RMS), maximum

(MAX) and minimum (MIN) heading error (in degrees) using the chained form controller (CF) and the fuzzy logic controller (FUZZY) for three reference velocities (6, 7, 8 ms^{-1})

system $\mathbf{f}(\cdot)$ function:

$$\mathbf{A}_k = \begin{bmatrix} 1 & 0 & -\Delta_k \sin\left(\theta_k + \frac{\omega_k}{2}\right) \\ 0 & 1 & \Delta_k \cos\left(\theta_k + \frac{\omega_k}{2}\right) \\ 0 & 0 & 1 \end{bmatrix} \quad (43)$$

$$\mathbf{B}_k = \begin{bmatrix} \cos\left(\theta_k + \frac{\omega_k}{2}\right) - \frac{\Delta_k}{2} \sin\left(\theta_k + \frac{\omega_k}{2}\right) \\ \sin\left(\theta_k + \frac{\omega_k}{2}\right) \quad \frac{\Delta_k}{2} \cos\left(\theta_k + \frac{\omega_k}{2}\right) \\ 0 & 1 \end{bmatrix}$$

Correction stage

Once measurements (a, α) become available the following correction stage is done:

$$\begin{aligned} \mathbf{S}_k &= (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \\ \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1} \\ \hat{\mathbf{x}}_k &= \mathbf{x}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)) \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \end{aligned} \quad (44)$$

where \mathbf{I} is the identity matrix and \mathbf{H}_k is the Jacobian of the measurement $\mathbf{h}(\cdot)$ function:

$$\mathbf{H}_k = \nabla_x \mathbf{h}(\mathbf{x}_k) \quad (45)$$

4.3.1 Data association

In this work, we have adopted the conventional *nearest neighbour* data association method, using the normalized innovation distance

$$\bar{d}_k = \mathbf{v}_k^T \mathbf{S}_k^{-1} \mathbf{v}_k \quad (46)$$

where \mathbf{v}_k denotes the innovation sequence $\mathbf{v}_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)$ and \mathbf{S}_k its predicted covariance, for accepting/rejecting observations.

5 Simulation results

5.1 Chained form controller vs. fuzzy logic controller

In simulations, the following gains in (27) were used as in [15]: $k_1 = \lambda^3$, $k_2 = 3\lambda^2$ and $k_3 = 3\lambda$ with $\lambda = 5$. The value of λ was obtained iteratively starting from an initial guess $\lambda = 8$.

From Figs. 11, 12, 13 and Table 3 one can observe the effectiveness of both controllers in guiding the car along a predefined path shown in Fig. 14. From Figs. 11(a) and 12(a), it is clear that the chained form controller attempts to reduce the errors with a faster response, but the reduction is only partially achieved since afterwards the errors rise again. The

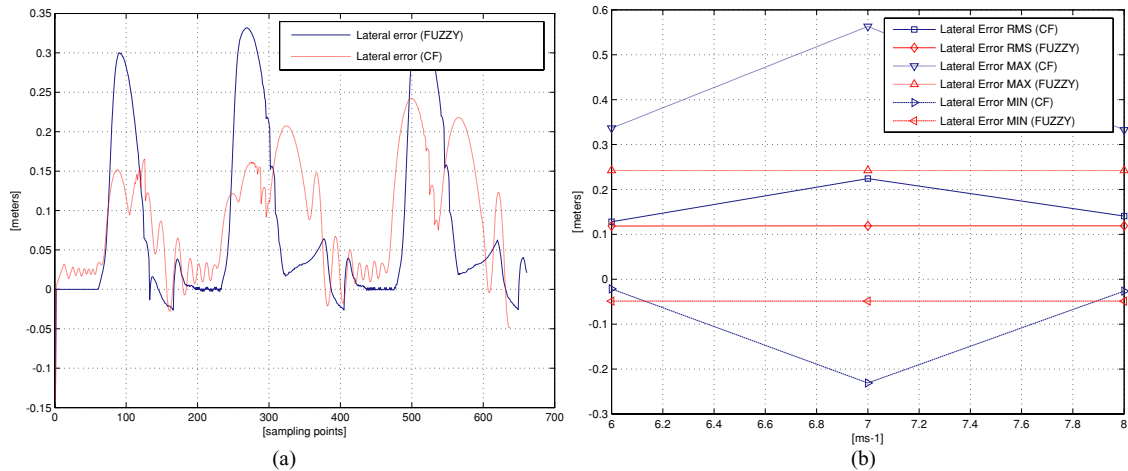


Fig. 12 (a) d_e^{lg} : Lateral error using the chained form controller (*dashed line*) and lateral error using the fuzzy logic controller (*solid line*) in meters; (b) root mean square (RMS), maximum

(MAX) and minimum (MIN) lateral error (in meters) using the chained form controller (CF) and the fuzzy logic controller (FUZZY) for three reference velocities (6, 7, 8 ms⁻¹)

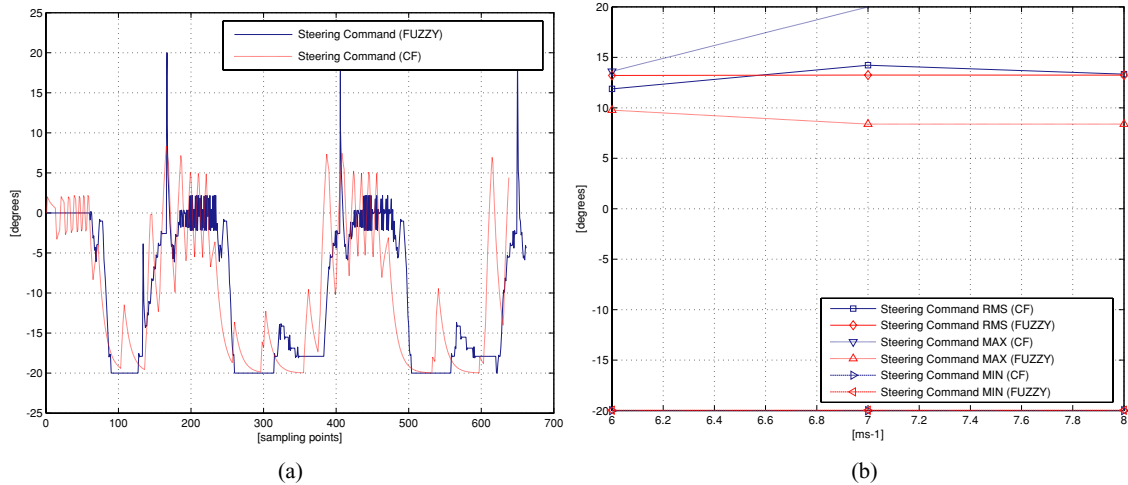


Fig. 13 (a) φ_c : Steering command using the chained form controller (*dashed line*) and steering command using the fuzzy logic controller (*solid line*); (b) root mean square (RMS), maximum

(MAX) and minimum(MIN) steering command (in degrees) using the chained form controller (CF) and the fuzzy logic controller (FUZZY) for three reference velocities (6, 7, 8 ms⁻¹)

previous errors dynamics reveals a two-lobe shape when analysed over time, which does not occur with the fuzzy controller.

Although the fuzzy logic has a better performance in convergence with a predefined path it also has some drawbacks, it shows a more oscillatory behaviour on the steering command (Fig. 13(a)).

From the analysis of Table 3, Figs. 11(b), 12(b) and 13(b) one can deduce that the fuzzy controller is generally slightly better than the chained form controller on most of the reference velocities used in the test. The

RMS data presented in Table 3 also reveals that the steering command effort in the fuzzy logic controller is not greater than in the chained form controller as it would be expected by observing (Fig. 13(b)).

Figure 14 shows the path followed by both controllers: the solid line is the predefined path, dashdot line is the path followed by the vehicle when using the fuzzy controller and the path followed when using the chained form controller is the dashed one. The fuzzy controller behaves better on the curves than does the chained form controller but it is worst when the path

Table 3 Comparative effectiveness (reference velocity = 7 m s^{-1})

	RMS	Max	Min (max negative)
Orientation error θ_e^{lg} (rad)			
Chained form	0.1223	0.2862	-0.1292
Fuzzy logic	0.1146	0.1920	-0.0426
Lateral error d_e^{lg} (m)			
Chained form	0.2241	0.5631	-0.2311
Fuzzy logic	0.1190	0.2423	-0.0487
Steering command φ_c (rad)			
Chained form	0.2482	0.3491	-0.3491
Fuzzy logic	0.2313	0.1463	-0.3486

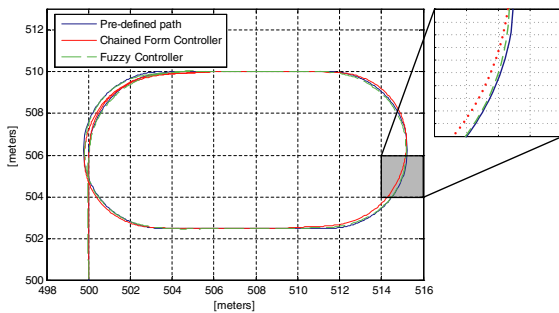


Fig. 14 Path-following simulation results, assuming no odometry errors (neither measurement noise nor cumulative errors)

is a straight line. The chained form analysed here does not embody the same prediction behaviour, described in Section 3.2.4, as the fuzzy controller, which may be one of the reasons of its inefficiency in curve.

Results presented in the following section concern only with the fuzzy logic controller because when performing simulations or experiments using recalibration, the heading and lateral errors fed as inputs are not continuous in time, showing significant values change as a result of a recalibration, for which the chained form controller is not able to cope with.

5.2 Magnetic guidance

The VPE based on fusion of odometry and landmarks, described in Section 4.3, has been extensively simulated. Some results are shown and discussed in this section. In the reported simulations two types of disturbances are considered: systematic errors and Gaussian sensors measurement noise.

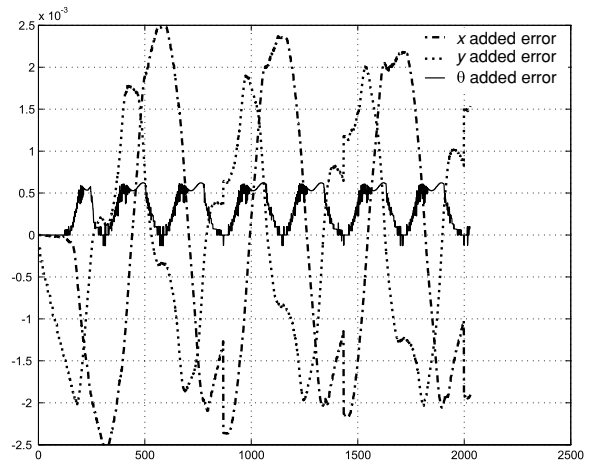


Fig. 15 Systematic noise added to the pose (47) with $K = 1.03$, $[x_e, y_e]$ in meters and $[\theta_e^{\text{sp}}]$ in radians

Systematic errors were applied in the process by multiplying Δ with a factor K , yielding

$$\begin{cases} x_{k+1} = x_k + \Delta \times K \times \cos(\theta_k + \omega/2) \\ y_{k+1} = y_k + \Delta \times K \times \sin(\theta_k + \omega/2) \\ \theta_{k+1} = \theta_k + \omega \end{cases} \quad (47)$$

and so, uncertainty is introduced in the pose (x_k, y_k, θ_k) . In order to evaluate the errors introduced by systematic errors, the magnitude of the disturbance in the vehicle's pose $(x_e, y_e, \theta_e^{\text{cp}})$ is displayed in Fig. 15.

Additionally, Gaussian noise was added, denoted by C , resulting in

$$\begin{cases} x_{k+1} = x_k + \Delta \times K \times \cos(\theta_k + \omega/2) + C \\ y_{k+1} = y_k + \Delta \times K \times \sin(\theta_k + \omega/2) + C \\ \theta_{k+1} = \theta_k + \omega + C \end{cases} \quad (48)$$

In real environments, the detection of the magnets does not return the exact center of the magnet, so in order to have simulated measures similar to real ones, a representative model of the magnetic field radiated by the magnetic marker was used in simulations. Thus, a magnetic marker was modelled as a magnetic dipole with the magnetic field, $B(x, y, z)$, at an arbitrary point $P(x, y, z)$, expressed as follows (in cgs units):

$$B = \frac{\mu_0 M}{4\pi r^5} (3xz\hat{i} + 3yz\hat{j} + (2z^2 - x^2 - y^2)\hat{k}) \quad (49)$$

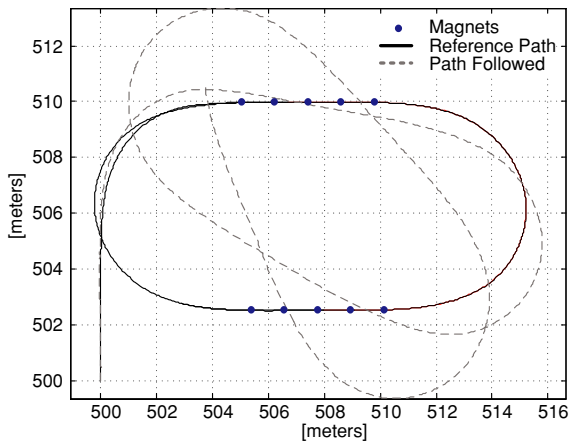


Fig. 16 Odometry results without EKF corrections

where $r = \sqrt{x^2 + y^2 + z^2}$, M is the magnetic moment of the magnetic marker and the z -axis corresponds to the height relative to the marker center.

Results shown in Figs. 16 and 17 exemplify how the fusion process leads to a correct vehicle path-following. If no correction is done, the path actually followed is much different from the reference path. In Fig. 17 the EKF copes with disturbances by using the magnets located at the marked points. Although the errors are accumulated during the curves, on the straight lines it recovers by using the detected magnets. The fusion method also handles false detections either coming from hardware anomaly or from incorrectly positioned magnets.

6 Experimental results

Extensive simulations have been done, showing the effectiveness of the proposed VPE data fusion module. Field experiments have also been done, with the purpose of analysing the localization system behaviour. Whenever a sensor ruler detects a magnetic marker, the measure (d_m) enters the data fusion algorithm and is accepted or not depending on the validation gate result.

Two types of experiments are reported in this section. Both concern the path-following control of a Robucar moving along a predefined closed path (see Figs. 18–20). Figure 18 shows the test field environment where the virtual line represents a rough approximation of the planned trajectory. In both cases the same fuzzy path-following controller, described here and in

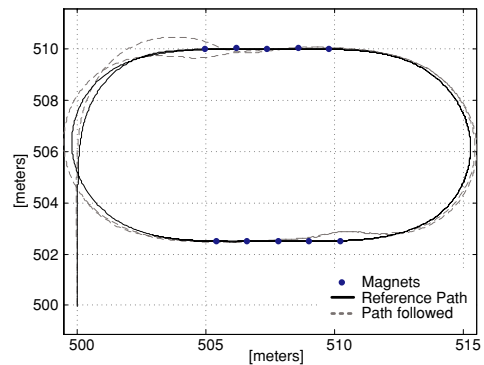


Fig. 17 VPE using EKF fusion of odometry and ten magnetic markers (five markers EKF on each side of the loop)

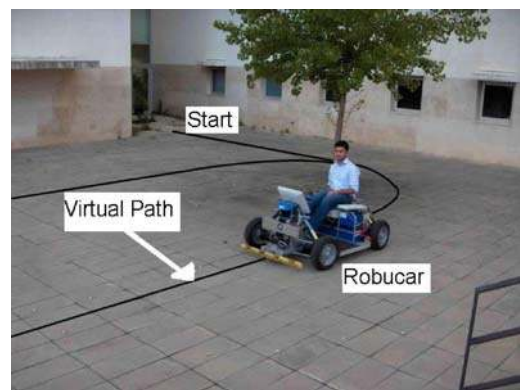


Fig. 18 Field test environment

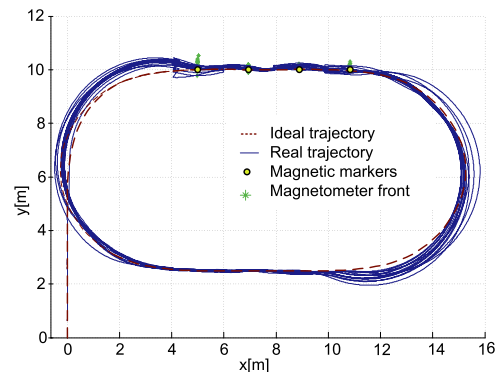


Fig. 19 Experimental results obtained using only the front magnetic ruler – Robucar moving autonomously under a fuzzy path-following controller. The four *bullets* represent the four physical magnetic markers used in the experiment

detail in [2], was used. However, in one case, the detected magnetic markers information was used in the on-line computation of the vehicle’s pose, and so used in the calculation of the errors to the controller, and in the other case it was not used either in the vehicle’s pose

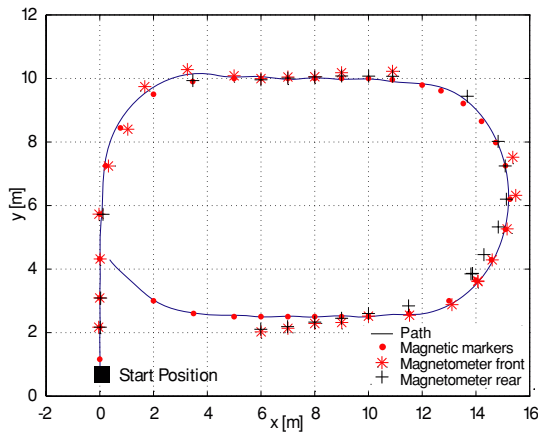


Fig. 20 Experiment without odometry calibration – one run, Robucar moving autonomously. In this experiment 36 magnets were used. The path followed by the Robucar was recorded using data from the front magnetic ruler (*), rear magnetic ruler (+) and data from the encoders (solid line) (standard odometry (3))

estimation or in the controller. As shown in Fig. 19, when the vehicle passes through a magnetic marker, the odometry is calibrated based on the detected lateral deviation of the vehicle. In the experimental result shown in Fig. 19, only four magnets, all aligned on one side of the loop, were used. This simple configuration was enough to keep the Robucar tracking the path. So as to compare the performance of the navigation system, with the calibration method based on the detected markers, Fig. 20 illustrates an experimental test where no calibration was done to the odometry system. In the same figure, the solid line represents the computed path as obtained by the odometry system and the star marks represent the path followed by the center of the front magnetic ruler, while the cross marks represent the path followed by the center of the rear magnetic ruler. As can be observed, the Robucar lost track of the path on the first loop run, when finishing the loop. Notice that the main error is in its orientation, hence the car before losing the track of the path, was following a straight line but with a wrong orientation.

7 Conclusions and future work

The fuzzy controller revealed to be very robust, this means that it was able to cope with the two types of errors presented in the simulations and experiments: the first ones arriving from a normal closed loop con-

troller in path-following and the second ones when it was submitted to sudden changes in the vehicle position, which arise in the odometry recalibration process using the magnets. If the magnets used in the simulation were disposed in such a way that it would almost be a continuous line of magnets then the chained form controller would also had a good performance in coping the first and second types of errors, otherwise with less recalibration data over time its use revealed to be unsuitable on coping with the second type of errors.

The magnetic guidance system revealed good results in experimental tests. We are now testing extensively the complete fusion process, integrating also range-bearing data from laser detected natural features [23].

The majority of systematic errors associated to the odometry relying only on encoders are eliminated by the markers calibration. However, that procedure alone does not solve the slippage (or high-slippage) problem, which can be reduced by applying confidence tests as proposed in Section 4.2.

More field experiments are being carried out to deeply characterize the performance of the overall VPE data fusion module.

Acknowledgements This work was supported by Institute of Systems and Robotics and Fundação para a Ciência e Tecnologia under contract NCT04:POSC/EEA-SRI/58016/2004.

References

- Bento, L.C.: Fuzzy logic lateral controller of a bi-steerable four-wheels actuated vehicle. Technical Report IS-RLM2004/01, Institute of Systems and Robotics, Portugal (2004)
- Bento, L.C., Nunes, U.: Autonomous navigation control with magnetic markers guidance of a cybernetic car using fuzzy logic. *Mach. Intell. Robotic Control* **6**(1), 1–10 (2004)
- Bishop, R.: *Intelligent Vehicle Technology and Trends*. Artech House, London (2005)
- Bonnifait, P., Crubill, P., Meizel, D.: Data fusion of four ABS sensors and GPS for an enhanced localization of car-like vehicles. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, pp. 1597–1602, (2001)
- Cordesses, L., Martinet, P., Thuilot, B., Berducat, M.: Robot motion planning and control. In: *Proceedings of the 16th IAARC/IFAC/IEEE International Symposium on Automation and Robotics in Construction*, Madrid, Spain, pp. 41–46, (1999)
- Cybercars: Cybernetic technologies for the car in the city. Available via www.cybercars.org (2001)

7. TranSafety: Canadian researchers test driver response to horizontal curves. *Road Manage. and Eng. J.*, TranSafety, Inc. (Sept. 1998)
8. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte carlo localization: Efficient position estimation for mobile robots. In: *Proceedings of the 16th National Conference on Artificial Intelligence*, Orlando, FL (1999)
9. Fox, D., Burgard, W., Thrun, S.: Markov localization for mobile robots in dynamic environments. *J. Artif. Intell. Res.* **11**, 343–349 (1999)
10. Fraichard, Th., Garnier, Ph.: Fuzzy control to drive car-like vehicles. *Int. J. Robotics Autonomous Syst.* **34**, 1–22 (2001)
11. Hessburg, T., Tomizuka, M.: Fuzzy control for lateral vehicle guidance. *IEEE Control Syst. Mag.* **14**, 55–63 (1994)
12. Kiencke, U., Nielsen, L.: *Automotive control systems*. SAE-Soc. Automotive Eng., ISBN 3-540-66922-1 (2000)
13. Leonard, J., Durrant-Whyte, H.F.: Mobile robot localization by tracking geometric beacons. *IEEE Trans. Robotics Automat.* **7**(3), 376–382 (1999)
14. Luca, A., Oriolo, G., Samson, C.: Feedback control of a nonholonomic car-like robot. In: *Robot Motion Planning and Control*, Laumond, J.-P. (ed.) LNCIS, Vol. 229, pp. 171–253, Springer, Berlin Heidelberg New York (1998)
15. Mellodge, P.: *Feedback Control for a Path Following Robotic Car*. M.Sc. thesis in Electrical Engineering, Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, VA (2002)
16. Mendes, A., Nunes, U.: Situation-based multi-target detection and tracking with laserscanner in outdoor semi-structured environment. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, pp. 88–93, (2004)
17. Moita, F., Nunes, U.: *Magnetic ruler version 1.0. configuration, software structure and characterization*. Technical Report ISRLM2004/03, Institute of Systems and Robotics, Portugal (2004)
18. Parent, M., Gallais, G., Alessandrini, A., Chanard, T.: *CyberCars: review of first projects*. In: *Proceedings of the 9th International Conference on Automated People Movers*, Singapore (2003)
19. Sekhavat, S., Hermosillo, J.: The cycab robot: a differentially flat system. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Japan, Vol. 1, pp. 312–318 (2000)
20. Sika, J., Hilgert, J., Bertram, T., Pauwelussen, J.P., Hiller, M.: Test facility for lateral control of scaled vehicle in an automated highway system. In: *Proceedings of the 8th Mechatronics Forum International Conference*, The Netherlands, pp. 24–26, (2002)
21. Solea, R., Nunes, U.: Trajectory planning with velocity planner for fully-automated passenger vehicles. In: *Proceedings of the IEEE Intelligent Transportation Systems Conference*. Toronto, Canada (2006)
22. Sotelo, M.A.: Nonlinear lateral control of vision driven autonomous vehicles. *Int. J. Mach. Intell. Robotic Control* **5**(3), 87–93 (2003)
23. Surrecio, A., Nunes, U., Araujo, R.: Fusion of odometry with magnetic sensors using Kalman filters and augmented system models for mobile robot navigation. In: *Proceedings of the IEEE International Conference on Industrial Electronics*, Dubrovnik, Croatia (2005)
24. Tan, A., Guldner, J., Patwardhan, S., Chen, C., Bougler, B.: Development of an automated steering vehicle based on roadway magnets-A case study of mechatronics system design. *IEEE/ASME Mechatron.* **4**(3), 258–272 (1999)
25. Taylor, C.J., Kosecka, J., Blasi, R., Malik, J.: A comparative study of vision-based lateral control strategies for autonomous highway driving. *Int. J. Robotic Res.* **18**(5), 442–453 (1999)
26. TranSafety: Simulated on-the-road emergencies used to test stopping sight distance assumptions. *Road Manage. and Eng. J.*, TranSafety, Inc. (July 1997)
27. Zhang, W., Parson, R.E.: An intelligent roadway reference system for vehicle lateral guidance/control. In: *Proceedings of the 1990 American Control Conference*, San Diego, CA pp. 281–286 (1990)