FUZZY PREDICTIVE CONTROLLER FOR MOBILE ROBOT PATH TRACKING

J. Gómez Ortega, F.R. Rubio and E. F. Camacho

Dpto. Ingeniería de Sistemas y Automática, Univ. de Sevilla
Escuela Superior de Ingenieros, Avda. reina Mercedes s/n, 41012
Sevilla, Spain.
Fax: +34-5-4556849, email: juango@cartuja.us.es

Abstract: This paper presents a way of implementing a Model Based Predictive Controller (MBPC) for mobile robot path-tracking. The method uses a non-linear model of mobile robot dynamics and thus allows an accurate prediction of the future trajectories. Constraints on the maximum attainable angular velocity is also considered by the algorithm. A fuzzy approach is used to implement the MBPC. The fuzzy controller has been trained using a lookup-table scheme, where the database of fuzzy-rules has been obtained automatically from a set of input-output training patterns, computed with the predictive controller. Experimental results obtained when applying the fuzzy controller to a TRC labmate mobile platform are given in the paper.

Keywords: Mobile robots, fuzzy control, predictive control, path-tracking.

1. INTRODUCTION

One of the most important fields in the design of autonomous mobile robots for indoor navigation in a partially structured environment is the path-tracking problem (PTP). It can be defined as the ability of a vehicle to follow precisely a predefined desired path.

This reference path is usually precomputed using an off-line path-planning algorithm, which makes use of the knowledge of the objects placed in the environment in order to generate a free colision reference path.

Many approaches can be found in the literature for the PTP. In Nelson and Cox (1990), a proportional control strategy is applied, where the angular velocity of the driving wheels and the orientation of the steering wheel is computed as a weighted sum of a set of tracking errors.

The PTP, also can be considered from a geometrical point of view. In this approach, the control actions drive the robot in such a way that the vehicle's curvature is the same, at each sampling instant, as the curvature of a geometric function, $y = f(x, P)$, where P is a parameter vector. This vector is computed applying the problem geometrical constraints, that is, considering the initial coordinates and curvature of the function to be the same as the robot position and curvature, and the final point and curvature of the function to be the same as the goal point coordinates and curvature. In Amidi (1990) and Shin and Singh (1990) this technique is applied, using a quintic polinomial function. A similar approach is presented in Nelson (1989) using splines functions. Another geometric approach, proposed by Amidi (1990), is known as *pure pursuit*. This technique uses the curvature of the circunference traced between the mobile robot position and the goal position as the reference curvature for the controller.

The approach presented in this paper is based on optimal control techniques. The problem that is raised here is that of driving a mobile robot to follow a previously calculated desired path. As the desired future reference is known, it would seem that a predictive control technique is a suitable approach for these problem.

Model-based predictive control (MBPC) methods are a family of optimal control techniques that are characterized by the following common elements: a) previous knowledge of future references; b) explicit use of a system model for future system output prediction; c) minimization of a cost function to obtain the control law; d) a receding-horizon strategy.

The use of MBPC strategies for solving the path tracking and the navigation problems in a partially structured environment has been referenced in the literature (Papageorgiou and Steinkogler, 1993), (Gómez Ortega and Camacho, 1994) and (Gómez Ortega and Camacho, 1996) where a neural-network approach was formulated.

In this paper, a predictive strategy is used for the path-tracking module. A nonlinear model of the robot kinematics is used. Constraints in the control variables are also considered and a quadratic cost function is proposed for computing the control signals. A fuzzy scheme is presented for the implementation of this complex predictive controller to achieve real-time performance. The fuzzy controller will be trained using a lookup-table scheme, with a set of training patterns obtained from an off-line simulation of a predictive controller, computed with a numerical optimization algorithm.

2. MBPC TECHNIQUES FOR MOBILE ROBOT PATH-TRACKING

2.1 *MBPC strategy*

The MBPC algorithm consists of applying a control sequence that minimizes a multistage cost function of the form:

$$J(N_1, N_2, N_u) = \sum_{i=N_1}^{N_2} \mu(i)[\hat{Y}(k+i|k) - Y_d(k+i)]^2 + \sum_{i=1}^{N_u} \lambda(i)[\Delta u(k+i-1|k)]^2$$

where $N = N_2 - N_1$ is the *prediction horizon* and N_u is the *control horizon*.

The notation $x(k+i|k)$ indicates that $x(k+i)$ is calculated with the data known in sample time k. $\mu(i)$ and $\lambda(i)$ are penalty factors, which are usually chosen to be constant along the time. The future system outputs, $\hat{Y}(k+i|k)$ for $i = N_1, ..., N_2$, are predicted from a model of the process, from the inputs and outputs before instant k, and from the control actions foreseen for the future, $u(k+i|k)$ for i=0,...,$N_u - 1$, which are the unknown variables. In this way, J can be expressed as a function of only the future control actions. It is usual to suppose that the control actions are constant after a predefined time instant.

The objective of predictive control is to obtain a future control action sequence $(u(k), u(k+1|k), ..., u(k+N_u-1|k))$ in such a way that the future predicted outputs $\hat{Y}(k+i|k)$ will be as close as possible to the desired references $Y_d(k+i)$ over the prediction horizon. This is accomplished by the minimization of J with respect to the control variables. After this sequence is obtained, a *receding horizon* approach is considered. This consists of applying only the first control action $u(k)$ calculated. This process is repeated at every sampling interval in such a way that the calculated open loop control law is applied in a closed-loop manner.

The problem raised in this paper is that of driving a mobile robot to follow a reference path generated on-line by a global planner. This path can be the preplanned global path or, if an obstacle is encountered, a modified one. A predictive control technique is used as the robot path-tracking module, being the robot's angular velocity the control variable.

The cost function used here is:

$$J(N_1, N_2, N_u) = \sum_{i=N_1}^{N_2} [\hat{Y}(k+i|k) - Y_d(k+i)]^2 + \sum_{i=1}^{N_u} \lambda_1 \dot{\theta}^2(k+i-1)$$

where $\hat{Y}(k+i|k) = \{\hat{x}(k+i|k), \hat{y}(k+i|k)\}$ is an i-step prediction of the robot position made at instant k, $\dot{\theta}$ is the robot's angular velocity, which is the control variable, and λ_1 is a constant weighting factor. For the i-step predictions, a nonlinear model of the robot's kinematics has been used.

In J, the first term penalizes the position error and the second term penalizes the robot's angular velocity. This last term ensure smooth robot guidance. An error in the robot heading could be considered in J, but it has been noticed that this is not necessary when the control horizon N_u is sufficiently large. A block diagram of the system is shown in Fig. 1.

In what follows, N_1 and N_2 will be considered to be $N_1 = d+1$ and $N_2 = N$, and N_u will be given a value of $N_2 - d$, where d is the dead time of the system. In this formulation it is assumed that

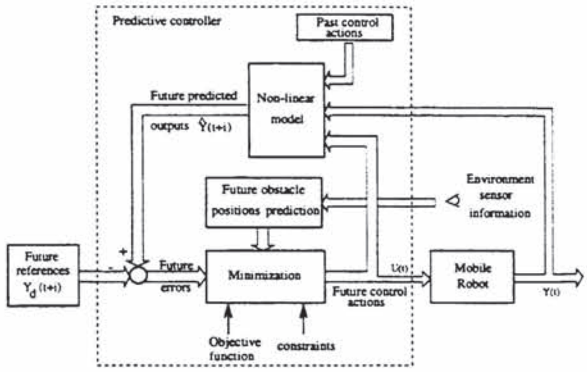

Fig. 1. The predictive controller scheme.

after the control horizon N_u, further increments in control are zero. So the controller has only one free parameter, N.

The predictive problem, formulated under these circumstances, has to be solved using numerical optimization methods, which are not acceptable for real-time control. The controller proposed in this work will be implemented using a fuzzy predictive scheme, which allows real-time implementation.

2.2 *Prediction model*

For an MBPC formulation, a dynamic model of the mobile platform is needed to predict the future positions and headings of the robot. As a testbed for the experiments, a *TRC LABMATE* mobile robot has been used (Fig. 2).

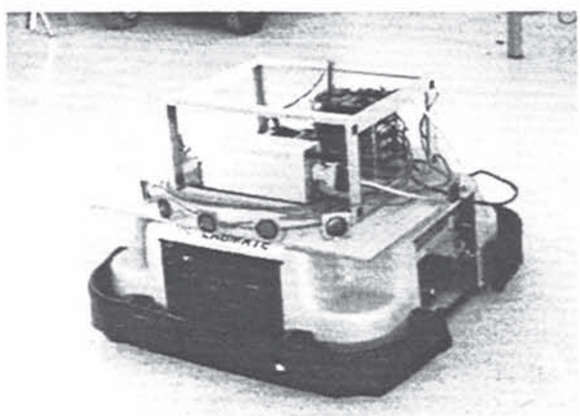

Fig. 2. The *LABMATE* mobile robot.

A model of the *LABMATE* mobile robot which takes account of low-level servocontrol dynamics, as well as the dead time produced by communications with the host computer, was obtained by using kinematic equations and identification tests. A more detailed model can be found in (Gómez Ortega, 1994).

The following kinematic model (which corresponds to a differential-drive vehicle) is used for computing the predictions:

$$
\begin{aligned}
\theta(k+1) &= \theta(k) + \dot{\theta}(k-1)T \\
x(k+1) &= x(k) + \frac{V}{\dot{\theta}(k-1)}\{\sin(\theta(k) + \dot{\theta}(k-1)T) \\
&\quad - \sin\theta(k)\} \\
y(k+1) &= y(k) - \frac{V}{\dot{\theta}(k-1)}\{\cos(\theta(k) + \dot{\theta}(k-1)T) \\
&\quad - \cos\theta(k)\} \\
\dot{\theta}(k-1) &= R\ \frac{\omega_r(k-1) - \omega_l(k-1)}{2W} \\
V &= R\ \frac{\omega_r(k-1) + \omega_l(k-1)}{2}
\end{aligned}
$$

where x, y, θ are the position and heading of the robot in a fixed reference frame (see Fig. 3), T is the sample interval and W is the half-distance between the wheels, which value has been estimated to be 168 mm (Fig. 3). V is the linear velocity of the mobile robot, which is considered to be constant, $\dot{\theta}$ is the steering speed, and $\omega_r(k-1), \omega_l(k-1)$ and R are the right and left wheel angular velocities (which are considered to be constant for each sample interval) and the wheel radius, respectively. These equations are valid in the case of a steering speed $\dot{\theta}(k-1) \neq 0$. In the case of a linear trajectory, the equations of motion are given by:

$$
\begin{aligned}
\theta(k+1) &= \theta(k) \\
x(k+1) &= x(k) + VT\ cos\ \theta(k) \\
y(k+1) &= y(k) + VT\ sin\ \theta(k)
\end{aligned}
$$

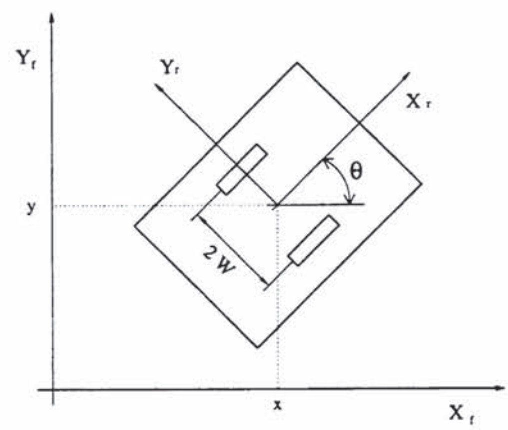

Fig. 3. Reference frame

Using the maximum acceleration value , the velocities of both wheels have been considered to be constant for each sample period.

2.3 *Desired path parametrization*

The reference path is given to the MBPC fuzzy controller as a set of straight lines and circular arcs. The MBPC approach needs the desired positions and headings of the mobile platform at the next N time instants. So, given the current position and heading of the robot, it is necessary to parametrize the desired path for the next N periods of time, in order to calculate the N future path points desired. As is shown in Fig. 4, the desired point for the current instant $(x_d(k), y_d(k))$ is obtained first. It is located at the intersection between the desired path and its perpendicular, traced from the actual robot position $(x(k), y(k))$. The next N points are spaced equally along the path, with a separation between them of ΔS, which is a design parameter. By using this approach, no approximation trajectory is needed when the robot position is not located on the desired path.

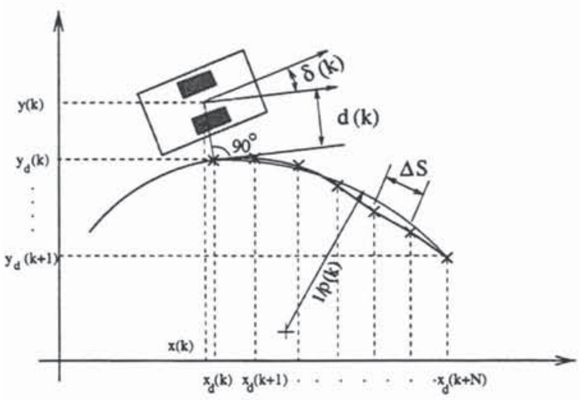

Fig. 4. Desired path parametrization.

3. THE FUZZY PREDICTIVE APPROACH

As was mentioned before, the minimization of the cost function J has to be carried out by a numerical optimization method which requires too much computation time to be used in real time. A fuzzy predictive solution is proposed, which guarantees real time for the robot control. Once the training stage is over, the fuzzy controller can reproduce the behaviour of the predictive controller in real-time.

The training stage of the fuzzy controller is carried out using a lookup-table scheme. The database of rules needed for the fuzzy controller is calculated automatically from a set of training patterns obtained with a predictive controller, computed off-line with a numerical optimization algorithm. The key idea of this method, proposed in Wang (1994), is to generate fuzzy rules from input-output pairs, collect the generated rules into a common fuzzy rule base, and construct a final fuzzy logic controller based on this combined fuzzy rule base.

The modules of the control scheme used in this work (see Fig. 5) are:

Fuzzy controller. The controller has four inputs. The first one corresponds to the previous angular

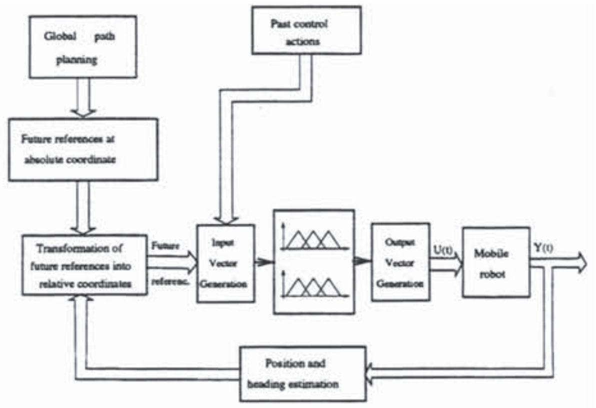

Fig. 5. Fuzzy predictive scheme for mobile robot path-tracking.

velocity of the robot. The last three inputs are associated with the parametrization of the desired trajectory over the prediction horizon. The parameters given to the fuzzy controller are the distance d from the robot guide point to the path, the angle δ between the robot heading and the path orientation, and an average of the inverse of the curvature of the future desired points ($c = 1/\rho$) (see Fig. 4). The output is the robot angular velocity for the next sampling interval.

Input-Output vector-generation modules. The main function of the first module is to compute the values for the input layer of the controller from different sources (the local reference path-generation module and the past control actions). Also, a symmetry analysis is made here in order to reduce the number of training patterns needed to provide good performance of the fuzzy controller. The objective of the second module is to perform the inverse symmetry transformation, when required.

Reference-path coordinate transformation module. The desired path coordinates are transformed from a global reference system to a local reference system, attached to the mobile robot. This avoids the use of additional fuzzy controller inputs for the robot position and heading, which are implicitly given to the controller in the reference path.

Past control actions. These are needed for the fuzzy controller to consider the delay time of the robot system.

3.1 *Fuzzy Rule Base generation*

The set of training items is composed of 3500 patterns, each one with four desired inputs and one desired output. These patterns have been computed automatically with an off-line predictive controller for different combinations of the four inputs over their ranges of values. Now, the task

Input	Region	Limits	
$\dot{\theta}(k-1)$ (rad/s)	L2	-19.5	-4.5
	L1	-9	0
	C	-4.5	4.5
	R1	0	9
	R2	4.5	19.5
$d(k)$ (m)	L	-0.2625	-0.0375
	C	-0.15	0
	R	-0.0375	0.0375
$\delta(k)$ (deg)	L1	-67.5	-4.5
	L2	-36	0
	C	-4.5	4.5
	R1	0	36
	R2	4.5	67.5
$c(k)$ (m^{-1})	L2	-2.25	-0.75
	L1	-1.5	0
	C	-0.75	0.75
	R1	0	1.5
	R2	0.75	2.25

Table 1. Fuzzy regions definition.

is to generate a set of fuzzy IF-THEN rules from this desired input-output patterns, and use these fuzzy rules to determine a fuzzy logic controller

$$f \; : \; (\dot{\theta}(k-1), d(k), \delta(k), c(k)) \rightarrow \dot{\theta}(k)$$

The algorithm for obtaining the fuzzy rule base from the set of training patterns is divided in five steps (Wang, 1994):

Step 1.- *Division of the input and output spaces into fuzzy regions.* The range of each input variable is divided into n_i fuzzy regions, where $i = 1,..,4$. The shape of each fuzzy region is triangular, and the maximum value of the membership function, $\mu(x)$, is equal to one for the center of the region. In Table 1, the fuzzy regions limits defined for the inputs of the fuzzy predictive controller are shown. The limits for the output fuzzy regions are the same as the limits for the input $\dot{\theta}(k-1)$.

Step 2.- *Generation of fuzzy rules from given training patterns set.* First, for all the training patterns, the membership degree of each component of each pattern to different fuzzy regions are calculated. Second, each training pattern component is assigned to the region with maximum membership degree.

Step 3.- *Assign a degree to each fuzzy rule.* As the number of training patterns is great, it is probable that the rules generated from two or more different patterns have the same IF part but different THEN part. In order to resolve this conflict, a degree will be assigned to each rule and only the rule with maximum degree will be considered as a component of the combined fuzzy rule base.

The degree assigned to each rule is the product of the membership degrees of each component of the pattern to the fuzzy regions chosen in step 2 (that is, the regions whith maximum membership degree).

Step 4.- *Create a combined fuzzy rule base.* The fuzzy rule base is created as follows. For each possible combination of different fuzzy regions, for all the input vector component, an output fuzzy region is assigned, which is the fuzzy region of the output component of the rule with given inputs regions and with a maximum assigned fuzzy rule-degree.

Step 5.- *Determine a defuzzification mapping based on the fuzzy rule.* The following defuzzification strategy is used to determine the output control. First, the antecedents of the *ith* fuzzy rule for given inputs $(\dot{\theta}(k-1), d(k), \delta(k), c(k))$ are combined using a product operation to determine the degree, $\mu^i_{O^i}$, of the output control corresponding to $(\dot{\theta}(k-1), d(k), \delta(k), c(k))$:

$$\mu^i_{O^i} = \mu_{I^i_1}(\dot{\theta}(k-1))\mu_{I^i_2}(d(k))\mu_{I^i_3}(\delta(k))\mu_{I^i_4}(c(k))$$

where O^i is the output region of rule i, and I^i_j is the input region of rule i for the jth component.

Finally, a center average defuzzification mapping is used to determine the control output:

$$\dot{\theta}(k) = \frac{\sum_{i=1}^{M} \mu^i_{O^i} \overline{y}^i}{\sum_{i=1}^{M} \mu^i_{O^i}}$$

where $\overline{y}^i$ is the center of value of region O^i. This will give less control effort. M is the number of fuzzy rules in the combined fuzzy rule base.

4. RESULTS

The proposed control structure has been tested by experimental tests when applying the fuzzy predictive controller to the *Labmate* mobile robot.

The controller was trained in a lookup-table manner, as described previously. The value of N chosen for the MBPC was experimentally made equal to seven; thus N_1, N_2 and N_u were given the values 2, 7 and 6, respectively, and the weighting factor was given the value: $\lambda_1 = 5$. The maximum and minimum angular velocity were given the following values respectively: -20 $^o/s$ and 20 $^o/s$. For Δs, a value of 0.15 m was chosen, which leads to a linear robot velocity of 0.25 m/s.

Figure 6 shows two experiments carried out with the LABMATE, where the behaviour of the fuzzy predictive controller is tested. In the figure, the dashed-lines show the desired trajectories and the solid lines show the real trajectories followed by the mobile robot. In the first one, the mobile robot moves through a narrow corridor and a door. As can be seen, the mobile robot follows the desired trajectory in spite of being in an initial position separated about 400 mm from the desired path. In the second test, a reference path with shorter curvature radii is given to the controller.

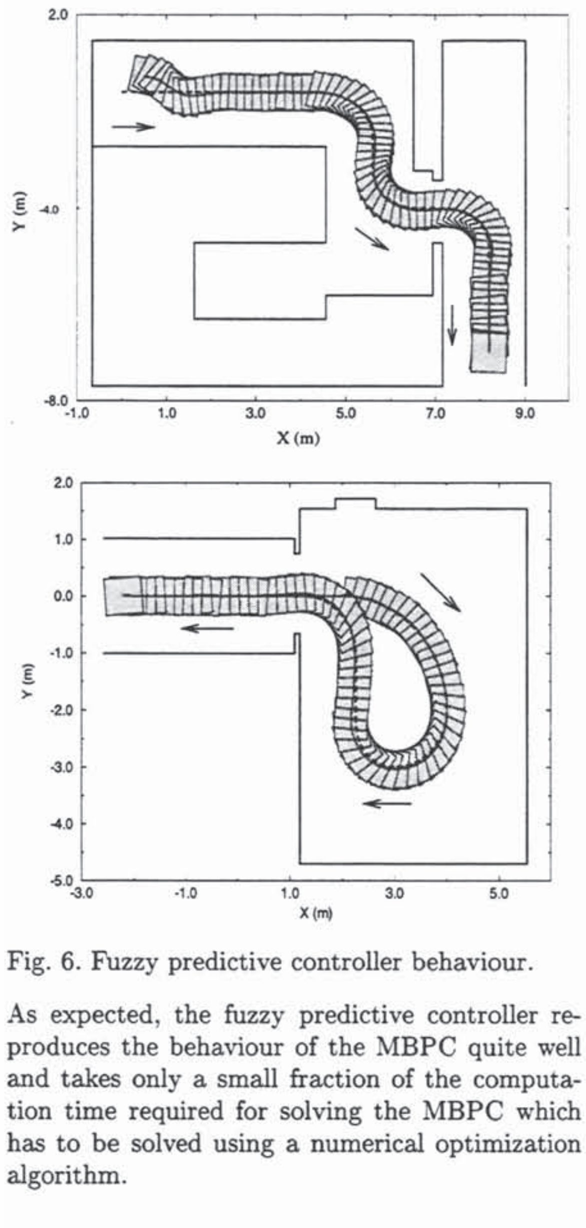

Fig. 6. Fuzzy predictive controller behaviour.

As expected, the fuzzy predictive controller reproduces the behaviour of the MBPC quite well and takes only a small fraction of the computation time required for solving the MBPC which has to be solved using a numerical optimization algorithm.

5. CONCLUSIONS

A fuzzy predictive path tracking controller for mobile robots has been presented. The fuzzy controller has been trained using a lookup-table scheme. The desired fuzzy output was computed off line by a predictive controller. Control signal saturations and non linearities of the model were considered in order to obtain accurate predictions of the robot trajectories. The computation time required to solve this MBPC problem under these circumstances would be prohibitive for real time. The fuzzy predictive approach has proved to be an effective way of implementing the path tracking predictive algorithm as shown by the experimental results.

6. ACKNOWLEDGEMENT

The authors would like to acknowledge the *CICYT* for funding this work under TAP95-0307 and TAP96-884CP.

7. REFERENCES

Amidi, O. (1990). *Integrated Mobile Robot Control.* Carnegie Mellon University Robotic Institute (Pittsburgh). Informe interno CMU-RI-TR-90-17.

Gómez Ortega, J. (1994). Navegación en obots móviles basada en técnicas de control predictivo neuronal. PhD thesis. Univ. of Seville.

Gómez Ortega, J. and E. F. Camacho (1994). Neural Network MBPC for Mobile Robots Path Tracking. *Robotics and Computer Integrated Manufacturing Journal* **11**(4), 271–278.

Gómez Ortega, J. and E. F. Camacho (1996). Mobile Robot Navigation in a Partially Structured Static Environment, using Neural Predictive Control. *Control Engineering Practice* **4**(12), 1669–1680.

Nelson, W. and I. Cox (1990). *Autonomous Robot Vehicles.* Chap. Local Path Control for an Autonomous Vehicle, pp. 38–44. Springer-Verlag.

Nelson, W.L. (1989). Continous Steering-Function Control of Robot Carts. *IEEE Transactions on Industrial Electronics* **36**(3), 330–337.

Papageorgiou, M. and A. Steinkogler (1993). Real-Time Optimal Control of Moving Vehicles in Changing Environments. In: *Proc. of the 12th IFAC World Congress.* Sydney. pp. 107–112.

Shin, D. H. and S. Singh (1990). *Vision and Navigation. The Carnegie Mellon Navlab.* Chap. Vehicle and Path Models for Autonomous Navigation, pp. 283–307. Kluwer Academic Publishers.

Wang, Li Xin (1994). *Adaptive Fuzzy Systems and Control. Design and Stability Analysis.* Prentice-Hall.