PREDICTIVE PATH TRACKING OF MOBILE ROBOTS. APPLICATION TO THE CMU NAVLAB.

A. Ollero†
Field Robotics Center

O.Amidi
Vision and Autonomous System Center

Robotics Institute
Carnegie Mellon University.
Pittsburgh, Pennsylvania 15213-3890, USA

Abstract - **In this paper we present a new path tracking method and its application to an autonomous vehicle, the NavLab (road-worthy truck modified for research in outdoor navigation). The method applies predictive control theory to obtain the appropriated steering angle taking into account the vehicle velocity. We consider both the single-input single output case and the multivariable formulation of the predictive control problem. Moreover we describe the implementation of the NavLab controller.**

I. INTRODUCTION

The control of Mobile Robots involves several tasks such as perception, path planning, path tracking, and low-level control. The estimation of the mobile robot position can be performed using several sources of data including the perception system (image, range data, sonars) as well as internal sensors (inclinometers, gyroscopes, acelerometers) and dead reckoning techniques. The general purpose of a robot path planner is to find a trajectory from a start position to a goal position with no collisions while minimizing a cost measure. This paper is mainly concerned with path tracking. The objective is to follow a previously defined path by taking into account the actual position and the constraints imposed by the vehicle and its lower level motion controllers. The controller architecture and the position estimation technique are discussed to clarify the implementation issues of the tracking strategies.

There are path tracking formulations [1] which respond to a controller incorporating real time image processing to estimate the vehicle position. On the other hand, there are architectures in which the perception and path planning are integrated with path tracking and low level control in order to provide fast response for real time obstacle avoidance [2]. In this paper we overview the general characteristics of the NavLab controller in section 2.

Most path tracking methods are based on the estimated error between the current vehicle position and the path to follow. In several approaches the robot motion between the current position and a previously defined point in the path is defined by using straight lines [3], circles [4], clothoids curves [5][6], and quintic polynomials [7]. From the point of view of the control law, linear proportional feedback of errors [4][8], and PI control algorithms [9], have been implemented by using error coordinate systems. In [10] a non-linear control law is presented and the stability is proved under the assumption of perfect velocity tracking.

In this paper we propose the application of the Generalized Predictive Control (GPC) method as presented in [11]. The motivation and formulation of the control problem in this method is particularly interesting for path tracking. In fact, it is intended to minimize a cost index with the errors between future desired outputs and predicted outputs, and the future increments in control. The method is based on a receding horizon approach similar to that used by the human operator when driving. Predictive control has been recently proposed for rotorcraft terrain-following flight [12] and manipulator control [13]. In section 3 we describe its formulation for mobile robot path tracking. Section 4 is devoted to the implementation. The conclusions and references are in sections 5 and 6.

Fig.1. The NavLab

II. THE NAVLAB CONTROLLER.

The NavLab (see Fig. 1) is a self-contained truck modified so that humans or computers can control it as occasion demands. Its core is the vehicle controller. Figure 2

† On leave from Universidad de Malaga, Spain.

This work has was supported by DARPA, under contract DAAE07-09--RO59. The work of the first author has been performed under grants from the spanish "Programa de Perfeccionamiento y Movilidad del Personal Investigador" and from the "Junta de Andalucia".

shows the architectural overview of this controller [4]. The State Maintenance System uses the data from different sensors to produce an accurate estimate of the vehicle state. The system has a state buffer which maintains positioning (coordinates x,y,z) and orientation (angles ψ, θ, ϕ) data. The buffer is updated by the sensor drivers which interface to the available sensors. The distance traveled (s) and the velocity (v) are computed from the data provided by optical shaft encoders in the transmission. An optical encoder on the steering column reports the steering wheel position which is used to compute vehicle's radius of curvature ($\gamma = d\phi/ds$). An Inertial Navigation System (INS) reports accurate high precision heading (ϕ), roll (ψ) and pitch (θ). The State Maintenance System also includes a processor which maintains the vehicle status and particularly the position estimate from all the sensor data in the state buffer. Dead reckoning can be used to compute changes in vehicle coordinates from the exact curvature and traveled distance. Provided that the computations are done in short intervals, it can be shown that the path between two points is given by:

$$\begin{aligned} dx &= -\sin(\phi)ds \\ dy &= \cos(\phi)\,ds \\ d\phi &= \gamma\,ds \end{aligned} \qquad (1)$$

The relation between the curvature γ and the position of the steering wheel in encoder counts has been determined experimentally by recording the vehicle's heading in short intervals as the vehicle was driven on a wide variety of paths at different speeds. The INS is used to eliminate the error accumulation of dead reckoning. Taking into account the calibration errors in the INS position data, only its orientation data, which is very accurate, is used. Thus, the position is obtained by looking up the initial position in a map and computing the small changes in vehicle coordinates with a small change in distance traveled. External sensors and maps (landmark recognition system) can be used for the INS initialization. Alternatively, data from Global Positioning Systems can also be incorporated.

The clients in Fig.2 are internal and external to the controller. There are three internal clients: the mapper, the velocity regulator and the tracker. The mapper records motion profiles (x,y,z, ψ, θ, ϕ, γ) at regular distance intervals (0.5 m for the NavLab). The velocity regulator commands the vehicle velocity by considering the sharpness of the path to follow in addition to the vehicle's physical constraints. The smoothness of the velocity profile is considered the most important issue. The tracker uses the mapper's path points and the current vehicle status to command a new steering angle. The next section is concerned with the path tracking.

A software interface server allows multiple external clients to communicate with the controller. Thus, the external clients are charged with the perception (interpreting image and range data) and path planning tasks. There is also a hardware interface server to process signals from other external sensors (sonar) and to support man-machine interfaces (joystick to maneuver the vehicle). The Actuation System in Fig. 2 uses a Hardware Configurator process for delivering consistent requests to two low-level feedback controllers. The steering wheel axis control loop has a high precision optical encoder and a PID motion controller board. Also a PID motion controller is used in the drive control loop to command the servo-valve of the NavLab's hydraulic transmission system. The Watch Dog examines the vehicle state and performs security functions, and the Request Maintenance System is the communication link handling the incoming and outgoing requests.

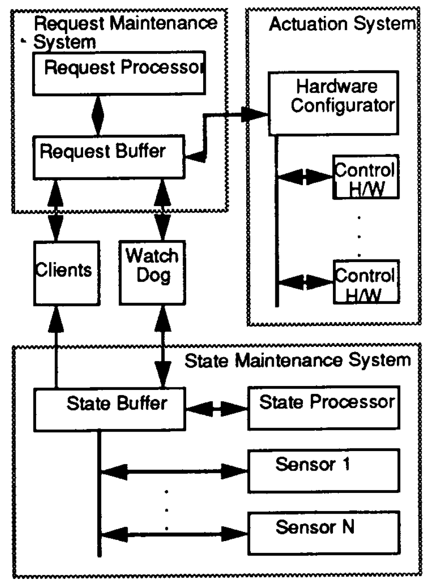


Fig. 2. Architectural view of the controller

III. PREDICTIVE PATH TRACKING

Consider the following locally linearized Controlled Autoregresive and Moving Average (CARIMA) time-discrete model:

$$A(q^{-1})\,y(k) = B(q^{-1})\,u(k-1) + \zeta(k)/\Delta\,, \qquad (2)$$

where u(k), y(k) and $\zeta(k)$ are respectively the control input, the controlled variable, and an uncorrelated random sequence at time k; q^{-1} is the backward shift operator, Δ is the differencing operator ($\Delta = 1-q^{-1}$); and $A(q^{-1})$, $B(q^{-1})$ are the polynomials:

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \ldots + a_{na} q^{-na} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \ldots + b_{nb} q^{-nb} \end{aligned} \qquad (3)$$

This model can be used to represent the path tracking problem if we linearize the motion equations by using vehicle's fixed coordinates, or path dependent coordinates [14][15]. A convenient choice is an axis tangent to the desired path. In this case the coordinates of path points in vehicle frame give the distance to be minimized. The control input is the required curvature or steering angle (γ_r) which is different from the vehicle's actual curvature due to the dynamics of the steering system.

Taking into account the NavLab's navigation limitations (top speed 18 MPH) and the data update rate of the controller (0.1 sec.), linearized models can be used for a constant velocity even if we consider vehicle-ground interaction [16]. On the other hand notice how ζ in (2) may represent the noise in the measurements.

The predictive control method is based on the minimization of the cost function given by:

$$\hat{y}(N_1,N_2) = E\left\{ \sum_{j=N_1}^{N_2} (\hat{y}(k+j) - y_d(k+j))^2 + \sum_{j=N_1}^{N_2} \lambda(j) (\Delta u(k+j-1))^2 \right\} \quad (4)$$

where E {.} represents the expectation and :

- N_1 and N_2 are respectively the minimum and maximum costing horizon: window defined in the path to follow (lookahead).

- $\hat{y}(k+j)$ and $y_d(k+j)$ are respectively the predicted output and the desired values of the output in the future interval $[N_1, N_2]$. Thus, it is intended to minimize the distance to the path in local coordinates (taking into account the vehicle heading). It is clear that data needs to be transformed to this coordinate system by:

$$y_d(k)=(x^*_d(k)-x^*(k))\cos\phi_d(0)-(y^*_d(k)-y^*(k))\mathrm{sen}\phi_d(0)$$

$$x_d(k) = (x^*_d(k)-x^*(k)) \sin \phi_d (0) + (y^*_d(k)-y^*(k)) \cos \phi_d(0) \quad (5)$$

where (x^*,y^*) and (x^*_d,y^*_d) are global coordinates, and ϕ_d (0) is the path heading at k=0.

- λ(j) is a control weighting sequence that can be used to prevent excessive changes in steering.

Thus, the objective is to drive the future outputs y(k+j) close to y_d(k+j) bearing in mind the control activity to do so. For N_1=1 and N_2= N , the prediction vector:

$$\hat{y} = \left[\hat{y}(k+1|k)\ (\hat{y}(k+2|k)) \ldots\ \hat{y}(k+N|k)\right]^T$$

Is given by

$$\hat{y} = G\Delta u + f \quad (6)$$

where $\Delta u=[\Delta u(k), \Delta u(k+1), \ldots, \Delta u(k+N-1)]^T$ is the vector of projected control increments (**steering increments**), G is an NxN lower triangular matrix, and

$$f=[f(k+1), f(k+2), \ldots f(k+N)]^T$$

are the predictions of the output by assuming that future control increments are all zero. The generalized predictive steering control law is given by:

$$\Delta u = (G^TG + \lambda I)^{-1} G^T (y_d - f), \quad (7)$$

where, $y_d= [y_d(k+1), y_d(k+2), \ldots, y_d(k+N)]^T$ is the future desired sequence.

If, after a certain number of samples NU<N_2, the projected steering increments are assumed to be zero

$$\Delta u(k+j-1)=0, j>NU ,$$

then the corresponding steering control law is

$$\Delta u = (G_1^TG_1 + \lambda I)^{-1} G_1^T (y_d - f), \quad (8)$$

where

$$G_1 = \begin{bmatrix} g_0 & 0 & \ldots 0 \\ g_1 & g_0 & \ldots 0 \\ \ldots & \ldots & \ldots \\ g_{N-1} & g_{N-2} & \ldots g_{N-NU} \end{bmatrix} \quad (9)$$

and the matrix

$$(G_1^TG_1 + \lambda I)$$

is NU x NU. Thus, for the particular case NU=1 only one control change (Δu(k)) is considered and the inversion in (7) or (8) is reduced to a scalar computation.

The coefficients of the matrix G_1 in (8)-(9), or G in (7), can be obtained from polynomials G_j given by

$$G_j(q^{-1})= E_j(q^{-1}) B(q^{-1}) \quad (10)$$

and $E_j(q^{-1})$ results from the recursive solution of the Diophantine equation

$$1+ E_j(q^{-1}) A\Delta + q^{-j} F_j(q^{-1}) \quad (11)$$

where $E_j(q^{-1})$and $F_j(q^{-1})$ are polynomials uniquely de

fined given $A(q^{-1})$ and j.

Now consider the **vehicle heading and curvature** as additional variables to be controlled. Thus, the above formulation needs to be generalized to cope with models with more than one output. Moreover, there are formulations of the path tracking problem in which there is more than one control variables. For example in [10] the control variables are the required vehicle linear and rotational velocities.

Thus we have considered a **multivariable formulation** of the predictive control problem. Let m and p be the number of outputs and control variables respectively. In this case $A(q^{-1})$ and $B(q^{-1})$ in (2) are the following mxm and mxp polynomial matrices:

$$A(q^{-1})= \mathbf{I}_{mxm} + \mathbf{A}_1 q^{-1} + \ldots + \mathbf{A}_{na} q^{-na} \quad (12)$$

$$B(q^{-1})= \mathbf{B}_0 + \mathbf{B}_1 q^{-1} + \ldots + \mathbf{B}_{nb} q^{-nb}$$

The Diophantine equation (11) is now an mxm matrix equation, that can also be solved recursively, and the polynomials G_j in (10) are the polynomial matrices given by

$$G_j (q^{-1})= (\varepsilon_0 + \varepsilon_1 q^{-1} + \ldots + \varepsilon_{nb} q^{-(j-1)}) (\mathbf{B}_0 + \mathbf{B}_1 q^{-1} + \ldots + \mathbf{B}_{nb} q^{-nb}) \quad (13)$$

where ε_i are mxm matrices. The method easily applies for computing the projected vectors Δu_j (j=1 ..p) for every control input from the prediction vectors $\mathbf{f}_i$ (i= 1..m) of every output.

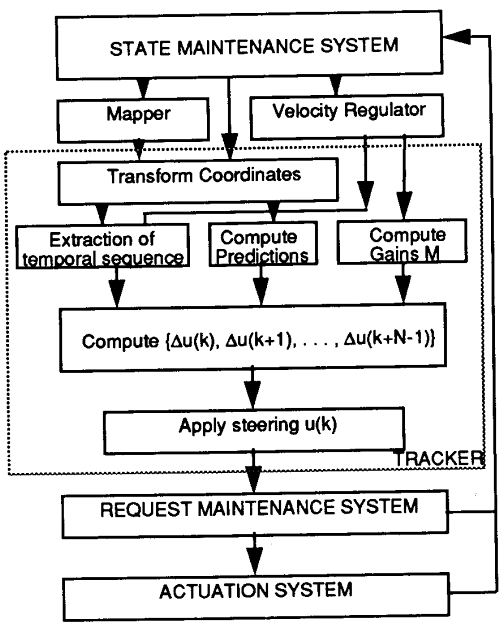


Fig. 3. Implementation of the predictive path tracking method.

IV. IMPLEMENTATION

Figure 3. shows the implementation of the predictive path tracking method in the NavLab's controller. The procedure to compute the gain matrix

$$\mathbf{M}_1=(\mathbf{G}_1{}^T\mathbf{G}_1 + \lambda \mathbf{I})^{-1} \mathbf{G}_1{}^T$$

in equation (8) or

$$\mathbf{M} = (\mathbf{G}^T\mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T$$

in (7) is the following: first compute the model parameters a_i(i=1..na) and b_i(i=1..nb) in (3) for a particular velocity; then solve the Diophantine equation (11) recursively; then compute the polynomials G_j using (10) and form the matrix $\mathbf{G}_1$ or $\mathbf{G}$; and finally compute $\mathbf{M}_1$ or $\mathbf{M}$.

Observe that if we apply all the projected steering angles in the sequence, we have an open loop strategy during the whole period $[N_1, N_2]$. Instead the mechanism of the predictive steering applies only the first increment $\Delta u(k)$ and at the next control instant recomputes a whole sequence from the new sequence of path points provided by the mapper. This feedback mechanism is quite similar to the one used by humans when driving.

In order to reduce steering wheel variations it is possible to use approach trajectories to the path as suggested in the predictive control literature [11]. We have implemented several approach trajectories including the exponential [12]:

$$y_c(k+j)= y_d(k+j)-\exp(-\tau_e j)[y_d(k+j)- y(k+j)] \quad (14)$$

Thus, we substitute in (7) the values of the trajectory on the path $y_d(k+j)$ for $y_c(k+j)$ given by (14).

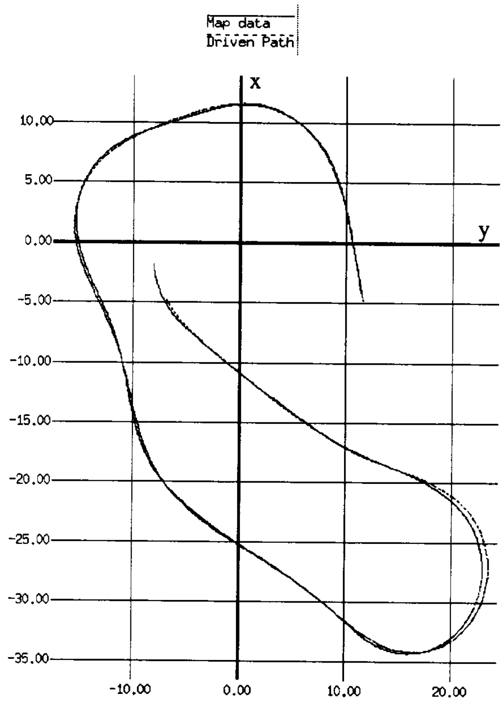
We have also implemented the multivariable formulation. Particularly we have used this formulation with three outputs (distance to the path, heading, and vehicle curvature) and the required steering angle as the only control variable.

In Fig. 4 we show the results of the implementation of the predictive control method by using a very simple model obtained by local linearization and discretization of the dead reckoning equations (1) for constant velocities, and considering first order dynamics for both the steering and the drive systems. The steering time constant has a significant influence on the dynamic behaviour. The steering time constant for the NavLab was measured to be 0.4 sec. for the velocity range of interest.

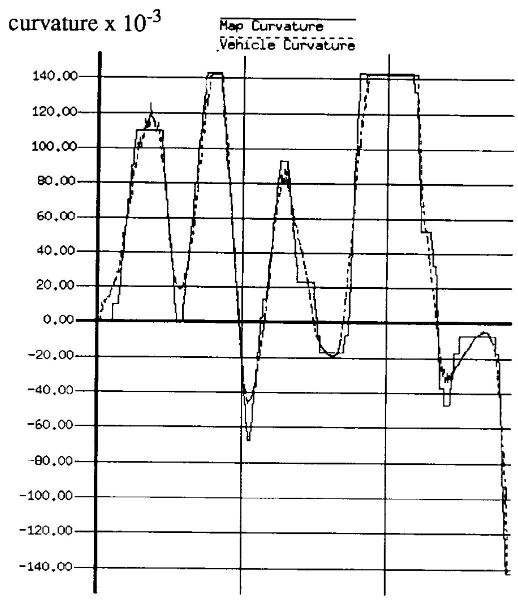
In this experiment we have used a temporal sequence of 18 points, and NU=1 for a velocity of 3.4 m/s. The path (Fig. 4a) is very difficult to follow due to the sharpness of its curves, especially the one on the bottom right of the figure which is almost at the minimum turning radius of the NavLab. However, the tracking is good and the steering wheel movements are minimized. In Fig. 4b the evolution of the vehicle steering angle is shown. A low value of τ_e in (14) is needed for the heading ϕ. If τ_e is decreased, the tracking is improved but undesirable high frequency oscillations of the steering wheel are observed. These oscillations are not present in the experiment shown in the Fig. 4.

In the experiment shown in the figures, the wheight λ in (4) is zero. Increasing λ the steering activity is reduced but the tracking deteriorates.

Moreover, in the experiment we have used N_1=1 in (4). The horizon N= N_2 in (4) (lookahead) was tuned automatically taking into account the vehicle velocity. Indeed if we have a constant control period it is advantageous



(a) position tracking



(b) curvature tracking

Fig. 4 Tracker performance.

to modify the number of path points ahead when changing the velocity. Increasing N has an anticipative effect in the tracking of curves and the opposite occurs when decreasing N. This effect is shown in Fig. 5.

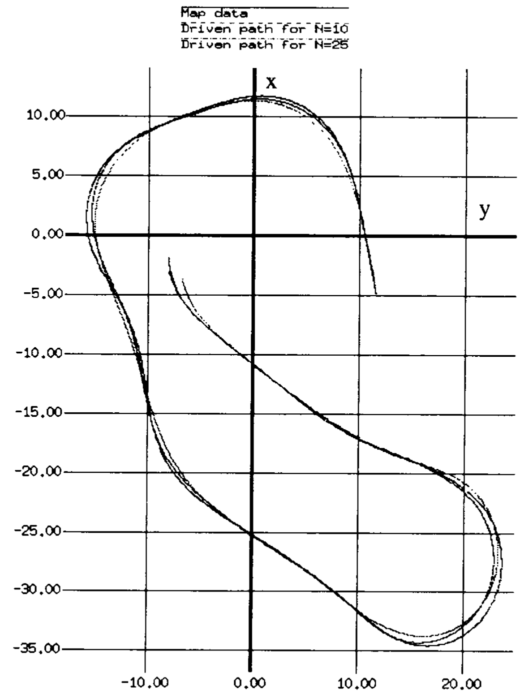


Fig.5. Modification of $N=N_2$ for $N_1=1$

It is also possible to use NU>1. However this increases the computational time of the adaptive implementation because it is necessary to perform the matrix inversion involved in (8) when the velocity is changed. Moreover, values of NU>1 may increase the control activity producing an undesirable steering behaviour.

V. CONCLUSIONS

Predictive control theory offers promising results for path tracking of mobile robots and autonomous vehicles. The method has been implemented and tested for the NavLab and it produced good results for both tracking and smoothness of steering. The method is robust even for poorly modelled systems and the stability can be proven. Adaptive implementations may cope with changes in velociy and other model parameters. More research effort must be devoted to investigating the effects of perturbations and unmodelled dynamics particularly for high speed and rugged terrain navigation. The relation between the design parameters and the navigation conditions is another subject to be studied.

ACKNOWLEDGEMENTS

We appreciate the comments and suggestions of William "Red" Whittaker, Director of the Field Robotics Center.

REFERENCES

[1] E. Dickmans and A. Zapp. Autonomous high-speed road vehicle gidance by computer vision. In Proc. 10 IFAC World Congress, Munich, 1987.

[2] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. I. J. Robotic Research, Vol. 5, No 1, pp 90-98, 1986.

[3] I. Kanayama and S. Yuta. Vehicle path specification by a sequence of straight lines. IEEE Journal of Robotics and Automation, Vol. 4, No. 3, pp 265-276, 1988.

[4] O. Amidi. Integrated mobile robot control, Carnegie Mellon Univ. Robotics Institute. Technical Report CMU-RI-TR-90-17, 1990.

[5] I. Kanayama and N. Miyake. Trajectory generation for mobile robots. Robotic Research, Vol. 3, pp 333-340, The MIT Press, 1986.

[6] J. Crowley. Asycronous control of orientation and displacement in a robotic vehicle. Proc. IEEE Conference on Robotics and Automation, pp 1277-1282, 1989.

[7] S. Singh and D. H. Shin, Position based path tracking for wheeled mobile robots, Proc. IEEE International Workshop on Intelligent Robots and Systems, pp 386-391, Tsukuba, Japan, September 1989.

[8] W. Nelson and I. Cox. Local path control for an autonomous vehicle. Proc. IEEE Conference on Robotics and Automation, pp 1315-1317, 1988.

[9] Y. Kanayama, A. Nilipour, and C. Lelm. A locomotion control method for autonomous vehicles. Proc. IEEE Conference on Robotics and Automation, pp 1315-1317, 1988.

[10] Y. Kanayama, Y. Kimura, T. Noguchi and F. Miyazaki. A stable control method for an autonomous mobile robot. Proc. IEEE Conference on Robotics and Automation, pp 384-389, 1990.

[11] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized predictive control, Parts I and II. Automatica, Vol. 23, No. 2, pp 137-160, 1987.

[12] R. Hess and Y. Jung. An application of generalized predictive control to rotorcraft terrain-following flight. IEEE Trans. on SMC, Vol. 19, No 5, pp 955-960, 1989.

[13] D. Zhang, L. Van Gool and A. Oosterlinck. Stochastic predictive control of robot tracking systems with dynamic visual feeback. Proc. IEEE Conference on Robotics and Automation, pp 610-615, 1990.

[14] R. Fenton, G. Melocik, K. Olson. On the steering of automated vehicles: theory and experiment. IEEE Trans. AC, pp 306-315, 1976.

[15] W. H. Cormier and R. Fenton. On the steering of automated vehicles- a velocity-adaptive controller. IEEE Trans. AC, pp 375-385, 1980.

[16] D. Shin. High performance tracking of explicit paths by roadworthy mobile robots. Ph. D. Dissertation. Carnegie Mellon University, 1990.